Stan Z. Li · Anil K. Jain · Jiankang Deng *Editors*

Handbook of Face Recognition

Third Edition

Springer

Handbook of Face Recognition

Stan Z. Li · Anil K. Jain · Jiankang Deng
Editors

Handbook of Face Recognition

Third Edition

Springer

Editors
Stan Z. Li
Westlake University
Hangzhou, China

Anil K. Jain
Department of Computer Science
and Engineering
Michigan State University
East Lansing, MI, USA

Jiankang Deng
Department of Computing
Imperial College London
London, UK

ISBN 978-3-031-43566-9 ISBN 978-3-031-43567-6 (eBook)
https://doi.org/10.1007/978-3-031-43567-6

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

This handbook is dedicated to all the researchers, scientists, and engineers who have worked tirelessly to advance the field of face recognition using deep learning. Your dedication, passion, and hard work have contributed to the development of cutting-edge technologies that have the potential to revolutionize the way we live, work, and interact with each other. We also dedicate this handbook to the countless individuals who have contributed to the field of face recognition in various ways, whether through their participation in research studies, their advocacy for privacy and security, or their support of emerging technologies. Your insights, feedback, and contributions have helped shape the direction of this field and continue to inspire innovations. Finally, we dedicate this handbook to the future generations of researchers and innovators who will build on our work and take the field of face recognition to new heights. May you continue to push the boundaries of what is possible and create a brighter, more inclusive, and more equitable future for all.

Foreword

Over the past decade, deep learning has emerged as a powerful tool for solving a wide range of complex problems in computer vision, speech recognition, and natural language processing. One area where deep learning has shown particularly promising results is in face recognition.

Face recognition is a critical technology with applications in security, surveillance, biometrics, and human-computer interaction. Deep learning-based approaches have achieved state-of-the-art performance in face recognition tasks, enabling accurate and efficient recognition of faces in a variety of settings.

This handbook brings together some of the leading experts in the field of deep learning-based face recognition to provide a comprehensive overview of the current state of the art.

The chapters cover a broad range of topics, such as deep learning fundamentals, face detection, facial landmark localization, facial attribute analysis, face presentation attack detection, face feature embedding, video-based face recognition, face recognition with synthetic data, uncertainty-aware face recognition, reducing bias in face recognition, adversarial attacks on face recognition, heterogeneous face recognition, and 3D face recognition.

I believe this handbook will be an invaluable resource for researchers and practitioners interested in deep learning-based face recognition. It provides a comprehensive overview of the field, from the fundamentals to the latest advances, and offers guidance on how to develop and deploy these technologies in a responsible and ethical manner. I am honored to have the opportunity to introduce this important work, and I hope it will inspire innovations and help shape the future of face recognition.

Surrey, UK
December 2023

Josef Kittler

Preface to the Third Edition

As the leading biometric technique for identity authentication, face recognition is widely utilized in areas such as access control, finance, law enforcement, and public security. Over its 50-year history, research and development in this field have been groundbreaking. The emergence of deep learning and neural networks has dramatically reshaped face recognition research and applications in almost every aspect since the publication of the first two editions of this Handbook.

The third edition of the Handbook of Face Recognition presents an entirely new collection of content emphasizing the latest face recognition methodologies and technologies within the deep neural network framework. Featuring contributions from leading experts in the field, this comprehensive handbook offers a current overview of the state-of-the-art while examining the newest developments and emerging trends. The chapters cover a broad range of topics, from the fundamentals of deep learning to the latest advances in face recognition algorithms and applications. This book serves as an all-encompassing resource, providing theoretical underpinnings, algorithms, and implementations to guide students, researchers, and practitioners across all aspects of face recognition. In addition to showcasing the most recent advancements in methods and algorithms, the book also supplies code and data to facilitate hands-on learning and the creation of reproducible face recognition algorithms and systems (Appendix) through deep learning programming. The code and data will be accessible on GitHub and will be updated regularly to keep the materials up to date.

This handbook will be a valuable resource for researchers, and practitioners interested in face recognition. It provides a comprehensive overview of the field and guidance on the responsible development and implementation of these technologies. We extend our gratitude to all the authors for their contributions and the editorial team for their tireless efforts in bringing this handbook to fruition. We hope it will inspire innovation and help shape the future of face recognition.

Hangzhou, China — Stan Z. Li
East Lansing, USA — Anil K. Jain
London, UK — Jiankang Deng

Preface to the Second Edition

Over the past decade, deep learning has emerged as a powerful tool for solving a wide range of complex problems in computer vision, speech recognition, and natural language processing. One area where deep learning has shown particularly promising results is in face recognition.

Face recognition is a critical technology with applications in security, surveillance, biometrics, and human-computer interaction. Deep learning-based approaches have achieved state-of-the-art performance in face recognition tasks, enabling accurate and efficient recognition of faces in a variety of settings.

This handbook brings together some of the leading experts in the field of deep learning-based face recognition to provide a comprehensive overview of the current state of the art. The chapters cover a broad range of topics, such as deep learning fundamentals, face detection, facial landmark localization, facial attribute analysis, face presentation attack detection, face feature embedding, video-based face recognition, face recognition with synthetic data, uncertainty-aware face recognition, reducing bias in face recognition, adversarial attacks on face recognition, heterogeneous face recognition, and 3D face recognition.

I believe this handbook will be an invaluable resource for researchers and practitioners interested in deep learning-based face recognition. It provides a comprehensive overview of the field, from the fundamentals to the latest advances, and offers guidance on how to develop and deploy these technologies in a responsible and ethical manner. I am honored to have the opportunity to introduce this important work, and I hope it will inspire innovations and help shape the future of face recognition.

Hangzhou, China
March 2023

Stan Z. Li

Acknowledgements

We would like to express our sincere gratitude to all the authors who contributed to this handbook. Their expertise, dedication, and hard work made this handbook possible. We are grateful for their willingness to share their knowledge and insights with the broader community. We would also like to thank the editorial team at Springer, who provided invaluable support and guidance throughout the editorial process. Their professionalism, expertise, and attention to detail were critical to the success of this project. We would like to acknowledge the many researchers and practitioners who have contributed to the field of face recognition over the years. Their insights, feedback, and contributions have helped shape the direction of this field and continue to inspire innovations. We hope that this handbook will be a valuable resource for the community and contribute to the development of responsible, ethical, and innovative face recognition technologies.

Contents

Part I Introduction and Fundamentals

Part II Face Processing Modules

Contributors

Rama Chellappa Johns Hopkins University, Baltimore, USA

Shouhong Ding Tencent, Shanghai, P.R. China

Yuantao Feng Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

Zhenhua Feng School of Computer Science and Electronic Engineering, University of Surrey, Guildford, UK

Xinbo Gao Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing, China

Dihong Gong Tencent Data Platform, Shenzhen, China

Sixue Gong Department of Computer Science & Engineering, Michigan State University, Michigan, USA

Guodong Guo Institute of Deep Learning, Baidu Research, Beijing, China

Di Huang Beihang University, Beijing, China

Yuge Huang Tencent, Shanghai, P.R. China

Ziqi Huang S-Lab, Nanyang Technological University, Singapore, Singapore

Anil K. Jain East Lansing, MI, USA;
Department of Computer Science & Engineering, Michigan State University, Michigan, USA

Yuming Jiang S-Lab, Nanyang Technological University, Singapore, Singapore

Zhen Lei National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China;
School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing, China;

Centre for Artificial Intelligence and Robotics, Hong Kong Institute of Science & Innovation, Chinese Academy of Sciences, Hong Kong, China

Yan-ran Li College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

Zhifeng Li Tencent Data Platform, Shenzhen, China

Ajian Liu Institute of Automation, Chinese Academy of Sciences, Beijing, China

Decheng Liu State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, Shaanxi, China

Wei Liu Tencent Data Platform, Shenzhen, China

Xiaoming Liu Department of Computer Science & Engineering, Michigan State University, Michigan, USA

Ziwei Liu S-Lab, Nanyang Technological University, Singapore, Singapore

Anirudh Nanduri University of Maryland, College Park, USA

Hanyang Peng Pengcheng Laboratory, Shenzhen, China

Haibo Qiu The University of Sydney, Sydney, Australia

Hailin Shi Nio Inc., Beijing, China

Yichun Shi ByteDance, San Jose, USA

Zichang Tan Institute of Deep Learning, Baidu Research, Beijing, China

Dacheng Tao The University of Sydney, Sydney, Australia

Jun Wan Institute of Automation, Chinese Academy of Sciences, Beijing, China

Nannan Wang State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi, China

Jianqing Xu Tencent, Shanghai, P.R. China

Hongyu Yang Beihang University, Beijing, China

Shuai Yang S-Lab, Nanyang Technological University, Singapore, Singapore

Xiao Yang Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China

Baosheng Yu The University of Sydney, Sydney, Australia

Shiqi Yu Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

Zitong Yu School of Computing and Information Technology, Great Bay University, Dongguan, China

Jianguo Zhang Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

Xiangyu Zhang MEGVII Technology, Beijing, China

Chenxu Zhao MiningLamp Technology, Beijing, China

Jingxiao Zheng Waymo, Mountain View, USA

Jun Zhu Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China

Xiangyu Zhu Institute of Automation, Chinese Academy of Sciences, Beijing, China

Part I

Introduction and Fundamentals

1 Face Recognition Research and Development

Zichang Tan and Guodong Guo

1.1 Introduction

Face recognition aims to identify or verify the identity of a person through his or her face images or videos. It is one of the most important research topics in computer vision with great commercial applications [37, 59, 86, 210], like biometric authentication, financial security, access control, intelligent surveillance, etc. Because of its commercial potential and practical value, face recognition has attracted great interest from both academia and industry.

The concept of face recognition probably appeared as early as 1960s [10], when researchers tried to use a computer to recognize the human face. In the 1990s and early 2000s, face recognition had rapid development and methodologies were dominated by holistic approaches (e.g., linear space [11], manifold learning [70], and sparse representations [227, 253]), which extract low-dimensional features by taking the whole face as the input. Later, in the 2000s and early 2010s, local descriptors (like Gabor [103, 129], LBP [3, 118], DFD [104], etc.), and Support Vector Machine (SVM) [58] were applied to face recognition, which further improved the recognition performance. However, these traditional methods [3, 11, 70, 104, 129, 227, 238, 253] suffer from elaborate design and shallow representations and hardly achieve a robust recognition performance against complex variations in head poses, occlusions, illuminations, expressions, etc.

In 2014, deep learning [67, 98, 101, 185, 191] was applied to the problem of face recognition and made remarkable achievements. For example, DeepFace [192] and DeepID [190],

Z. Tan · G. Guo (✉)
Institute of Deep Learning, Baidu Research, Beijing, China
e-mail: guoguodong01@baidu.com; guodong.guo@mail.wvu.edu

Z. Tan
e-mail: tanzichang@baidu.com

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_1

which are constructed based on several Convolutional Neural Networks (CNN) layers, reached human-like performance (99% accuracy in LFW benchmark) for the first time. Compared with the traditional methods, the deep learning-based methods show an overwhelming advantage in recognition accuracy. Inspired by this, many researchers have been actively involved in the research of deep face recognition, and have developed a series of algorithms to push the state-of-the-art performance. The success of deep learning for face recognition comes from several aspects. First, the proper structure of stacked CNN layers allows learning discriminative features with a strong invariance against relatively large variations in head pose, occlusions, illuminations, and so on. Second, training deep learning-based methods with massive learnable parameters on large-scale datasets (e.g., CASIA-WebFace [241] and MS-Celeb-1M [63]) allows the extraction of discriminative representations in an efficient way. Last but not least, the development of computing technologies and hardware (e.g., GPUs) provides strong support for large-scale training.

This survey mainly focuses on the latest advances in face recognition using deep learning technologies [28, 60, 62, 131, 135, 205], as well as the development of specific recognition tasks and databases [8, 63, 167, 241, 280].

Generally, a complete face recognition system consists of face detection, alignment, feature extraction, and face matching. For face detection [256, 257] and face alignment [277], they can be regarded as pre-processing steps. For these pre-processing steps, we briefly review the related works, while we will put our focus on face representation and matching.

In this chapter, we present a comprehensive overview of deep face recognition. Figure 1.1 illustrates the structure of this survey. The key insights are as follows:

- The advances of deep methods in face recognition are investigated, including the network architectures, loss functions, face recognition with Generative Adversarial Networks (GAN), and multi-task learning methods.
- Challenges in many specific face recognition tasks are presented, including masked face recognition (FR), large-scale FR, cross-factor FR, heterogeneous FR, low-resolution FR, and FR against adversarial attacks.
- The face datasets for training and evaluation are examined, especially the developments related to noisy data and data imbalance problems.

1.2 Processing Workflow

Generally, the workflow of the face recognition process has four stages, including face detection, face alignment, face anti-spoofing, feature extraction, and face matching. The diagram of the workflow can be found in Fig. 1.2.

Face detection is the basic step of face recognition and other face analysis applications (such as face attribute analysis [193, 194], expression recognition [232, 233], face forgery

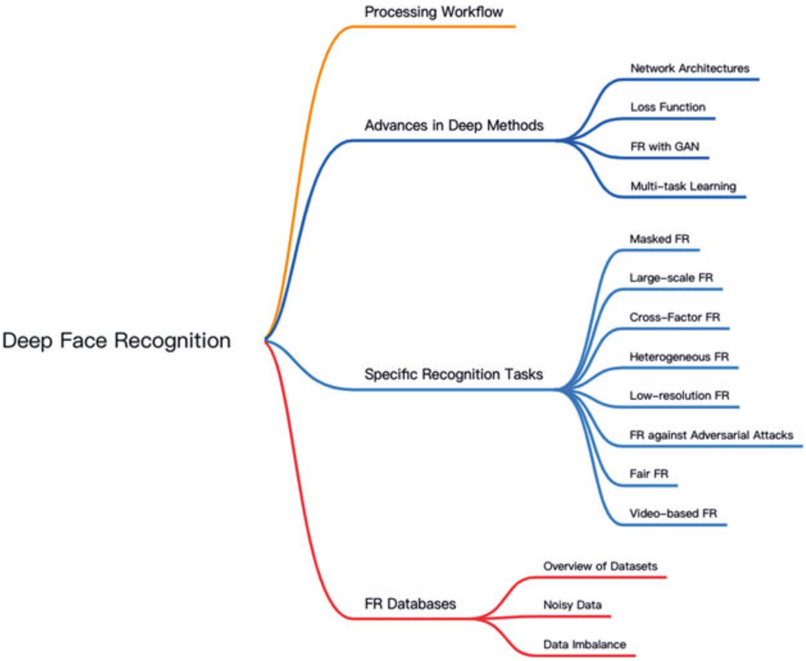


Fig. 1.1 The structure of this survey. The survey focuses on three aspects: the advances in deep methods, specific recognition tasks, and databases

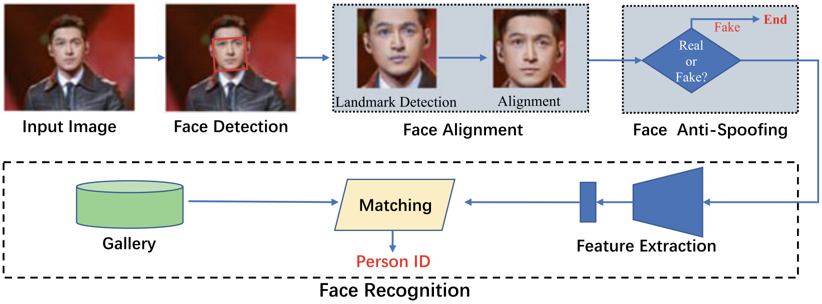


Fig. 1.2 The processing workflow of face recognition. For a given face image, face detection is first employed to locate the face, and then landmark detection is taken to find the locations of some key facial landmarks, which are then used to align the face. In some commercial face recognition systems, face anti-spoofing is also often used to filter fake faces to ensure the security of the system. The real face would be input to the next stage for face recognition, where a deep network is used to extract its face feature and then match the extracted feature with the face features in the gallery

detection [155], etc.). Face detection is to detect the face in a given image, in other words, to find the location of the face. With the development of deep learning and general object detection, face detection has also made great progress in recent years. At the early stages of deep learning, the multi-stage detectors played a leading role in the field of face detection, like MT-CNN [251] and FA-RPN [161]. Multi-stage detectors first generate a number of candidate boxes, and then refine these candidate boxes in the later stages. Multi-stage face detectors usually can achieve good performance but with a low efficiency. To improve the efficiency, some single-stage face detectors are proposed, like S^3FD [257] and RefineFace [254]. Singe-stage face detectors could conduct the classification and bounding box regression from the feature maps directly in a single stage. They remove the stage of generating candidate boxes, which improves the detection efficiency. Moreover, some researchers propose the CPU real-time detectors for real-world applications, like Faceboxes [256] and RetinaFace [27]. Those methods are designed with lots of efficient components like a lightweight network, rapidly digested convolutional layer, knowledge distillation, and so on, which largely improves the efficiency.

Face alignment [61, 277] is the next step after face detection. Its goal is to calibrate unconstrained faces and it facilitates the later stages of face feature extraction. Most existing methods align the face through some pre-defined facial landmarks, which is called landmark-based face alignment. For example, given five points at the center of two eyes, tip of the nose, and two corners of the mouth, the face is then aligned by an affine transformation. Therefore, the quality of face alignment mainly depends on the quality of facial landmark detection, and the core of face alignment is how to accurately detect facial landmarks. Existing methods of landmark regression can be classified into three categories, namely regression-based [143], heatmap-based [228], and 3D model fitting [61, 277] methods. Regression-based methods usually employ several regression layers to regress the landmark locations with L1, L2, or smoothed L1 loss functions. Regarding heatmap-based methods, they predict the score maps of all landmarks, which is inspired by the works of human pose estimation [164, 207]. For 3D model fitting, it aims to improve the accuracy of landmark locations by exploring the relationship between 2D facial landmarks and 3D facial shapes.

The detailed review on face anti-spoofing can be found in Sect. 1.6. Feature extraction and face matching are the core parts of our survey. In recent years, many methods have been proposed to study how to extract better facial features or improve face matching, like designing network architectures, modifying loss functions, improve FR with GAN, and multi-task learning strategies. The related works will be introduced in the following.

1.3 Advances in Deep Methods

In the past several years, a large number of researchers have been devoted to the research on face recognition and numerous deep learning based methods have been proposed [28, 60, 62, 131, 135, 205]. In these studies, Convolutional Neural Networks (CNN) [102] are the most popular architecture and have brought great improvement to face recognition. Thus, we first present a detailed overview of the employed CNN architectures. Face recognition is different from the general object classification problem, where there are massive classes but the inter-class differences are very small. How to design an effective loss function to train the deep networks has became one of the hottest research directions. Thus, the progress of loss function modification will be checked extensively in the following. With the development of Generative Adversarial Networks (GAN) [56], applying GAN for face recognition has gained promising developments, especially in domain-invariant face recognition. Moreover, we present the advances in multi-task learning methods, which learns the task of face recognition with other tasks, such as pose estimation and gender recognition.

1.3.1 Network Architectures

The early CNN architectures in face recognition, like DeepFace [192] and DeepID [190], only contained a few neural layers due to the limited computing capability. For example, DeepFace consists of nine layers including convolutional, local-connected and fully connected layers. In the following years, the architecture evolved along with the evolution of networks for general object classification. For example, in 2015, FaceNet [178] and VGG-Face [167] utilized the GoogleNet [191] and VGGNet [185] for extracting face features, respectively. Later, SphereFace [135] designed a 64-layer residual networks with an angular loss named A-softmax to extract discriminative features. In 2019, ArcFace [28] proposed an improved residual networks named IResNet. IResNet was constructed based on the standard ResNet [67], but replacing the redisual unit with an improved residual unit, which has a BN-Conv-BN-PReLu-Conv-BN structure. Experiments show that this improved unit can obviously improve the verification performance. This made IResNet the most popular structure in face recognition, widely used in both academia and industry. Han et al. [65] proposed a personalized convolution for face recognition, which aims to enhance the individual characteristics while suppressing common characteristics for each person. Recently, transformer [35] was also taken for extracting face representations [106, 273]. However, Zhong et al. [273] took various transformers and a classic CNN architecture (ResNet-100) with similar number of parameters for comparisons. Experiments show that the transformer could achieve comparable performance with the CNN when sufficient images are accessible for training.

Besides designing network architectures, it is also important to enhance the networks' capability by developing attention mechanisms for face recognition [28, 120, 121, 211, 213]. In ArcFace [28], an attention-based network, namely SE-IResNet, was constructed with applying Squeeze-and-Excitation (SE) [73] attention to IResNet. SE attention is a kind of channel-wise attention, and it recalibrates channel-wise feature responses by learning attentive weights. Spatial attention is also widely used in face recognition. For example, Wang et al. [213] proposed a pyramid diverse attention (PDA) to adaptively learn multi-scale diverse local representations. DSA-Face [212] presented diverse and sparse attentions, which extract diverse discriminative local representations while suppressing the responses on noisy regions. Moreover, Ling et al. [121] proposed SRANet that consists of self channel attention (SCA) and self spatial attention (SSA) blocks to learn channel and spatial attentions simultaneously for better capturing discriminative feature embeddings. In addition to channel and spatial attentions, Kang et al. [89] proposed an Attentional Feature-pair Relation Network (AFRN), which represents a face by exploring the relations of pairs of local appearance block features. These attention mechanisms usually contain only a small number of calculations and parameters but they can bring considerable performance gains. Therefore, the attention mechanism has attracted great research interests and lots of attention mechanisms have been constructed and applied to face recognition in the past several years.

In the early stages of deep face recognition, the employed networks only contained a few layers [190, 192]. However, with the development of related technologies and the pursuit of high recognition performance, the networks became deeper and deeper. For example, the popular network IResNet100 in ArcFace [28] consists of 100 layers (with a model size of 249M), which is difficult to put into practical use. From a practical point of view, how to design a lightweight architecture for face recognition is also an important task in the community. In 2018, Wu et al. [229] proposed a Light CNN framework, which aims to learn a compact embedding against large-scale data with noisy labels. In this work, three networks, namely Light CNN-4, Light CNN-9 and Light CNN-29, were carefully designed to obtain good performance while reducing the complexity and computational costs. In 2019, Deng et al. [30] holded a challenge/workshop named *Lightweight Face Recognition Challenge* in conjunction with ICCV 2019, which attracted a large number of researchers to participate in the competition. Most importantly, lots of insightful solutions, like VarGFaceNet [235], AirFace [111] and ShuffleFaceNet [148], were proposed and promoted the progress of lightweight face recognition. For example, VarGFaceNet [235] proposed a variable group convolution to support large-scale face recognition while reduced the computational cost and parameters. VarGFaceNet finally won the first place in DeepGlint-Light track with an accuracy of 88.78% at FPR=1e-8 while containing only 1G FLOPs.

1.3.2 Loss Function

Loss function [28, 51, 135, 136, 168, 201, 205, 217] plays an important role in deep learning, especially for face recognition. Early works [178, 190, 192] adopted the Softmax loss or Triplet loss for face recognition. However, lots of large face datasets have been assembled in recent years (e.g., WebFace42M [280] contains over 200K identities), the plain softmax and triplet losses are not satisfactory to extract discriminative face features. In recent years, lots of works focus on how to design an effective loss function, which have brought great progress to face recognition. The newly developed losses mainly can be divided into four main categories: including classification-based loss, metric learning loss, set-based loss and adaptive loss, which will be presented in detail in the following.

Face recognition can be regarded as a classification problem, and some early works [190, 192] used the softmax loss for deriving face representations. In verification stage, the cosine similarity or equivalently L2 normalized Euclidean distance for a pair of face representations is calculated for matching. However, the features are not normalized in the standard softmax loss, which results in an inconsistency between the training and testing stages. To eliminate this inconsistency, NormFace [202] was proposed to add feature normalization to the softmax loss. Later, angular-margin-based loss was proposed to explicitly encourage intra-class compactness and inter-class separability, e.g., L-softmax [136], A-softmax [135], AM-softmax/CosFace [201, 205] and ArcFace [28]. In L-softmax, the separability between the sample $\boldsymbol{x}$ and the parameter $\boldsymbol{W}$ is transformed to angular similarity: $\boldsymbol{W}\boldsymbol{x} = ||\boldsymbol{W}||_2||\boldsymbol{x}||_2 cos(\theta)$, and then it produces the angular margin by multiplying a constant m with the angle θ (i.e., $cos(m\theta)$), which learns compact and separated features. Then, A-softmax further constrains the learned features lying on a hypersphere manifold by setting $||\boldsymbol{W}||_2 = 1$. For AM-softmax/CosFace and ArcFace, they produce the angular margin by computing $cos(\theta) - m$ and $cos(\theta + m)$, respectively, which are easy to be trained and achieve more compact features. Inspired by these losses, lots of variants [85, 139, 188, 216, 244] have been developed in recent years as well, showing clearly the effectiveness of angular-margin-based softmax loss for face recognition.

The metric learning loss [64, 178] optimizes the networks based on a pair-wise distance or similarity. In 2015, Schroff et al. [178] proposed the Triplet loss, which learns feature representations based on the triplet of one anchor, one positive and one negative samples. The triplet loss aims to narrow the distance between the positive pair (intra-variation) and enlarge the distance between the negative pair (inter-variation). Later, Sohn et al. [186] generalized the triplet loss by associating each sample with more than one negative samples. Hierarchical triplet Loss [48] collects informative triplets according to a defined hierarchical tree. There are also some works focusing on hard example mining [91, 282] to select effective sample pairs for training. For example, Tan et al. [282] extended the selection space of hard sample pairs by taking samples in previous batches as a reference. Other losses, e.g. SFace [274], CDT [42], Circle loss [189], also promote the accuracies of face recognition.

The performance of the developed losses usually highly depends on the hyperparameter settings. In the works [28, 93, 135, 201, 205], the parameters are usually fixed for all classes, which ignores the differences between different categories. Some researchers [83, 128, 132, 153, 165, 219, 259] propose adaptive losses by adjusting parameters dynamically, which aims to obtain more effective supervisions during training. For example, Liu et al. [132] proposed an Adaptive Margin Softmax Loss (AdaM-Softmax) to adaptively find the appropriate margins for different classes. Zhang et al. [259] proposed the AdaCos to dynamically change the scale and margin parameters. Moreover, Liu et al. [128] adopted a margin-aware reinforcement learning to adaptively learn margins. Kim et al. [93] proposed the AdaFace to adaptively adjust the margins based on the image quality.

The set-based losses [39, 80, 224, 225] were designed based on a set of samples rather than a single example. For example, Center loss [224, 225] tried to enhance the discriminative capability by learning a center for each class, and then adding an extra regularization term to narrow the distance between the face feature and the corresponding center. Later, Git loss [14] improved the Center loss by adding an additional term to maximize the distance between the face feature and other negative centers. Range loss [258] proposed to reduce intra-class variations while enlarging inter-class variations based on the range and center of each class. Moreover, other losses, e.g. Contrastive-center loss [171] and Cosmos-Loss [80], were also constructed based on feature centers and gain some improvements. Liu et al. [134] proposed a Feature Consistency Distillation (FCD) loss to fully transfer the relation-aware knowledge from the teacher to student.

The loss function is essential in deep face recognition, and it has made a significant progress in recent years. For detailed descriptions and mathematical forms about the losses, please refer to the previous survey [59].

1.3.3 Face Recognition with GAN

In 2014, Goodfellow et al. [56] proposed the Generative Adversarial Networks (GAN), which employs a generator and a discriminator to perform adversarial learning in the form of zero-sum game. Since then, GAN has attracted great research interest and successfully applied to various tasks. In face recognition, GAN is mainly taken for face synthesis and feature disentanglement in order to achieving a better performance. We introduce the related advances of adversarial learning in these sub-fields.

Face synthesis is a straightforward solution to assist face recognition through the use of GAN. In video face recognition, one common practice is to generate one or more high-quality face images [157, 174, 175] by a given input video. Then, the high-quality face picture is taken for feature extraction and thus the recognition performance can be improved. Some researchers [197] adopted synthetic faces to augment the face dataset, which led to an increased recognition accuracy. Others synthesized faces in various cross-factor settings, e.g. cross-domain [107, 158], cross-pose [16, 23, 82, 117, 133, 264, 265], cross-spectral [44, 68, 245], cross-age [84, 263, 267], cross-makeup [114, 250], etc. A common way in these methods is to generate an appropriate face image from other imaging conditions, e.g. generating frontal-view faces [23, 133, 265], cross-spectral face generation [44, 68, 187, 245], facial de-makeup [114, 250] and so on. For example, Zhao et al. [265] proposed a Face Frontalization sub-Net (FFN) to normalize profile face images to frontal pose, which facilitates pose-invariant feature extraction. Note that face synthesis is not the whole in those methods. Most of them only took the face synthesis as an intermediate step and then combined it with other modules to achieve a better performance. Let us take the work [265] as an example again. In addition to the face frontalization part, it further proposed a Discriminative Learning sub-Net (DLN) to capture pose-invariant features. However, using synthetic faces for training is not always useful. For example, previous works [25, 97] trained networks on both the real dataset and the synthetic data. Both showed that the synthetic data could give a reduced performance on real-world face recognition, compared with training on real datasets. The reduced performance may be caused by the poor intra-class variations in synthetic data and the domain gap between synthetic and real face images. Therefore, although various works have demonstrated that synthetic data could improve the performance of face recognition, it still requires care on how to select appropriate algorithms and strategies to use the synthetic data.

The feature disentanglement with GAN also received lots of attention. In 2017, Chen et al. [22] proposed an InfoGAN to learn interpretable representations, which is probably the first work of applying GAN for feature disentanglement. The disentangled representation is useful for face recognition, which usually only requires the identity knowledge of the face data. One intuitive approach of feature disentanglement [142] in face recognition is to disentangle the identity-related features from other irrelevant features (e.g., pose, age, makeup and modality information) [113, 196, 204, 263, 268], and thus reduces the impact of irrelevant information on recognition. Such an approach is widely used in cross-pose [196, 204], cross-age [263, 268], cross-makeup [113] and heterogeneous [130] face recognition. For example, DR-GAN [196] disentangled the face representations from pose variations through providing pose code to the decoder and pose estimation in the discriminator. In this way, the discriminative pose-invariant identity representations are captured because the pose variations are explicitly disentangled. Besides, feature disentanglement is also employed to unbiased face recognition [52]. More specifically, DebFace [52] extracted disentangled feature representations (including identity, age, gender and race representations) via adversarial learning, and then employed the disentangled features for face recognition and demographics estimation with abating bias influence.

1.3.4 Multi-task learning

A face contains a wealth of information, e.g., identity, gender, race, pose, etc, which are highly coupled and correlated and may increase the difficulty for face recognition. Therefore, some researchers [40, 52, 84, 163, 221, 242, 247] proposed to use the multi-task learning to do multiple tasks together, such as face recognition, age estimation, gender classification and pose estimation. In such a way, different tasks can be exploited and interacted with each other, which may facilitate to capture complementary features. For example, Yin et al. [242] proposed a multi-task network by jointly learning identity classification and other side tasks including pose, illumination, and expression estimations. By jointly learning them, the side tasks help reduce pose, illumination and expression variations in the learned identity features. Besides boosting the recognition performance, multi-task learning also helps in reducing computation cost and improving efficiency, since multiple tasks (e.g., identity and expression) can be accomplished by using a single network.

1.4 Recent Development in Specific Face Recognition Tasks

1.4.1 Large-Scale Face Recognition

With the wide application of face recognition technology and the requirements of high performance, the face database has become larger and larger in recent years, as shown in Fig. 1.4. For example, the largest academic face dataset, i.e., WebFace260M [280], consists of 260M images of over 200K identities. Moreover, some companies use more face data than this to train the model for commercial use. Although training on such large-scale datasets usually can lead to a good performance it also poses a great pressure on computing resources, where a large-scale distributed cluster is needed to conduct model training. For example, when classifying on millions of identities, hundreds of millions of parameters would be produced in the classifier layer, which requires many computational resources especially the GPU memory. However, such high requirements on devices are infeasible for most academic laboratories. How to train a face model on a large-scale face recognition dataset with limited computational resources is still a challenge [8, 110, 278, 282]. To address this problem, ArcFace [28] took a distributed learning and Automatic Mixed Precision (AMP) to reduce computational cost. Further, An et al. [7, 8] proposed a Partial-FC and Li et al. [110] propose Virtual-FC to reduce the computational consumption especially the GPU memory when classifying millions of identities. More specifically, Partial-FC randomly samples partial classes (e.g., 10%) rather than taking the full classes for training at each step, while Virtual-FC divides N identities into M groups ($M \ll N$) with each group of identities sharing the parameter vector in the projection matrix (W). Moreover, Zhu et al. [278] developed a Dominant Prototype Softmax (DP-softmax) to only select important prototypes for training. Recently, Tan et al. [282] trained the model on ID vs. Spot (IvS) face dataset, containing millions of identities using metric learning, where the classification layer is removed and massive parameters can be avoided. There is room left to improve large-scale face recognition. How to train a large-scale face model faster, better and more efficiently will be a long-term topic for the whole FR community.

1.4.2 Cross-Factor Face Recognition

Some common research topics of cross-factor face recognition are cross-pose FR, cross-age FR and cross-makeup FR. We review them separately in the following.

Cross-pose FR: Face recognition still suffers from pose variations although cross-pose face recognition has been studied for many years. Cross-pose face recognition is also known as pose-invariant face recognition, which aims to improve face recognition when exhibiting large poses variations. In cross-pose face recognition, using synthetic faces (e.g., frontal view) [16, 23, 82, 117, 133, 141, 147, 243, 261, 264–266] to assist recognition has aroused great interest. More specifically, these methods synthesize faces with different views by using

GAN [23, 82, 147, 264, 265] or 3D Morphable Model (3DMM) [147, 243, 266], which brings better identity information for the self-occluded missing part and thus boosts the performance in recognition. Taking a recent work as an example, Marriott et al. [147] incorporated a 3D morphable model into a GAN, which allows to manipulate a person's expression, illumination and pose without compromising the identity. Then, the manipulated faces were employed to augment the dataset and enhance the performance. As we have mentioned in Sect. 1.3.3, feature disentanglement [152, 170, 196, 204] is also a reliable way for cross-pose face recognition, where the identity features used for recognition are disentangled from other features (e.g., pose). Besides synthesizing faces and feature disentanglement, other methods learn pose-invariant features by multi-task learning [40, 242], e.g. jointly learning identity classification and pose estimation [242], or designing effective networks [18, 119, 198] like using attention mechanism [198] and Deep Residual EquivAriant Mapping (DREAM) block [18].

Cross-age FR: Face recognition across age is still a challenging problem due to large variations caused by face aging. Similar to cross-pose face recognition, both synthesizing faces [84, 263, 267] and feature disentanglement [204, 220, 263, 268] are effective ways to capture age-invariant features. Somewhat different, both synthesizing faces and feature disentanglement aim to reduce aging rather than pose variations. In particular, different from employing the GAN for feature disentanglement [204, 263, 268], Wang et al. [220] decomposed features in a spherical coordinate system with the angular component representing identity and the radial component representing age information. Note that the angular component and radial component are orthogonal. Moreover, some other works captured age-invariant features by exploring multi-layer fusion [234] or cascading networks utilizing cross-age identity difference [38]. Experiments show that those methods gained promising improvements.

Cross-makeup FR: Facial makeup can greatly change facial characteristics, especially in this era that the makeup technology is so developed. Some researchers employed GAN to first remove makeup [114, 250], and then the synthesized non-makeup faces were employed for identity recognition or verification. Li et al. [113] decomposed the face representations into a makeup code and an identity code, which reduce the effect of makeup for recognition. Considering that the face makeup is mainly presented in the eyes and mouth area, some works [115, 214] employ a multi-branch CNN to capture global and local information concurrently, where the one corresponds to the global region and the others correspond to local regions, e.g. eyes and mouth.

1.4.3 Masked Face Recognition

With the outbreak of the COVID-19 epidemic, people usually wear masks to hinder the spread of the virus. Wearing masks poses a great challenge to face recognition, because a large part of the face area is occluded, causing the general face recognition systems

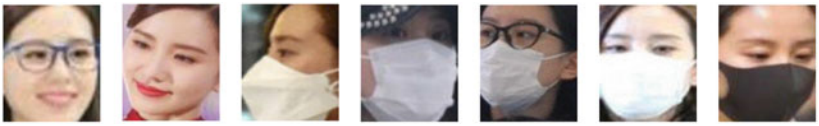


Fig. 1.3 Face examples from RMFRD [77, 222] with and without facial masks. For the faces wearing masks, a large part of the face area is blocked, which brings great difficulties to recognition

and algorithms to fail. Some examples of masked face recognition are shown in Fig. 1.3. Since 2019, masked face recognition has become an active research topic, and various new algorithms [9, 12, 21, 36, 49, 72, 87, 105, 112, 150, 159, 163, 172, 173, 208, 215, 246, 271] emerged endlessly. In masked face recognition, one obvious challenge is the lack of training face images with masks. A simple and effective approach against this problem is to synthesize masked face images for training [21, 172, 208, 215, 246]. However, synthesized images are still different from real masked face images. There are still some defects when applying the model trained on synthesized images to real-world applications. Considering that the masked faces are easily available on the web, some researchers [271] proposed a webly supervised meta-learning approach to utilize web images of containing label noise. Many other studies proposed to focus on learning attentive features against mask occlusions [21, 112, 212], improving the fairness of masked face recognition among different people groups [172, 246], designing effective losses [12, 173] or unmasking [33]. Moreover, the work [87] conducted an empirical study on masked face recognition, in which lots of experiments were executed to investigate the impacts of network architectures, loss functions and training strategies on recognition performance.

In addition to algorithms, lots of databases and benchmarks [77, 78, 156, 199, 222] have been developed in recent years, including Real-world Masked Face Recognition Dataset (RMFRD) [77, 222], Synthetic Masked Face Recognition Dataset (SMFRD) [77, 222], Masked LFW (MLFW) [199] and Indian Masked Faces in the Wild (IMFW) [156]. RMFRD and IMFW consist of images from real-world scenes, while SMFRD and MLFW are synthetic masked face datasets, which were created by wearing virtual masks on faces in existing datasets. These datasets will be introduced in detail in Sect. 1.5.

Several competitions were held to promote the development of masked face recognition, such as Masked Face Recognition Competitions in IJCB 2021 (IJCB-MFR-2021) [13], Masked Face Recognition Challenge (MFR) in ICCV 2021 [26, 279] and Face Recognition Vendor Test (FRVT): Face Mask Effects.[1] IJCB-MFR-2021 only focuses on the performance of masked face recognition and evaluates models on a private masked face dataset. MFR ICCV 2021 is more comprehensive and realistic, as it employs a mixed metric that considers both masked and standard faces. In MFR ICCV 2021, two tracks were held, namely InsightFace Track and WebFace260 Track. The differences between the two tracks mainly exist in training and evaluation datasets. For example, the models were evaluated with more

[1] https://pages.nist.gov/frvt/html/frvt_facemask.html.

comprehensive datasets including a mask set, a children set and a multi-racial set in Insight-Face Track. Both competitions attracted teams from all over the world and lots of insightful solutions [173, 208] were proposed for masked face recognition, dramatically improving the state-of-the-art performance. For the contest in FRVT, it mainly quantifies the accuracy of 1:1 verification for people wearing masks. The masked faces were generated by applying a synthetic mask to the existing raw images. Different from the above two competitions, the FRVT contest is an ongoing test that remains open to new participations, which means that the participating teams can submit their results at any time, and the ranking list is also dynamically updated. By March 2022, over 300 algorithms have been submitted to FRVT.

1.4.4 Heterogeneous FR

Heterogeneous Face Recognition (HFR) objective is to recognize and match faces from different visual domains or modalities. Typical HFR tasks include VIS-NIR face recognition (VISble face vs. Near InfRared face) [32, 44, 51, 75, 100, 239], Photo-Sketch face recognition (face photo vs. face sketch) [41, 160, 166], and IvS face recognition (ID face vs. Spot face) [183, 184, 278, 282]. There are many challenges in the task of HFR, because of the large modality discrepancy between different sensors. The lack of data in some specific modalities is another issue. For example, there are significant discrepancies between the RGB images and the near infrared images in VIS-NIR face recognition. Besides, near infrared faces are not often collected, which also brings difficulty in training. Although HFR has been studied for many years, the relevant research is still valuable and plays an important role in daily lives. For example, the VIS-NIR face recognition provides an effective solution for low-light recognition scenarios. Photo-Sketch face recognition has a wide range of applications in both social entertainment and police enforcement. For IvS face recognition, it verifies the identity of a person by comparing the live face to the photo of an ID Document, which is widely used in our daily life, e.g. ID card gates in railway stations or ePassport gates.

In HFR, a popular approach is to translate the images from one domain/modality to the another to reduce the domain/modality variations [32, 41, 44, 68, 71, 88, 140, 145]. For example, Mallat et al. [145] propose the cascaded refinement networks to generate visible-like colored images of high visual quality for cross-spectrum face recognition, which reduces the need for large amounts of training data. Fang et al. [41] proposed an Identity-Aware CycleGAN (IACycleGAN), focusing on key facial regions (e.g., eyes and noses), which improved the quality of synthetic faces in both sketch-to-photo and photo-to-sketch transfer tasks, and thus improved performance in photo-sketch face recognition. A lot of efforts have been invested in other works [32, 44, 68, 71, 88, 140] to improve the quality of the generated images.

Some works [36, 43, 44, 230] addressed the problem of HFR by mapping data from different domains to a common latent space, or learning domain-invariant features from different domains/modalities. For example, He et al. [69] narrowed the domain gap by

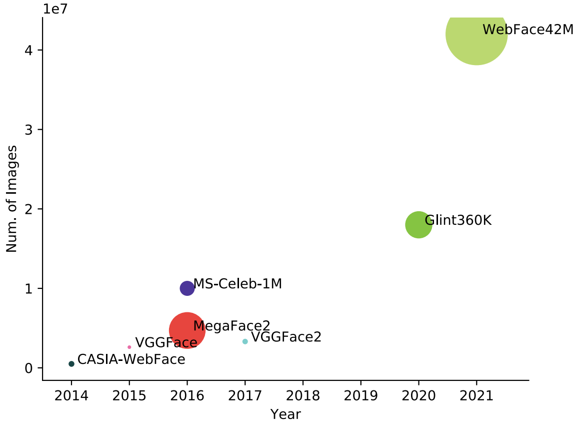


Fig. 1.4 Comparisons of dataset sizes of several popular training datasets published in the deep learning era. The y-axis indicates the number of images in the dataset, and the size of the scatter denotes the number of identities of the corresponding dataset. The collected datasets are getting larger and larger, which helps improve the performance while also poses some challenges in training

Wasserstein distance to obtain domain-invariant features for VIS-NIR face recognition. Further, some researchers addressed HFR from other perspectives [74, 169, 230, 278]. Specifically, Wu et al. [230] made use of the Disentangled Variational Representation (DVR) to treat the HFR as a cross-modal matching task. Tan et al. [282] and Zhu et al. [278] extracted discriminative features for IvS face recognition through a large-scale training with millions of faces.

1.4.5 Low-Resolution Face Recognition

Low-Resolution (LR) face recognition [2, 24, 46, 47, 71, 88, 108, 140, 149] is another important task in real-world applications, e.g. long-distance surveillance. The low-resolution face images are usually captured in non-ideal conditions with low quality, noise and occlusions, which makes the recognition problem quite challenging. As stated in the work [71], one main challenge in low-resolution face recognition is the deviation (domain shift) in gallery and probe sets, in which the probe images are in low resolution, while the gallery images are clear and of high quality. Some researchers addressed this problem by domain transferring [71] or domain translation [88]. For example, Hong et al. [71] proposed an unsupervised face domain transfer method for capturing domain-invariant features. Jiao et al. [88] proposed a dual domain adaptive structure to generate high-quality faces. Recently, Li et al. [109] developed a Rival Penalized Competitive Learning (RPCL) to add penalty to not only the target category but also the rival categories, which further enlarges inter-class variations and narrows intra-class variations.

1.4.6 FR Under Atmospheric Turbulence

Remote face recognition has also been on the spotlight. The challenge is that the face image is usually degraded due to the inconsistent atmospheric refractive indexes when capturing it at long distances. This phenomenon is called atmospheric turbulence and it can lead to a decline in face recognition performance. Some researchers address the atmospheric turbulence by restoring face images. In detail, Lau et al. [99] proposed an ATFaceGAN to reconstruct the restored image and eliminate the blur and deformation caused by atmospheric turbulence. Moreover, Yasarla et al. [240] proposed an AT-Net to remove the atmospheric turbulence distortion from a single image. Recently, Wes Robbins and Terrance Boult presented a different view on the problem of face recognition against atmospheric turbulence [177]. Robbins et al. [177] first studied the influence of atmospheric turbulence on deep features for face recognition, and found that feature magnitudes would increase for a certain degree of atmospheric turbulence. Based on these results, the authors have made more in-depth investigations, including the reason of feature defection for low face recognition performance and proposed several future research directions.

1.4.7 Face Recognition Against Adversarial Attacks

Although a very high performance has been achieved in deep face recognition, researchers have found that those models are vulnerable to adversarial attacks [34, 96, 182, 195, 231, 248, 272, 281]. In general, there are two types of attacks, namely digital attack and physical attack. Concerning the digital attack [34, 96, 195, 248, 272, 281], it creates an attack image by adding imperceptible perturbations on the raw images to mislead the identification. Regarding the physical attack [182], it produces the actual entity for attack rather than the digital image. For example, Adversarial Generative Nets (AGNs) [182] can produce physical eyeglasses, which enables an attacker to either evade correct identification or to impersonate a specific target. Such attacks have threaten the applications of face recognition in security systems, and some researchers [57, 116] have studied defense strategies against them. For example, Li et al. [116] proposed to improve model's robustness by using a denoising neural network with a carefully designed loss function.

1.4.8 Fair Face Recognition

Although the accuracy of face recognition has been improved greatly in recent years, many studies [45, 53, 209, 249] have shown the demographic bias in deep face recognition systems. For example, the error rate of non-Caucasians is usually higher than that of Caucasians in face recognition. The face recognition bias stems from two aspects. One is the data bias. In other words, different groups have different quality or quantity of training data. Training the network on biased data may lead to the recognition bias. The other is the inherent bias as discussed by the work [94], where certain groups are inherently very difficult in face matching. Many fairness technologies were proposed by preventing the model from learning data bias, including the design of novel loss functions [128, 180], data pre-processing [6, 249] and adversarial training [52, 92]. In addition to that, Gong et al. [53] mitigated the bias by learning the feature representations on every demographic group with adaptive convolution kernels and attention mechanisms. Wang et al. [209] employed deep reinforcement learning to increase the fairness of face recognition systems. Despite the many efforts over the years, face recognition bias persists and concerns about it are often expressed. How to effectively eliminate the bias in face recognition systems is still a challenge to further study.

1.4.9 Video-Based Face Recognition

Video-based FR [20, 54, 55, 175, 176, 237] is to extract face representations based on a series of video frames rather than still images. Compared to still images, a video presents temporal and multi-view information, which may facilitate the model to learn better features. One straightforward approach for video-based FR is to aggregate the information from a set of video frames. Intuitively, the aggregation is two-fold: image-level and feature-level aggregations. Image-level aggregation, merges multiple frames to a single frame [174, 175]. For example, Rao et al. [175] proposed a discriminative aggregation network (DAN) to synthesize a high-quality image from multiple low-quality frames. Then, only a small number of aggregated high-quality frames would be sent to the network for feature extraction, thus improving the efficiency and performance of face recognition system. Feature-level aggregation [54, 55, 237, 262] merges multiple face vectors to a single discriminative vector. One widely used scheme to conduct feature-level aggregation is the aggregation through attention mechanisms [54, 55, 237]. For example, Gong et al. [54] employed the recurrent networks to generate attentive fusing weights based on contextual quality information, while some other researchers achieved the aggregation through Graph Convolutional Networks [262]. In recent years, the research in video-based face recognition has not attracted much attention. One reason might be that image-based face recognition has achieved a very high performance (near 100% accuracy), and there is not much room for improvement in video-based face recognition. Moreover, video-based face recognition has larger computation burdens compared to image-based approaches.

1.5 Databases

1.5.1 Overview of FR Datasets

Data plays an important role in deep face recognition. The commonly used training datasets in the deep learning era are CASIA-WebFace [241], VGG-Face [167], VGG-Face2 [19], MS-Celeb-1M [63], MegaFace [90], MegaFace2 [162], Glint360K [8], WebFace42M [280]

Table 1.1 A summary of training and evaluation sets for general face recognition

Datasets	Year	Identities	Images	Description
CASIA-WebFace [241]	2014	10,575	0.5M	Usually as training set
VGG-Face [167]	2015	2,622	2.6M	Usually as training set
VGG-Face2 [19]	2018	9,131	3.31M	Large-scale training set; pose, age, illumination
MS-Celeb-1M [63]	2016	100K	10M	Large-scale training set; noisy data
MegaFace2 [162]	2017	672,057	4.7M	Large-scale training set
Glint360K [8]	2021	360K	18M	Large-scale training set
WebFace42M [280]	2021	2M	42M	Large-scale training set; automatically cleaned data
WebFace260M [280]	2021	4M	260M	Large-scale training set; noisy data
LFW [81]	2008	5,749	13,233	The classic evaluation set
IJB-A [95]	2015	25,809	500	Evaluation set
IJB-B [226]	2017	11,754	1,845	Evaluation set
IJB-C [151]	2018	3,531	21,294	Evaluation set
MegaFace [90]	2016	690K	1M	Used as gallery for evaluation
CFP [179]	2016	7,000	500	Frontal to profile face verification
CPLFW [269]	2018	11,652	3968	Cross-pose evaluation dataset
CALFW [270]	2017	4,025	–	Cross-age evaluation datasets, 6,000 pairs
IFRT [29]	2021	242K	1.6M	Multiple races and large-scale evaluation dataset
FRUITS [280]	2021	2,225	38,578	Cross-age, multiple races, various scenarios

and WebFace260M [280]. The commonly used evaluation datasets include LFW [81], IJB-A [95], IJB-B [226], MegaFace [90], CFP [179], CPLFW [269] and so on. A summary of these datasets is shown in Table 1.1. Regarding training datasets, all of them contain a massive amount of images to ensure the performance of face recognition. For example, the dataset WebFace42M contains about 42M images of over 200K identities. Some other training and evaluation datasets, e.g. IMDb-Face [200] and Megvii Face Classification (MFC) [276], have been detailed summarized by previous surveys [59, 210], and readers can refer to those surveys for more details.

Moreover, the masked face recognition has received increasing attentios in recent years. However, there is no work that systematically summarizes these masked face databases. Due to the high cost of collecting masked face data, researchers prefer to synthesize massive masked faces by adding virtual face masks on existing large-scale datasets. For example, two large masked face datasets, namely SMFRD [77, 222] and Webface-OCC [78], were

Table 1.2 A summary of training and evaluation sets for masked face recognition. For the images column, the symbol ‘A’ indicates the number of all masked and non-masked images. The symbols ‘Y’ and ‘N’ denote the number of masked and non-masked images, respectively

Datasets	Year	Identities	Images (A/N/Y)	Description
RMFRD [77, 222]	2020	525	95K/5K/90K	Containing both masked and unmasked real-world faces
Deng et al. [26]	2021	7K	21K/7K/14K	Evaluation dataset
MFR [279]	2021	2,478	61K/3K/58K	Real-world masked face evaluation dataset

generated based on CASIA-WebFace [241]. These synthesized faces are usually treated as the training data since they are still different from the real-data. Moreover, some works [26, 206] released some tools for wearing masks, which is convenient for generating masked faces for any face dataset. Table 1.2 summarizes the real-world masked face datasets. These datasets usually contain both masked and unmasked faces, which are usually treated as test sets. Here we do not give a summary on the virtual masked datasets because these datasets are uncertain and depend on the employed synthetic strategies. For example, the synthesized masked faces can be more or less, depending on the setting of generating control.

1.5.2 Noisy Data

As mentioned above, face datasets are growing tremendously in recent years. Most of these large-scale datasets, e.g., CASIA-WebFace [241] and MS-Celeb-1M [63], are automatically collected via image search engines or movies, where the labor costs are significantly reduced but noisy labels are also inevitably introduced.

According to previous works [275], the noise can be divided into the following types: (1) Label flips: the image of an identity is wrongly labeled as another identity in the dataset; (2) Outliers: the images, which do not belong to anyone in the dataset, are incorrectly labeled as the identity in the dataset. (3) Entirely dirty data: this noise mainly refers to non face images, which are mainly caused by wrong face detection or annotation. Intuitively, data plays an important role in network training and the noisy data can hurt the performance of the model. Wang et al. [200] conducted a comprehensive investigation on the source of label noise and its consequences for face recognition. The authors found that the performance of face recognition drops rapidly with the increase of label noise. This result also promotes researchers to conduct data cleaning or propose robust algorithms against label noise.

A straightforward way for data cleaning is to clean up the data manually. For example, Wang et al. [200] hired 50 annotators to clean 1.7M images from 2M raw images in a month. Although a clean and high-quality dataset named IMDb-Face was collected, it also cost a lot of human effort. More recently, Zhu et al. [280] proposed a Cleaning Automatically by Self-Training (CAST) method for automatic data cleaning without human intervention. With this algorithm, a large-scale high-quality 2M identities and 42M images (WebFace42M) was collected from the Internet. Besides, lots of robust algorithms [76, 218, 229, 260, 275] against label noise were proposed in recent years. Most previous works learn to detect the noisy data according to output probabilities [275], the distribution of classifier parameters [76] and loss values [218]. After that, the noisy samples will be discarded or their importance will be reduced during training, which forces the clean faces to dominate during training. For example, Wang et al. [218] proposed a co-mining framework, which employed two peer networks to detect the noisy faces. Al Jazaery et al. [4] proposed an approach for detecting identity label noise by incorporating the face image quality measure. In addition to filtering noisy data, some researchers [229, 260] aim at designing novel network structures, which are more robust to noise. For example, LightCNN [229] proposed a Max-Feature-Map (MFM) at every convolutional layer, which can well separate noisy from informative signals.

1.5.3 Data Imbalance

The imbalance phenomenon in face recognition datasets usually exists in the distribution of class (identity) and domain attributes (age, race, etc.). This skewed data distribution can adversely affect the training process of the network, biasing it toward the majority classes. Focusing on the class imbalance, [258] proposed a novel loss function called range loss, which reduces overall intrapersonal variations while enlarges interpersonal differences simultaneously. A similar idea was also proposed in [79], where a Cluster-based Large Margin Local Embedding (CLMLE) was introduced to improve performance. For domain imbalance, Cao et al. [15] and Albiero et al. [5] did several works. For example, [15] proposed a Domain Frequency Indicator (DFI) to judge whether a sample is from head domains or tail domains. Then, they formulated a Residual Balancing Mapping (RBM) block to balance the domain distribution according to the result of DFI. Finally, a Domain Balancing Margin (DBM) was utilized in the loss function to further optimize the feature space of the tail domains. Moreover, Albiero et al. [5] investigated how gender balance in training data affects the test accuracy. Nowadays, the data imbalance problem is still a great challenge for face recognition and more research is needed.

1.6 Other Related Topics

In this section, we briefly discuss some topics closely related to face recognition, including facial attribute recognition [17, 66, 193, 194], face anti-spoofing [123, 124, 223], and face privacy issue [144, 146]. Besides, face detection [257] and face alignment [277] are also closely related to face recognition. The corresponding review on face detection and face alignment has been presented in Sect. 1.2.

Facial attribute recognition: In some applications, a complete face recognition system may function as attribute analysis, e.g. age estimation, gender classification, race recognition and so on, which provides detailed descriptions about what the input face looks like. Although the accuracy of related attribute analysis tasks have been greatly improved [66, 194, 233] in the era of deep learning, some of these attributes still hardly achieve an accurate recognition due to the inherent ambiguity of those attributes themselves. For example, the faces from adjacent ages (e.g., 20 and 21 years old) look very similar and it is difficult to distinguish them. Due to the ambiguity, the attribute recognition tasks can be challenging. For age estimation and expression recognition, the lowest Mean Absolute Error (MAE) and highest recognition rate are only about 1–3 years [31] and 90% [233], respectively, which is far from the near 100% accuracy of face recognition. From the perspective of recognition accuracy, more efforts on facial attribute recognition are urgently needed.

Face anti-spoofing: Face Anti-Spoofing (FAS) [123, 124, 223] is a technology of defending the face recognition system from a variety of presentation attacks (PAs), such as print attack, replay attack, or 3D mask attack. It has been widely incorporated in face recognition systems for face unlocking, payment and self-security inspection. Due to its importance, Presentation Attacks Detection (PAD) has been studied a lot in recent years. A series of datasets have been proposed as well, e.g. CeFA [123] and HiFiMask [127]. At the same time, the face anti-spoofing algorithms were developed from the early binary classification [236] to face depth fitting [137], multimodal fusion [50, 255], cross-modal translation [124], feature disentangling [138, 252] and domain generalization [181, 203]. In order to improve the practicality of face anti-spoofing algorithms, many competitions [122, 125, 126] were successfully held, which promoted the face anti-spoofing algorithm development from academic research laboratory to industry.

Privacy issue: With the expansion of face recognition technology, the face privacy issue has been raised. A human face contains rich personal information, including identity, demographic information and even health status. The face recognition systems collect personal face data, which can be easily copied, shared and abused for illegal purposes, which may lead to the users' concerns regarding their privacy. One approach to regulate the use of FR technology and face data is through legislation. For example, in 2021, New York City enacted the biometric information law to regulate the collection, retention, storage or sharing of biometric information. Moreover, some researchers [1, 154] addressed the data privacy issue in the training stage with federated learning techniques, which learn face recognition models in a privacy aware manner. Even with a lot of efforts, there is still a risk of face data leakage and abuse, so the privacy issue still exists.

1.7 Conclusions

We have conducted a comprehensive survey on deep face recognition from three aspects, deep methods, specific recognition tasks and data. For deep methods, we have reviewed the main advances in network architectures, loss functions, FR with GAN and multi-task learning. Regarding specific recognition tasks, we have presented some challenges with specific tasks, e.g. masked FR, large-scale FR and so on. Concerning datasets, several recent face databases and the developments of addressing data-related issues (noisy data and data imbalance) have been given. The survey shows that great progress has been made in face recognition over the recent years, especially the performance on various applications has been significantly improved, and a large number of databases have been developed. In the future, face recognition will still be a topic of great interest in both academia and industry. Future face recognition works may focus on the following challenges:

- Under the continuous influence of COVID-19, masked face recognition will still be a focus of research. At present, the accuracy of masked face recognition is not very high, and thus, how to improve the accuracy is especially important in the coming years.
- Data plays an important role in face recognition, and collecting large-scale face datasets is still the trend. Large-scale datasets could further promote the development of face recognition models, while they also bring some challenges in training as mentioned in Sect. 1.4. Therefore, the large-scale face recognition will continue to draw attention, especially with big models.
- The FR in complex scenes (e.g., cross-factor FR, low-resolution FR and so on) needs further investigation. These FR tasks are full of challenges and the corresponding performance is far from that of state-of-the-art in the usual recognition scenario.
- The way of collecting large-scale datasets has gradually evolved from full manual annotation to semi-automatic and even full-automatic annotation. The collected datasets inevitably contain noise. Therefore, addressing recognition with noisy data is still worth of investigation in the future.

References

1. Aggarwal, D., Zhou, J., Jain, A.K.: Fedface: collaborative learning of face recognition model. In: IJCB, pp. 1–8 (2021)
2. Aghdam, O.A., Bozorgtabar, B., Ekenel, H.K., Thiran, J.P.: Exploring factors for improving low resolution face recognition. In: CVPRW, pp. 2363–2370 (2019)

3. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. IEEE TPAMI **28**(12), 2037–2041 (2006)
4. Al Jazaery, M., Guo, G.: Automated cleaning of identity label noise in a large face dataset with quality control. IET Biometrics **9**(1) (2020)
5. Albiero, V., Zhang, K., Bowyer, K.W.: How does gender balance in training data affect face recognition accuracy? In: IJCB, pp. 1–10 (2020)
6. Ali-Gombe, A., Elyan, E.: Mfc-gan: class-imbalanced dataset classification using multiple fake class generative adversarial network. Neurocomputing **361**, 212–221 (2019)
7. An, X., Deng, J., Guo, J., Feng, Z., Zhu, X., Yang, J., Liu, T.: Killing two birds with one stone: efficient and robust training of face recognition cnns by partial fc. arXiv preprint arXiv:2203.15565 (2022)
8. An, X., Zhu, X., Gao, Y., Xiao, Y., Zhao, Y., Feng, Z., Wu, L., Qin, B., Zhang, M., Zhang, D., et al.: Partial fc: training 10 million identities on a single machine. In: ICCV, pp. 1445–1449 (2021)
9. Anwar, A., Raychowdhury, A.: Masked face recognition for secure authentication. arXiv preprint arXiv:2008.11104 (2020)
10. Ballantyne, M., Boyer, R.S., Hines, L.: Woody bledsoe: his life and legacy. AI Mag. **17**(1), 7–7 (1996)
11. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE TPAMI **19**(7), 711–720 (1997)
12. Boutros, F., Damer, N., Kirchbuchner, F., Kuijper, A.: Self-restrained triplet loss for accurate masked face recognition. Pattern Recogn. **124**, 108473 (2022)
13. Boutros, F., Damer, N., Kolf, J.N., Raja, K., Kirchbuchner, F., Ramachandra, R., Kuijper, A., Fang, P., Zhang, C., Wang, F., et al.: Mfr 2021: masked face recognition competition. In: IJCB, pp. 1–10. IEEE (2021)
14. Calefati, A., Janjua, M.K., Nawaz, S., Gallo, I.: Git loss for deep face recognition. arXiv preprint arXiv:1807.08512 (2018)
15. Cao, D., Zhu, X., Huang, X., Guo, J., Lei, Z.: Domain balancing: face recognition on long-tailed domains. In: CVPR, pp. 5671–5679 (2020)
16. Cao, J., Hu, Y., Zhang, H., He, R., Sun, Z.: Towards high fidelity face frontalization in the wild. IJCV **128**(5), 1485–1504 (2020)
17. Cao, J., Li, Y., Zhang, Z.: Partially shared multi-task convolutional neural network with local constraint for face attribute learning. In: CVPR, pp. 4290–4299 (2018)
18. Cao, K., Rong, Y., Li, C., Tang, X., Loy, C.C.: Pose-robust face recognition via deep residual equivariant mapping. In: CVPR, pp. 5187–5196 (2018)
19. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: a dataset for recognising faces across pose and age. In: IEEE FG, pp. 67–74 (2018)
20. Cevikalp, H., Dordinejad, G.G.: Video based face recognition by using discriminatively learned convex models. IJCV **128**(12), 3000–3014 (2020)
21. Chang, W.Y., Tsai, M.Y., Lo, S.C.: Ressanet: a hybrid backbone of residual block and self-attention module for masked face recognition. In: ICCVW, pp. 1468–1476 (2021)
22. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: interpretable representation learning by information maximizing generative adversarial nets. NeurIPS **29** (2016)
23. Chen, Z.L., He, Q.H., Pang, W.F., Li, Y.X.: Frontal face generation from multiple pose-variant faces with cgan in real-world surveillance scene. In: ICASSP, pp. 1308–1312 (2018)
24. Cheng, Z., Zhu, X., Gong, S.: Low-resolution face recognition. In: ACCV, pp. 605–621 (2018)
25. Conway, D., Simon, L., Lechervy, A., Jurie, F.: Training face verification models from generated face identity data. arXiv preprint arXiv:2108.00800 (2021)

26. Deng, J., Guo, J., An, X., Zhu, Z., Zafeiriou, S.: Masked face recognition challenge: the insightface track report. In: ICCVW, pp. 1437–1444 (2021)
27. Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: Retinaface: single-shot multi-level face localisation in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5203–5212 (2020)
28. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: additive angular margin loss for deep face recognition. In: CVPR, pp. 4690–4699 (2019)
29. Deng, J., Guo, J., Yang, J., Lattas, A., Zafeiriou, S.: Variational prototype learning for deep face recognition. In: CVPR, pp. 11906–11915 (2021)
30. Deng, J., Guo, J., Zhang, D., Deng, Y., Lu, X., Shi, S.: Lightweight face recognition challenge. In: ICCVW, pp. 0–0 (2019)
31. Deng, Z., Liu, H., Wang, Y., Wang, C., Yu, Z., Sun, X.: Pml: progressive margin loss for long-tailed age classification. In: CVPR, pp. 10503–10512 (2021)
32. Deng, Z., Peng, X., Qiao, Y.: Residual compensation networks for heterogeneous face recognition. In: AAAI, pp. 8239–8246 (2019)
33. Din, N.U., Javed, K., Bae, S., Yi, J.: A novel gan-based network for unmasking of masked face. IEEE Access **8**, 44276–44287 (2020)
34. Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., Zhu, J.: Efficient decision-based black-box adversarial attacks on face recognition. In: CVPR, pp. 7714–7722 (2019)
35. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
36. Du, H., Shi, H., Liu, Y., Zeng, D., Mei, T.: Towards nir-vis masked face recognition. IEEE SPL **28**, 768–772 (2021)
37. Du, H., Shi, H., Zeng, D., Zhang, X.P., Mei, T.: The elements of end-to-end deep face recognition: a survey of recent advances. ACM Comput. Surv. (CSUR) (2020)
38. Du, L., Hu, H.: Cross-age identity difference analysis model based on image pairs for age invariant face verification. IEEE TCSVT **31**(7), 2675–2685 (2020)
39. Duan, Y., Lu, J., Zhou, J.: Uniformface: learning deep equidistributed representation for face recognition. In: CVPR, pp. 3415–3424 (2019)
40. Elharrouss, O., Almaadeed, N., Al-Maadeed, S., Khelifi, F.: Pose-invariant face recognition with multitask cascade networks. Neural Comput. Appl. 1–14 (2022)
41. Fang, Y., Deng, W., Du, J., Hu, J.: Identity-aware cyclegan for face photo-sketch synthesis and recognition. Pattern Recogn. **102**, 107249 (2020)
42. Faraki, M., Yu, X., Tsai, Y.H., Suh, Y., Chandraker, M.: Cross-domain similarity learning for face recognition in unseen domains. In: CVPR, pp. 15292–15301 (2021)
43. Fondje, C.N., Hu, S., Short, N.J., Riggan, B.S.: Cross-domain identification for thermal-to-visible face recognition. In: IJCB, pp. 1–9 (2020)
44. Fu, C., Wu, X., Hu, Y., Huang, H., He, R.: Dvg-face: dual variational generation for heterogeneous face recognition. IEEE TPAMI (2021)
45. Garcia, R.V., Wandzik, L., Grabner, L., Krueger, J.: The harms of demographic bias in deep face recognition research. In: ICB, pp. 1–6 (2019)
46. Ge, S., Zhao, S., Li, C., Li, J.: Low-resolution face recognition in the wild via selective knowledge distillation. IEEE TIP **28**(4), 2051–2062 (2018)
47. Ge, S., Zhao, S., Li, C., Zhang, Y., Li, J.: Efficient low-resolution face recognition via bridge distillation. IEEE TIP **29**, 6898–6908 (2020)
48. Ge, W.: Deep metric learning with hierarchical triplet loss. In: ECCV, pp. 269–285 (2018)
49. Geng, M., Peng, P., Huang, Y., Tian, Y.: Masked face recognition with generative data augmentation and domain constrained ranking. In: ACM MM, pp. 2246–2254 (2020)

50. George, A., Mostaani, Z., Geissenbuhler, D., Nikisins, O., Anjos, A., Marcel, S.: Biometric face presentation attack detection with multi-channel convolutional neural network. TIFS (2019)
51. Ghosh, S., Singh, R., Vatsa, M.: Subclass heterogeneity aware loss for cross-spectral cross-resolution face recognition. IEEE TBIOM **2**(3), 245–256 (2020)
52. Gong, S., Liu, X., Jain, A.K.: Jointly de-biasing face recognition and demographic attribute estimation. In: ECCV, pp. 330–347 (2020)
53. Gong, S., Liu, X., Jain, A.K.: Mitigating face recognition bias via group adaptive classifier. In: CVPR, pp. 3414–3424 (2021)
54. Gong, S., Shi, Y., Jain, A.: Low quality video face recognition: multi-mode aggregation recurrent network (marn). In: ICCVW, pp. 0–0 (2019)
55. Gong, S., Shi, Y., Kalka, N.D., Jain, A.K.: Video face recognition: component-wise feature aggregation network (c-fan). In: ICB, pp. 1–8 (2019)
56. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS **27** (2014)
57. Goswami, G., Agarwal, A., Ratha, N., Singh, R., Vatsa, M.: Detecting and mitigating adversarial perturbations for robust face recognition. IJCV **127**(6), 719–742 (2019)
58. Guo, G., Li, S.Z., Chan, K.: Face recognition by support vector machines. In: IEEE FG, pp. 196–201. IEEE (2000)
59. Guo, G., Zhang, N.: A survey on deep learning based face recognition. Comput. Vis. Image Underst. **189**, 102805 (2019)
60. Guo, J., Zhu, X., Lei, Z., Li, S.Z.: Decomposed meta batch normalization for fast domain adaptation in face recognition. IEEE TIFS **16**, 3082–3095 (2021)
61. Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z., Li, S.Z.: Towards fast, accurate and stable 3d dense face alignment. In: European Conference on Computer Vision, pp. 152–168. Springer (2020)
62. Guo, J., Zhu, X., Zhao, C., Cao, D., Lei, Z., Li, S.Z.: Learning meta face recognition in unseen domains. In: CVPR, pp. 6163–6172 (2020)
63. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: a dataset and benchmark for large-scale face recognition. In: ECCV, pp. 87–102. Springer (2016)
64. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR, vol. 2, pp. 1735–1742 (2006)
65. Han, C., Shan, S., Kan, M., Wu, S., Chen, X.: Personalized convolution for face recognition. IJCV pp. 1–19 (2022)
66. Han, H., Jain, A.K., Wang, F., Shan, S., Chen, X.: Heterogeneous face attribute estimation: a deep multi-task learning approach. IEEE TPAMI **40**(11), 2597–2609 (2017)
67. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
68. He, R., Cao, J., Song, L., Sun, Z., Tan, T.: Adversarial cross-spectral face completion for nir-vis face recognition. IEEE TPAMI **42**(5), 1025–1037 (2019)
69. He, R., Wu, X., Sun, Z., Tan, T.: Wasserstein cnn: learning invariant features for nir-vis face recognition. IEEE TPAMI **41**(7), 1761–1773 (2018)
70. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. IEEE TPAMI **27**(3), 328–340 (2005)
71. Hong, S., Ryu, J.: Unsupervised face domain transfer for low-resolution face recognition. IEEE SPL **27**, 156–160 (2019)
72. Hsu, G.S.J., Wu, H.Y., Tsai, C.H., Yanushkevich, S., Gavrilova, M.: Masked face recognition from synthesis to reality. IEEE Access (2022)
73. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR, pp. 7132–7141 (2018)
74. Hu, W., Hu, H.: Disentangled spectrum variations networks for nir-vis face recognition. IEEE TMM **22**(5), 1234–1248 (2020)

75. Hu, W., Hu, H.: Domain-private factor detachment network for nir-vis face recognition. IEEE TIFS (2022)
76. Hu, W., Huang, Y., Zhang, F., Li, R.: Noise-tolerant paradigm for training face recognition cnns. In: CVPR, pp. 11887–11896 (2019)
77. Huang, B., Wang, Z., Wang, G., Jiang, K., He, Z., Zou, H., Zou, Q.: Masked face recognition datasets and validation. In: ICCVW, pp. 1487–1491 (2021)
78. Huang, B., Wang, Z., Wang, G., Jiang, K., Zeng, K., Han, Z., Tian, X., Yang, Y.: When face recognition meets occlusion: A new benchmark. In: ICASSP, pp. 4240–4244. IEEE (2021)
79. Huang, C., Li, Y., Loy, C.C., Tang, X.: Deep imbalanced learning for face recognition and attribute prediction. IEEE TPAMI **42**(11), 2781–2794 (2019)
80. Huang, F., Yang, M., Lv, X., Wu, F.: Cosmos-loss: a face representation approach with independent supervision. IEEE Access **9**, 36819–36826 (2021)
81. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition (2008)
82. Huang, R., Zhang, S., Li, T., He, R.: Beyond face rotation: global and local perception gan for photorealistic and identity preserving frontal view synthesis. In: ICCV, pp. 2439–2448 (2017)
83. Huang, Y., Wang, Y., Tai, Y., Liu, X., Shen, P., Li, S., Li, J., Huang, F.: Curricularface: adaptive curriculum learning loss for deep face recognition. In: CVPR, pp. 5901–5910 (2020)
84. Huang, Z., Zhang, J., Shan, H.: When age-invariant face recognition meets face age synthesis: a multi-task learning framework. In: CVPR, pp. 7282–7291 (2021)
85. Iqbal, M., Sameem, M.S.I., Naqvi, N., Kanwal, S., Ye, Z.: A deep learning approach for face recognition based on angularly discriminative features. Pattern Recogn. Lett. **128**, 414–419 (2019)
86. Jain, A.K., Li, S.Z.: Handbook of face recognition, vol. 1. Springer (2011)
87. Jeevan, G., Zacharias, G.C., Nair, M.S., Rajan, J.: An empirical study of the impact of masks on face recognition. Pattern Recogn. **122**, 108308 (2022)
88. Jiao, Q., Li, R., Cao, W., Zhong, J., Wu, S., Wong, H.S.: Ddat: dual domain adaptive translation for low-resolution face verification in the wild. Pattern Recognit. 108107 (2021)
89. Kang, B.N., Kim, Y., Jun, B., Kim, D.: Attentional feature-pair relation networks for accurate face recognition. In: ICCV, pp. 5472–5481 (2019)
90. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: CVPR, pp. 4873–4882 (2016)
91. Khalid, S.S., Awais, M., Chan, C.H., Feng, Z., Farooq, A., Akbari, A., Kittler, J.: Npt-loss: a metric loss with implicit mining for face recognition. arXiv preprint arXiv:2103.03503 (2021)
92. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Learning not to learn: training deep neural networks with biased data. In: CVPR, pp. 9012–9020 (2019)
93. Kim, M., Jain, A.K., Liu, X.: Adaface: quality adaptive margin for face recognition. arXiv preprint arXiv:2204.00964 (2022)
94. Klare, B.F., Burge, M.J., Klontz, J.C., Bruegge, R.W.V., Jain, A.K.: Face recognition performance: role of demographic information. IEEE TIFS **7**(6), 1789–1801 (2012)
95. Klare, B.F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Jain, A.K.: Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In: CVPR, pp. 1931–1939 (2015)
96. Komkov, S., Petiushko, A.: Advhat: real-world adversarial attack on arcface face id system. In: ICPR, pp. 819–826 (2021)
97. Kortylewski, A., Egger, B., Schneider, A., Gerig, T., Morel-Forster, A., Vetter, T.: Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In: CVPRW, pp. 0–0 (2019)

98. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. NeurIPS **25** (2012)
99. Lau, C.P., Souri, H., Chellappa, R.: Atfacegan: single face image restoration and recognition from atmospheric turbulence. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), pp. 32–39. IEEE (2020)
100. Le, H.A., Kakadiaris, I.A.: Dblface: domain-based labels for nir-vis heterogeneous face recognition. In: IJCB, pp. 1–10 (2020)
101. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
102. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
103. Lei, Z., Li, S.Z., Chu, R., Zhu, X.: Face recognition with local gabor textons. In: ICB, pp. 49–57. Springer (2007)
104. Lei, Z., Pietikäinen, M., Li, S.Z.: Learning discriminant face descriptor. IEEE TPAMI **36**(2), 289–302 (2013)
105. Li, C., Ge, S., Zhang, D., Li, J.: Look through masks: Towards masked face recognition with de-occlusion distillation. In: ACM MM, pp. 3016–3024 (2020)
106. Li, C., Huang, Y., Huang, W., Qin, F.: Learning features from covariance matrix of gabor wavelet for face recognition under adverse conditions. Pattern Recognit. 108085 (2021)
107. Li, J., Li, Z., Cao, J., Song, X., He, R.: Faceinpainter: high fidelity face adaptation to heterogeneous domains. In: CVPR, pp. 5089–5098 (2021)
108. Li, P., Prieto, L., Mery, D., Flynn, P.J.: On low-resolution face recognition in the wild: comparisons and new techniques. IEEE TIFS **14**(8), 2000–2012 (2019)
109. Li, P., Tu, S., Xu, L.: Deep rival penalized competitive learning for low-resolution face recognition. Neural Netw. (2022)
110. Li, P., Wang, B., Zhang, L.: Virtual fully-connected layer: training a large-scale face recognition dataset with limited computational resources. In: CVPR, pp. 13315–13324 (2021)
111. Li, X., Wang, F., Hu, Q., Leng, C.: Airface: lightweight and efficient model for face recognition. In: ICCVW, pp. 0–0 (2019)
112. Li, Y., Guo, K., Lu, Y., Liu, L.: Cropping and attention based approach for masked face recognition. Appl. Intell. **51**(5), 3012–3025 (2021)
113. Li, Y., Huang, H., Cao, J., He, R., Tan, T.: Disentangled representation learning of makeup portraits in the wild. IJCV **128**(8), 2166–2184 (2020)
114. Li, Y., Song, L., Wu, X., He, R., Tan, T.: Anti-makeup: learning a bi-level adversarial network for makeup-invariant face verification. In: AAAI (2018)
115. Li, Y., Song, L., Wu, X., He, R., Tan, T.: Learning a bi-level adversarial network with global and local perception for makeup-invariant face verification. Pattern Recogn. **90**, 99–108 (2019)
116. Li, Y., Wang, Y.: Defense against adversarial attacks in deep learning. Appl. Sci. **9**(1), 76 (2019)
117. Liao, J., Kot, A., Guha, T., Sanchez, V.: Attention selective network for face synthesis and pose-invariant face recognition. In: ICIP, pp. 748–752 (2020)
118. Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: ICB, pp. 828–837. Springer (2007)
119. Lin, C.H., Huang, W.J., Wu, B.F.: Deep representation alignment network for pose-invariant face recognition. Neurocomputing **464**, 485–496 (2021)
120. Ling, H., Wu, J., Huang, J., Chen, J., Li, P.: Attention-based convolutional neural network for deep face recognition. Multimed. Tools Appl. **79**(9), 5595–5616 (2020)
121. Ling, H., Wu, J., Wu, L., Huang, J., Chen, J., Li, P.: Self residual attention network for deep face recognition. IEEE Access **7**, 55159–55168 (2019)

122. Liu, A., Li, X., Wan, J., Liang, Y., Escalera, S., Escalante, H.J., Madadi, M., Jin, Y., Wu, Z., Yu, X., et al.: Cross-ethnicity face anti-spoofing recognition challenge: a review. IET Biometrics (2020)
123. Liu, A., Tan, Z., Wan, J., Escalera, S., Guo, G., Li, S.Z.: Casia-surf cefa: a benchmark for multi-modal cross-ethnicity face anti-spoofing. In: WACV (2021)
124. Liu, A., Tan, Z., Wan, J., Liang, Y., Lei, Z., Guo, G., Li, S.Z.: Face anti-spoofing via adversarial cross-modality translation. IEEE TIFS **16**, 2759–2772 (2021)
125. Liu, A., Wan, J., Escalera, S., Jair Escalante, H., Tan, Z., Yuan, Q., Wang, K., Lin, C., Guo, G., Guyon, I., et al.: Multi-modal face anti-spoofing attack detection challenge at cvpr2019. In: CVPRW, pp. 0–0 (2019)
126. Liu, A., Zhao, C., Yu, Z., Su, A., Liu, X., Kong, Z., Wan, J., Escalera, S., Escalante, H.J., Lei, Z., et al.: 3d high-fidelity mask face presentation attack detection challenge. In: ICCVW, pp. 814–823 (2021)
127. Liu, A., Zhao, C., Yu, Z., Wan, J., Su, A., Liu, X., Tan, Z., Escalera, S., Xing, J., Liang, Y., et al.: Contrastive context-aware learning for 3d high-fidelity mask face presentation attack detection. arXiv preprint arXiv:2104.06148 (2021)
128. Liu, B., Deng, W., Zhong, Y., Wang, M., Hu, J., Tao, X., Huang, Y.: Fair loss: margin-aware reinforcement learning for deep face recognition. In: ICCV, pp. 10052–10061 (2019)
129. Liu, C., Wechsler, H.: Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE TIP **11**(4), 467–476 (2002)
130. Liu, D., Gao, X., Peng, C., Wang, N., Li, J.: Heterogeneous face interpretable disentangled representation for joint face recognition and synthesis. IEEE TNNLS (2021)
131. Liu, H., Zhu, X., Lei, Z., Cao, D., Li, S.Z.: Fast adapting without forgetting for face recognition. IEEE TCSVT **31**(8), 3093–3104 (2020)
132. Liu, H., Zhu, X., Lei, Z., Li, S.Z.: Adaptiveface: adaptive margin and sampling for face recognition. In: CVPR, pp. 11947–11956 (2019)
133. Liu, J., Li, Q., Liu, M., Wei, T.: Cp-gan: a cross-pose profile face frontalization boosting pose-invariant face recognition. IEEE Access **8**, 198659–198667 (2020)
134. Liu, J., Qin, H., Wu, Y., Guo, J., Liang, D., Xu, K.: Coupleface: relation matters for face recognition distillation. arXiv preprint arXiv:2204.05502 (2022)
135. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: deep hypersphere embedding for face recognition. In: CVPR, pp. 212–220 (2017)
136. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: ICML, p. 7 (2016)
137. Liu, Y., Jourabloo, A., Liu, X.: Learning deep models for face anti-spoofing: binary or auxiliary supervision. In: CVPR (2018)
138. Liu, Y., Stehouwer, J., Liu, X.: On disentangling spoof trace for generic face anti-spoofing. In: ECCV, pp. 406–422 (2020)
139. Liu, Y., et al.: Towards flops-constrained face recognition. In: ICCVW, pp. 0–0 (2019)
140. Low, C.Y., Teoh, A.B.J., Park, J.: Mind-net: a deep mutual information distillation network for realistic low-resolution face recognition. IEEE SPL **28**, 354–358 (2021)
141. Luan, X., Geng, H., Liu, L., Li, W., Zhao, Y., Ren, M.: Geometry structure preserving based gan for multi-pose face frontalization and recognition. IEEE Access **8**, 104676–104687 (2020)
142. Luo, M., Cao, J., Ma, X., Zhang, X., He, R.: Fa-gan: face augmentation gan for deformation-invariant face recognition. IEEE TIFS **16**, 2341–2355 (2021)
143. Lv, J., Shao, X., Xing, J., Cheng, C., Zhou, X.: A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3317–3326 (2017)

144. Ma, Z., Liu, Y., Liu, X., Ma, J., Ren, K.: Lightweight privacy-preserving ensemble classification for face recognition. IEEE Internet Things J. **6**(3), 5778–5790 (2019)
145. Mallat, K., Damer, N., Boutros, F., Kuijper, A., Dugelay, J.L.: Cross-spectrum thermal to visible face recognition based on cascaded image synthesis. In: ICB, pp. 1–8 (2019)
146. Mao, Y., Yi, S., Li, Q., Feng, J., Xu, F., Zhong, S.: A privacy-preserving deep learning approach for face recognition with edge computing. In: Proceedings of USENIX Workshop Hot Topics Edge Computing (HotEdge), pp. 1–6 (2018)
147. Marriott, R.T., Romdhani, S., Chen, L.: A 3d gan for improved large-pose facial recognition. In: CVPR, pp. 13445–13455 (2021)
148. Martindez-Diaz, Y., Luevano, L.S., Mendez-Vazquez, H., Nicolas-Diaz, M., Chang, L., Gonzalez-Mendoza, M.: Shufflefacenet: a lightweight face architecture for efficient and highly-accurate face recognition. In: ICCVW (2019)
149. Martínez-Díaz, Y., Méndez-Vázquez, H., Luevano, L.S., Chang, L., Gonzalez-Mendoza, M.: Lightweight low-resolution face recognition for surveillance applications. In: ICPR, pp. 5421–5428 (2021)
150. Martínez-Díaz, Y., Méndez-Vázquez, H., Luevano, L.S., Nicolás-Díaz, M., Chang, L., Gonzalez-Mendoza, M.: Towards accurate and lightweight masked face recognition: an experimental evaluation. IEEE Access (2021)
151. Maze, B., Adams, J., Duncan, J.A., Kalka, N., Miller, T., Otto, C., Jain, A.K., Niggel, W.T., Anderson, J., Cheney, J., et al.: Iarpa janus benchmark-c: Face dataset and protocol. In: ICB, pp. 158–165 (2018)
152. Meng, Q., Xu, X., Wang, X., Qian, Y., Qin, Y., Wang, Z., Zhao, C., Zhou, F., Lei, Z.: Poseface: Pose-invariant features and pose-adaptive loss for face recognition. arXiv preprint arXiv:2107.11721 (2021)
153. Meng, Q., Zhao, S., Huang, Z., Zhou, F.: Magface: a universal representation for face recognition and quality assessment. In: CVPR, pp. 14225–14234 (2021)
154. Meng, Q., Zhou, F., Ren, H., Feng, T., Liu, G., Lin, Y.: Improving federated learning face recognition via privacy-agnostic clusters. arXiv preprint arXiv:2201.12467 (2022)
155. Miao, C., Tan, Z., Chu, Q., Yu, N., Guo, G.: Hierarchical frequency-assisted interactive networks for face manipulation detection. IEEE Trans. Inf. Forensics Secur. **17**, 3008–3021 (2022)
156. Mishra, S., Majumdar, P., Singh, R., Vatsa, M.: Indian masked faces in the wild dataset. In: ICIP, pp. 884–888 (2021)
157. Mokhayeri, F., Granger, E., Bilodeau, G.A.: Domain-specific face synthesis for video face recognition from a single sample per person. IEEE TIFS **14**(3), 757–772 (2018)
158. Mokhayeri, F., Kamali, K., Granger, E.: Cross-domain face synthesis using a controllable gan. In: WACV, pp. 252–260 (2020)
159. Montero, D., Nieto, M., Leskovsky, P., Aginako, N.: Boosting masked face recognition with multi-task arcface. arXiv preprint arXiv:2104.09874 (2021)
160. Nagpal, S., Singh, M., Singh, R., Vatsa, M.: Discriminative shared transform learning for sketch to image matching. Pattern Recogn. **114**, 107815 (2021)
161. Najibi, M., Singh, B., Davis, L.S.: Fa-rpn: floating region proposals for face detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7723–7732 (2019)
162. Nech, A., Kemelmacher-Shlizerman, I.: Level playing field for million scale face recognition. In: CVPR, pp. 7044–7053 (2017)
163. Neto, P.C., Boutros, F., Pinto, J.R., Darner, N., Sequeira, A.F., Cardoso, J.S.: Focusface: multi-task contrastive learning for masked face recognition. In: IEEE FG, pp. 01–08 (2021)
164. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision, pp. 483–499. Springer (2016)

165. Oinar, C., Le, B.M., Woo, S.S.: Kappaface: adaptive additive angular margin loss for deep face recognition. arXiv preprint arXiv:2201.07394 (2022)
166. Osahor, U., Kazemi, H., Dabouei, A., Nasrabadi, N.: Quality guided sketch-to-photo image synthesis. In: CVPRW, pp. 820–821 (2020)
167. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC (2015)
168. Peng, C., Wang, N., Li, J., Gao, X.: Dlface: deep local descriptor for cross-modality face recognition. Pattern Recogn. **90**, 161–171 (2019)
169. Peng, C., Wang, N., Li, J., Gao, X.: Re-ranking high-dimensional deep local representation for nir-vis face recognition. IEEE TIP **28**(9), 4553–4565 (2019)
170. Peng, X., Yu, X., Sohn, K., Metaxas, D.N., Chandraker, M.: Reconstruction-based disentanglement for pose-invariant face recognition. In: ICCV, pp. 1623–1632 (2017)
171. Qi, C., Su, F.: Contrastive-center loss for deep neural networks. In: ICIP, pp. 2851–2855 (2017)
172. Qi, D., Hu, K., Tan, W., Yao, Q., Liu, J.: Balanced masked and standard face recognition. In: ICCVW, pp. 1497–1502 (2021)
173. Qian, H., Zhang, P., Ji, S., Cao, S., Xu, Y.: Improving representation consistency with pairwise loss for masked face recognition. In: ICCVW, pp. 1462–1467 (2021)
174. Rao, Y., Lin, J., Lu, J., Zhou, J.: Learning discriminative aggregation network for video-based face recognition. In: ICCV, pp. 3781–3790 (2017)
175. Rao, Y., Lu, J., Zhou, J.: Learning discriminative aggregation network for video-based face recognition and person re-identification. IJCV **127**(6), 701–718 (2019)
176. Rivero-Hernández, J., Morales-González, A., Denis, L.G., Méndez-Vázquez, H.: Ordered weighted aggregation networks for video face recognition. Pattern Recogn. Lett. **146**, 237–243 (2021)
177. Robbins, W., Boult, T.E.: On the effect of atmospheric turbulence in the feature space of deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1618–1626 (2022)
178. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: CVPR, pp. 815–823 (2015)
179. Sengupta, S., Chen, J.C., Castillo, C., Patel, V.M., Chellappa, R., Jacobs, D.W.: Frontal to profile face verification in the wild. In: WACV, pp. 1–9 (2016)
180. Serna, I., Morales, A., Fierrez, J., Obradovich, N.: Sensitive loss: improving accuracy and fairness of face representations with discrimination-aware deep learning. Artif. Intell. **305**, 103682 (2022)
181. Shao, R., Lan, X., Li, J., Yuen, P.C.: Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In: CVPR (2019)
182. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Adversarial generative nets: neural network attacks on state-of-the-art face recognition. arXiv preprint arXiv:1801.00349, **2**(3) (2017)
183. Shi, Y., Jain, A.K.: Docface: Matching id document photos to selfies. In: IEEE BTAS, pp. 1–8 (2018)
184. Shi, Y., Jain, A.K.: Docface+: Id document to selfie matching. IEEE TBIOM **1**(1), 56–67 (2019)
185. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
186. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. NeurIPS **29** (2016)
187. Song, L., Zhang, M., Wu, X., He, R.: Adversarial discriminative heterogeneous face recognition. In: AAAI (2018)
188. Sun, J., Yang, W., Xue, J.H., Liao, Q.: An equalized margin loss for face recognition. IEEE TMM **22**(11), 2833–2843 (2020)

189. Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y.: Circle loss: a unified perspective of pair similarity optimization. In: CVPR, pp. 6398–6407 (2020)
190. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1891–1898 (2014)
191. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR, pp. 1–9 (2015)
192. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: CVPR, pp. 1701–1708 (2014)
193. Tan, Z., Wan, J., Lei, Z., Zhi, R., Guo, G., Li, S.Z.: Efficient group-n encoding and decoding for facial age estimation. IEEE TPAMI **40**(11), 2610–2623 (2018)
194. Tan, Z., Yang, Y., Wan, J., Guo, G., Li, S.Z.: Deeply-learned hybrid representations for facial age estimation. In: IJCAI, pp. 3548–3554 (2019)
195. Tong, L., Chen, Z., Ni, J., Cheng, W., Song, D., Chen, H., Vorobeychik, Y.: Facesec: A fine-grained robustness evaluation framework for face recognition systems. In: CVPR, pp. 13254–13263 (2021)
196. Tran, L., Yin, X., Liu, X.: Disentangled representation learning gan for pose-invariant face recognition. In: CVPR, pp. 1415–1424 (2017)
197. Trigueros, D.S., Meng, L., Hartnett, M.: Generating photo-realistic training data to improve face recognition accuracy. Neural Netw. **134**, 86–94 (2021)
198. Tsai, E.J., Yeh, W.C.: Pam: pose attention module for pose-invariant face recognition. arXiv preprint arXiv:2111.11940 (2021)
199. Wang, C., Fang, H., Zhong, Y., Deng, W.: Mlfw: a database for face recognition on masked faces. arXiv preprint arXiv:2109.05804 (2021)
200. Wang, F., Chen, L., Li, C., Huang, S., Chen, Y., Qian, C., Loy, C.C.: The devil of face recognition is in the noise. In: ECCV, pp. 765–780 (2018)
201. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE SPL **25**(7), 926–930 (2018)
202. Wang, F., Xiang, X., Cheng, J., Yuille, A.L.: Normface: L2 hypersphere embedding for face verification. In: ACM MM, pp. 1041–1049 (2017)
203. Wang, G., Han, H., Shan, S., Chen, X.: Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In: CVPR, pp. 6678–6687 (2020)
204. Wang, H., Gong, D., Li, Z., Liu, W.: Decorrelated adversarial learning for age-invariant face recognition. In: CVPR, pp. 3527–3536 (2019)
205. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: large margin cosine loss for deep face recognition. In: CVPR, pp. 5265–5274 (2018)
206. Wang, J., Liu, Y., Hu, Y., Shi, H., Mei, T.: Facex-zoo: a pytorch toolbox for face recognition. In: ACM MM, pp. 3779–3782 (2021)
207. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3349–3364 (2020)
208. Wang, K., Wang, S., Yang, J., Wang, X., Sun, B., Li, H., You, Y.: Mask aware network for masked face recognition in the wild. In: ICCVW, pp. 1456–1461 (2021)
209. Wang, M., Deng, W.: Mitigating bias in face recognition using skewness-aware reinforcement learning. In: CVPR, pp. 9322–9331 (2020)
210. Wang, M., Deng, W.: Deep face recognition: a survey. Neurocomputing **429**, 215–244 (2021)
211. Wang, Q., Guo, G.: Aan-face: attention augmented networks for face recognition. IEEE TIP **30**, 7636–7648 (2021)

212. Wang, Q., Guo, G.: Dsa-face: diverse and sparse attentions for face recognition robust to pose variation and occlusion. IEEE TIFS **16**, 4534–4543 (2021)
213. Wang, Q., Wu, T., Zheng, H., Guo, G.: Hierarchical pyramid diverse attention networks for face recognition. In: CVPR, pp. 8326–8335 (2020)
214. Wang, W., Fu, Y., Qian, X., Jiang, Y.G., Tian, Q., Xue, X.: Fm2u-net: face morphological multi-branch network for makeup-invariant face verification. In: CVPR, pp. 5730–5740 (2020)
215. Wang, W., Zhao, Z., Zhang, H., Wang, Z., Su, F.: Maskout: a data augmentation method for masked face recognition. In: ICCVW, pp. 1450–1455 (2021)
216. Wang, X., Wang, S., Chi, C., Zhang, S., Mei, T.: Loss function search for face recognition. In: ICML, pp. 10029–10038 (2020)
217. Wang, X., Wang, S., Liang, Y., Gu, L., Lei, Z.: Rvface: reliable vector guided softmax loss for face recognition. IEEE TIP **31**, 2337–2351 (2022)
218. Wang, X., Wang, S., Wang, J., Shi, H., Mei, T.: Co-mining: deep face recognition with noisy labels. In: ICCV, pp. 9358–9367 (2019)
219. Wang, X., Zhang, S., Wang, S., Fu, T., Shi, H., Mei, T.: Mis-classified vector guided softmax loss for face recognition. In: AAAI, pp. 12241–12248 (2020)
220. Wang, Y., Gong, D., Zhou, Z., Ji, X., Wang, H., Li, Z., Liu, W., Zhang, T.: Orthogonal deep features decomposition for age-invariant face recognition. In: ECCV, pp. 738–753 (2018)
221. Wang, Z., He, K., Fu, Y., Feng, R., Jiang, Y.G., Xue, X.: Multi-task deep neural network for joint face recognition and facial attribute prediction. In: ICMR, pp. 365–374 (2017)
222. Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., Yi, P., Jiang, K., Wang, N., Pei, Y., et al.: Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093 (2020)
223. Wang, Z., Wang, Q., Deng, W., Guo, G.: Learning multi-granularity temporal characteristics for face anti-spoofing. IEEE TIFS **17**, 1254–1269 (2022)
224. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV, pp. 499–515 (2016)
225. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A comprehensive study on center loss for deep face recognition. IJCV **127**(6), 668–683 (2019)
226. Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A.K., Duncan, J.A., Allen, K., et al.: Iarpa janus benchmark-b face dataset. In: CVPRW, pp. 90–98 (2017)
227. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE TPAMI **31**(2), 210–227 (2008)
228. Wu, W., Qian, C., Yang, S., Wang, Q., Cai, Y., Zhou, Q.: Look at boundary: a boundary-aware face alignment algorithm. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2129–2138 (2018)
229. Wu, X., He, R., Sun, Z., Tan, T.: A light cnn for deep face representation with noisy labels. IEEE TIFS **13**(11), 2884–2896 (2018)
230. Wu, X., Huang, H., Patel, V.M., He, R., Sun, Z.: Disentangled variational representation for heterogeneous face recognition. In: AAAI, pp. 9005–9012 (2019)
231. Xu, Y., Raja, K., Ramachandra, R., Busch, C.: Adversarial attacks on face recognition systems. In: Handbook of Digital Face Manipulation and Detection, pp. 139–161. Springer, Cham (2022)
232. Xue, F., Tan, Z., Zhu, Y., Ma, Z., Guo, G.: Coarse-to-fine cascaded networks with smooth predicting for video facial expression recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2412–2418 (2022)
233. Xue, F., Wang, Q., Guo, G.: Transfer: learning relation-aware facial expression representations with transformers. In: ICCV, pp. 3601–3610 (2021)

234. Yan, C., Meng, L., Li, L., Zhang, J., Wang, Z., Yin, J., Zhang, J., Sun, Y., Zheng, B.: Age-invariant face recognition by multi-feature fusionand decomposition with self-attention. ACM TOMM **18**(1s), 1–18 (2022)
235. Yan, M., Zhao, M., Xu, Z., Zhang, Q., Wang, G., Su, Z.: Vargfacenet: an efficient variable group convolutional neural network for lightweight face recognition. In: ICCVW, pp. 0–0 (2019)
236. Yang, J., Lei, Z., Li, S.Z.: Learn convolutional neural network for face anti-spoofing. arXiv (2014)
237. Yang, J., Ren, P., Zhang, D., Chen, D., Wen, F., Li, H., Hua, G.: Neural aggregation network for video face recognition. In: CVPR, pp. 4362–4371 (2017)
238. Yang, Y., Liao, S., Lei, Z., Li, S.Z.: Large scale similarity learning using similar pairs for person verification. In: AAAI (2016)
239. Yang, Z., Liang, J., Fu, C., Luo, M., Zhang, X.Y.: Heterogeneous face recognition via face synthesis with identity-attribute disentanglement. IEEE TIFS (2022)
240. Yasarla, R., Patel, V.M.: Learning to restore images degraded by atmospheric turbulence using uncertainty. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 1694–1698. IEEE (2021)
241. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
242. Yin, X., Liu, X.: Multi-task convolutional neural network for pose-invariant face recognition. IEEE TIP **27**(2), 964–975 (2017)
243. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Towards large-pose face frontalization in the wild. In: ICCV, pp. 3990–3999 (2017)
244. Yu, H., Fan, Y., Chen, K., Yan, H., Lu, X., Liu, J., Xie, D.: Unknown identity rejection loss: utilizing unlabeled data for face recognition. In: ICCVW, pp. 0–0 (2019)
245. Yu, J., Cao, J., Li, Y., Jia, X., He, R.: Pose-preserving cross spectral face hallucination. In: IJCAI, pp. 1018–1024 (2019)
246. Yu, J., Hao, X., Cui, Z., He, P., Liu, T.: Boosting fairness for masked face recognition. In: ICCVW, pp. 1531–1540 (2021)
247. Yu, J., Jing, L.: A joint multi-task cnn for cross-age face recognition. In: ICIP, pp. 2411–2415 (2018)
248. Yuan, H., Chu, Q., Zhu, F., Zhao, R., Liu, B., Yu, N.: Efficient open-set adversarial attacks on deep face recognition. In: ICME, pp. 1–6 (2021)
249. Yucer, S., Akçay, S., Al-Moubayed, N., Breckon, T.P.: Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation. In: CVPRW, pp. 18–19 (2020)
250. Zhang, H., Wang, Z., Hou, J.: Makeup removal for face verification based upon deep learning. In: ICSIP, pp. 446–450 (2021)
251. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. **23**(10), 1499–1503 (2016)
252. Zhang, K.Y., Yao, T., Zhang, J., Tai, Y., Ding, S., Li, J., Huang, F., Song, H., Ma, L.: Face anti-spoofing via disentangled representation learning. In: ECCV, pp. 641–657 (2020)
253. Zhang, L., Yang, M., Feng, X.: Sparse representation or collaborative representation: which helps face recognition? In: ICCV, pp. 471–478 (2011)
254. Zhang, S., Chi, C., Lei, Z., Li, S.Z.: Refineface: refinement neural network for high performance face detection. IEEE Trans. Pattern Anal. Mach. Intell. **43**(11), 4008–4020 (2020)
255. Zhang, S., Liu, A., Wan, J., Liang, Y., Li, S.Z.: Casia-surf: a large-scale multi-modal benchmark for face anti-spoofing. TBIOM (2019)
256. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z.: Faceboxes: a cpu real-time face detector with high accuracy. In: IJCB, pp. 1–9 (2017)

257. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z.: S3fd: single shot scale-invariant face detector. In: ICCV, pp. 192–201 (2017)
258. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: ICCV, pp. 5409–5418 (2017)
259. Zhang, X., Zhao, R., Qiao, Y., Wang, X., Li, H.: Adacos: adaptively scaling cosine logits for effectively learning deep face representations. In: CVPR, pp. 10823–10832 (2019)
260. Zhang, Y., Deng, W., Wang, M., Hu, J., Li, X., Zhao, D., Wen, D.: Global-local gcn: large-scale label noise cleansing for face recognition. In: CVPR, pp. 7731–7740 (2020)
261. Zhang, Z., Chen, Y., Yang, W., Wang, G., Liao, Q.: Pose-invariant face recognition via adaptive angular distillation. In: AAAI (2022)
262. Zhao, H., Shi, Y., Tong, X., Wen, J., Ying, X., Zha, H.: G-fan: graph-based feature aggregation network for video face recognition. In: ICPR, pp. 1672–1678 (2021)
263. Zhao, J., Cheng, Y., Cheng, Y., Yang, Y., Zhao, F., Li, J., Liu, H., Yan, S., Feng, J.: Look across elapse: disentangled representation learning and photorealistic cross-age face synthesis for age-invariant face recognition. In: AAAI, pp. 9251–9258 (2019)
264. Zhao, J., Cheng, Y., Xu, Y., Xiong, L., Li, J., Zhao, F., Jayashree, K., Pranata, S., Shen, S., Xing, J., et al.: Towards pose invariant face recognition in the wild. In: CVPR, pp. 2207–2216 (2018)
265. Zhao, J., Xing, J., Xiong, L., Yan, S., Feng, J.: Recognizing profile faces by imagining frontal view. IJCV **128**(2), 460–478 (2020)
266. Zhao, J., Xiong, L., Cheng, Y., Cheng, Y., Li, J., Zhou, L., Xu, Y., Karlekar, J., Pranata, S., Shen, S., et al.: 3d-aided deep pose-invariant face recognition. In: IJCAI, p. 11 (2018)
267. Zhao, J., Yan, S., Feng, J.: Towards age-invariant face recognition. IEEE TPAMI (2020)
268. Zhao, S., Li, J., Wang, J.: Disentangled representation learning and residual gan for age-invariant face verification. Pattern Recogn. **100**, 107097 (2020)
269. Zheng, T., Deng, W.: Cross-pose lfw: a database for studying cross-pose face recognition in unconstrained environments. Beijing University of Posts and Telecommunications, Tech. Rep **5**, 7 (2018)
270. Zheng, T., Deng, W., Hu, J.: Cross-age lfw: a database for studying cross-age face recognition in unconstrained environments. arXiv preprint arXiv:1708.08197 (2017)
271. Zheng, W., Yan, L., Wang, F.Y., Gou, C.: Learning from the web: webly supervised meta-learning for masked face recognition. In: CVPR, pp. 4304–4313 (2021)
272. Zhong, Y., Deng, W.: Towards transferable adversarial attack against deep face recognition. IEEE TIFS **16**, 1452–1466 (2020)
273. Zhong, Y., Deng, W.: Face transformer for recognition. arXiv preprint arXiv:2103.14803 (2021)
274. Zhong, Y., Deng, W., Hu, J., Zhao, D., Li, X., Wen, D.: Sface: sigmoid-constrained hypersphere loss for robust face recognition. IEEE TIP **30**, 2587–2598 (2021)
275. Zhong, Y., Deng, W., Wang, M., Hu, J., Peng, J., Tao, X., Huang, Y.: Unequal-training for deep face recognition with long-tailed noisy data. In: CVPR, pp. 7812–7821 (2019)
276. Zhou, E., Cao, Z., Yin, Q.: Naive-deep face recognition: Touching the limit of lfw benchmark or not? arXiv preprint arXiv:1501.04690 (2015)
277. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: a 3d solution. In: CVPR, pp. 146–155 (2016)
278. Zhu, X., Liu, H., Lei, Z., Shi, H., Yang, F., Yi, D., Qi, G., Li, S.Z.: Large-scale bisample learning on id versus spot face recognition. IJCV **127**(6), 684–700 (2019)
279. Zhu, Z., Huang, G., Deng, J., Ye, Y., Huang, J., Chen, X., Zhu, J., Yang, T., Guo, J., Lu, J., et al.: Masked face recognition challenge: the webface260m track report. arXiv preprint arXiv:2108.07189 (2021)

280. Zhu, Z., Huang, G., Deng, J., Ye, Y., Huang, J., Chen, X., Zhu, J., Yang, T., Lu, J., Du, D., et al.: Webface260m: a benchmark unveiling the power of million-scale deep face recognition. In: CVPR, pp. 10492–10502 (2021)
281. Zhu, Z.A., Lu, Y.Z., Chiang, C.K.: Generating adversarial examples by makeup attacks on face recognition. In: ICIP, pp. 2516–2520 (2019)
282. Zichang, T., Ajian, L., Jun, W., Hao, L., Zhen, L., Guodong, G., Stan Z., L.: Cross-batch hard example mining with pseudo large batch for id vs. spot face recognition. IEEE TIP (2022)

Convolutional Neural Networks and Architectures

2

Xiangyu Zhang

2.1 Convolutional Neural Network Basics

This chapter briefly introduces *Convolutional Neural Networks (CNNs)*. One of the first CNNs is proposed in [41] (known as *LeNet*) to deal with handwriting recognition task. After that, CNN becomes the most popular deep neural network model to process visual data, including images and videos. Until now, a lot of modern fundamental computer vision systems, e.g., image classification and face recognition, are usually built upon convolutional networks.

2.1.1 Motivation: Idea Behind Convolutional Layer

Let us consider how to design a neural network layer to process a digital image. Suppose the image data is represented in a three-channel matrix: $X \in \mathbb{R}^{3\times H\times W}$, where H and W are the height and the width of the image, respectively. To fit the data with the simplest neural network (i.e., perceptron), one straightforward way is to *flatten* the tensor into a vector $vec(X)$. Hence, the i-th output of the perceptron is

$$\begin{aligned} y_i &= \mathbf{w}_i^\top vec(X) \\ z_i &= \sigma(y_i), \end{aligned} \tag{2.1}$$

where $\mathbf{w}_i$ is the weight associated with the i-th neuron and $\sigma(\cdot)$ is the (nonlinear) activation function. For simplicity, we omit the bias term unless otherwise specified here and in the

X. Zhang (✉)
MEGVII Technology, No. 2 Kexueyuan South Road, Haidian District, Beijing, China
e-mail: zhangxiangyu@megvii.com

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_2

next text. We name the above perceptron **fully connected layer** (FC layer), as each of the outputs is connected to each component of the input via an independent weight.

Deep neural networks typically involve a lot of layers. When directly applying an FC layer to the input image, a drawback comes up: the number of parameters could greatly blow up. For example, suppose the image size is $3 \times 1000 \times 1000$; in order to maintain the representative capacity of the hidden layer, we assume the number of neurons is the same as the input dimensions. Hence, the total number of the parameters is as many as 9×10^{12}. The same issue occurs in subsequent layers unless the number of hidden neurons is greatly reduced, however, large parameter reduction may cause too much information loss, especially for the early layers in the neural network.

To save parameters, we have to investigate some unique properties of image data. One important property is *locality*, whose insight is that in most visual tasks each semantic concept only relates to a local patch rather than the whole image. Figure 2.1a illustrates the property: the red neuron responsible for dog detection only needs to connect the region in the red box; so does the "cat" neuron (green). Inspired by the locality property, FC layer can be simplified into **locally-connected layer** (or local layer), in which each output is only related to a specified window of the input rather than the whole. Local layer is formulated as follows:

$$y_{o,i,j} = \sum_{c} \sum_{k=0}^{K-1} \sum_{l=0}^{K-1} w_{o,c,k,l}^{i,j} \times x_{c,i+k,j+l}. \tag{2.2}$$

Note that, in the above formulation, we reorganize the outputs into a tensor $Y \in \mathbb{R}^{O \times H \times W}$, and $y_{o,i,j}$ is the corresponding component. Nonlinear activation function is omitted here for simplicity. K is named *filter size*, which controls the window size each output neuron is connected to. Comparing Eq. 2.2 with Eq. 2.1, it can be inferred that a local layer requires

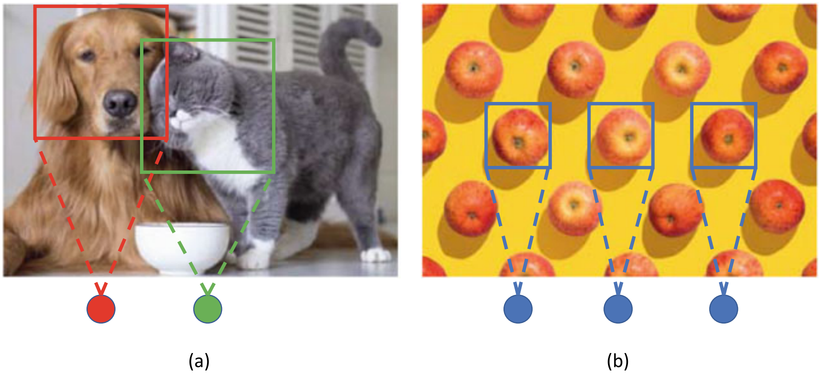


Fig. 2.1 **a** *Locality* property. The "dog" neuron (red) and the "cat" neuron (green) only relate to a local patch of the image, respectively, rather than the whole. **b** *Translational equivariant* property. "Apple" neurons (blue) at different locations share the same weight, since location does not matter in recognizing apples

far less parameters than that of an FC layer if the filter size is small. For example, given the input and output channels $C = O = 3$ and $H = W = 1000$, $K = 10$, the total number of parameters in the local layer is 9×10^8, much smaller than 9×10^{12} in a fully connected layer.

Can we further reduce the number of parameters in a local layer? We further explore another important property: *translational equivariance*. In signal processing, translational equivariance means that when the input sequence shifts by an offset, the output sequence thus shifts in the same way. Interestingly, many visual tasks share such property. Figure 2.1b gives an example: for "apple detection" task, suppose a neuron in a local layer captures an apple in the left box; when the apple moves to the right, it will be captured by another neuron. We definitely want the two neurons output the same activation value, since the location of the apple should not affect the recognition result. In other words, the "apple detection" task requires the network to be translational equivariant.[1]

How to make a local layer translational equivariant? One simple but effective way is to make neurons at different locations share the same weight, or formally, for any i, j, i', j', let $w^{i,j}_{o,c,k,l} = w^{i',j'}_{o,c,k,l} \triangleq w_{o,c,k,l}$ in Eq. 2.2. Figure 2.1b illustrates the idea: if all the blue neurons share the same parameters, no matter where the apple appears, the corresponding neuron will produce the same output value, hence the layer becomes translational equivariant. Therefore, Eq. 2.2 can be simplified as

$$y_{o,i,j} = \sum_{c} \sum_{k=0}^{K-1} \sum_{l=0}^{K-1} w_{o,c,k,l} \times x_{c,i+k,j+l}. \tag{2.3}$$

We say that Eq. 2.3 defines a **convolutional layer**, or more precisely, named **2D-convolutional layer**. The layer got its name because if we let the number of input and output channels to be one ($O = C = 1$), Eq. 2.3 satisfies the definition of 2D convolution in mathematics.[2] Thanks to weight sharing, the convolutional layer further reduces the number of parameters from the local layer. For example, we still let $C = O = 3$, $H = W = 1000$, and $K = 10$, the total number of weights significantly decreases from 9×10^8 to 9×10^2. In practice, we usually adopt more output channels to maintain the representative capability, e.g., 100 or up to 1000. Nevertheless, convolutional layers are still much more efficient in parameters than the counterpart local layers.

In summary, the convolutional layer is proposed to overcome the inefficiency of the fully connected layer on image data. The reason why the convolutional layer requires far less parameters is that it utilizes *locality* and *translational equivariance* properties in many

[1] Another class of tasks, e.g., image classification, requires the network to be translational invariant, i.e., invariant to the shift of the input. It can be simply done by adding a global pooling layer [62] on top of the network if it is already translational equivariant.

[2] Strictly, Eq. 2.3 defines a *correlation* operation. Nevertheless, literature usually does not distinguish convolution from correlation in neural networks since they can be simply bridged by vertically and horizontally flipping the weights.

visual tasks, termed *inductive bias* of the model. In deep learning methods, introducing a proper inductive bias is an important methodology, which not only helps to save parameters but also eases the optimization and gains more generalization capability.

! Attention

Keep in mind the *No Free Lunch* principle. Although widely used in computer vision world, CNNs could be unexpectedly inefficient or not proper if locality or translational equivariance properties break, unless some workarounds are introduced. For example, [42] shows an extreme case that CNN is even not as good as a simple multi-layer perceptron (MLP) if they shuffle the pixels in each image with a fixed pattern before training, since the locality property thoroughly breaks. In general, if the target task prefers global or long-term interaction properties instead of locality properties, consider introducing deeper architectures [25, 60], large convolutional kernels [5, 14, 54], non-local structures [69], transformers [16, 66] or MLP-Mixer [65] into your CNN models.

Accordingly, some tasks may not satisfy translational equivariance, for instance, in face recognition, the input face image is usually aligned according to the landmark label, therefore the absolute position of the eye or mouth matters to the recognition result. Some early face recognition networks, e.g., [64], employ local layers on top of convolutional backbones. Recent research [14, 34] suggests large padding can help CNNs learn absolute positions; CoordConv [46] also works in the similar way, so does vision transformer [16] with absolute positional embeddings. All those methods help CNNs to deal with those tasks where translational equivariance does not strictly satisfy.

2.1.2 Convolutional Layer: Concepts and Variants

In this subsection, we define convolution in formal. Given an input tensor $X \in \mathbb{R}^{C \times I_h \times I_w}$, where C is the number of input channels and $I_h \times I_w$ is the spatial size. A convolutional layer can be viewed as a *linear mapping* that derives the output tensor $Y \in \mathbb{R}^{O \times F_h \times F_w}$. The tensors X and Y are also named *input feature map* and *output feature map* accordingly. Convolutional operation is generally noted as:

$$Y = X * w, \tag{2.4}$$

where the tensor $w \in \mathbb{R}^{O \times C \times K_h \times K_w}$ is the weights of the convolution, named *filter* or *kernel*; and $K_h \times K_w$ is the *kernel size*.

Figure 2.2 intuitively shows how a convolutional layer operates. To compute the o-th output channel, we first slice out the corresponding kernel $w_o \triangleq w_{o,:,:,:}$, viewed as a 3D-volume (see the green or orange box in Fig. 2.2, left). Then, we allow the kernel volume to slide from the left-top corner to the right-bottom corner in the input feature map. Each

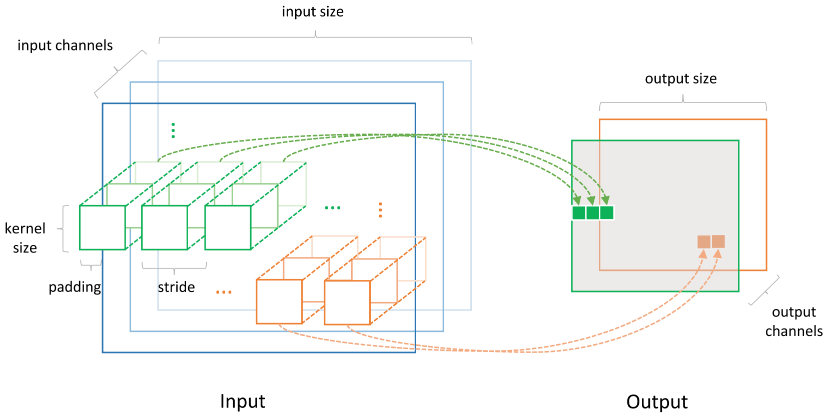


Fig. 2.2 Illustration on the convolutional layer operation

time the kernel moves to a certain position, we compute the dot product between the kernel weights and the corresponding elements of the feature map, which derives the output value at the position. The computation can also be interpreted as a kind of *template matching*: suppose we have O templates (namely kernels); we compute the similarity between each template and each local patch of the input image, thus generating O similarity score maps (i.e., output feature map).

The following formulation describes the computation of each output exactly:

$$y_{o,i,j} = \sum_{c=0}^{C-1} \sum_{k=0}^{K_h-1} \sum_{l=0}^{K_w-1} x_{c,i\times s_h-p_h+k,j\times s_w-p_w+l} \times w_{o,c,k,l}. \tag{2.5}$$

Readers may have already found that the above equation is an extension to Eq. 2.3. Two additional parameters control the behavior of the sliding convolutional kernel[3] (illustrated in Fig. 2.2):

Stride (s) It indicates the step length where the filter moves along the vertical or horizontal axis. Letting $s = 1$ derives a roughly equal-sized output feature map from the input, while $s = 2$ could reduce the output size by half. Besides, if stride is smaller than kernel size K, we say the convolutional operation is *overlapped*; otherwise, it is *non-overlapped*.

Padding (p) It controls the behavior of convolution around the edge or corner. If $p > 0$, the input feature map will first be padded with zeros along the spatial dimensions before computation, whose padding width is specified by p.

[3] In this chapter, the subscript form, i.e., s_h or s_w indicates the index along height or width direction, respectively.

It can be derived that the relation between input size (I), kernel size (K), stride (s), padding (p), and output size (F) is

$$F = \lfloor (I - K + 2p)/s \rfloor + 1 \tag{2.6}$$

Equation 2.6 is very useful in designing convolutional networks. For example, if the spatial size of the feature map must remain unchanged after convolution, one could use odd-sized kernel (e.g., $K = 3$) as well as let $p = \lfloor K/2 \rfloor$ and $s = 1$.

Complexity analysis. Literature often uses *FLOPs* (i.e., the number of floating-point operations) and *the number of parameters* to measure the computation cost and the storage footprint of a neural network, respectively. For a typical convolutional layer, they can be estimated as follows[4]:

$$\begin{aligned} FLOPs &= F_h \times F_w \times K_h \times K_w \times C \times O \\ Parameters &= K_h \times K_w \times C \times O. \end{aligned} \tag{2.7}$$

It is easy to see that the number of parameters in a convolutional layer has nothing to do with the feature map size, but the computation does. That is why convolution is very parameter efficient compared with the counterpart fully connected layer or local layer. Instead, shrinking the *output* feature size (e.g., setting $s > 1$ according to Eq. 2.6) may greatly save computations. In most convolutional networks, convolutional layers consume most of the FLOPs and parameters.

! Attention

A common misunderstanding on FLOPs is that a network with smaller FLOPs must run faster. It is actually not always true because the running speed also heavily depends on the hardware architecture and implementation, especially memory access cost and degree of parallelism [53]. Similarly, the number of parameters is not the only factor that affects the running memory footprint either, because the feature maps between layers also consume a lot of memory.

Variants of Convolutional Layers

Point-wise convolution. Namely, the kernel size is 1 × 1, which is introduced in [45] for the first time. Point-wise convolution transforms a feature map without spatial interaction. Besides, point-wise convolution is often employed to increase or decrease the number of channels in the feature map. In some vision transformers [16], the so-named "feed-forward network" (FFN) is actually composed of point-wise convolutions too.

[4] For convolutions under special configurations, there may exist clever implementations such as FFT and Winograd [39] involving fewer FLOPs.

Group convolution. As illustrated in Fig. 2.2, each filter can be viewed as a 3D-volume, whose number of channels (C) equals to that of input feature map. If the channel number is big, the complexity will also increase. One solution is to introduce *group convolution* [38, 72], which divides the input channels as well as the filters into g groups, then computes convolutions within each group accordingly. Therefore, the channel number of each kernel decreases to C/g so that the complexity also drops by a factor of g. As an extreme case, **depth-wise convolution** [8] is a special group convolution where $C = O = g$, hence i-th output channel of a depth-wise convolution only depends on i-th input channel. Depth-wise convolution is widely used in lightweight CNNs [27, 80].

Factorized convolution. According to Eq. 2.7, the complexity of a convolution layer rapidly increases if the kernel size and input/output channels scale up. One common way to decrease the complexity is to decompose a large convolution into several small convolutions. There are two major factorization schemes: in serial or in parallel:

$$\begin{aligned} X * w &\approx X * w^{(1)} * w^{(2)} * \ldots \\ &\approx X * w^{(1)} + X * w^{(2)} + \ldots \end{aligned}$$

The decomposition method varies in different works, for example, low-rank decomposition [36, 81], depth-wise separable convolution [8], CP-decomposition [40], criss-cross decomposition [54] and etc. Notice that several works add nonlinearities between the factorized components [27, 62, 63]; even though the decomposition helps to reduce the complexity, those architectures cannot be simply viewed as a factorization of a big convolution.

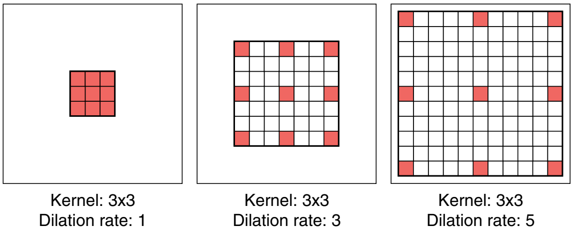
Dilated convolution. Some applications (e.g., semantic segmentation) may prefer big convolution kernels to obtain large *receptive field*.[5] However, according to Eq. 2.7, enlarging the kernel size brings a lot of complexity. One workaround is to introduce *dilated convolution* [5, 75] (also known as *atrous convolution*). Figure 2.3 illustrates how it works. Dilated convolution can be viewed as a kind of sparse convolution (or convolution with "holes"), whose sparsity is specified by *dilation rate* d. For example, if the kernel size is 3×3 and $d = 2$, it behaves like a 5 convolution, however, only $9/25$ elements in the filter could be non-zero. If $d = 1$, dilated convolution degenerates to normal convolution. The computation follows the formulation:

$$y_{o,i,j} = \sum_{c=0}^{C-1} \sum_{k=0}^{K_h-1} \sum_{l=0}^{K_w-1} x_{c,i\times s_h - p_h + k\times d_h, j\times s_w - p_w + l\times d_w} \times w_{o,c,k,l}, \tag{2.8}$$

where d_h and d_w are the dilation rates along height and width axes accordingly.

[5] Receptive field means the largest possible region in which the perturbation of any input pixel could affect the output of a given neuron. For a single convolutional layer, the size of the receptive field

Fig. 2.3 Dilated convolutions. The chart shows 3 × 3 convolutional kernels with different dilation rates (image credit to [85])



Transposed convolution. In some literature it has an alias *deconvolution* [48, 77]. Note that transposed convolution (or deconvolution) does not really inverse the convolution operation; instead, it just inverses the *connectivity* between the input and the output. Figure 2.4 illustrates the "inversion of connectivity" in transposed convolution. Before starting, suppose we have a conventional convolution layer (noted by "original convolution", not marked in the figure) that predicts the blue feature map from the green. Hence the transposed convolution takes the blue feature map as the input and predicts the green back.

First, let us consider the case of $s = 1$ for the original convolution. Figure 2.4a shows the configuration of $p = 2$ and $K = 3$. The spatial size of the green map is 5×5, hence the blue map size is 7×7 according to Eq. 2.6. Then we analyze the connectivity between the two feature maps: since each pixel in the blue map is derived from 3×3 pixels in the green map, to inverse the connectivity, each pixel in the green map should also relate to 3×3 pixels in the blue map. Therefore, in this case, the transposed convolution acts like a normal convolution of configuration $K = 3$, $p = 0$, and $s = 1$, from the blue map back to the green map (however, we should flip the filter vertically and horizontally before computing).

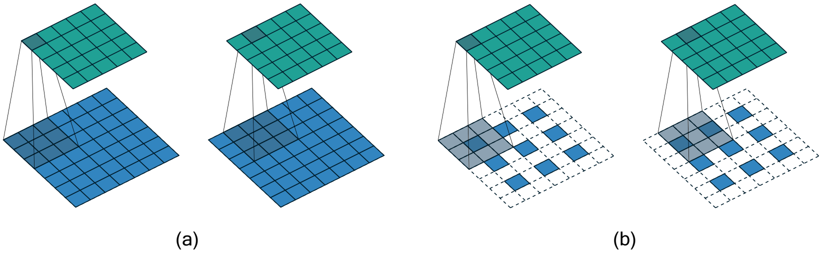


Fig. 2.4 Transposed convolutions. Suppose there is a conventional convolution taking the green feature map to predict the blue accordingly. Then the corresponding transposed convolution otherwise takes the blue feature map as the input and predicts the green. (1) $s = 1$, $K = 3$ and $p = 2$ for the original convolution; (2) $s = 2$, $K = 3$, and $p = 1$ for the original convolution. Images are taken from [17]

equals to the kernel size. There is another related concept: *effective receptive field (ERF)* [51], which further considers the content of the kernel. Readers may refer to [51] for more details.

Second, consider the configuration of $s = 2$, $K = 3$, and $p = 1$ for the original convolution, as in Fig. 2.4b. Still, each value in the blue map corresponds to 3×3 pixels in the green map. However, when considering the inverse connection, things will be a little different: each pixel in the green map could relate to 1, 2, or 4 pixels in the blue map, depending on the coordinate of the pixel. It is because the original convolution uses a stride larger than 1, hence to perform down-sampling; so accordingly, the transposed convolution has to upsample the feature map to inverse the connectivity. Nevertheless, a clever trick helps to make it simpler: first we "dilate" the blue map with zeros, as illustrated in Fig. 2.4b, then the green map can be derived from a normal convolution with $K = 3$, $s = 1$, and $p = 1$ on the dilated blue map (we still require to flip the kernel first).

In mathematics, transposed convolution can be rigorously defined as a linear transformation by the transposed kernel matrix. Generally speaking, for a conventional convolution operation $Y = X * w$, $X \in \mathbb{R}^{C \times I_h \times I_w}$, $Y \in \mathbb{R}^{O \times F_h \times F_w}$, $w \in \mathbb{R}^{O \times C \times K_h \times K_w}$, there exists a *kernel matrix* $\Omega \in \mathbb{R}^{(O F_h F_w) \times (C I_h I_w)}$, such that

$$vec(Y) = \Omega \times vec(X).$$

Readers are recommended referring to [17] to understand how to construct the matrix Ω from the weight w. Then, transposed convolution ($*^\top$) is defined as

$$Z = Y *^\top w \quad \Longleftrightarrow \quad vec(Z) = \Omega^\top \times vec(Y),$$

where $Z \in \mathbb{R}^{C \times I_h \times I_w}$. The following equation computes the output of a transposed convolution:

$$z_{c,i,j} = \sum_{o=0}^{O-1} \sum_{k=0}^{K_h-1} \sum_{l=0}^{K_w-1} y_{o,(i+k-K_h+p_h+1)/s_h,(j+l-K_w+p_w+1)/s_w} \times w_{o,c,K_h-k-1,K_w-l-1}. \tag{2.9}$$

Note that in the above formulation, if the subscript is not an integer or out of the dimension range, the result is regarded as 0.

Transposed convolution is very useful in convolutional neural networks. First, it can be used in computing the gradient *w.r.t.* the input of a convolutional layer, because we can prove

$$\frac{\partial \mathcal{L}}{\partial X} = \frac{\partial \mathcal{L}}{\partial Y} *^\top w,$$

where $Y = X * w$ and $\mathcal{L}$ is the loss. Second, since transposed convolution will predict a bigger feature map if the stride is greater than 1, we can employ it to upsample a feature map [48].

Global convolution. It indicates a convolution whose kernel size is as big as or even larger than the size of the feature map. Therefore, global convolution is good at summarizing global information from the whole image. Due to the huge computational cost, global convolution is often designed in a *depth-wise* [14, 56] or factorized [54] manner. There are a few variants.

For example, if the padding is zero, the spatial size of the resulted feature map will be 1 × 1, thus the convolution degenerates to weighted global pooling [29]. Moreover, some works [56] implement global convolution with circular convolution instead of the conventional zero-padded counterpart, which is convenient to speed up with fast Fourier transform (FFT).

Dynamic convolution. Up until now, all mentioned convolutions store weights on their own. The weights are learned during training and kept fixed in inference. These convolutions are known as *static*. *Dynamic convolutions* [52], on the other hand, the filter weights can be generated by another network:

$$Y = X * f(X'), \tag{2.10}$$

where we name the weight prediction function $f(\cdot)$ *hyper-network* [22] or *meta-network*. The input of the hyper-network X' can directly relate to the image (e.g., directly letting $(X' = X)$ [52]) or relies on additional input (e.g., [30, 47, 73]). Therefore, the weights of dynamic convolution can change with the inputs during inference.

3D/1D convolution. All convolutional layers mentioned above are *2D convolutions*. As shown in Fig. 2.2, even though each convolution filter appears to be a 3D volume, since it can only slide along two spatial dimensions (height and width), we still name it 2D convolution. Nevertheless in video processing, sometimes, we need "true" 3D convolution [4] to process both spatial and temporal dimensions. Given the input feature map $X \in \mathbb{R}^{C \times I_t \times I_w \times I_h}$, 3D convolution predicts the result $Y = X * w$, $Y \in \mathbb{R}^{O \times F_t \times F_w \times F_h}$, $w \in \mathbb{R}^{O \times C \times K_t \times K_h \times K_w}$, in which each element is computed as follows:

$$y_{o,u,i,j} = \sum_{c=0}^{C-1} \sum_{m=0}^{K_t-1} \sum_{k=0}^{K_h-1} \sum_{l=0}^{K_w-1} x_{c,u \times s_t - p_t + m, i \times s_h - p_h + k, j \times s_w - p_w + l} \times w_{o,c,m,k,l}. \tag{2.11}$$

Accordingly, to process 1D sequence (e.g., data for natural language processing (NLP) [9]) we can define 1D convolution in the similar way. Details are omitted.

2.1.3 CNN Example: AlexNet

AlexNet [38] is one of the cutting-edge convolutional neural networks in history, which is also the first deep learning architecture achieving state-of-the-art results on large-scale image classification dataset *ImageNet* [12]. Although the network was proposed over 10 years ago, some of the design choices are still being adopted in the follow-up works.

Next, we go over the elements in AlexNet and briefly discuss the relation between its design choices and more recent related works.

Macro design. AlexNet is a typical *plain network*, i.e., in which all layers are stacked one by one in series. Table 2.1 lists the architecture details. We say the depth of AlexNet is 8, since

Table 2.1 AlexNet architecture details

Layer	Kernel size	Stride	Group	Input dim	Output dim
Conv1	11 × 11	4	1	3 × 224 × 224	96 × 55 × 55
LRN	–	–	–	96 × 55 × 55	96 × 55 × 55
Max Pool	3 × 3	2	–	96 × 55 × 55	96 × 27 × 27
Conv2	5 × 5	1	2	96 × 27 × 27	256 × 27 × 27
LRN	–	–	–	256 × 27 × 27	256 × 27 × 27
Max Pool	3 × 3	2	–	256 × 27 × 27	256 × 13 × 13
Conv3	3 × 3	1	1	256 × 13 × 13	384 × 13 × 13
Conv4	3 × 3	1	2	384 × 13 × 13	384 × 13 × 13
Conv5	3 × 3	1	2	384 × 13 × 13	256 × 13 × 13
Max Pool	3 × 3	2	–	256 × 13 × 13	256 × 6 × 6
FC6	–	–	1	256 × 6 × 6	4096
FC7	–	–	1	4096	4096
FC8	–	–	1	4096	1000

it contains eight weight layers: five convolutional layers and three fully connected layers. A few other non-parametric layers, e.g., max pooling, are involved in between.

Convolutional layers build up the main body of AlexNet as well as all other CNNs. AlexNet employs large kernel convolutions on bottom layers (`Conv1` and `Conv2`) and small convolutions with more channels on top layers (`Conv3`, `Conv4` and `Conv5`), which used to be a common fashion but was challenged afterward [14, 60]. It is worth noting that `Conv2`, `Conv4`, and `Conv5` introduce *group convolutions*, not only reducing the complexity but also allowing efficient computing on multiple devices—it is a form of *model parallelism*.

Fully connected (FC) layers compose the last three layers of AlexNet to derive the 1000-way classification scores. As the spatial resolution is sufficiently small, those FC layers are able to gather information from the whole feature map maintaining low levels of complexity. Nevertheless, they still take $\sim 58.6M$ parameters, which is much bigger than convolutional layers. To save parameters, following works like [25, 62] usually use global average pooling followed by an FC layer instead of multiple FC layers to predict the classification results.

Pooling layers are used in AlexNet to perform down-sampling. Like depth-wise convolution, pooling layer also works in the channel-wise manner, which predicts the results by applying the pooling function to the elements in the sliding window. There are two common pooling functions: average pooling (mean pooling) and max pooling, where average pooling can also be viewed as a special depth-wise convolution with a (fixed) uniform kernel. It was believed that pooling layers help to introduce *permutation invariance* in the local region. However, further research seems to indicate that pooling layers can be replaced by (stride) convolutions without performance loss.

Normalization layers can stabilize training and highlight the contrast of input signal. AlexNet utilizes *local response normalization (LRN)*, which is defined as

$$y_{c,i,j} = x_{c,i,j} / \left(k + \alpha \sum_{m=\max(0,c-n/2)}^{\min(C-1,c+n/2)} (x_{m,i,j})^2 \right)^{\beta}, \tag{2.12}$$

where $X, Y \in \mathbb{R}^{C \times I_h \times I_w}$ are input and output tensors, respectively; k, α, β and window size n are the hyper-parameters. Clearly, LRN layer normalizes the input along the channel axis in a sliding-window manner. In modern CNNs, simpler normalization techniques such as *batch normalization (BN)* [33], *layer normalization (LN)* [1], or *group normalization (GN)* [70] are often employed instead of LRN.

Activation function. AlexNet chooses *rectified linear unit (ReLU)*, i.e., $y = \max(x, 0)$, as the activation function, which is placed right after each of the weight layers except for `FC8`. ReLU could be the most popular activation function in CNN design as it is simple and effective. Compared with other counterparts like tanh or sigmoid, one benefit of ReLU is that it does not saturate for positive inputs, which speeds up the convergence significantly as demonstrated in the paper. As an alternative to ReLU, Leaky ReLU or PReLU [24] further solve the vanishing gradient issue in the negative part. More recently, inspired by the success of transformers and neural architecture search, more sophisticated activations like GeLU [66] or SiLU [55] are proposed to obtain higher performance, however, at the cost of extra computation.

Training. AlexNet uses back-propagation with momentum SGD to train the parameters in end-to-end manner. Although a lot of sophisticated optimizers have been proposed, until now momentum SGD and Adam [37] (or its variant AdamW [49]) are the most widely used solvers in training CNNs. Additionally, research shows that if the batch size becomes large, then some hyper-parameter tuning [21] or other tricks [74] may be required.

As CNNs typically involve a large quantity of parameters, it is important to mitigate over-fitting. AlexNet suggests random cropping and color jittering to augment data. Other common data augmentation techniques include scale augmentation [62], *cutout* [13], *mixup* [79], etc. Apart from data augmentation, AlexNet also adopts *dropout* [61] as a structural regularization. Other similar techniques also include *drop-path* [32], which is mainly used in *ResNet*-like architectures [25].

2.1.4 Concept Modeling in CNNs

It is important to learn how convolutional neural networks extract rich semantic concepts from the input image. As illustrated in Fig. 2.2, it appears that the convolution operation can be viewed as *template matching*: each filter serves as a template thus generating a similarity map via sliding on the input image. For example, to detect a car, one possible way is to store all kinds of cars as templates in the convolutional kernels, therefore the detection results are embedded in the outputs after convolution. However, such template matching interpretation has the following two fatal drawbacks. First, in modern CNNs many convolutional kernels are very small (e.g., 3 × 3), hence it is unable to store complex templates nor catch the long-range interactions in the image. Second but more importantly, the number of required templates can be exponentially large—a good car detector may need the template set to cover all kinds of views, styles, colors, etc., which is extremely inefficient or impossible in practice.

Instead, deep learning draws power from the composition of layers, so do CNNs. As shown in Fig. 2.5, research (e.g., [76]) demonstrates that a network stacked with multiple convolutional layers learns visual concepts in a *hierarchical* way: the bottom most layers (`Layer 1` and `Layer 2`) detect simple *low-level* concepts—edge, corner, local texture, or color patch; the medium layers (`Layer 3` and `Layer 4`) employ the low-level information as the input thus predicting *mid-level* concepts, for instance, wheels, animal's heads, and object parts; finally, the mid-level features are composed *high-level* into semantic concepts by the top most layer (`Layer 5`), thus generating the prediction of the whole network. It is worth noting that such hierarchical feature composition is much more efficient than template matching. Still taking car detector as an example, to deal with a new deformation on the shape, rather than designing (exponentially) many new templates, we only need to adjust the way of predicting the highest level concepts, while low-level and mid-level features can still be *reused*. Therefore, hierarchical concept modeling is the key to the powerful capability of CNN models.

The fact that CNNs model hierarchical concepts further inspires researchers to design new architectures. For example, many dense prediction tasks in computer vision (e.g., semantic segmentation) require to predict semantic label for each input pixel. Therefore, those tasks not only require high-level features to generate semantic information but also rely on low-level details to align the results to the pixels. A widely used architectural paradigm for dense prediction is *U-net* [57], see Fig. 2.6. U-net introduces a decoder, which starts with the top layer in a pretrained network, then progressively up-samples the feature map and fuses the information with earlier feature maps. During up-sampling, the decoder takes rich concepts from different semantic levels, which helps to generate high-quality dense predictions.

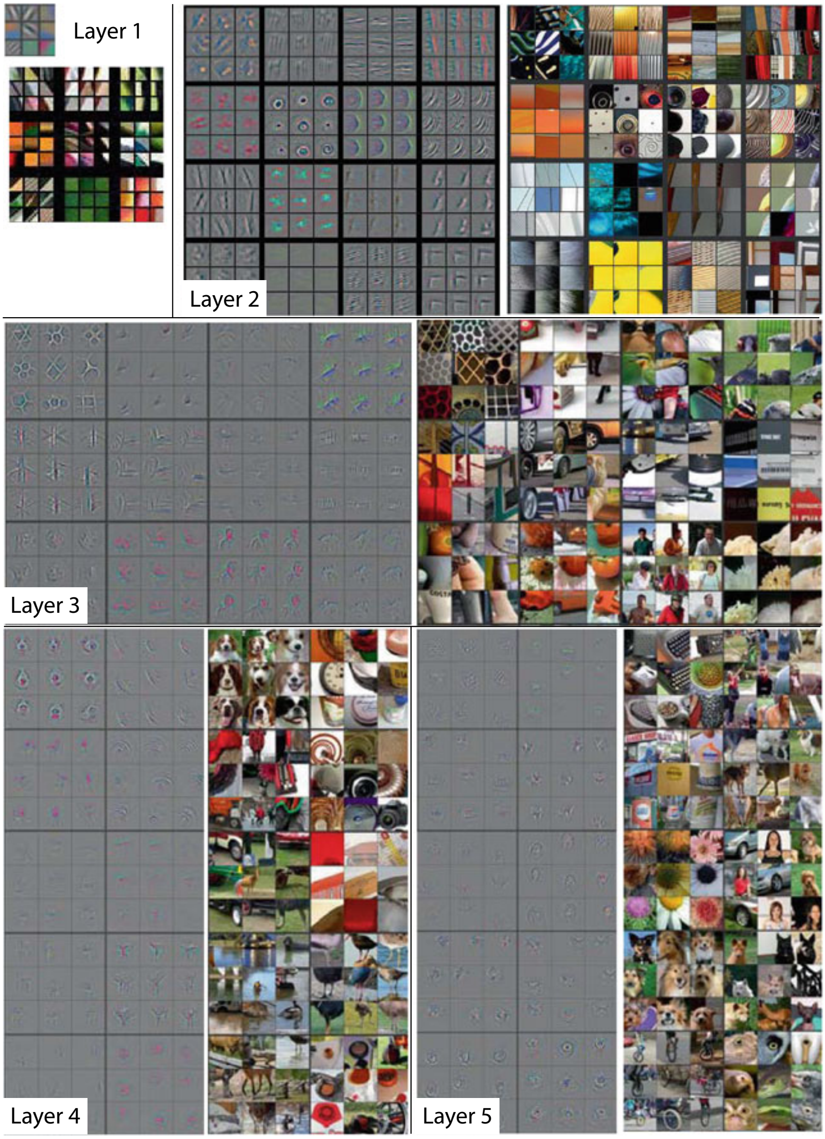


Fig. 2.5 Feature visualization in different CNN layers, as well as the corresponding image patch (courtesy of [76])

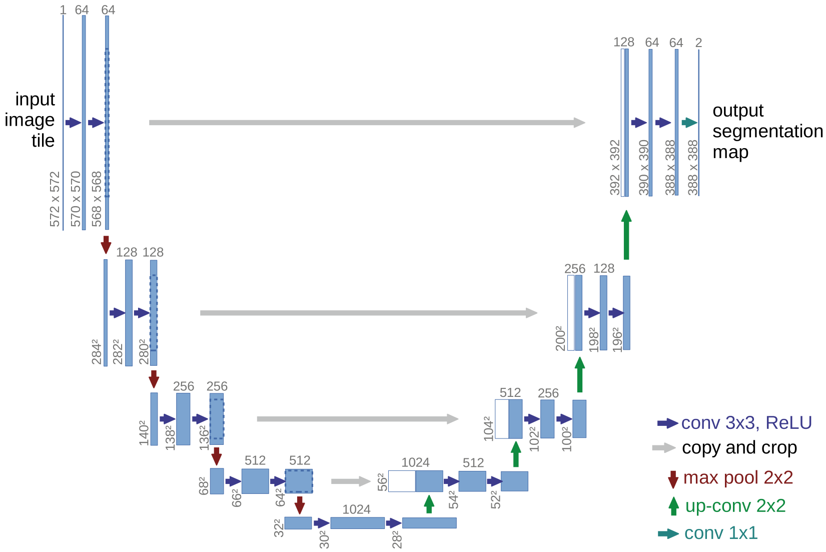


Fig. 2.6 U-net architecture (image credit to [57])

2.2 Convolutional Network Architectures

In the last decade, researchers have come up with enormous convolutional networks for different purposes. Why it is important to design new architectures? Generally speaking, architecture invention aims at:

- improving the representative capability;
- introducing new inductive bias for the target task;
- reducing the complexity for practical use;
- easing the optimization or overcoming the possible over-fitting in training.

In the next subsections, let us review some representative CNN architectures as well as the motivations behind them.

2.2.1 Going Deeper

The success of deep learning lies in deep composition of nonlinear functions. Therefore, to enhance the capability of CNNs, one of the most important directions is to scale up the depth. As mentioned in the last section, AlexNet includes eight weight layers. After that, a series of works successfully increased the depth toward a dozen or even more than a hundred layers.

VGG-Net

VGG-net [60] could be one of the earliest CNNs which gets improved performance via increasing the depth to more than 10 layers. VGG-net roughly follows the *plain* architecture of AlexNet [38], but with further simplifications. First, due to *universal approximation theorem* of neural networks, the authors believe a stack of *pure* convolutional layers is strong enough, hence they remove the local response layers suggested in AlexNet. Second, VGG-net abandons large kernel convolutions (e.g., 11×11 and 5×5 in AlexNet), instead, utilizes 3×3 convolutions only—the paper supposes that the capability of one large kernel can be implemented by stacking several small convolutions more efficiently.[6] The derived network is very simple and elegant, which contains only three different types of layers: 3×3 convolutions, 2×2 (non-overlapping) max-pooling layers, and fully connected layers. The deepest version of VGG-net is VGG-19 and it includes 19 weighted layers in total (16 convolutional layers in addition to 3 FC layers).

It is interesting to discuss how VGG-net manages to increase the depth over AlexNet. The authors find that when adding layers to AlexNet, the optimization becomes more and more difficult to converge. Therefore, they propose *two-step* training schedule, i.e., first to train a shallower network, then introducing additional layers with random initialization for further fine-tuning. In the following works, such training trick has developed to *Net2net* methodology [7], which provides a general approach to scale up a small network to a big one without training from scratch. Another line of research focuses on how to train deep VGG-style architectures directly. For example, [24] suggests initializing the convolutional kernels (followed by a ReLU activation) with Gaussian distribution, whose mean is zero and standard deviation is

$$\sigma = \sqrt{\frac{2}{N}}, \tag{2.13}$$

where N denotes the size of the filter $(K_h \times K_w \times C)$.[7] It is known as "MSRA initialization". The authors find that under such initialization, VGG-net can train from scratch without convergence problems such as vanishing or exploding gradient.

Although a lot of efforts have been done to overcome the optimization issue of VGG-net, the performance starts to saturate when the depth increases to around 20 layers. For example, VGG-19 obtains nearly the same accuracy as VGG-16. In [60], the authors hypothesize the saturation is resulted from the side effect of ReLU activation, which might block the information flow when the depth increases. However, [24] suggests that even when replacing ReLU with PReLU, it cannot significantly increase the depth without performance drop either. The problem remains unsolved until ResNet [25] comes up.

[6] However, it is challenged by the theory of *effective receptive field* [51].

[7] An alternative configuration is $N = K_h \times K_w \times O$. Please refer to [24] for details.

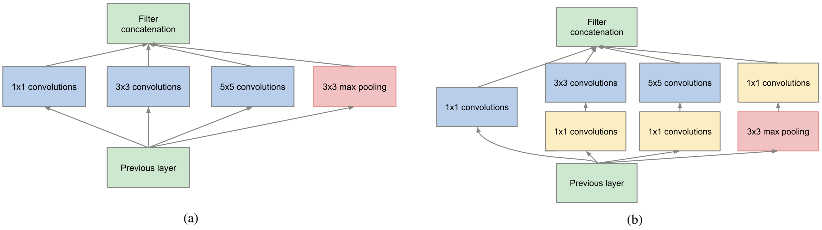


Fig. 2.7 Inception module in GoogleNet. **a** Naive version. **b** Module with dimensionality reduction. Image is taken from [62]

GoogleNet

Concurrent to VGG-net, *GoogleNet* [62] also aims to increase the depth but in a different way. The core of GoogleNet is the proposed *Inception module*, as shown in Fig. 2.7. Different from the *plain* network fashion as in AlexNet and VGG-net, the Inception module is a *multi-path* structure, in which the input feature map is passed into four branches and the corresponding results are concatenated up as the output. The four branches include different types of layers—1 × 1 convolution, 3 × 3 convolution, 5 × 5 convolution, and 3 × 3 max pooling, respectively, as illustrated in Fig. 2.7a. Furthermore, since large kernel branches are costly, the paper introduces additional 1 × 1 convolutional layers to all branches other than the 1 × 1 branch, in order to reduce the number of channels (see Fig. 2.7b). The authors believe the multi-path design helps to encode rich representations of multiple scales.

Based on Inception blocks, the resulted GoogleNet successfully increases the depth to 22 layers,[8] achieving outstanding performances and winning the ILSVRC 2014 competition [58]. Some researchers believe the isolated 1 × 1 branch (Fig. 2.7b, the left-most branch) is the key to success: although it is known that a network deeper than 20 layers suffer from severe optimization problem, thanks to the 1 × 1 branch, the shallowest path in GoogleNet contains only 12 layers, which may greatly ease the optimization. A number of empirical results support the conjecture. For example, the original paper [62] mentioned that purely stacking the naive version of the Inception block (Fig. 2.7a) does not work. And the following work [63] implies the 1 × 1 serves as a "shortcut" analogous to that in ResNet [25].

GoogleNet raises several interesting research topics. For example, for multi-branch structures like Inception, how to balance the output of each branch? The following work *Batch Normalization (BN)* [33] proposes a nice idea via normalizing the output of each convolutional layer in such a way:

$$y^{(c)} = \frac{x^{(c)} - \mu_{\mathcal{B}}^{(c)}}{\sigma_{\mathcal{B}}^{(c)}} \gamma^{(c)} + \beta^{(c)}, \tag{2.14}$$

[8] For multi-path architectures like GoogleNet, the depth is defined according to the longest path from the input to the output.

where x and y are the input and output, respectively; and the superscript "(c)" means c-th channel, $\mu_{\mathcal{B}}^{(c)}$ and $\sigma_{\mathcal{B}}^{(c)}$ are the mean and standard deviation accordingly within the batch; $\gamma^{(c)}$ and $\beta^{(c)}$ are learnable parameters. During inference, since "batch data" are not available, $\mu_{\mathcal{B}}^{(c)}$ and $\sigma_{\mathcal{B}}^{(c)}$ are replaced by the running statistics. BN contributes great benefits to CNN networks, for example, it helps to balance the magnitude of feature maps, prevent the variance shift as well as the saturation of activation functions, introduce additional regularization, and speed up the convergence. Therefore, BN has become one of the most popular normalization techniques in modern CNNs. Nevertheless, in theory, the mystery behind BN is still not fully clear yet [2, 44, 50, 59, 68, 78].

ResNet

In the race of going deeper, *ResNet* [25] could be the most influential CNN architecture, which first increases the depth to hundreds of layers with improved performances. The idea of ResNet is based on the following observation: when increasing the depth of a VGG-style plain network from 18 to 34 layers, even though the number of parameters increases, the accuracy unexpectedly drops (shown in Fig. 2.8). It cannot be attributed to over-fitting because training loss and validation accuracy degenerate at the same time. Such a phenomenon seems rather counter-intuitive because the capability of a 34-layer network is strictly stronger than the 18-layer counterpart, since the former can *exactly* represent the latter by setting the additional layers toward *identity*. Therefore, it is in fact the optimization problem that causes the degradation.

To overcome the optimization issue in deep networks, ResNet thus proposes a very simple solution: just adding the *identity prior* back. For each building block in a network, rather than predict a brand new feature map, ResNet suggests predicting the *residue* subject to the input:

$$y = \sigma(\mathcal{F}(x) + x), \tag{2.15}$$

where $\mathcal{F}(\cdot)$ is the *residual block* and $\sigma(\cdot)$ is the activation function (ReLU by default). The design of the residual function $\mathcal{F}(\cdot)$ can be arbitrary. For example, the original paper of

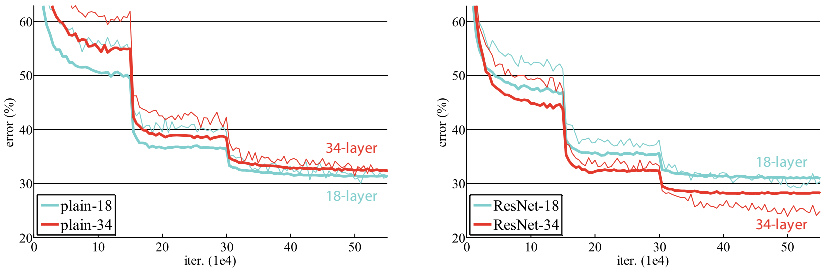


Fig. 2.8 Comparison of training curves between plain network and ResNet (courtesy of [25])

ResNet proposes two residual blocks: one is composed of two successive 3×3 convolutional layers (following VGG-style structure); the other is so-named “bottleneck” block stacking $1 \times 1, 3 \times 3$ and another 1×1 convolutions, in which the two 1×1 layers shrink the number of channels to reduce the cost. The latter design obtains slightly better performance under the given complexity budget.

In Eq. 2.15, the *identity shortcut* term (plus x) plays a key role. From the perspective of representative capability, this term seems to be unnecessary because it can be absorbed into the residual function $\mathcal{F}(\cdot)$. However, the experiment in Fig. 2.8 indicates the difference during *optimization*: after the shortcut is introduced, a 34-layer ResNet starts to outperform the 18-layer counterpart, while a 34-layer plain network cannot. It strongly suggests that ResNet greatly overcomes the optimization problem when network becomes deep, even though ResNet may share the same functional space with conventional VGG-style networks.

The original ResNet paper finally manages to train a 152-layer network on ImageNet, which also wins ILSVRC 2015 challenge [58]. The depth of a smaller ResNet on Cifar-10 even exceeds 1000 layers, however, it does not further improve the performance. Afterward, an improved version of ResNet [26] overcomes the drawback via keeping a “clean” shortcut path. The idea of residual connection has been widely accepted in the community—not only in computer vision but also in natural language processing and many other fields, for example, transformers [66] utilize identity shortcuts in each of the attention blocks. Moreover, researchers further find the shortcut is neither necessary in identity form, nor in additive manner. For example, [31, 53] demonstrate that shortcuts of 1×1 convolutions, or fused by concatenation, still work for deep architectures. A recent work *RepVGG* [15] manages to train a VGG-style plain network toward ResNet’s performance. The proposed methodology, *i.e., structural re-parameterization*, suggests training the network with shortcuts while merging the additional branches into an equivalent plain structure in inference time, which is also inspired by the optimization bonus of ResNet.

ResNet has attracted a lot of attention from researchers to study how it works. Veit et al. [67] conjectures that ResNet may actually behave like an ensemble of many shallower networks; since the underlying subnetworks are shallow, it avoids optimization issue brought by deep architectures. The conjecture can explain some empirical results, e.g., [14] finds the effective receptive field of deep ResNet is not that large, which implies ResNet may be intrinsically shallow. De and Smith [11] further points that *batch normalization* [33] layers in ResNet may be the key to make it shallow. Additionally, there exist other related directions. For example, [43] suggests that the shortcut connections in ResNet make the loss landscape smooth. [23] finds that the identity shortcut changes the distribution of local optima. Chen et al. [6] relates ResNet to *Ordinary Differential Equations*, etc.

2.2.2 Spatial Modeling

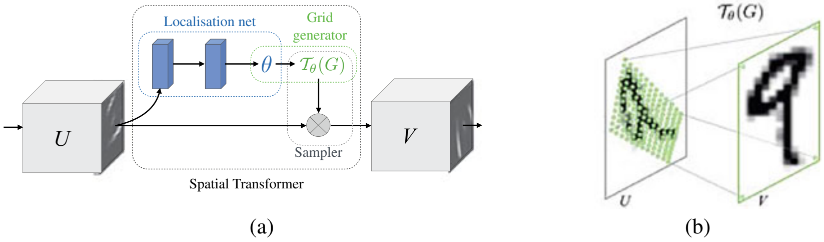
As mentioned above, CNNs model spatial relations via *spatial convolutions* (i.e., whose kernel size is larger than 1 × 1), which is built upon *locality* and *translational equivariant* properties. However, not all vision tasks strictly satisfy these properties, such as face recognition. Fortunately, researchers have come up with many spatial modeling techniques to make CNN effective on those tasks.

Spatial Transformer Network

Spatial Transformer Network (STN) [35] is a plug-and-play module for convolutional networks. It is known that objects in real world may be deformed or transformed in geometry. For example, a face image can be taken from either the front view or the side view; thus, the network should be robust to these transformations. However, it is difficult for vanilla CNNs to learn the invariance by design: CNNs process image with *regular* sampling grid, while if global transformations are applied (e.g., rotation or affine transformations), the desired sampling grid could also be transformed or become *irregular*, as illustrated in Fig. 2.9b. In other words, the default inductive bias for convolutional layers is not suitable to represent global transformations.

To overcome such drawback, the solution proposed by [35] is simple and straightforward: introducing a building block named STN to learn global transformations explicitly. As shown in Fig. 2.9a, the block mainly consists of two parts: the localization network that predicts a transformation matrix Θ from the input feature map, followed by the grid generator as well as the sampler that performs transformation according to Θ. To make the whole network end-to-end trainable, every component in STN has to be differentiable. STN warps the feature map in the following formulation:

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m) k(y_i^s - n), \quad \forall i \in [1 ... H'W'], \tag{2.16}$$



(a)

(b)

Fig. 2.9 Illustration of Spatial Transformer Network (courtesy of [35]). **a** Overall architecture. **b** Image sampling mechanism according to the learned transformation

where $U \in \mathbb{R}^{C\times H\times W}$ and $V \in \mathbb{R}^{C\times H'\times W'}$ are the input and output feature maps, respectively; the function $k(\cdot)$ denotes the interpolation kernel; the coordinate (x_i^s, y_i^s) is the *sampling point* corresponding to the output position i, which is predicted by the grid generator $\mathcal{T}_\Theta(G)$:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\Theta(G) = \Theta \times \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}; \tag{2.17}$$

$\Theta \in \mathbb{R}^{2\times 3}$ is the affine matrix generated by the localization network; (x_i^t, y_i^t) is the coordinate of i-th position in the output feature map V.

In detail, Eq. 2.16 can be interpreted as extracting a feature point at (x_i^s, y_i^s) for the position i, i.e., $V_i^c = U^c_{y_i^s, x_i^s}$. However, there are two problems: first, the coordinates (x_i^s, y_i^s) could be non-integers; second, the operation of extracting features is not differentiable *w.r.t.* x_i^s and y_i^s. To solve this problem, Eq. 2.16 proposes to sum up several surrounding points weighted by the kernel $k(\cdot)$ centered at (x_i^s, y_i^s), instead of directly extracting the features exactly at the location. For example, the kernel function could be chosen as a *bilinear kernel*:

$$k(x) \triangleq \max(0, 1 - |x|).$$

Hence, the partial derivatives *w.r.t.* the input feature map U and the sampling point (x_i^s, y_i^s) are as follows:

$$\frac{\partial V_i^c}{\partial U_{nm}^c} = \sum_n^H \sum_m^W \max(0, 1 - |x_i^s - m|)\max(0, 1 - |y_i^s - n|),$$

$$\frac{\partial V_i^c}{\partial x_i^s} = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |y_i^s - n|) \begin{cases} 0 & |m - x_i^s| \geq 1 \\ 1 & m \geq x_i^s \\ -1 & m < x_i^s \end{cases}$$

and the case of $\partial V_i^c / \partial y_i^s$ is similar.

In conclusion, STN is a very powerful plugin for CNNs to deal with global transformation or deformation. The general idea of STN has inspired many follow-ups, e.g., [10]. In some tasks like face recognition, introducing an STN block to the bottom layer can significantly improve the performance and robustness to pose variations [82].

Deformable Convolutional Network

Besides global transformations, in computer vision, it is also very important to model *local* transformation or deformation. For example, consider an image with multiple objects, each of which can have different poses and be deformed in many ways, respectively. On the one hand, introducing a learned global transformation like [35] is clearly insufficient to deal with the deformation of each object individually. On the other hand, the vanilla convolutional layer is good at modeling local objects, however, it cannot produce robust features for

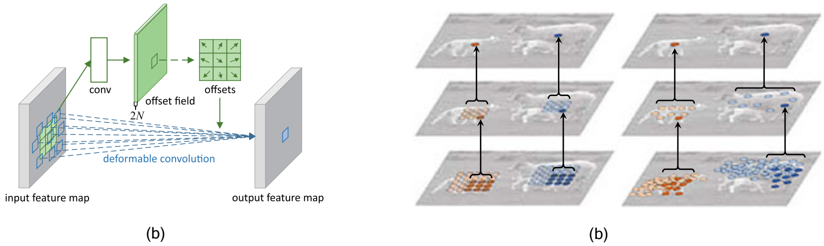


Fig. 2.10 Deformable convolutional network (DCN). **a** Deformable convolutional layer. **b** Comparison between regular convolution and deformable convolution. Image is taken from [10]

object's deformation. Is there any way to combine the advantages of convolutions and STNs? *Deformable convolutional network (DCN)* [10] is one of the solutions following such motivation.

Figure 2.10a gives an illustration of the structure of deformable convolution. Like vanilla convolution, DCN also includes $K_h \times K_w$ convolutional kernels; however, whose shape is data-dependent and context-aware—for various inputs and different locations on the feature map, the kernel shapes are also different. Specifically, DCN includes an auxiliary branch (usually implemented with a conventional convolution) to predict the spatial *offsets* for each element in the kernel, thus the kernels can be *deformable* with the input. In mathematics, a deformable convolution is formulated as[9]:

$$Y[o, i, j] = \sum_{c} \sum_{k=0}^{K_h-1} \sum_{l=0}^{K_w-1} W[o, c, k, l] \times X[c, i + k + \Delta_h^{i,j}(k, l), j + l + \Delta_w^{i,j}(k, l)], \tag{2.18}$$

where $X \in \mathbb{R}^{C\times I_h\times I_w}$, $Y \in \mathbb{R}^{O\times F_h\times F_w}, and W \in \mathbb{R}^{O\times C\times K_h\times K_w}$ are the input tensor, output tensor, and weight tensor, respectively, similar to the vanilla convolutions, while $\Delta_h^{i,j}(\cdot)$ and $\Delta_w^{i,j}(\cdot)$ specify the kernel offsets at location (i, j) vertically and horizontally, which is the key to deformation and produced by an auxiliary convolutional subnetwork from the input. Similar to that in STN, the offset values Δ_h and Δ_w are not integers necessarily; a bilinear interpolation can be introduced to make the formulation differentiable and compatible with non-integer indexes, i.e.,

$$X[c, d_y, d_x] \triangleq \sum_{n}^{I_h} \sum_{m}^{I_w} X[c, n, m] \max(0, 1 - |d_y - n|) \max(0, 1 - |d_x - m|).$$

Readers may refer to STN [35] in the previous text for more details about how the interpolation works.

[9] For simplicity, in the formulation we omit common hyper-parameters of convolutions such as stride, padding, and dilation, although they are still applicable in DCNs.

DCN enables convolutional kernels to sample feature map *adaptively* according to the context. As shown in Fig. 2.10b, in a vanilla convolution, the kernel shape is always regular and irrelevant to the input content; while in deformable convolution, the kernel shape is *context-aware*: a convolutional filter sliding over different objects will have different shapes, which enhances the capability to model the deformation of each individual object, respectively. For example, for animals with various poses, deformable convolutions could adapt their kernel shapes with the deformation of the contours, thus help to derive robust features.

The idea of DCN can be further extended. For example, as pointed in [10], *RoIPooling* [20], a common operator in CNN-based object detection frameworks, can also be generalized to the deformable counterpart. Research in [83] further suggests that deformable convolution can be viewed as a special but efficient self-attention [66] layer. More recently, [84] takes the idea of DCN to propose a new attention mechanism named *deformable attention*, which yields outstanding performance on end-to-end object detection frameworks.

Non-local Network

In visual understanding tasks, sometimes we not only need to recognize each object individually but also require to learn their *relations* along spatial or temporal dimensions. For example, to track a quick-moving object, the network has to look at not only the surrounding pixels around the object but also far-away patches which have similar appearances in other frames. Yet again, vanilla convolutions (including 3D convolution layers) are not suitable to model such non-local relations due to the intrinsic locality prior. Hence, *non-local neural networks (NLNet)* [69] are proposed to address the drawback of CNNs.

The building block design of NLNet is greatly inspired by *multi-head self-attention (MHSA)* [66], which is firstly applied in natural language processing tasks to deal with the long-term dependency problem. The following equation gives the general form of the proposed non-local block:

$$\mathbf{y}_i = \frac{1}{C(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j), \tag{2.19}$$

where $\mathbf{x}_i$ is the feature point vector (whose dimension equals to the number of channels) at location i,[10] and $\mathbf{y}_i$ is the corresponding output; $f(\mathbf{x}_i, \mathbf{x}_j)$ indicates the *similarity* between $\mathbf{x}_i$ and $\mathbf{x}_j$; $g(\cdot)$ is a unary mapping and $C(\mathbf{x})$ is the normalization term. The summation in Eq. 2.19 is performed over all possible locations—not only limited to the positions near i—therefore we say the operator is “non-local”. It is worth noting that some classic computer vision algorithms such as non-local means [3] and bilateral filter [18] also share the similar formulation as Eq. 2.19.

[10] In non-local network, the “location” refers to both spatial position and temporal frame (if any).

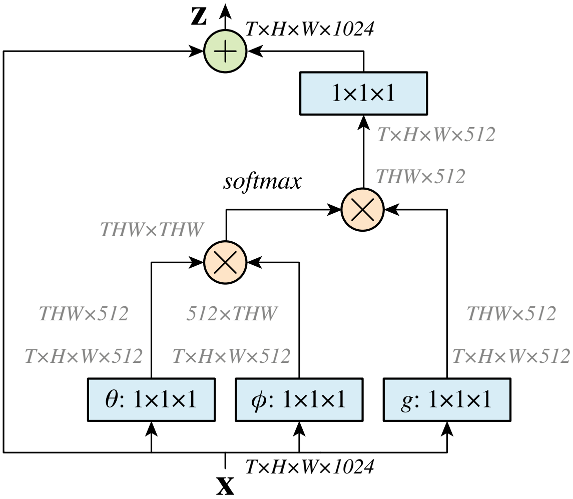


Fig. 2.11 Non-local neural network. Image is taken from [10]

As a special case of non-local module, letting $f(\mathbf{x}_i, \mathbf{x}_j) = \exp(< W_\theta \mathbf{x}_i, W_\phi \mathbf{x}_j >)$, $g(\mathbf{x}_i) = W_g \mathbf{x}_i$ and $C(\mathbf{x}_i) = \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j)$, the proposed building block can be formulated in the following matrix form (also shown in Fig. 2.11):

$$
\begin{aligned}
Y &= \text{Softmax}(X W_\theta W_\phi^\top X^\top) X W_g \\
Z &= X + Y W_z^\top .
\end{aligned}
\tag{2.20}
$$

Here, $X, Z \in \mathbb{R}^{(T \times H \times W) \times C}$ are the input and output feature maps of NLNet block (in matrix form), in which T is the number of frames in the video ($T = 1$ if performed on a single image) and $H \times W$ is the spatial size. $W_\theta, W_\phi, W_g, W_z \in \mathbb{R}^{C \times C'}$ are linear mappings which can be implemented by 1×1 convolutions, respectively. Notice that Eq. 2.20 extends Eq. 2.19 with an additional shortcut so that it can be applied in deeper network architecture. The formulation is very similar to that in self-attention [66], while in the latter work, the dimension of X is $N \times C$, where N is the sequence length of the input sentences. NLNet utilizes Eq. 2.20 as the default block design.

Further research works suggest that blocks like NLNet or self-attention give many advantages to CNNs, such as enabling object relation modeling [28], feature denoising [71], and enlarging the effective receptive field [19]. More recently, [16] finds that given more training data and complexity budget, *pure* attention networks (i.e., vision transformers) could even outperform CNN counterparts, suggesting non-local mechanism is a strong and general spatial modeling paradigm.

Acknowledgements Special thanks to my colleague Qi Han for his help in proofreading.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Bjorck, N., Gomes, C.P., Selman, B., Weinberger, K.Q.: Understanding batch normalization. Adv. Neural Inf. Process. Syst. **31** (2018)
3. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, pp. 60–65. IEEE (2005)
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
6. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. Adv. Neural Inf. Process. Syst. **31** (2018)
7. Chen, T., Goodfellow, I., Shlens, J.: Net2net: Accelerating learning via knowledge transfer. arXiv preprint arXiv:1511.05641 (2015)
8. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
9. Collins, M., Duffy, N.: Convolution kernels for natural language. Adv. Neural Inf. Process. Syst. **14** (2001)
10. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)
11. De, S., Smith, S.: Batch normalization biases residual blocks towards the identity function in deep networks. Adv. Neural. Inf. Process. Syst. **33**, 19964–19975 (2020)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
13. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
14. Ding, X., Zhang, X., Han, J., Ding, G.: Scaling up your kernels to 31x31: revisiting large kernel design in cnns. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11963–11975 (2022)
15. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: Repvgg: making vgg-style convnets great again. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13733–13742 (2021)
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
17. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285 (2016)
18. Elad, M.: On the origin of the bilateral filter and ways to improve it. IEEE Trans. Image Process. **11**(10), 1141–1151 (2002)

19. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146–3154 (2019)
20. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
21. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
22. Ha, D., Dai, A., Le, Q.V.: Hypernetworks. arXiv preprint arXiv:1609.09106 (2016)
23. Hardt, M., Ma, T.: Identity matters in deep learning. arXiv preprint arXiv:1611.04231 (2016)
24. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
26. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European Conference on Computer Vision, pp. 630–645. Springer (2016)
27. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
28. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3588–3597 (2018)
29. Hu, J., Shen, L., Albanie, S., Sun, G., Vedaldi, A.: Gather-excite: exploiting feature context in convolutional neural networks. Adv. Neural Inf. Process. Syst. **31** (2018)
30. Hu, X., Mu, H., Zhang, X., Wang, Z., Tan, T., Sun, J.: Meta-sr: a magnification-arbitrary network for super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1575–1584 (2019)
31. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
32. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: European Conference on Computer Vision, pp. 646–661. Springer (2016)
33. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456. PMLR (2015)
34. Islam, M.A., Jia, S., Bruce, N.D.: How much position information do convolutional neural networks encode? arXiv preprint arXiv:2001.08248 (2020)
35. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Adv. Neural Inf. Process. Syst. **28** (2015)
36. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866 (2014)
37. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
38. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
39. Lavin, A., Gray, S.: Fast algorithms for convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4013–4021 (2016)
40. Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., Lempitsky, V.: Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv preprint arXiv:1412.6553 (2014)

41. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
42. Li, C., Farkhoor, H., Liu, R., Yosinski, J.: Measuring the intrinsic dimension of objective landscapes. arXiv preprint arXiv:1804.08838 (2018)
43. Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the loss landscape of neural nets. Adv. Neural Inf. Process. Syst. **31** (2018)
44. Li, X., Chen, S., Hu, X., Yang, J.: Understanding the disharmony between dropout and batch normalization by variance shift. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2682–2690 (2019)
45. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
46. Liu, R., Lehman, J., Molino, P., Petroski Such, F., Frank, E., Sergeev, A., Yosinski, J.: An intriguing failing of convolutional neural networks and the coordconv solution. Adv. Neural Inf. Process. Syst. **31** (2018)
47. Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, K.T., Sun, J.: Metapruning: meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3296–3305 (2019)
48. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
49. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
50. Luo, P., Wang, X., Shao, W., Peng, Z.: Towards understanding regularization in batch normalization. arXiv preprint arXiv:1809.00846 (2018)
51. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **29** (2016)
52. Ma, N., Zhang, X., Huang, J., Sun, J.: Weightnet: revisiting the design space of weight networks. In: European Conference on Computer Vision, pp. 776–792. Springer (2020)
53. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 116–131 (2018)
54. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters–improve semantic segmentation by global convolutional network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4353–4361 (2017)
55. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017)
56. Rao, Y., Zhao, W., Zhu, Z., Lu, J., Zhou, J.: Global filter networks for image classification. Adv. Neural. Inf. Process. Syst. **34**, 980–993 (2021)
57. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
58. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
59. Santurkar, S., Tsipras, D., Ilyas, A., Madry, A.: How does batch normalization help optimization? Adv. Neural Inf. Process. Syst. **31** (2018)
60. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
61. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

62. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
63. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
64. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014)
65. Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al.: Mlp-mixer: an all-mlp architecture for vision. Adv. Neural. Inf. Process. Syst. **34**, 24261–24272 (2021)
66. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
67. Veit, A., Wilber, M.J., Belongie, S.: Residual networks behave like ensembles of relatively shallow networks. Adv. Neural Inf. Process. Syst. **29** (2016)
68. Wan, R., Zhu, Z., Zhang, X., Sun, J.: Spherical motion dynamics: learning dynamics of normalized neural network using sgd and weight decay. Adv. Neural. Inf. Process. Syst. **34**, 6380–6391 (2021)
69. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)
70. Wu, Y., He, K.: Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
71. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 501–509 (2019)
72. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)
73. Yang, T., Zhang, X., Li, Z., Zhang, W., Sun, J.: Metaanchor: learning to detect objects with customized anchors. Adv. Neural Inf. Process. Syst. **31** (2018)
74. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
75. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 472–480 (2017)
76. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer (2014)
77. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2528–2535. IEEE (2010)
78. Zhang, G., Wang, C., Xu, B., Grosse, R.: Three mechanisms of weight decay regularization. arXiv preprint arXiv:1810.12281 (2018)
79. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
80. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
81. Zhang, X., Zou, J., He, K., Sun, J.: Accelerating very deep convolutional networks for classification and detection. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 1943–1955 (2015)

82. Zhou, E., Cao, Z., Sun, J.: Gridface: Face rectification via learning local homography transformations. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
83. Zhu, X., Cheng, D., Zhang, Z., Lin, S., Dai, J.: An empirical study of spatial attention mechanisms in deep networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6688–6697 (2019)
84. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
85. Ziegler, T., Fritsche, M., Kuhn, L., Donhauser, K.: Efficient smoothing of dilated convolutions for image segmentation. arXiv preprint arXiv:1903.07992 (2019)

Generative Networks

3

Ziwei Liu, Shuai Yang, Yuming Jiang, and Ziqi Huang

3.1 Introduction

Synthesizing photorealistic human faces is an appealing yet challenging task. Before the advent of deep learning, researchers used predefined 3D face models to design generative models for facial images. However, the abstraction and distortion of predefined models hinder the realism of the generated faces. With the development of deep learning, a large number of generative models have been proposed, especially in the field of face image generation. These generative models do not rely on the predefined 3D models and are purely data-driven. The generative face models can successfully capture the features of human faces. Recent years have witnessed the great progress in various generative frameworks and training paradigms, as well as the large collections of face datasets from LFW [28], CelebA [47], CelebA-HQ [37] to FFHQ [40]. As shown in Fig. 3.1, the quality and resolution of the generated faces have gradually increased. By the end of 2020, the state-of-the-art generative model [41] has the capability to generate 1024 × 1024 photorealistic human faces.

Face generation has many exciting applications. In the filming and gaming industry, we can synthesize virtual avatars to create interactions that are beyond the capabilities of humans

Z. Liu (✉) · S. Yang · Y. Jiang · Z. Huang
S-Lab, Nanyang Technological University, Singapore, Singapore
e-mail: ziwei.liu@ntu.edu.sg

S. Yang
e-mail: shuai.yang@ntu.edu.sg

Y. Jiang
e-mail: yuming002@e.ntu.edu.sg

Z. Huang
e-mail: ziqi002@e.ntu.edu.sg

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_3

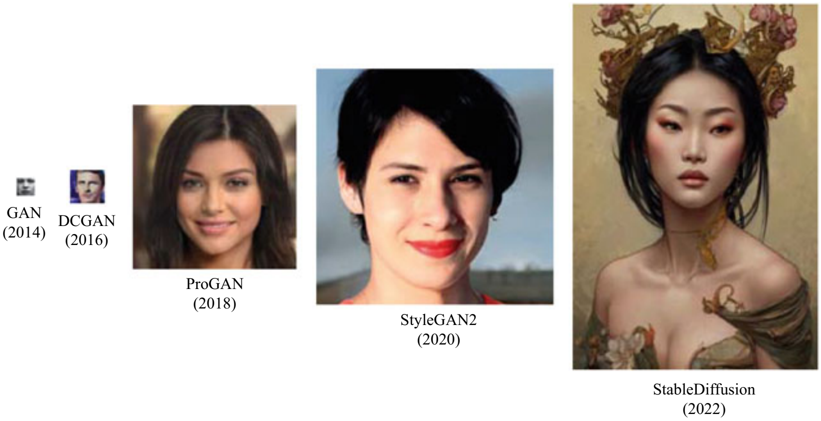


Fig. 3.1 Progress in face generation every 2 years. With the rapid development of advanced generative models and the collection of large datasets, the quality and resolution of face generation have improved greatly over the past 10 years

in the real world. For example, in the fantasy world of Cyberpunk, players can use wonderful special effects to become a cyborg and interact with NPCs. In social media, users can create their own personalized avatars. In the security domain, we can protect privacy by replacing real faces with generated ones. In traditional media, TV stations can create virtual digital hosts, and entertainment companies can create virtual singers and virtual idols. Self-media can also customize the image they present to the public in videos.

Besides, in face-related computer vision tasks, face generation and face recognition can complement each other. On the one hand, face generation can provide thousands of synthetic data for face recognition, and these synthetic data do not have privacy or copyright issues. On the other hand, face recognition can provide various semantic labels for face generation, assisting the conditional face generation, thereby supporting multiple application tasks. For example, we can edit the facial expressions given a face image. In fact, researches [45] have employed face generation to support face recognition.

In this chapter, we will dive into two successful generative models developed in the past decade: Variational Autoencoder (VAE) and Generative Adversarial Networks (GAN). VAE adopts the idea of variational inference and learns to align the latent space to a fixed distribution. The final image is synthesized by sampling from the fixed distribution. Later, quantization techniques and Transformers are introduced to boost the performance of VAE, making it have comparable performance to GAN. As for GAN, it is the most popular generative model in the recent decade. Since 2017, it has dominated the field of generative models. The core idea of adversarial training enables the network learn to generate faces implicitly without any hand-crafted priors, which promotes the spirit of data-driven learning. We will introduce the adversarial idea of GAN and give a brief review of the development of GAN models, both in terms of the architectures and optimization goals. Then, we will present the development of generative models in face-related tasks, including unconditional face generation and conditional face generation. Finally, we will briefly explain the metrics for evaluating the quality of face generation.

3.2 Variational Autoencoder

3.2.1 Vanilla VAE

Autoencoders are commonly adopted in unsupervised learning to extract features from raw data. The extracted features are used in many downstream tasks, such as image recognition and segmentation. The idea behind autoencoders is to use an encoder to extract features, which are then reconstructed by a decoder. The extracted features learned in this way contain compressed information of the raw data. Mathematically, the training process of autoencoders can be expressed as follows:

$$\begin{aligned} \hat{x} &= D(E(x)), \\ L &= \left\| \hat{x} - x \right\|_2^2, \end{aligned} \tag{3.1}$$

where $E(\cdot)$ is the encoder, $D(\cdot)$ is the decoder, and L is the training objectives.

The autoencoders have the ability to generate images from a given feature. However, this ability is limited to reconstruction, i.e., the learned autoencoder cannot generate new data. Variational autoencoders overcome the aforementioned limitation by incorporating probabilistic concepts.

Variational autoencoders are trained to reconstruct training data $\{x^{(i)}\}_{i=1}^{N}$ from a latent representation z, which is optimized to align with the fixed probabilistic prior, e.g., Gaussian distribution. Once the model is trained, new data can be sampled as follows:

$$x \sim p(x|z^{(i)}), z^{(i)} \sim p(z), \tag{3.2}$$

where $p(z)$ is assumed as simple priors, e.g., Gaussian distribution, and $p(x|z)$ is represented as a neural network, e.g., the decoder in variational autoencoders.

To this end, the decoder needs to be probabilistic. The decoder takes the latent code z as input and outputs the mean $\mu_{x|z}$ and the covariance $\Sigma_{x|z}$. The new data x is then sampled from the Gaussian with mean and covariance. The intuition of training the decoder is to maximize the likelihood of the raw data. For each x, if its corresponding z could be observed, then a conditional generative model $p(x|z)$ could be trained. However, when training variational autoencoders, the latent representation z is unobserved, so we need to marginalize $p_\theta(x)$:

$$p_\theta(x) = \int p_\theta(x, z)dz = \int p_\theta(x|z)p_\theta(z)dz, \tag{3.3}$$

where $p_\theta(x|z)$ can be implemented as a decoder network, and $p_\theta(z)$ is usually a simple Gaussian distribution in practice.

However, it is not ideal to perform the integration for all z. Another alternative solution is to use the Bayes' Rule to represent the $p_\theta(x)$ as follows:

$$p_\theta(x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)}. \tag{3.4}$$

It is hard to compute the posterior probability $p_\theta(z|x)$, therefore, we use another network to mimic the posterior probability. In this way, the $p_\theta(x)$ can be further represented as follows:

$$p_\theta(x) = \frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|x)}. \tag{3.5}$$

The $q_\phi(z|x)$ can be implemented as the encoder network in variational autoencoders.

Therefore, the training of variational autoencoders involves the training of two networks: an encoder and a decoder. The encoder receives raw data x and then outputs the distribution over latent codes z, and the decoder transforms the latent code z into data x. The high-level idea is to maximize $p_\theta(x)$. The training goal is derived by maximizing $p_\theta(x)$ as follows:

$$\begin{aligned}
\log p_\theta(x) &= \log\frac{p_\theta(x|z)p(z)}{p_\theta(z|x)} = \log\frac{p_\theta(x|z)p(z)q_\phi(z|x)}{p_\theta(z|x)q_\phi(z|x)} \\
&= \log p_\theta(x|z) - \log\frac{q_\phi(z|x)}{p(z)} + \log\frac{q_\phi(z|x)}{p_\theta(z|x)} \\
&= E_z[\log p_\theta(x|z)] - E_z[\log\frac{q_\phi(z|x)}{p(z)}] + E_z[\log\frac{q_\phi(z|x)}{p_\theta(z|x)}] \\
&= E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z)) + D_{KL}(q_\phi(z|x), p_\theta(z|x)).
\end{aligned} \tag{3.6}$$

Since the KL divergence is always greater or equal to 0, we can drop the last term, and then obtain the lower bound of $\log p_\theta(x)$:

$$\log p_\theta(x) \geqslant E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z)), \tag{3.7}$$

where the first term can be regarded as the data reconstruction objective, and the second term is the KL divergence between the prior distribution and the output of the encoder network. The variational autoencoder is trained by maximizing the lower bound as indicated in Eq. (3.7).

In practice, the variational autoencoders are implemented as Fig. 3.2. The image x is firstly encoded by an encoder to the latent representation z. In the decoder part, the latent representation z is transformed to μ and Σ by two fully-connected layers, respectively. A reparameterization trick is then applied to sample $\hat{z}$ from the distribution $\mathcal{N}(\mu, \Sigma)$. Finally, the sampled $\hat{z}$ is fed into the decoder to reconstruct the image x. The loss function in Eq. (3.7) is then formulated as the follows:

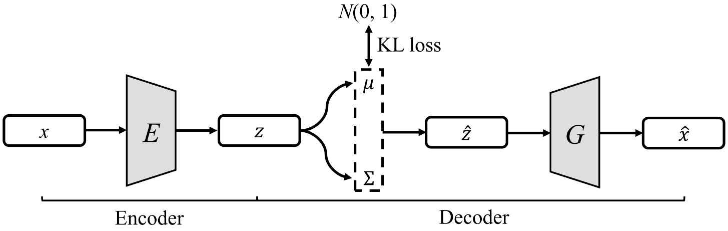


Fig. 3.2 Illustration of the VAE pipeline. VAE consists of an encoder and a decoder. The encoder extracts the input image x into the latent representation z. As for the decoder part, z is firstly transformed to μ and Σ, which are then reparameterized into $\hat{z}$. $\hat{z}$ is finally fed into the generator to reconstruct the image. KL loss is applied to align the z with the standard normal distribution

$$\mathcal{L} = \left\|x - \hat{x}\right\|_2^2 - KL(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mathbf{0}, \mathbf{1})) \tag{3.8}$$

3.2.2 Vector-Quantized Variational AutoEncoder

Vector-Quantized Variational AutoEncoder (VQVAE) is a variant of variational autoencoder. Compared to VAE, it has two main modifications: 1) the output of the encoder network is discretized; 2) the prior distribution $p(z)$ is explicitly learned rather than fixed. The motivation for these modifications is the "posterior collapse" issue in VAE. In the original VAE, if the decoder is too powerful, the sampled latent z will be ignored.

As shown in Fig. 3.3, the overall architecture of VQVAE is also an encoder–decoder-based one. The encoder takes the raw image as the input, and outputs an embedding $z_e(x)$. The continuous embedding $z_e(x)$ is then discretized by finding its nearest neighbor in the codebook $e \in R^{K \times D}$. Then the decoder takes the discrete embedding $z_q(x)$ as the input and reconstructs the raw data. Mathematically, $z_q(x)$ is expressed as

$$z_q(x) = \{e_k, k = \text{argmin}_j \left\|z_e(x) - e_j\right\|_2\}. \tag{3.9}$$

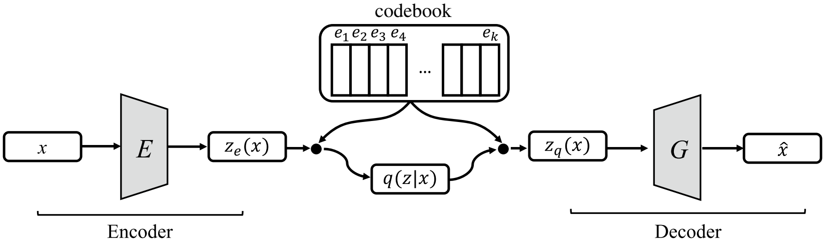


Fig. 3.3 Illustration of VQVAE. VQVAE consists of an encoder, a codebook, and a decoder. The encoder E extracts the input image x into $z(x)$. The $z(x)$ is quantized by the codebook by finding the nearest representation stored in the codebook. The quantized feature $z_q(x)$ is finally reconstructed by the decoder G

Since the $argmin$ operation is involved, there is no gradient for Eq. (3.9). The straight-through estimator [5] is employed to approximate the gradient. The gradient from decoder to $z_q(x)$ is copied to the output of encoder $z_e(x)$. The overall training objective is defined as follows:

$$\mathcal{L} = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta \|z_e(x) - \text{sg}[e]\|_2^2, \tag{3.10}$$

where $\text{sg}(\cdot)$ is the stop-gradient operation. The objective consists of three terms. The first term is the reconstruction loss to optimize the encoder and decoder. The second term is to learn the embeddings in the codebook. The third term is the commitment loss to add constraints to the output of the encoder.

Once the VQVAE is trained, images can be generated by sampling from the latent embeddings of the well-trained VQVAE. Recall that in VAE, the goal of data generation is to maximize $p(x)$ in Eq. (3.4).

In VQVAE, the posterior distribution $p(z|x)$ is defined as follows:

$$p(z|x) = \sum_k q(z = k|x), \tag{3.11}$$

$$q(z = k|x) = \begin{cases} 1 & \text{for } k = \text{argmin}_j \left\| z_e(x) - e_j \right\|_2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

Since the $q(z|x)$ is a constant, the log-likelihood of $p(x)$ can be expressed as follows:

$$\begin{aligned} \log p(x) &= \log \sum_k p(x|z_k)p(z_k), \\ &\approx \log p(x|z_q(x))p(z_q(x)). \end{aligned} \tag{3.13}$$

The $p(x|z_k)$ is the decoder of the VQVAE. To sample images, the rest is to train a network to approximate $p(z_q(x))$. In vanilla VQVAE, PixelCNN [59, 91], a type of autoregressive model, is employed to learn the distribution of $p(z)$. PixelCNN is trained to autoregressively predict the tokens, i.e., the indices of the learned codebook, to sample the desired images.

3.2.3 Variants of VAE and VQVAE

A variant of VAE, β-VAE [27], is proposed to make the latent space of VAE interpretable in an unsupervised manner. In the learning of β-VAE, higher weights are assigned to the KL divergence in Eq. (3.7). The core idea is that the larger weight limits the capacity of z, forcing the model to disentangle the most representative features from the original data. CVAE [80] enables VAE to generate data x conditioned on the class labels c. An intuitive idea is to feed the class labels together with the input x into the encoder and decoder during the training. Then the optimization of $p_\theta(x)$ becomes the optimization of $p_\theta(x|c)$. Tomczak et al. [87]

extended the VAE into a hierarchical architecture with multiple priors, which learns more powerful latent representation and avoids unused dimensions in the latent space.

Many variants of VQVAE are proposed to boost the performance. The modifications fall into two main categories: (1) improving the training of the encoder and decoder of VQVAE and (2) improving the learning of the prior distribution. Esser et al. [18] added adversarial loss (explained in Sect. 1.3) to the training of VQVAE to make the reconstructed images contain more high-frequency details. Other works [34, 69] improve the reconstruction quality by using hierarchical encoders and decoders. As for the improvement of prior learning, transformers [18] are employed to replace the PixelCNN design. In transformers, the self-attention mechanism is adopted to better model the correlations between features and thus enables better results than PixelCNN. Some works [8, 11, 17, 23, 34] propose to use the diffusion models instead of the autoregressive models. The diffusion models gradually predict the tokens in a bidirectional way. Compared to the unidirectional prediction in the autoregressive models, which only predicts the incoming pixels conditioned on the previous contexts, the bi-directional prediction allows the prediction conditioned on the whole contexts, thus making the sampled image more coherent in content.

One application of VQVAE is DALLE [68], a large pretrained model for text-to-image generation. The training set of DALLE contains 3.3 million text–image pairs and it achieves a wonderful performance in text–image generation. The training of DALLE consists of two stages: (1) Stage 1 for a VQVAE to compress images into tokens, i.e., indices in the codebook; (2) Stage 2 for an autoregressive transformer to model the joint distribution over text and images.

In stage 1, the codebook contains 8192 elements. And then $p(z|x)$ in Eq. (3.4) becomes the categorical distribution. To optimize the encoder and decoder, Gumbel-softmax strategy [32, 49] is adopted in DALLE instead of the straight-through estimator in vanilla VQVAE.

In stage 2, DALLE adopts the Byte Pair Encoding (BPE) [19, 75] to encode the texts into tokens, and uses the encoder trained in stage 1 to encode the images into tokens. Then, these two types of tokens are concatenated and modeled together in an autoregressive way. The transformer used is a decoder-only model. Three kinds of self-attention masks are used in the model. For text-to-text attention, standard causal attention masks are used. For image-to-image attention, row, column, or convolutional attention masks are used. Normalized cross-entropy loss is used to train the transformer.

3.3 Generative Adversarial Networks

3.3.1 Overview

In this decade, the Generative Adversarial Network (GAN) has become one of the most popular generative models and has taken dominance in this field.

In 2014, Goodfellow et al. first proposed the idea of GAN [20]. The core idea is the two-player adversarial game, where a generative model G captures the data distribution, and a discriminative model D estimates the probability that a sample is real or synthetic. Let's consider this analogy to better understand GAN: the generator is a counterfeiter trying to produce counterfeit money, while the discriminator is a cop trying to seize the counterfeit money. In the beginning, both the counterfeiter and the cop have little experience, so the counterfeit money appeared to be very different from the real money. After we tell the cop which money is fake or real, the cop can easily identify the difference between real and fake according to the color for example. Meanwhile, the counterfeiter would learn that the cop distinguish the counterfeit money according to color, so they would try to produce money that has the same color as the real one. In the next round of competition, the cop would find other features to distinguish real and fake money, and in turn, the counterfeiter would try to eliminate the gap of those features between fake and real money. The two players, counterfeiter (generator) and cop (discriminator), adversarially compete against each other and both do an increasingly better job. Goodfellow et al. proved theoretically that the adversarial game has a single outcome, in which the counterfeiter eventually masters the technique of producing counterfeit money that the cop cannot distinguish from the real money. That is, the generator will learn to synthesize samples that fall into the distribution of the real data so that the cop (discriminator) cannot tell.

Prior to the GAN era, researchers used carefully designed loss functions to guide network training. For tasks like classification, detection, and reconstruction, where ground truths are available, there exist well-defined and commonly accepted evaluation metrics. However, for generative models, it remained an open question of how to evaluate the quality of synthesized samples. The birth of GAN provides a new paradigm by introducing a discriminiator network to bypass the need for an explicitly defined sample quality assessment metric. GAN turns the unsolved quality assessment problem to a binary classification problem with well-defined loss and metric. Specifically, the discriminator network predicts whether a sample is real or fake (i.e., synthesized), by which the generator network is encouraged to generate realistic samples that fool the discriminator. GAN learns a good evaluation metric (i.e., the discriminator) from unlabeled data, again unveiling the appeal of the data-driven spirit in deep learning.

3.3.2 Architectures and Losses

As with VAE, GAN generates samples from the latent space. Normally, the generator and the discriminator are denoted by G and D, respectively, as shown in Fig. 3.4a. We would like G's generative distribution p_g to approximate the real data distribution p_{data}. To learn p_g over data $x \sim p_{\text{data}}$, we define a prior on input latent variables p_z, which is usually a standard normal distribution, then G aims to map $z \sim p_z$ to data space. D receives an input sample and outputs a single scalar, representing the probability that the sample comes from

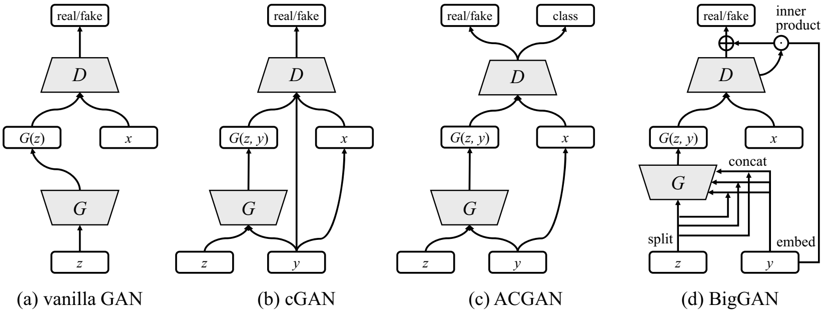


Fig. 3.4 Illustration of network architectures of different GANs. The generator G aims to map random noises z to images that satisfy the distribution of real images x as much as possible so that the discriminator D cannot distinguish the authenticity. For conditional generation, side information y is provided to control the generation process to make the generated images match y

p_{data} rather than p_g. The goal of D is to maximize the probability of assigning the correct label to both real data and fake data, i.e., $D(x) = 1$ and $D(G(z)) = 0$. Meanwhile, G aims to confuse D so that $D(G(z)) = 1$. Therefore, G and D play the following two-player minimax game with the loss function $\mathcal{L}(G, D)$:

$$\min_G \max_D \mathcal{L}_{adv}(G, D) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]. \tag{3.14}$$

To minimize this loss function, the training of GAN is implemented as an iterative approach, alternately optimizing G and D. Each iteration first samples z and x to optimize D while keeping G fixed,

$$\max_D \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]. \tag{3.15}$$

Then z is sampled to optimize G with D fixed,

$$\min_G \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]. \tag{3.16}$$

In the beginning of training when G is poor, D can easily tell the fake data from the real one. In this case, $\log(1 - D(G(z)))$ saturates with tiny gradients to update G. To solve this problem, instead of optimizing Eq. (3.16), we can optimize G with

$$\max_G \mathbb{E}_{z \sim p_z}[\log D(G(z))], \tag{3.17}$$

which is known as the non-saturating GAN loss.

(1) Variation of the loss function. Goodfellow has presented a theoretical analysis of the convergence of GAN in [20]. If G and D have infinite capacity, the adversarial training will converge to a global optimum (Nash-equilibrium) for $p_g = p_{\text{data}}$ and the discriminator is unable to differentiate between the fake data and real data, i.e., $D(\cdot) = 0.5$. However, in practice, the models have limited capacity and the training of GAN is known to be unstable, leading to model collapse where G generates unreasonable and monotonous outputs. Researches have been devoted to improving the loss functions to stabilize the training.

R1 regularization [51]. Mescheder et al. analyzed the instabilities come from oscillation near the Nash-equilibrium [51]. When G is far from p_{data}, D pushes G toward p_{data}. And D becomes more certain, which increases D's gradients. When G is near the Nash-equilibrium, D accumulates strong gradients that pushes G away from the Nash-equilibrium. As a result, training is unstable near the equilibrium point. To solve this issue, Mescheder et al. proposed an R1 regularization [51] to prevent D from creating harmful non-zero gradients orthogonal to the data manifold,

$$\min_D \mathbb{E}_{x \sim p_{data}}[\|\nabla_\psi D(x)\|^2], \tag{3.18}$$

where ψ is the weights of D and the gradients are penalized on real data alone.

Least Squares GAN (LSGAN) [50]. Besides non-saturated GAN loss, LSGAN [50] is proposed to solve the problem of vanishing gradients. The main idea is to replace the binary cross-entropy loss with an L_2 loss. D directly predicts the label to approach the ground truth class label (0 for fake data and 1 for real data) in terms of mean squared error

$$\min_D \mathbb{E}_{x \sim p_{data}}[(D(x) - 1)^2] + \mathbb{E}_{z \sim p_z}[(D(G(z))^2], \tag{3.19}$$

while G aims to confuse D to predict 1 for fake data

$$\min_G \mathbb{E}_{z \sim p_z}[((D(G(z)) - 1)^2]. \tag{3.20}$$

The benefit is that the L_2 loss gives more penalty to large errors, allowing for large model corrections in the beginning of training.

Wasserstein GAN (WGAN) [4]. The original GAN loss uses Jensen–Shannon divergence to evaluate the distribution distance between the real data and fake data. Instead, WGAN proposes to use the Wasserstein distance (also known as Earth-Mover distance), which measures the cost of turning one distribution into another. In WGAN, weight clipping is applied to D to enforce the Lipschitz constraint. Under this constraint, the Wasserstein distance can be approximated by minimizing the WGAN loss

$$\min_D \mathbb{E}_{z \sim p_z}[D(G(z))] - \mathbb{E}_{x \sim p_{data}}[D(x)], \tag{3.21}$$

$$\min_G -\mathbb{E}_{z \sim p_z}[D(G(z))]. \tag{3.22}$$

This measurement provides a useful gradient almost everywhere, allowing continuous and stable training. However, the weight clipping sometimes makes the model hard to converge. To solve this problem, WGAN-GP [24] eliminates weight clipping by introducing a gradient penalty regularization to enforce the Lipschitz constraint. This term penalizes the norm of gradient of D with respect to its input. In real implementation, WGAN-GP enforces a soft version of the constraint with a penalty on the gradient norm for random samples $\hat{x} \sim p_{\hat{x}}$:

$$\min_{D} \mathbb{E}_{z\sim p_z}[D(G(z))] - \mathbb{E}_{x\sim p_{data}}[D(x)] + \lambda_{gp}\mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{3.23}$$

where $\hat{x}$ is defined as a uniform sampling along the straight line between the sampled real data and the dummy data.

Spectral Normalization [54]. Apart from loss functions, the Lipschitz constraint can be also realized by weight normalization. The spectral norm of a matrix is its maximum singular value. Normalize a matrix with its spectral norm can enforce 1-Lipschitz continuity Therefore, spectral normalization that normalizes the weight for each layer of D with its spectral norm is proposed to stabilize training. The advantage is that the computational cost of spectral normalization is small.

(2) Architecture. The vanilla GAN [20] builds both G and D with multi-layer perceptrons, which are not suitable to handle high-dimensional images as in Fig. 3.1.

Deep Convolutional GAN (DCGAN) [67]. DCGAN proposes a successful fully convolutional GAN architecture especially good at dealing with images, where some of the design ideas become the paradigm of the subsequent GAN design:

- Use fractional-strided convolution and strided convolution for upsampling and downsampling. It allows the network to learn its own spatial resampling.
- Remove fully connected layers to speed up convergence.
- Use batch normalization to stabilize training, which helps gradient flow in deeper layers and increase robustness to poor network initialization.
- Use ReLU/LeakyReLU to make the network learn more quickly to cover the color space of the real data distribution.

Therefore, the generator of DCGAN is mainly composed of a set of fractional-strided convolutional (also known as transposed convolution) layer, batch normalization layer, and ReLU layer (the last layer uses Tanh instead). The discriminator mainly contains a set of strided convolutional layer, batch normalization layer, and LeakyReLU layer.

Conditional GAN (cGAN) [53]. Vanilla GAN generates random images without control. By comparison, cGAN introduces extra conditions y such as class labels to control the generation process. Besides the increased controllability, additional information can also decrease the training difficulty and significantly improve the quality of generated samples. Correspondingly, G and D in cGAN are additionally conditioned by y

$$\min_{G}\max_{D} \mathbb{E}_{x,y\sim p_{data}}[\log D(x, y)] + \mathbb{E}_{z\sim p_z, y\sim p_y}[\log(1 - D(G(z, y), y))]. \tag{3.24}$$

For example, cGAN [53] uses an class label y. G is conditioned by receiving a combination of z and y and maps them to an output face image. y is also combined with the real/fake data before sending to D as illustrated in Fig. 3.4b. The combination can be simply concatenations when z, y, and the data have similar dimension, i.e., one-dimensional vectors.

When it comes to images, special combination should considered. In Auxiliary Classifier GAN (ACGAN) [57], the class label y and the noise z are combined as the input of G. The difference is that D is conditioned by y in the form of loss function, as shown in Fig. 3.4c. Specifically, the last layer of D has two branches, one gives a probability distribution over data sources and another gives a probability distribution over the class labels, denoted as $P(C|\cdot)$. Therefore, the loss function has two parts: the original adversarial loss $\mathcal{L}_{adv}$, and the log-likelihood of the correct class $\mathcal{L}_C$:

$$\mathcal{L}_C(G, D) = \mathbb{E}_{z \sim p_z}[\log P(C = y|G(z, y))] + \mathbb{E}_{x \sim p_{data}}[\log P(C = y|x))]. \tag{3.25}$$

Both G and D are trained to maximize $\mathcal{L}_C$.

BigGAN [10]. BigGAN suggests a comprehensive cGAN framework that integrates various branches of GAN improvement. This framework shows its power by being successfully trained on images of 256 × 256 resolution in 1,000 classes of ImageNet [14]. It can generate very different yet plausible objects like animals, vehicles, tools, and foods within a single network. After experimenting with a large number of GAN models and hyperparameters, BigGAN presents several useful suggestions for training GAN:

- Increase batch size: Large batch size allows the network to see more diverse objects, thus obtaining more accurate gradients to speed up the training.
- Increase channels: Increasing the number of channels is equivalent to increasing the model complexity (capability). Another observation of BigGAN is that increasing the model depth does not bring much improvement.
- Noise in each layer: Besides the input layer of G, adding z into all layers makes G learn a hierarchical impact of z on features in different semantic levels.
- Condition in every layer: Instead of conditioning only the input layer of G, y can be added into each layer to enforce the condition. Specifically, as shown in Fig. 3.4d, y is embedded into a condition vector and is concatenated with z_l at layer l. The resulting condition vector of layer l is projected to a gain vector and a bias vector through two fully connected layers. The batch-normalized feature map of layer l is adjusted by multiplying the gain and adding the bias (also known as conditional batch normalization) to realize the condition.

Besides, BigGAN uses projection discriminator [55] for class condition. It takes an inner product between the embedded condition vector and the last-layer feature vector of D, which significantly improves the quality of the class conditional image generation.

3.4 Generative Model in Faces

Previous sections introduce the basic ideas of two kinds of popular generative networks. This section will introduce specialized generative networks for the face generation task, including unconditional face generation and conditional face generation.

The goal of unconditional generation is to synthesize face samples from random Gaussian noises. This task maps the low-dimensional noise vector to the high-dimensional image, which is rather difficult for high-resolution face images. In Sect. 3.4.1, we will introduce a series of high-resolution face image generation models proposed by Karras et al., of which StyleGAN [40] is the most successful face generation model in recent years, regarded as the paradigm by researchers.

The conditional generation task aims to synthesize a face according to the conditional inputs. With different types of inputs as conditions, we can enforce different intensities of controllability of the output images. Therefore, conditional face generation has rich application scenarios and the corresponding researches have also blossomed in recent years. In Sect. 3.4.2, we will give a brief review of their representative ones sequentially according to the type of the condition.

3.4.1 Unconditional Generation

3.4.1.1 ProgressiveGAN

ProgressiveGAN (ProGAN) [37] has greatly improved the quality and variations of generated faces as shown in Fig. 3.1. The three key contributions of ProgressiveGAN are (1) scaling up both the generator network and discriminator network progressively, (2) introducing minibatch standard deviation to increase the variations, and (3) adopting normalization to discourage unhealthy competition between G and D.

As shown in Fig. 3.5, the training of ProgressiveGAN starts from low-resolution images, and gradually shifts to higher resolutions by introducing additional layers to the network. The intuition behind this is that the network focuses on the learning of coarse features first and then learns the fine details. The generator and discriminator have mirror designs and they are scaled up at the same time. The benefits of progressive training lie in that (1) it decomposes the training into several easier sub-procedures and (2) it reduces the training time.

In ProgressiveGAN, the averaged standard deviation over the minibatch is calculated for each feature at each spatial location. The averaged standard deviation is then replicated to all locations in a minibatch, resulting in a one-channel feature map. The feature map is then concatenated with the original features and inserted at the beginning of the last discriminator block, which is empirically found to be optimal.

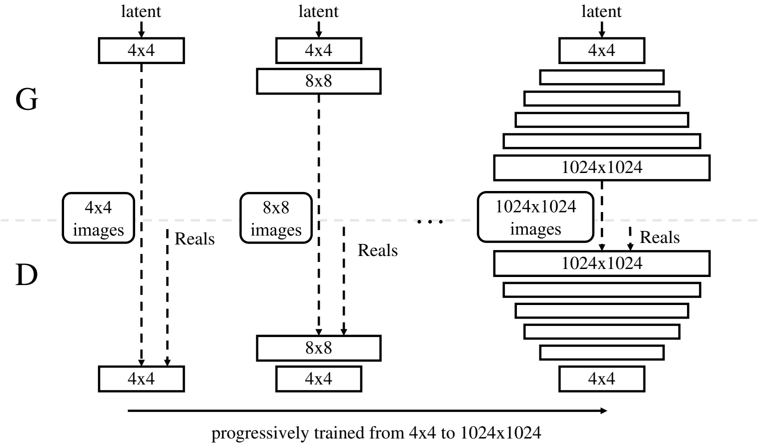


Fig. 3.5 Illustration of Progressive Training Scheme of ProGAN. The generator and discriminator of ProGAN are progressively trained from the smallest resolution (4 × 4) to the largest resolution (1024 × 1024)

To constrain the magnitudes of the features of G and D, pixel-wise feature vector normalization is introduced in the generator. The normalized features are computed by dividing it by the square root of the average of sum of squares of pixel-wise feature values.

3.4.1.2 StyleGAN Series

After ProgressiveGAN, StyleGAN [40] further boosts the quality of generated images. The key contributions of StyleGAN can be summarized as follows:

- Interpretable latent space: StyleGAN makes the learned latent space meaningful. Specifically, by shifting the latent code in one dimension, some semantic attributes of the generated images change accordingly, while others remain unchanged.
- Unsupervised separation of high-level attributes: In StyleGAN, at each convolution layer, there exists a latent code adjusting the "style" of the image.
- Stochastic variation: At each layer, there is also a noise input controlling the stochastic details of the images, e.g., freckles and hair.
- Mixable and interpolatable latent space: The latent space supports the operation of mixing and interpolation of latent codes. By mixing and interpolating of latent codes, the generated images are still photorealistic as shown in Fig. 3.6.

As shown in Fig. 3.7, the StyleGAN generator consists of four key components: (1) A learnable constant input, (2) a mapping network, (3) AdaIN for style modulation, and (4) noise inputs. The whole network starts from a learned constant input. The latent codes z

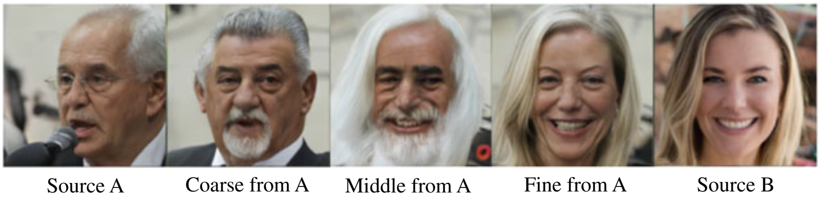


Fig. 3.6 Illustration of Style Mixing. By mixing different levels of latent codes, we can bring different aspects from source B, ranging from high-level information to color schemes and microstructure

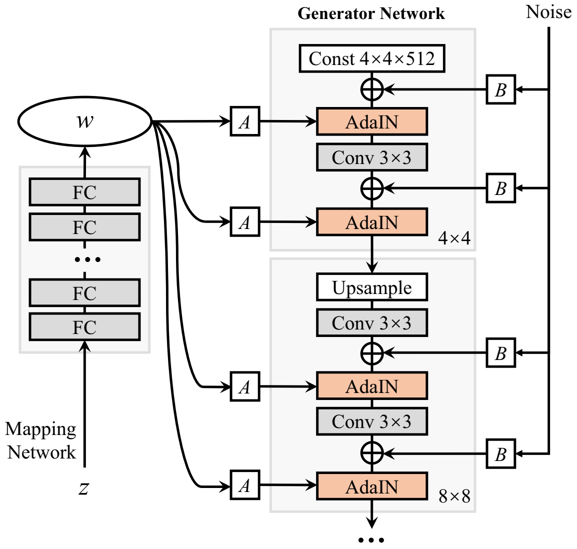


Fig. 3.7 Illustration of the network architecture of StyleGAN. StyleGAN first maps the input z to an intermediate latent space W with a mapping network containing fully convolutional (FC) layers. The mapped latent codes control the generator through AdaIN at each convolution layer. Gaussian noise is added after each convolution. Here, "A" stands for a learned affine transform to map the latent codes to AdaIN modulation parameters, and "B" applies learned per-channel scaling factors to the noise input. "Const" is the constant input feature

are used for layer-wise control. They are first fed into a mapping network to a W space and then modulate the features through AdaIN. There is also a noise input branch to control the stochastic variations. StyleGAN generates images starting from a resolution of 4×4. At each level, the generated images are upsampled $2\times$.

The mapping network maps the latent code z into an intermediate latent space, i.e., W space, and is implemented as an eight-layer MLP. In traditional GANs, the latent code is directly projected to a synthesized image. The necessity of the additional W space is to avoid entanglement of different features. Assuming we only have Z space, the mapping from Z

to the feature must be curved since Z space is sampled from a fixed distribution and some combinations of characteristics may not be present in the training data. If we were to change one property using the curved mapping, other irrelevant properties would also be updated along the curved route. However, the W space does not follow a fixed distribution, and it is sampled from a learned piece-wise continuous mapping. The mapping network "unwarp" the curved mapping to some extent. Thus, the factors of variation become more linear.

Inspired by style transfer, StyleGAN uses an Adaptive Instance Normalization (AdaIN) [29] to make the style vector explicitly control the synthesis. A learned affine transformation is employed to transform the latent code w into AdaIN modulation parameters. With the style modulation module, the StyleGAN has the capability of style localization. It means that when we modify a specific subset of the styles, only certain aspects of the images would be changed.

The noise inputs branch introduces stochastic variation. It provides a direct means to generate stochastic details. If we changed the noise inputs, only some fine details would change. The overall appearance of the images would remain the same. Only the high-frequency details are affected.

One year after StyleGAN was proposed, StyleGAN2 [41] comes out to solve the characteristic blob-like artifacts of StyleGAN. StyleGAN2 points out that StyleGAN exhibits blob-like artifacts that resemble water droplets in the generated face images. These artifacts are caused by the AdaIN. AdaIN normalizes the mean and variance of each feature map separately. Thus, it eliminates the relative size discrepancy of the features, which captures the semantic information between the features. To retain this information, StyleGAN learns to generate some blob-like artifacts with high activation to recover the mean magnitude as well as to deceive the discriminator at the same time (since the artifacts are small). To this end, StyleGAN2 redesigns the AdaIN operation, by removing the normalization to the mean (only normalize variance) and moving the noise outside the AdaIN operation. In addition to the AdaIN, StyleGAN2 points out that the progressive growing tends to have a strong location preference, leading to detail stuck problem. Correspondingly, StyleGAN2 obtains the high-resolution output by upsampling and summing the contributions of the outputs from different low-resolution layers. This allows StyleGAN2 to be trained without changing the network topology to gradually shift its focus from low-resolution images to high-resolution images. An example of the generated face image is shown in Fig. 3.1.

One year later, Karras et al. introduced StyleGAN3 [39] that solves the alias artifacts of StyleGAN series. StyleGAN and StyleGAN2 tend to generate the facial detail glued to pixel coordinates instead of the surfaces of faces, which is not suitable for moving faces. The reason is that operations like upsampling and ReLU in StyleGAN and StyleGAN2 can provide pixel coordinate information for the network to reference when synthesizing details. To solve this problem, StyleGAN3 redesigns these operations to enforce continuous equivariance to sub-pixel translation, i.e., all details are generated equally well regardless of pixel coordinates.

In terms of data, Karras et al. proposed a StyleGAN-ADA [38] to allow for training a StyleGAN model with limited data. StyleGAN-ADA proposes an adaptive discriminator augmentation mechanism to augment data with a wide range of image transformations against overfitting, and adaptively adjust the probability of executing the augmentation to prevent augmentations leak to the generated images (e.g., generator learns to generate unrealistic color-augmented faces). With this mechanism, StyleGAN-ADA can be trained to generate satisfying face images with only a few thousand training images.

3.4.1.3 Advanced Progress of StyleGAN

The success of StyleGAN has attracted much attention, and in recent years a number of advanced advances have been made in the development and application of StyleGAN. In this section, representative work on introducing new style modeling to StyleGAN, finding the latent code of real-world faces, and extending StyleGAN to new domains will be briefly presented. Later in Sect. 3.4.2, face editing with StyleGAN will be further introduced. For a more comprehensive study of StyleGAN, readers may refer to the survey [6].

(1) Style control. StyleGAN provides a flexible control over facial style in hierarchical semantic levels. DiagonalGAN [43] further disentangles the facial style into the content (spatial information such as face pose, direction, expression) and style (other features such as color, makeup, gender). The style follows the original StyleGAN to be modulated with AdaIN. For content feature, DiagonalGAN proposes diagonal spatial attention. Randomly sampled content latent code z_c is first mapped and reshaped to an attention map through MLPs. The attention map is then applied pixel-wisely to manipulate the feature values of a specific location, thus it can control the spatial information of the face. By varying z_c while keeping the style feature fixed, the generated face changes its pose and expression while maintaining the identity, enabling fine-grained control. DualStyleGAN [98], on the other hand, introduces new styles to StyleGAN. The original real face style is defined as intrinsic style while the new cartoon face style is defined as extrinsic style. The extrinsic style latent code is extracted from a reference cartoon image, and is used to generate AdaIN parameters. A modulative ResBlock is proposed to predict the structure adjustment features from StyleGAN features, which are modulated by the AdaIN parameters and are added back to the StyleGAN features to deform the face structures. In this way, random cartoon faces can be generated by varying the extrinsic style latent code and the intrinsic style latent code, as shown in Fig. 3.8c.

(2) StyleGAN inversion. StyleGAN inversion is the inverse process of StyleGAN face generation. It aims to find the latent code of a given real face image. Then flexible face editing can be realized by simply editing its latent code. Image2StyleGAN [1] first explores the latent space of StyleGAN and finds that $W+$ space can better reconstruct real faces. $W+$ space is an extended W space. Recall that the noise vector z is mapped to a 512-dimensional vector w in W space. All layers of StyleGAN uses this same w for style modulation. For $W+$ space, on the other hand, each layer has its own w. For N-layer StyleGAN, all ws constitute

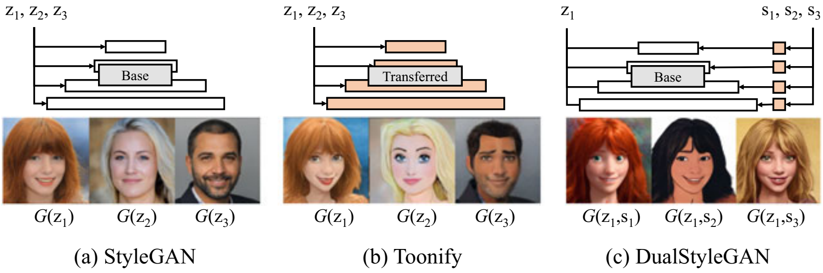


Fig. 3.8 Style control and transfer learning in StyleGAN. **a** Three faces generated by StyleGAN from three latent codes. **b** Toonify fine-tunes the original StyleGAN and generates the corresponding stylized faces using the same latent codes in (**a**). **c** DualStyleGAN introduces extrinsic style codes s into the base StyleGAN model and can render cartoon faces under different styles based on the extrinsic style codes. The white box means the original base StyleGAN model. The orange box means the trained or fine-tuned part of the model

a $N \times 512$ tensor $w+$. In addition, Image2StyleGAN [1] proposes an optimization-based algorithm to find $w+$ of an image I. $w+$ is initialized by a mean latent vector and is optimized via gradient descent to minimize the reconstruction error:

$$\mathcal{L}_{perc}(G(w+), I) + \|G(w+) - I\|_2, \tag{3.26}$$

where $\mathcal{L}_{perc}$ is the perceptual loss [36]. PIE [85] additionally considers the face quality after editing during the optimization to ensure high-fidelity and editibility. IDinvert [108] trains an encoder to map the face to an initial highly editable latent code and requires the optimized $w+$ to be close to the initial one to maintain editibility. However, the optimization makes inversion time-consuming. To speed up, pSp [70] proposes to train an encoder to directly map face images to the $W+$ space. It follows the hierarchical characteristics of StyleGAN to extract multi-level features to predict the latent code in corresponding layers. Later, e4e [88] improves pSp [70] by constraining the predicted $w+$ to lie close to the W space, which avoids over-fitting to the target that hampers editibility. The encoder-based methods are hard to capture the uncommon details like accessories and background. To preciously reconstruct the details, Restyle [2] iteratively predicts the residue of $w+$ wile HFGI [95] predicts the residual mid-level StyleGAN feature maps. Instead of predicting residues, Pivotal Tuning [71] proposes to fine-tune StyleGAN over the target face image. Hyperinverter [16] and Hyperstyle [3] accelerate Pivotal Tuning with a hyper network to directly predict offsets of StyleGAN weights to simulate the fine-tuning process.

(3) Transfer learning. Besides real faces, pretrained StyleGAN can be efficiently extended to face-related domains like cartoon faces and oil-painting portraits with limited data. Toonify [63] fine-tunes a pretrained StyleGAN on cartoon face dataset. After a few iterations, the network is able to generate plausible cartoon faces. Moreover, StyleGAN models before and after fine-tuning are well aligned as systematically analyzed in StyleAlign [96].

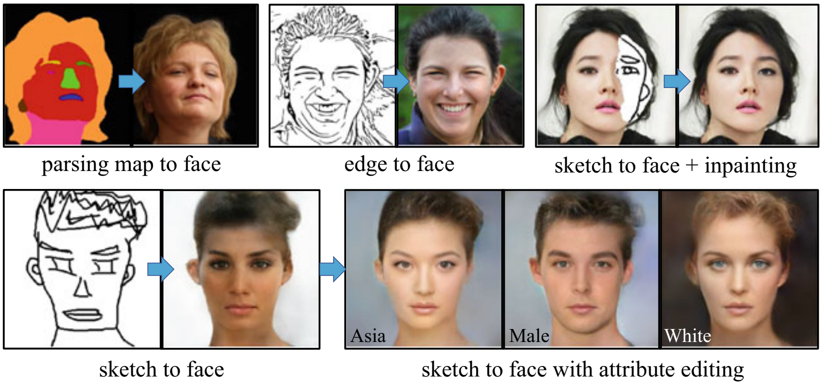
For example, real eyes will be smoothly transformed to cartoon-like eyes during transfer learning, as shown in Fig. 3.8a, b. Therefore, one can perform StyleGAN inversion over a real face with the original StyleGAN and apply the resulting latent code to the fine-tuned StyleGAN to obtain its cartoon version, which is called “toonify”. AgileGAN [81] extends z to $z+$ just as w to $w+$ and experimentally finds that $z+$ strikes good balance between the fidelity and the quality of toonify. FS-Ada [58] investigates a more challenging task of fine-tuning StyleGAN with extremely limited images (e.g., 10) of the target domain. The main idea is to preserve the relative pairwise distances before and after fine-tuning and to combine global discriminator and local patch discriminator, where global discriminator is only used near the real samples. Although real images are limited, real patches extracted from the images are abundant to train the local discriminator. The two strategies effectively prevent mode collapse.

3.4.2 Conditional Generation

With increasingly promising unconditional face synthesis techniques, it is natural to explore how to synthesize a face image according to user requirements. To make face synthesis more user-friendly, recent works have been studying the face synthesis task conditioned on various modalities, such as semantic segmentation masks [110], sketches [100], and texts [97]. Conditional face generation has potential applications in avatar generation, criminal profiling and photo editing.

3.4.2.1 Reference-to-Face

Reference-to-face generation aims to synthesize a face image based on reference images. The commonly used reference images contain face parsing maps, sketch/edge images and partially masked face image as shown in Fig. 3.9.


parsing map to face

edge to face

sketch to face + inpainting

sketch to face

sketch to face with attribute editing

Fig. 3.9 Applications of reference-to-face translation. Images from [101] and [44]

(1) Parsing map to face. Face parsing is an important topic in face recognition to predict the parsing map of a face. The parsing map is the semantic mask of a face image, where each channel is a binary mask indicating the region of a facial component such as the nose or eyes. Generating a face image from a face parsing map, also known as parsing-to-face translation, is the reverse process of face parsing, where parsing maps can serve as a suitable intermediate representation with strong spatial constraints for flexible face manipulation. With the collection of large-scale face parsing dataset like CelebA-Mask-HQ [44], this task can be achieved by simply applying general image-to-image translation model like pix2pix [31]. Parsing-to-face translation is an ill-posed problem where the conditional parsing map only provides facial structure information and may correspond to different appearances. To tackle this problem, a common practice is to simultaneously model the appearances information and combine it with the structure information. For example, in mask-guided portrait editing [22], the authors train an Autoencoder for each facial component to extract its appearance features. These features are rearranged to the target position according to parsing map, and are decoded to generate the corresponding face. Once trained, the method can synthesize novel faces with the appearance from one face and the structure from another. With the introduction of StyleGAN [40], AdaIN is one of the most popular way of modeling appearance feature and combining it with the structure feature. The pSp [70] directly trains an encoder to project the parsing map into the latent codes of StyleGAN, which are then used to reconstruct the original face image. Since the structure style and the texture style are mainly modeled by the first 7 layers and the last 11 layers of StyleGAN, applying different appearance features to the same parsing map can be effectively achieved by altering the latent codes of the last 11 layers. Since the structure and appearance are modeled as latent codes globally and independently in different layers, fine-grained local editing such as only changing the hair styles is not feasible. SEAN [110] proposes a semantic region-adaptive normalization for local editing. In terms of appearance features, the style encoder encodes the input face image into a style feature map. The feature map's elements within the parsing region of each facial component are globally averaged to form a style vector for the corresponding component. In terms of the structure features, the parsing map is mapped to a structure feature map. Within each layer of the generative network, the style vectors are broadcast to the corresponding positions based on the facial component they belong to. The broadcast style vectors and the structure feature map are projected to mean and variance style codes to adjust the current layer's feature map with AdaIN. By this implementation, each facial component has its unique style code that only functions in its own region. User can conveniently edit the style code or region shape to precisely adjust the local appearance.

(2) Sketch/edge to face. Sketch is a kind of more abstract intermediate representation to provide facial structure constraints than parsing maps. With the increasing popularity of touch-screen devices, drawing sketches become more convenient to reflect users' idea. Therefore, researches have been devoted to the problem of sketch-to-face translation. Since it is difficult to collect large-scale hand-drawn sketches, most sketch-to-face translation methods use edges extracted from real faces with post-processing like simplification or vectorization instead. As with parsing-to-face translation, sketch-to-face translation is an ill-posed problem and needs conditions to specify the appearance information. The pSp [70] directly trains an encoder to project the simplified edge into the latent codes of StyleGAN. Structural styles in first 7 layers are controlled by the edges and other appearance styles in last 11 layers are modulated as in original StyleGAN. ArtEditing [90] explicitly simulates the artistic workflows to map a sketch image to a flat color face image and further to the detailed face image. The three-stage framework allows flexible editing in different stages. In this work, the appearance condition is the AdaIN style codes extracted from a real face by a style encoder or derived from Gaussian noise samplings. CocosNet [103] and CocosNetv2 [106] directly find the correspondence between the edge map and the reference face image by projecting them into an aligned feature domains. Then, pixels in the reference image can be warped to the corresponding positions in the edge map to synthesize a rough face image, which guides the generation of the final high-quality output.

When editing a local region of real faces, there is no need to draw the entire sketch. A more practical way is to mask out the target region and only draw the sketch in the masked region. This task is a combination of sketch-to-face translation and image inpainting. A common way is to train an inpainitng network to map the concatenation of the mask, the masked face and the sketch to the original face as in Deepfillv2 [102]. The network learns to generate face structures in the masked region based on the sketch, while maintaining appearance consistency with the unmasked face. Faceshop [64] and SC-FEGAN [35] further introduce color strokes as extra conditional inputs to guide the color styles in the masked region.

Networks trained on edges may fail on low-quality sketches. To bridge the domain gap between the edge and sketch, ContextualGAN [48] trains a GAN to learn a joint distribution of high-quality faces and edges. Then it searches nearest neighbors to the input sketch in this distribution. Similarly, DeepVideoFaceEditing [46] leverages the powerful StyleGAN for face editing. It extends StyleGAN to a joint face and edge generation, and optimizes the latent code of an image to make its edge approximate the edited edge. DeepPS [100], on the other hand, models sketches as dilated and distorted edges to learn a mapping between the low-quality sketches and the fine edges. It serves as a pre-processing tool for translation networks trained on edges.

3.4.2.2 Attribute-to-Face

Attribute-to-face translation aims to generate a high-quality face image that follows the given facial attributes such as smiling, open mouth, old age and black hair. Before the proposal of StyleGAN, researchers focus on training an image-to-image translation network to map an input face image to another that satisfies the target face attribute, i.e., attribute editing. Recently, interpreting the latent space of pretrained StyleGAN has drawn wide attention because this not only helps discover internal representation learned by StyleGAN but also allows semantic control over the generation process. The main idea of the StyleGAN-based attribute editing is to find an editing latent vector for each attribute like smile. By adding this vector to the random style codes, their generated faces will be the smiling versions of those generated from the original style codes.

(1) Conditional translation methods. Image-to-image translation normally handles mapping between two domains, which involves only two attributes. StarGAN [12] and StarGANv2 [13] proposes to train a single network for multiple attribute editing. Instead of learning a fixed translation between two domains, StarGAN takes in as inputs both image and attribute label, and learns to flexibly translate the input image to satisfy the corresponding attribute. StarGANv2 further follows StyleGAN to use AdaIN to inject the attribute label information, which efficiently and effectively learns the attribute-related face styles. In DeepPS [101], the authors further integrate AdaIN-based attribute editing into the sketch-to-face pipeline, which provides visual information like colors that is hard to characterize solely by the sketch as shown in Fig. 3.9. IC-Face [89] proposes a face editing framework where a face neutralization network neutralizes the input face to obtain a template face and a face reenactment network is used to map the template face to given attributes. The combination of neutralization and reenactment makes the network better model facial identity and pose/expression. Since it is difficult to obtain the paired face photos of the same person under different attributes, most methods are based on CycleGAN [109] to learn a two-way mapping between domains on unpaired data. Thanks to the invention of StyleGAN that generates high-quality and style controllable face images, StyleGAN2 distillation [93] proposes to train a one-way mapping with paired data generated by StyleGAN, which shows better attribute editing performance than the frameworks that rely on unpaired data.

(2) Unsupervised StyleGAN-based methods. Unsupervised methods [25, 78, 94, 107] statistically analyze the GAN latent space to discover semantically important and distinguishable directions, so that semantic attribute control can be achieved by manipulating the latent codes along the directions found. Low-rank subspaces [107] relates the GAN latent space to the image region with the Jacobian matrix and then uses low-rank factorization to discover steerable latent subspaces. Voynov and Babenko [94] discover interpretable directions in an iterative fashion. First, an edited image is synthesized according to a latent shift. Then a reconstructor predicts the direction and magnitude of the latent shift from the edited image. This process is repeated to find semantically interpretable directions. GANSpace [25] identifies important latent directions based on Principal Component Analysis (PCA) applied in latent space or feature space, and achieves interpretable controls by layer-wise perturbation along the principal directions. SeFa [78] proposes a closed-form factorization algorithm, which merely uses the pretrained weights of the GAN generator to find semantically meaning dimensions in GAN latent space.

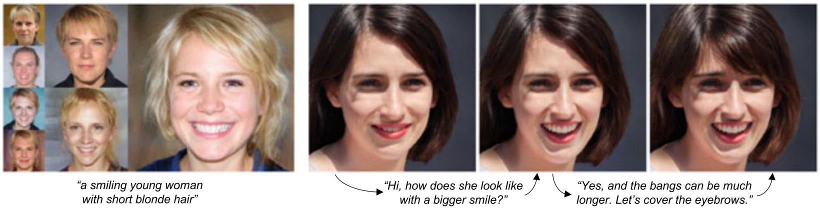
(3) Supervised StyleGAN-based methods. Supervised methods [33, 76, 77, 111] utilize attribute labels of face images, or off-the-shelf attribute classifiers to identify semantically meaningful latent subspaces. For each semantic attribute, InterFaceGAN [76, 77] finds a hyperplane in the latent space to separate its semantics into a binary state, and the normal vector of the hyperplane serves as an editing direction to manipulate the selected attribute. Enjoy-Your-Editing [111] learns a transformation supervised by binary attribute labels, and adds the transformation direction to the latent code to achieve one-step editing. Talk-to-Edit [33] models semantic attribute score as a scalar field with respect to the StyleGAN latent space, and trains a network to learn the gradient to the score scalar field, so that moving the latent code along the learned gradient field lines achieves fine-grained facial editing.

3.4.2.3 Text-to-Face

Given a text description of a face, text-to-face translation aims to generate a high-quality face image that is semantically consistent with the text input. Compared to visual guidance where the constraints are at pixel level, text conditions are more flexible and can be highly semantic. Text-to-face translation is challenging because one text description can lead to a large variety of corresponding images. Therefore, enforcing the image–text consistency plays a central role in text-to-face synthesis.

There are two major types of ways to enforce consistency between the synthesized face image and the input text: (1) train text-to-face translation models directly using image–text pairs (2) utilize a pretrained visual language model (i.e., CLIP [66]) to evaluate the consistency between image and text.

(1) Close-domain text-to-face generation. Close-domain text-to-face generation focuses on conditioning the face generation process on a pre-defined set of text conditions, and is usually achieved by directly training on image–text pairs. TediGAN [97] is a StyleGAN-based method for text-based face generation and manipulation, as shown in Fig. 3.10. It trains a text encoder that maps the text descriptions into StyleGAN's W latent space, where the text embedding is expected to be close to the corresponding image embedding (i.e., the W latent code of the image). The text and image embeddings are then mapped to the $W+$ latent space, and are mixed to obtain the $W+$ code which will be mapped the text-guided image using the StyleGAN generator. Collaborative Diffusion [30] trains text-conditional diffusion models in the latent space of VAE or VQ-VAE following Latent Diffusion Model (LDM) [72]. The diffusion model is implemented using a UNet [73] which gradually denoises the Gaussian


"a smiling young woman
with short blonde hair"
"Hi, how does she look like
with a bigger smile?"
"Yes, and the bangs can be much
longer. Let's cover the eyebrows."

(a) Text-Based Face Generation (b) Text-Based Face Editing

Fig. 3.10 Applications of text-to-face translation. **a** TediGAN [97] synthesizes faces consistent with the text descriptions. **b** Talk-to-Edit [33] performs text-guided facial editing via interactions with users

prior to a natural image. The text conditions are injected into the diffusion model via cross-attention with the UNet's intermediate activations. The denoising process is supervised by text–image pairs.

(2) Open-domain text-to-face generation. Close-domain text-to-face generation falls short when a text description is out of the training text distribution. A common workaround is using CLIP as the bridge between open-world language descriptions and the various facial images. Contrastive Language-Image pretraining (CLIP) [66] aims to efficiently learn visual concepts using natural language supervision. CLIP jointly optimizes an image encoder and a text encoder to encourage a high dot product between the image embedding and text embedding from a positive image–text pair, and a low dot product of that from a negative pair. Since CLIP provides a score measuring the similarity between a pair of image and text, this score can serve as supervision signals to ensure image–text consistency in the text-to-face generation task. StyleCLIP [62] is a text-based face manipulation framework, which first inverts a given face image into StyleGAN's latent code, then uses CLIP loss to optimize the latent code in response to a user-provided text prompt. For every text input, DiffusionCLIP [42] fine-tunes the pretrained diffusion model using CLIP loss to produce images consistent with the text prompt. GLIDE [56], a text-to-image diffusion model, explores two techniques to guide diffusion models toward text prompts: CLIP guidance and classifier-free guidance and finds that the latter produces samples with better photorealism and image–text consistency. For CLIP guidance, the diffusion process is adversarially pushed toward the image that has a higher CLIP score (i.e., higher similarity) with the text condition.

3.5 Evaluation of Generative Models

There are several dimensions in which a generative model can be evaluated, such as density estimation, sample quality, and latent representation quality. In the context of face synthesis, what we care about the most is the synthesized face samples. Therefore, this section mainly focuses on evaluation metrics that assess the sample quality. This section also briefly discusses some evaluation methods to assess conditional consistency.

3.5.1 Evaluation Metrics on Sample Quality

Unlike many other tasks where there exists a loss function which can directly evaluate the performance, generative models like GANs rely on a discriminator network to judge the photorealism and do not have a straightforward evaluation objective.

It is hard to define and assess generation, as memorizing the training set would give excellent samples but is clearly undesirable [82]. Human evaluations can be expensive, hard to reproduce, and potentially biased. Therefore, the community turns to the existing quantitative evaluation metrics such as the Inception Score (IS), Fréchet Inception Distance (FID), and Kernel Inception Distance (KID).

3.5.1.1 Inception Score (IS)

Inception Score (IS) [74] can be used if we have a pretrained classifier (usually the InceptionNet [84] pretrained on ImageNet [14]) that can predict the class probabilities of each generated sample. There are two components in IS:

$$IS = D \times S, \tag{3.27}$$

where D refers to Diversity and S refers to Sharpness. A higher IS implies better sample quality.

(1) Diversity (D). Diversity characterizes the level of uncertainty which class a generated sample belongs to.

$$D = exp(H(y)) = exp(-E_{x\sim p_g}[\int p(y|x)\ log\ p(y)dy]), \tag{3.28}$$

where p_g is the distribution of generated samples. A higher diversity means $p(y)$ has higher entropy $H(y)$, indicating less severe mode collapse.

(2) Sharpness (S). Sharpness is defined as the negative entropy of the distribution of labels predicted by the classifier on the generated images.

$$S = exp(-H(y|x)) = exp(E_{x\sim p_g}[\int p(y|x)\ log\ p(y|x)dy]), \tag{3.29}$$

A higher sharpness means the classifier's predictive distribution $p(y|x)$ has very low entropy. That is, individual-generated samples are classified into corresponding classes with high confidence.

IS can also be viewed as the KL-divergence (on the exponential scale) between the conditional class distribution $p(y|x)$ and the marginal class distribution over the generated data $p(y) = E_{x\sim p_g}[p(y|x)]$.

3.5.1.2 Fréchet Inception Distance (FID)

IS only requires samples from the generated distribution p_g and does not take the desired true data distribution p_{data} into consideration.

To solve this problem, Fréchet Inception Distance (FID) [26] measures the feature representation's similarities between generated samples from p_g and the real samples in the dataset p_{data}. The feature refers to those extracted by a pretrained classifier, usually the InceptionNet [84]. A lower FID implies better sample quality.

To compute FID, we extract the features F_g and F_{data}, and then fit a multivariate Gaussian to each feature representation, giving $\mathcal{N}(\mu_g, \Sigma_g)$ and $\mathcal{N}(\mu_{data}, \Sigma_{data})$. FID is defined as the Wasserstein-2 distance between the two Gaussians [82]:

$$FID = ||\mu_{data} - \mu_g||^2 + Tr(\Sigma_{data} + \Sigma_g - 2(\Sigma_{data}\Sigma_g)^{1/2}) \tag{3.30}$$

3.5.1.3 Kernel Inception Distance (KID)

Kernel Inception Distance (KID) [7] computes the Maximum Mean Discrepancy (MMD) over the feature representations extracted by a pretrained classifier (e.g., InceptionNet [84]). It is more computationally expensive than FID, but can provide a less biased estimation than FID. MMD is defined as follows:

$$\begin{aligned} MMD(p_g, p_{data}) = E_{x_1,x_2 \in p_g}[K(x_1, x_2)] + E_{x_1,x_2 \in p_{data}}[K(x_1, x_2)] \\ -2E_{x_1 \in p_g, x_2 \in p_{data}}[K(x_1, x_2)], \end{aligned} \tag{3.31}$$

where K is a kernel function that measures similarity between points, such as the Gaussian kernel.

3.5.1.4 Discussions on Evaluation Metrics

IS and FID are currently very commonly adopted. IS can capture inter-class diversity but fail to capture intra-class diversity, and is sensitive to the prior class distribution [9]. FID is very popular due to its consistency with human perception [9]. KID is a more recent technique and is becoming adopted.

As face synthesis is a qualitative task, these quantitative metrics are still not perfect indicators of the quality of face synthesis models. In many research publications, human evaluation on samples is also provided as an additional indicator for sample quality. There has also been research [61] that shows subtle differences in image processing can result in large variations in FID values. We look forward to the next decade for future research on stable and perceptually consistent quality assessment systems for generative models.

3.5.2 Evaluation Metrics on Consistency

In face synthesis applications, sample quality is not the only thing to care about. For example, in conditional synthesis tasks, it is important that the synthesized face is consistent with the provided condition. In face editing tasks, a good method should make sure that only the desired attributes are changed, while the other attributes and the face identity are preserved. With these considerations, there have been some evaluation metrics on consistency which are commonly adopted by the academia.

(1) Text–image consistency. For text-driven tasks, the CLIP [66] score is often used to assess image–text consistency.
(2) Attribute-image consistency. To evaluate image-attribute consistency, a classifier for the specific attributes is usually trained to predict the attributes of the synthesized image. The predictions are then compared with the target attribute labels to indicate consistency.
(3) Parsing-image consistency. For semantic parsing (i.e., segmentation masks) to face synthesis, a face parsing model is usually trained to measure the consistency between the input semantic layout and the synthesized image in a pixel-wise manner.
(4) Identity preservation. In reference-to-face tasks and most of the face editing tasks, it is desired to maintain the same identity (i.e., whose face it is). Most practices choose to use ArcFace [15] to measure the identity similarity between the synthesized face and the reference/input face. ArcFace uses Additive Angular Margin Loss to obtain highly discriminative features for face recognition, and is a commonly used method to distinguish different identities.
(5) Others. Image-condition consistency assessment is also accompanied by user study, where users are given questionnaires to judge the consistency between a pair of conditions and synthesized face.

3.6 Summary

This chapter reviews state-of-the-art generative network models, especially their progress on face generation tasks. Specifically, we introduce VAE and GAN, two most popular models in recent years. The former proposes to encode the image domain to a certain distribution, showing brilliant performance in the task of text-guided image generation. The latter provides researchers with the classic data-driven idea of learning a robust evaluation metric of realism, which greatly simplifies the design of training objectives for generative models, and is widely used in tasks such as image generation, image restoration, image enhancement, and image editing. Among them, StyleGAN [40] is the most successful face generation model in the second decade of the 21st century and has derived a series of impressive face-related models.

Although face image generation has made great progress in the past decades, there are still several open questions worth exploring.
(1) From 2D to 3D. While we are impressed by the 2D face images generated by StyleGAN, the generative models still have a lot of room for improvement in generating 3D faces and face videos. Compared with the large-scale datasets like CelebA [47] and FFHQ [40] that are widely used in 2D face image generation tasks, high-definition 3D face datasets are highly scarce. This is because the 3D scanning equipment is more expensive and harder to use than the cameras and mobile phones. In recent years, researchers have begun to study 3D generative models. For example, NeRF [52] builds 3D models based on multi-view 2D images of the same object. StyleNeRF [21] and StyleSDF [60] combine StyleGAN and 3D generative models to recover 3D information from 2D face images. Although these 3D

generation methods have made significant progress, the current 3D face model still lacks a comprehensive pipeline of generation, inversion, and editing like 2D faces. Therefore, a valuable research direction in the future is to directly reconstruct 3D structural information from 2D real face images and edit them.

(2) From images to videos. On the other hand, in terms of video generation, the face-swapping technology conditioned by a driving face video is relatively mature [83, 99, 104, 105], but it is still a critical challenge for the network to capture reasonable motion information for unconditional dynamic face generation. Existing works [65, 79, 86] have demonstrated the potential of generating dynamic face videos, but the performance is not as good as face images. Future research may focus on combining dynamic face generation with Transformers [92].

(3) New generative models. We have noticed that the diffusion model has emerged since 2020. The basic idea of the diffusion model is to learn a parameterized Markov chain that iteratively transits Gaussian noise to a plausible image. In Fig. 3.1, we show an example of an artistic portrait generated by stable diffusion [72]. The excellent performance of diffusion models has attracted the interest of researchers, being vibrantly discussed in social media and art communities. We anticipate that the next generation of generative models with new formulations will open up significant opportunities in the next decade.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: how to embed images into the stylegan latent space? In: Proceedings of International Conference on Computer Vision, pp. 4432–4441 (2019)
2. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Restyle: a residual-based stylegan encoder via iterative refinement. In: Proceedings of International Conference on Computer Vision, pp. 6711–6720 (2021)
3. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 18511–18521 (2022)
4. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of IEEE International Conference on Machine Learning, pp. 214–223. PMLR (2017)
5. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
6. Bermano, A.H., Gal, R., Alaluf, Y., Mokady, R., Nitzan, Y., Tov, O., Patashnik, O., Cohen-Or, D.: State-of-the-art in the architecture, methods and applications of stylegan. In: Computer Graphics Forum, pp. 591–611. Wiley Online Library (2022)
7. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. In: Proceedings of International Conference on Learning Representations (2018)
8. Bond-Taylor, S., Hessey, P., Sasaki, H., Breckon, T.P., Willcocks, C.G.: Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In: Proceedings of European Conference on Computer Vision (2022)
9. Borji, A.: Pros and cons of GAN evaluation measures. CoRR (2018)

10. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: Proceedings of International Conference on Learning Representations (2019)
11. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 11315–11325 (2022)
12. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2018)
13. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: diverse image synthesis for multiple domains. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 8188–8197 (2020)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
15. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: additive angular margin loss for deep face recognition. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
16. Dinh, T.M., Tran, A.T., Nguyen, R., Hua, B.S.: Hyperinverter: improving stylegan inversion via hypernetwork. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 11389–11398 (2022)
17. Esser, P., Rombach, R., Blattmann, A., Ommer, B.: Imagebart: bidirectional context with multinomial diffusion for autoregressive image synthesis. Adv. Neural. Inf. Process. Syst. **34**, 3518–3532 (2021)
18. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 12873–12883 (2021)
19. Gage, P.: A new algorithm for data compression. C Users J. **12**(2), 23–38 (1994)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
21. Gu, J., Liu, L., Wang, P., Theobalt, C.: Stylenerf: a style-based 3d aware generator for high-resolution image synthesis. In: Proceedings of International Conference on Learning Representations (2021)
22. Gu, S., Bao, J., Yang, H., Chen, D., Wen, F., Yuan, L.: Mask-guided portrait editing with conditional gans. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 3436–3445 (2019)
23. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 10696–10706 (2022)
24. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems, pp. 5767–5777 (2017)
25. Härkönen, E., Hertzman, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. In: Advances in Neural Information Processing Systems (2020)
26. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems (2017)

27. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations (2017)
28. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst (2007)
29. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of International Conference on Computer Vision, pp. 1510–1519 (2017)
30. Huang, Z., Chan, K.C.K., Jiang, Y., Liu, Z.: Collaborative diffusion for multi-modal face generation and editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
31. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 5967–5976 (2017)
32. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: Proceedings of International Conference on Learning Representations (2017)
33. Jiang, Y., Huang, Z., Pan, X., Loy, C.C., Liu, Z.: Talk-to-edit: Fine-grained facial editing via dialog. In: Proceedings of International Conference on Computer Vision (2021)
34. Jiang, Y., Yang, S., Qiu, H., Wu, W., Loy, C.C., Liu, Z.: Text2human: text-driven controllable human image generation. ACM Trans. Graph. (TOG) **41**(4), 1–11 (2022). https://doi.org/10.1145/3528223.3530104
35. Jo, Y., Park, J.: Sc-fegan: face editing generative adversarial network with user's sketch and color. In: Proceedings of International Conference on Computer Vision (2019)
36. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of European Conference on Computer Vision, pp. 694–711. Springer (2016)
37. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: Proceedings of International Conference on Learning Representations (2018)
38. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: Advances in Neural Information Processing Systems, vol. 33, pp. 12104–12114 (2020)
39. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. Adv. Neural Inf. Process. Syst. **34** (2021)
40. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 4401–4410 (2019)
41. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)
42. Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2022)
43. Kwon, G., Ye, J.C.: Diagonal attention and style-based gan for content-style disentanglement in image generation and translation. In: Proceedings of International Conference on Computer Vision (2021)
44. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: towards diverse and interactive facial image manipulation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 5549–5558 (2020)

45. Li, L., Peng, Y., Qiu, G., Sun, Z., Liu, S.: A survey of virtual sample generation technology for face recognition. Artif. Intell. Rev. **50**(1), 1–20 (2018)
46. Liu, F.L., Chen, S.Y., Lai, Y., Li, C., Jiang, Y.R., Fu, H., Gao, L.: Deepfacevideoediting: sketch-based deep editing of face videos. ACM Trans. Graph. **41**(4), 167 (2022)
47. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (2015)
48. Lu, Y., Wu, S., Tai, Y.W., Tang, C.K.: Image generation from sketch constraint using contextual gan. In: Proceedings of European Conference on Computer Vision, pp. 205–220. Springer (2018)
49. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: a continuous relaxation of discrete random variables. In: Proceedings of International Conference on Learning Representations (2017)
50. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of International Conference on Computer Vision, pp. 2794–2802 (2017)
51. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: Proceedings of IEEE International Conference on Machine Learning, pp. 3481–3490. PMLR (2018)
52. Mildenhall, B., Srinivasan, P., Tancik, M., Barron, J., Ramamoorthi, R., Ng, R.: Nerf: representing scenes as neural radiance fields for view synthesis. In: Proceedings of European Conference on Computer Vision (2020)
53. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
54. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: Proceedings of International Conference on Learning Representations (2018)
55. Miyato, T., Koyama, M.: cgans with projection discriminator. In: Proceedings of International Conference on Learning Representations (2018)
56. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: towards photorealistic image generation and editing with text-guided diffusion models. In: Proceedings of IEEE International Conference on Machine Learning (2022)
57. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of IEEE International Conference on Machine Learning, pp. 2642–2651. PMLR (2017)
58. Ojha, U., Li, Y., Lu, J., Efros, A.A., Lee, Y.J., Shechtman, E., Zhang, R.: Few-shot image generation via cross-domain correspondence. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 10743–10752 (2021)
59. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. Adv. Neural Inf. Process. Syst. **29** (2016)
60. Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, J.J., Kemelmacher-Shlizerman, I.: Stylesdf: high-resolution 3d-consistent image and geometry generation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 13503–13513 (2022)
61. Parmar, G., Zhang, R., Zhu, J.Y.: On aliased resizing and surprising subtleties in gan evaluation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2022)
62. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 2085–2094 (2021)

63. Pinkney, J.N., Adler, D.: Resolution dependent gan interpolation for controllable image synthesis between domains. arXiv preprint arXiv:2010.05334 (2020)
64. Portenier, T., Hu, Q., Szabo, A., Bigdeli, S.A., Favaro, P., Zwicker, M.: Faceshop: deep sketch-based face image editing. ACM Trans. Graph. **37**(4), 99 (2018)
65. Qiu, H., Jiang, Y., Zhou, H., Wu, W., Liu, Z.: Stylefacev: face video generation via decomposing and recomposing pretrained stylegan3. arXiv preprint arXiv:2208.07862 (2022)
66. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of IEEE International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
67. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proceedings of International Conference on Learning Representations (2016)
68. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: Proceedings of IEEE International Conference on Machine Learning, pp. 8821–8831. PMLR (2021)
69. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. Adv. Neural Inf. Process. Syst. **32** (2019)
70. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2021)
71. Roich, D., Mokady, R., Bermano, A.H., Cohen-Or, D.: Pivotal tuning for latent-based editing of real images. ACM Tran, Graph (2022)
72. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)
73. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
74. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems (2016)
75. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1715–1725. Association for Computational Linguistics (2016)
76. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 9243–9252 (2020)
77. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: interpreting the disentangled face representation learned by gans. IEEE Trans. Pattern Anal. Mach, Intell (2020)
78. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in gans. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1532–1540 (2021)
79. Skorokhodov, I., Tulyakov, S., Elhoseiny, M.: Stylegan-V: A continuous video generator with the price, image quality and perks of stylegan2. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 3626–3636 (2022)
80. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. Adv. Neural Inf. Process. Syst. **28** (2015)

81. Song, G., Luo, L., Liu, J., Ma, W.C., Lai, C., Zheng, C., Cham, T.J.: Agilegan: stylizing portraits by inversion-consistent transfer learning. ACM Trans. Graph. **40**(4), 1–13 (2021)
82. Ermon, S., Song, Y.: CS236 - deep generative models. In: Stanford (2021)
83. Sun, Y., Zhou, H., Liu, Z., Koike, H.: Speech2talking-face: inferring and driving a face with synchronized audio-visual representation. In: International Joint Conference on Artificial Intelligence, vol. 2, p. 4 (2021)
84. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2016)
85. Tewari, A., Elgharib, M., Bernard, F., Seidel, H.P., Pérez, P., Zollhöfer, M., Theobalt, C.: Pie: portrait image embedding for semantic control. ACM Trans. Graph. **39**(6), 1–14 (2020)
86. Tian, Y., Ren, J., Chai, M., Olszewski, K., Peng, X., Metaxas, D.N., Tulyakov, S.: A good image generator is what you need for high-resolution video synthesis. In: Proceedings of International Conference on Learning Representations (2021)
87. Tomczak, J., Welling, M.: Vae with a vampprior. In: International Conference on Artificial Intelligence and Statistics, pp. 1214–1223. PMLR (2018)
88. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. ACM Trans. Graph. **40**(4), 1–14 (2021)
89. Tripathy, S., Kannala, J., Rahtu, E.: Icface: interpretable and controllable face reenactment using gans. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision, pp. 3385–3394 (2020)
90. Tseng, H.Y., Fisher, M., Lu, J., Li, Y., Kim, V., Yang, M.H.: Modeling artistic workflows for image generation and editing. In: Proceedings of European Conference on Computer Vision, pp. 158–174. Springer (2020)
91. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: Proceedings of IEEE International Conference on Machine Learning, pp. 1747–1756. PMLR (2016)
92. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
93. Viazovetskyi, Y., Ivashkin, V., Kashin, E.: Stylegan2 distillation for feed-forward image manipulation. In: Proceedings of European Conference on Computer Vision, pp. 170–186. Springer (2020)
94. Voynov, A., Babenko, A.: Unsupervised discovery of interpretable directions in the gan latent space. In: Proceedings of IEEE International Conference on Machine Learning. PMLR (2020)
95. Wang, T., Zhang, Y., Fan, Y., Wang, J., Chen, Q.: High-fidelity gan inversion for image attribute editing. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 11379–11388 (2022)
96. Wu, Z., Nitzan, Y., Shechtman, E., Lischinski, D.: StyleAlign: Analysis and applications of aligned stylegan models. In: Proceedings of International Conference on Learning Representations (2022)
97. Xia, W., Yang, Y., Xue, J.H., Wu, B.: Tedigan: text-guided diverse face image generation and manipulation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2021)
98. Yang, S., Jiang, L., Liu, Z., Loy, C.C.: Pastiche master: exemplar-based high-resolution portrait style transfer. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2022)
99. Yang, S., Jiang, L., Liu, Z., Loy, C.C.: Vtoonify: controllable high-resolution portrait video style transfer. ACM Trans. Graph. (TOG) **41**(6), 1–15 (2022). https://doi.org/10.1145/3550454.3555437

100. Yang, S., Wang, Z., Liu, J., Guo, Z.: Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. In: Proceedings of European Conference on Computer Vision, pp. 601–617. Springer (2020)
101. Yang, S., Wang, Z., Liu, J., Guo, Z.: Controllable sketch-to-image translation for robust face synthesis. IEEE Trans. Image Process. **30**, 8797–8810 (2021)
102. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of International Conference on Computer Vision, pp. 4471–4480 (2019)
103. Zhang, P., Zhang, B., Chen, D., Yuan, L., Wen, F.: Cross-domain correspondence learning for exemplar-based image translation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 5143–5153 (2020)
104. Zhou, H., Liu, Y., Liu, Z., Luo, P., Wang, X.: Talking face generation by adversarially disentangled audio-visual representation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9299–9306 (2019)
105. Zhou, H., Sun, Y., Wu, W., Loy, C.C., Wang, X., Liu, Z.: Pose-controllable talking face generation by implicitly modularized audio-visual representation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 4176–4186 (2021)
106. Zhou, X., Zhang, B., Zhang, T., Zhang, P., Bao, J., Chen, D., Zhang, Z., Wen, F.: Cocosnet v2: full-resolution correspondence learning for image translation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 11465–11475 (2021)
107. Zhu, J., Feng, R., Shen, Y., Zhao, D., Zha, Z.J., Zhou, J., Chen, Q.: Low-rank subspaces in gans. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
108. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: Proceedings of European Conference on Computer Vision, pp. 592–608. Springer (2020)
109. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of International Conference on Computer Vision, pp. 2242–2251 (2017)
110. Zhu, P., Abdal, R., Qin, Y., Wonka, P.: Sean: Image synthesis with semantic region-adaptive normalization. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 5104–5113 (2020)
111. Zhuang, P., Koyejo, O., Schwing, A.G.: Enjoy your editing: controllable GANs for image editing via latent space navigation. In: Proceedings of International Conference on Learning Representations (2021)

Part II
Face Processing Modules

Face Detection

4

Shiqi Yu, Yuantao Feng, Hanyang Peng, Yan-ran Li, and Jianguo Zhang

4.1 Introduction

Face detection is the first step of most face-related applications such as face recognition, face tracking, facial expression recognition, facial landmarks detection, and so on. Face detection is to detect human faces from images and return the spatial locations of faces via bounding boxes as shown in Fig. 4.1. Starting with the Viola–Jones (V–J) detector [53] in 2001, the solution to face detection has been significantly improved from handcrafting features such as Haar-like features [53], to end-to-end convolutional neural networks (CNNs) for better feature extraction. The face detection algorithms have been much faster and more accurate than those of 10 years ago.

S. Yu (✉) · Y. Feng · J. Zhang
Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
e-mail: yusq@sustech.edu.cn

Y. Feng
e-mail: yuantao.feng@opencv.org.cn

J. Zhang
e-mail: zhangjg@sustech.edu.cn

H. Peng
Pengcheng Laboratory, Shenzhen, China
e-mail: philoso_phy0922@163.com

Y. Li
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
e-mail: lyran@szu.edu.cn

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_4

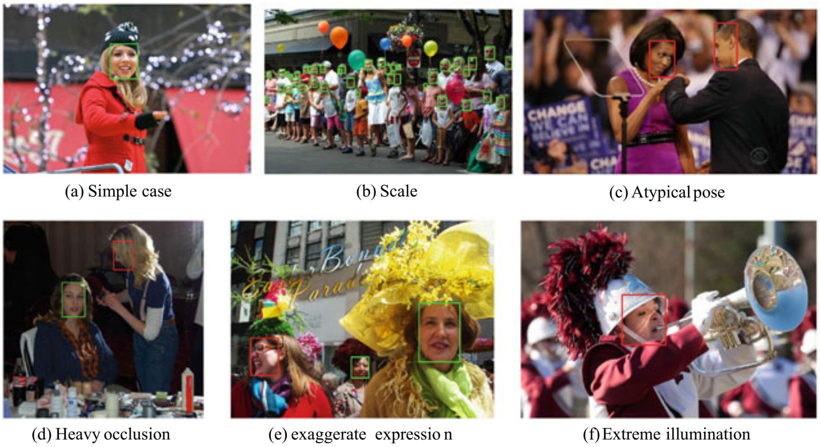


Fig. 4.1 Examples of face detection from WIDER Face [62]. A simple case (**a**) where there is only one clear frontal face. Common variations are in scale (**b**), pose (**c**), occlusion (**d**), expression (**e**), and illumination (**f**). Red boxes indicate faces in difficult conditions

Before deep learning was employed for face detection, the cascaded AdaBoost classifier was the dominant method for face detection. Some algorithms were specifically designed for face detection by using some kinds of features, such as Haar-like features [53], SURF [26] and multi-block LBP [67]. In recent years, deep learning has been proven to be more powerful for feature extraction and helps achieve impressive object detection accuracy. Numerous object detection deep models have been designed for generic object detection which is much more challenging than face detection. Therefore, many models for face detection are adopted from or inspired by models for generic object detection. We can train a deep face detector directly using Faster R-CNN [45], YOLO [44] or SSD [30], and much better detection results can be obtained than traditional cascaded classifiers. Some similar works can be found, such as Face R-CNN [55] and Face R-FCN [57] which are modified and improved based on Faster R-CNN, R-FCN [5], respectively. Additionally, some other detectors, such as MTCNN [66], HR [20], and SSH [41], are originally designed for face detection. Some techniques in generic object detection have also been adapted into face detection, such as the multi-scale mechanism from SSD, the feature enhancement from FPN [27], and the focal loss from RetineNet [28] according to the unique pattern of human faces for face detection. These techniques lead to the proposal of various outstanding face detectors such as S^{3}FD [69], PyramidBox [51], SRN [4], DSFD [25], and RetinaFace [9].

Face detection is sometimes considered a solved problem because of the high average precision (AP) achieved on many face detection datasets such as PASCAL Face [59],

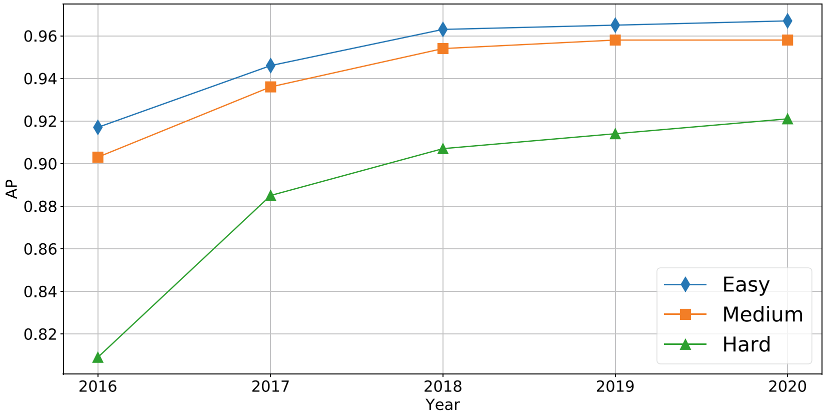


Fig. 4.2 The best APs on the easy, medium, and hard subsets of WIDER Face [62] test set in recent years

AFW [77], and FDDB [21], has reached or exceeded 0.990 since 2017.[1] On the most popular and challenging WIDER Face dataset [62], the AP has reached 0.921 even on the hard test set (Fig. 4.2).

If we look slightly deeper into the implementation of some recent models, we can find that multiple scaling is heavily used in the evaluations on the WIDER face benchmark. If we resize the input image with many different scales, such as 1/4, 1/2, 1, 3/2, 2, 4, and more, and feed all those resized images into a detector, the combined results will have a better AP. It is achieved by assembling and suppressing (using non-maximum suppression or NMS) the multi-scale outputs and is independent of the backbone of the underlying face detector. We listed the scales used by some models in Table 4.1. None of them tested an image using only one scale. It is difficult for users to know by which the improvement is achieved, a better backbone technology, or the follow-up computational-intensive multi-scale ensemble strategy.

We do expect a perfect face detector that is robust and accurate even for some faces in extremely difficult conditions while being extremely fast with low computational cost. However, we all know the *no free lunch theorem*. Therefore, in this chapter, we introduce the recent deep learning-based face detectors and evaluate them in terms of both accuracy and computational cost.

The rest of the chapter is organized as follows. Some key challenges in face detection are summarized in Sect. 4.2. In Sect. 4.3, we provide a roadmap to describe the development of deep learning-based face detection with detailed reviews. In Sect. 4.4, we review several fundamental subproblems including backbones, context modeling, the handling of face scale variations, and proposal generation. Popular datasets for face detection and state-of-

[1] State-of-the-art APs can be found in the official result pages of the datasets, and https://paperswithcode.com/task/face-detection which also collects results from published papers.

Table 4.1 The different strategies for better APs by some state-of-the-art face detectors. “0.25x” denotes shrinking the width and height by 0.25, and others follow. Specifically, “Sx” and “Ex” are shrinking and enlarging images accordingly, while “Fx” is enlarging the image into a fixed size. Test image sizes stand for re-scaling the smaller side of the image to the given value, and the other side follows the same ratio

Method	Test image scales
HR, 2017 [20]	0.25x, 0.5x, 1x, 2x
S3FD, 2017 [69]	0.5x, 1x, Sx, Ex
SRN, 2019 [4]	0.5x, 1x, 1.5x, 2.25x, Fx
DSFD, 2019 [25]	0.5x, 1x, 1.25x, 1.75x 2.25x, Sx, Ex
CSP, 2019 [31]	0.25x, 0.5x, 0.75x, 1x, 1.25x, 1.5x, 1.75x, 2x
Method	Test image sizes
SSH, 2017 [41]	500, 800, 1200, 1600
SFA, 2019 [36]	500, 600, 700, 800, 900, 1000, 1100, 1200, 1600
SHF, 2020 [70]	100, 300, 600, 1000, 1400
RetinaFace, 2020 [9]	500, 800, 1100, 1400, 1700

the-art performances are presented in Sect. 4.5. Section 4.6 reveals the relation between computational cost and AP by conducting extensive experiments on several open-source one-stage face detectors. In addition, speed-focusing face detectors collected from Github are reviewed in Sect. 4.7. Finally, we conclude the chapter with a discussion on future challenges in face detection in Sect. 4.8.

4.2 The Challenges in Face Detection

Most face-related applications need clear frontal faces. Detecting a clear frontal face is a relatively easy task. Some may argue that some faces are useless for the next step such as face recognition if the faces are tiny and with occlusions. But it is not. Effectively detecting any faces in extremely difficult conditions can greatly improve the perception capability of a computer, and it is still a challenging task. If a face can be detected and evaluated as a bad-quality sample, the subject can be suggested to be closer to the camera, or the camera can adjust automatically for a better image. Face detection is still a problem far from being well solved.

Accuracy-related challenges are from facial appearance and imaging conditions. In real-world scenes, there are many different kinds of facial appearances, varying in different skin color, makeup, expression, wearing glasses or facial masks, and so on. In unconstrained environments, imaging a face can be impacted by various lighting, viewing angles and distances, backgrounds, and weather conditions. The face images will vary in illumination, pose, scale, occlusion, blur, and distortion. Some face samples in difficult conditions can be found in Fig. 4.1. There are several datasets and competitions featuring face detection in unconstrained conditions, such as FDDB [21], WIDER Face [62] and WIDER Face Challenge 2019.[2] More than 45% of faces are smaller than 20 × 20 pixels in WIDER. In most face-related applications, we seldom need small faces whose sizes are less than 20. However, if we can detect small or even tiny faces, we can resize the original large images to smaller ones and send them to a face detector. Then, the computational cost can be greatly reduced since we only need to detect faces in smaller images. Therefore, better accuracy sometimes also means higher efficiency.

Masked face detection is becoming more important since people are wearing and will continuously wear masks to prevent COVID-19 in the next few years. Face-related applications did not consider this situation in the past. Wearing masks will reduce the detection accuracy obviously. Some masks are even printed with logos or cartoon figures. All of those can disrupt face detection. If a face is with a mask and sunglasses at the same time, face detection will be even more difficult. Therefore, in the next few years, masked face detection should be explored and studied.

Efficiency-related challenges are brought by the great demands on edge devices. Since the increasing demands on edge devices, such as smartphones and intelligent CCTV cameras, a massive amount of data is generated per day. We frequently take selfies, photos of others, long video meetings, etc. Modern CCTV cameras record 1080P videos constantly at 30 FPS. This results in great demand for facial data analysis, and the data is large. In contrast, edge devices have limited computational capability, storage, and battery to run advanced deep learning-based algorithms. In this case, efficient face detection is essential for face applications on edge devices.

4.3 Popular Face Detection Frameworks

Before deep learning was used for face detection, cascaded AdaBoost-based classifiers [53] were the most popular classifiers for face detection. The features used in AdaBoost were designed specifically for faces, not generic objects. For example, the Haar-like [53] feature can describe facial patterns of eyes, mouth, and others. In recent years, facial features can be automatically learned from data via deep learning techniques. Therefore, many deep learning-based face detectors are inspired by modern network architectures designed for object detection. Following the popular manner of organizing object detection frameworks, we organize deep learning-based face detectors into three main categories:

- Multi-stage face detection frameworks. They are inspired by cascaded classifiers in face detection and are an early exploration of applying deep learning techniques to face detection.

[2] https://competitions.codalab.org/competitions/20146.

- Two-stage face detection frameworks. The first stage generates some proposals for face regions, and the proposals are confirmed in the second stage. The efficiency is better than multi-stage ones most of the time.
- One-stage face detection frameworks. Feature extraction and proposal generation are performed in a single unified network. These frameworks can be further categorized into anchor-based methods and anchor-free methods.

To show how deep learning-based face detection evolves, milestone face detectors and some important object detectors are plotted in Fig. 4.3. The two-stage and multi-stage face detectors are on the top branch, and the single-stage ones are on the bottom branch. The generic object detectors are in the middle branch and in blue. A More detailed introduction to those detectors is provided in the following subsections.

4.3.1 Multi-stage and Two-Stage Face Detectors

In the early era when deep learning techniques entered face detection, face detectors were designed to have multiple stages, also known as the cascade structure which has been widely used in most early face detectors. With the remarkable breakthrough brought by Faster R-CNN [45], some researchers turned to improving Faster R-CNN based on face data.

In the cascade structure, features are usually extracted and refined one or multiple times before being fed into classifiers and regressors, so as to reject most of the sliding windows

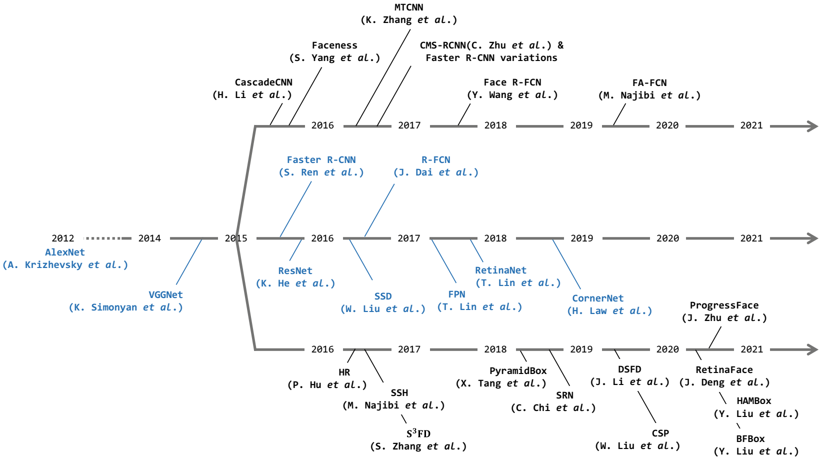


Fig. 4.3 Timeline of milestone face detectors [4, 9, 20, 24, 25, 31, 33, 34, 41, 42, 51, 57, 60, 66, 69, 74, 75], and remarkable works for object recognition [15, 47] and object detection [5, 23, 27, 28, 30, 45] (marked as blue, attached to the middle branch). The top branch is for two/multi-stage face detectors, and the bottom branch is for one-stage ones

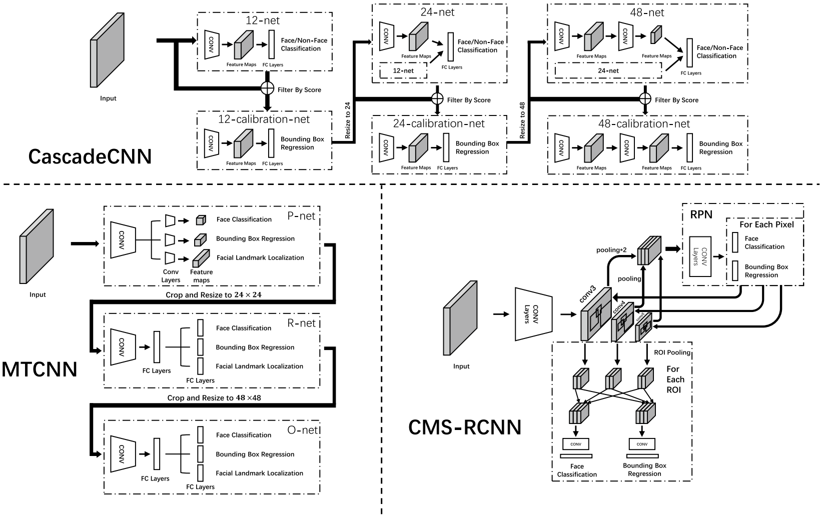


Fig. 4.4 Diagrams of three multi/two-stage face detectors [24, 66, 74]. Most others share similar architectures of these three

to improve efficiency. As shown on the result page[3] of FDDB [21], Li et al. made an early attempt and proposed their CNN-based face detector, named **CascadedCNN** [24]. CascadeCNN consists of 3 stages of CNNs, as shown in Fig. 4.4. Sliding windows are firstly designed to 12 × 12 pixels and fed into the shallow 12-net to reduce candidate windows by 90%. The remaining windows are then processed by the 12-calibration-net to refine the size for face localization. Retained windows are then resized to 24 × 24 as the input for the combination of 24-net and 24-calibration-net, and so on for the next CNNs combination. CascadeCNN achieved state-of-the-art performance on AFW [77] and FDDB [21] while reaching a compelling speed of 14 FPS for the typical 640 × 480 VGA images on a 2.0 GHz CPU. Another attempt at cascaded CNNs for face detection is **MTCNN** [66] proposed by Zhang et al. MTCNN is composed of three subnetworks, which are P-Net for obtaining candidate facial windows, R-Net for rejecting false candidates and refining remaining candidates, and O-Net for producing the final output with both face bounding boxes and landmarks in a multi-task manner. P-Net is a shallow fully convolutional network with six `CONV` layers, which can take images of any size as input. MTCNN was a great success with large and state-of-the-art advantages on WIDER Face, FDDB, and AFW while reaching 16 FPS on a 2.6 GHz CPU.

In the two-stage network architectures, a region proposal network (RPN) [45] is required to generate object proposals. RPN can be considered as a straightforward classification CNN, which generates proposals based on the preset anchors on CNN features, filters out non-object regions, and refines object proposals. However, as the CNNs shrink the image to

[3] http://vis-www.cs.umass.edu/fddb/results.html.

extract features, the corresponding output features for tiny faces can be less than 1 pixel in the feature maps, making it insufficient to encode rich information. To address this problem, Zhu et al. proposed **CMS-RCNN** [74], which is equipped with a contextual multi-scale design for both RPN and final detection. As shown in Fig. 4.4, multi-scale features from *conv3*, *conv4* and *conv5* are concatenated by shrinking them into the same shape with *conv5* as the input for RPN, so as to collect more information for tiny faces and also improve the localization capability from low-level layers. CMS-RCNN achieved an AP of 0.899, 0.874, and 0.624 on the easy, medium, and hard sets of WIDER Face dataset, respectively, and outperforms MTCNN by 0.051 (Easy), 0.049 (Medium), and 0.016 (Hard).

In addition to CMS-RCNN, there are some other improvements based on Faster R-CNN. **Bootstrapping Faster R-CNN** [54] builds a training dataset by iteratively adding false positives from a model's output to optimize Faster R-CNN. **Face R-CNN** [55] adopts the same architecture as Faster R-CNN with center loss, online hard example mining, and multi-scale training strategies. **FDNet** [65] exploits multi-scale training and testing and a vote-based NMS strategy on top of Faster R-CNN with a light-head design. Position-sensitive average pooling was proposed in **Face R-FCN** [57] to assign different weights to different parts of the face based on R-FCN [5]. With the improvements considering the special patterns of face data, those methods achieved better performance than their original version on the same WIDER Face dataset.

Whether it is the cascaded multi-stage or two-stage network design, its computation is heavily dependent on the number of faces in the image, the increase in which also increases proposals passed to the next stage in the interior of the network. Notably, the multi-scale test metric, which usually enlarges the images multiple times to make tiny faces detectable, can dramatically increase the computational cost on this basis. Considering that the number of faces in the image from the actual scene varies from one face in a selfie to many faces in a large group photo, the robustness of cascade or two-stage networks in terms of runtime may be not good.

4.3.2 One-Stage Face Detectors

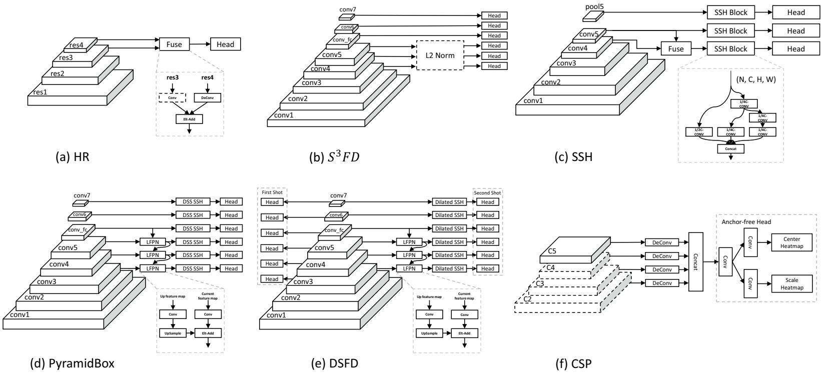
In real-time face-related applications, face detection must be in real time. If the system is deployed on edge devices, the computing capacity is low. In those kinds of situations, one-stage face detectors may be more suitable than others since their process time is stable regardless of the number of faces in an image. Different from the multi-stage or two-stage detectors, the one-stage face detectors perform feature extraction, proposal generation, and face detection in a single and unified neural network. The runtime is independent of the number of faces. Dense anchors are designed to replace proposals in two-stage detectors [41]. Starting from CornerNet [23], many works use the anchor-free mechanism in their frameworks.

4.3.2.1 Anchor-Based Face Detectors

HR [20] proposed by Hu et al. is one of the first to perform anchor-based face detection in a unified convolutional neural network. The backbone of HR is ResNet-101 [15] with layers truncated after `conv4_5`. Early feature fusion on layers `conv3_4` and `conv4_5` is performed to encode context since high-resolution features are beneficial for small face detection. Through experiments on faces clustered into 25 scales, 25 anchors are defined for 2X, 1X, and .5X inputs, to achieve the best performance of three input scales. HR outperformed CMS-RCNN [74] by 0.199 on the WIDER Face validation hard set, and more importantly, the run-time of HR is independent of the number of faces in the image, while CMS-RCNN's linearly scale up with the number of faces.

Different from HR, **SSH** [41] attempts to detect faces at different scales on different levels of features, as shown in Fig. 4.5. Taking VGG-16 [47] as the backbone, SSH detects faces on the enhanced features from `conv4_3`, `conv5_3`, and `pool5` for small, medium, and large faces, respectively. SSH introduces a module (SSH module) that greatly enriches receptive fields to better model the context of faces. The SSH module is widely adopted by later works [9, 25, 34, 51], which turns out to be efficient for performance boosting.

Since **S^{3}FD** [69], many one-stage face detectors [4, 9, 25, 31, 33, 34, 51, 75] fully utilize multi-scale features attempting to achieve scale-invariant face detection. S^{3}FD extends the headless VGG-16 [47] with more convolutional layers, whose stride gradually doubles from 4 to 128 pixels, so as to cover a larger range of face scales. **PyramidBox** [51] adopts the same backbone as S^{3}FD, integrates FPN [27] to fuse adjacent-level features for semantic enhancement, and improves the SSH module with wider and deeper convolutional layers inspired by Inception-ResNet [49] and DSSD [12]. **DSFD** [25] also inherits the backbone from S^{3}FD, but enhances the multi-scale features by the Feature Enhance Module (FEM), so that detection can be made on two shots—one from non-enhanced multi-scale features, and the other from the enhanced features. The same scale features from the second shot not only have larger RFs than those from the first shot but also have smaller RFs than the



(a) HR

(b) S^3FD

(c) SSH

(d) PyramidBox

(e) DSFD

(f) CSP

Fig. 4.5 Diagrams of some one-stage face detectors [20, 25, 31, 41, 51, 69]

next-level features from the first shot, indicating that the face scales are split more refined across these multi-scale detection layers. Similarly, **SRN** [4] has a dual-shot network but is trained differently on multi-scale features: low-level features need two-step classification to refine since they have higher resolution and contribute the vast majority of anchors and also negative samples; additionally, high-level features have a lower resolution which is worth two-step regression using the Cascade R-CNN [2] to have more accurate bounding boxes.

There are also some significant anchor-based methods using the FPN [27] as the backbone. **RetinaFace** adds one more pyramid layer on top of the FPN and replaces CONV layers with the deformable convolution network (DCN) [6, 76] within FPN's lateral connections and context module. RetinaFace models a face in three ways: a 3D mesh (1k points), a 5-landmark mask (5 points), and a bounding box (2 points). Cascade regression [2] is employed with multi-task loss in RetinaFace to achieve better localization. Instead of using the handcrafting structures, Liu et al. proposed **BFBox**, which explores face-appropriate FPN architectures using the successful Neural Architecture Search (NAS). Liu decouples FPN as the backbone and FPN connections, the former of which can be replaced by VGG [47], ResNet [15] or the backbone from NAS, and the latter of which can be top-down, bottom-up or cross-level fusion from NAS.

4.3.2.2 Anchor-Free Face Detectors

Since the proposal of CornerNet [23] back in 2018, which directly predicts the top left and bottom right points of bounding boxes instead of relying on prior anchors, many explorations [52, 63, 71, 72] have been made to remodel object detection more semantically using the anchor-free design. **CSP** models a face bounding box as a center point and the scale of the box as shown in Fig. 4.5. CSP takes multi-scale features from the modified ResNet-50 [15] and concatenates them to take the advantage of rich global and local information for detection heads using transpose convolution layers. In particular, the anchor-free detection head can also be an enhancement module for anchor-based heads. **ProgressFace** [75] appends an anchor-free module to provide more positive anchors for the highest resolution feature maps in FPN, so as to reduce the imbalance of positive and negative samples for small faces.

4.3.2.3 Summary of One-Stage Frameworks

One-stage frameworks are popular in face detection in recent years for the following three reasons. (a) The runtime of one-stage face detectors is independent of the number of faces in an image by design. Therefore, it enhances the robustness of runtime efficiency. (b) It is computationally efficient and straightforward for one-stage detectors to reach near-scale invariance by contextual modeling and multi-scale feature sampling. (c) Face detection is a relatively less complex task than general object detection. This means that innovations and advanced network designs in object detection can be quickly adjusted to face detection by considering the special pattern of faces.

4.4 Feature Extraction for Face Detection

The key idea of face detection has never changed whether it is in the traditional era or in the deep learning era. It tries to find the common patterns of all faces in the training set. In the traditional era, many of the handcrafted features, such as SIFT [35], Haar [53], and HOG [7], are employed to extract local features from the image, which are aggregated by approaches such as AdaBoost for a higher level representation of faces.

Different from traditional methods, which require rich prior knowledge to design handcrafted features, deep convolutional neural networks can directly learn features from face images. A deep learning-based face detection model can be considered as two parts, a CNN backbone and detection branches. Starting from some popular CNN backbones, the feature extraction methods that can handle face scale invariance are introduced as well as several strategies to generate proposals for face detection.

4.4.1 Popular CNN Backbones

In most deep face detectors, there is a CNN backbone for feature extraction. Some popular backbone networks are listed in Table 4.2. They are VGG-16 from the VGGNet [47] series, ResNet-50/101/152 from the ResNet [15] series, and MobileNet [18]. The models are powerful and can achieve good accuracy in face detection, but they are a little heavy since they were not designed directly for face detection.

Some early attempts at deep learning-based face detection are cascaded structures, and the above CNN architectures are not used. Even some simple structured CNN is computationally heavier than AdaBoost, cascaded CNN is even heavier. With breakthroughs in object detection, some of the techniques have been borrowed and applied to face detection.

Table 4.2 CNN backbones are commonly used by some deep learning-based face detectors. FC layers of these CNNs are ignored when calculating "#CONV Layers", "#Params", and "FLOPs". The input size for calculating "FLOPs" is 224 × 224. The calculation of FLOPs is discussed in Sect. 4.6. "Top-1 Error" refers to the performance on the ImangeNet [8] validation set. Note that 9 of the 20 CONV layers in MobileNet [18] are depth-wise

CNN backbones	#CONV layers	#Params ($\times 10^6$)	FLOPs ($\times 10^9$)	Top-1 error (%)
VGG-16	13	14.36	30.72	28.07
ResNet-50	52	23.45	8.25	22.85
ResNet-101	136	42.39	15.72	21.75
ResNet-152	188	56.87	23.19	21.43
MobileNet	20	3.22	1.28	29.40

VGG-16 [47] has 13 CONV layers, which is the first choice for the baseline backbones for many face detectors, such as SSH [41], S^3FD [69], and PyramidBox [51]. Performance improvements can easily be obtained by simply changing the backbone from VGG-16 to ResNet-50/101/152 [15], as shown in [25]. Since the state of the arts have achieved an AP of more than 0.900 even on the WIDER Face hard set, it is a straightforward idea to use a deeper and wider backbone for higher APs [25, 75, 78], such as ResNet-152 and ResNets with FPN [27] connections. Liu et al. employed Neural Architecture Search (NAS) to search face-appropriate backbones and FPN connections.

One inexpensive choice is ResNet-50 listed in Table 4.2. It has fewer parameters and fewer FLOPs while achieving very similar performance compared to some deeper ones. Another choice to reach a real-time speed is to change the backbone to MobileNet [18] and its variants, which have similar performance to VGG-16 but one order of magnitude less in the number of parameters and FLOPs.

4.4.2 Toward Face Scale Invariance

One of the major challenges for face detection is the large span of face scales. As statistics shown in Fig. 4.6, there are 157,025 and 39,123 face bounding boxes in the train set and the validation set, respectively. Both sets have more than 45% of face bounding boxes smaller than 16 × 16, and a non-negligible 1% larger than 256 × 256. We also present the visual differences among scales in Fig. 4.7. It is challenging even for humans to tell whether the image of size 16 × 16 contains a face. In the following, we describe the mechanism of face detectors toward face scale invariance even with tiny faces.

Most modern face detectors are anchor-based. Anchors are predefined boxes of different scales and aspect ratios attached to each pixel in the feature maps. The anchors serve as the

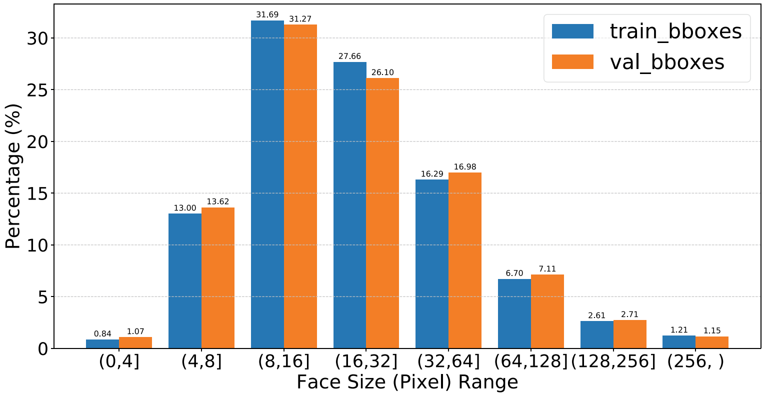


Fig. 4.6 The distribution of face scales on WIDER Face [62] dataset

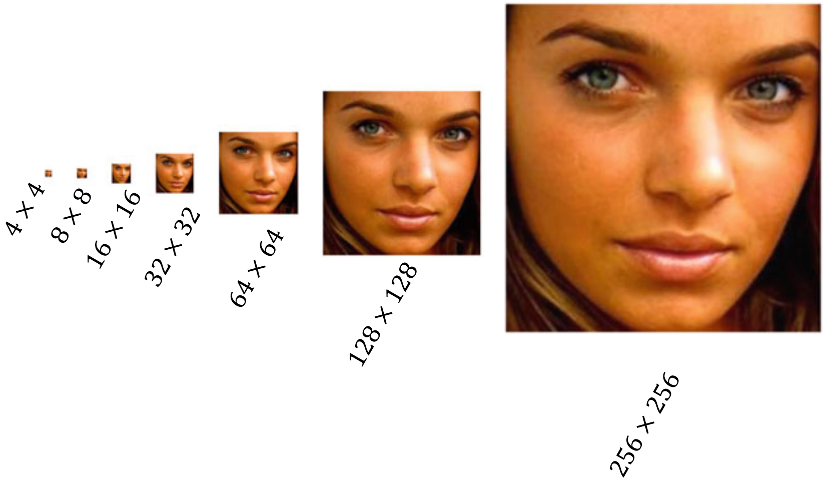


Fig. 4.7 A face in different scales. It is difficult to recognize faces in the images of sizes 4 × 4, 8 × 8

proposals to match with the ground truth faces. More details about anchors are provided in Sect. 4.4.3. As described in [69], since the predefined anchor scales are discrete while the face scales in the wild change continuously, the outer faces whose scales are distributed away from anchor scales cannot match enough anchors. It will result in a low recall rate. A simple solution for a trained face detector is to perform a multi-scale test on an image pyramid, which is built by progressively resizing the original image. It is equal to re-scale faces and hopefully brings outer faces back into the detectable range of scales. This solution does not require retraining the detector. But it may come with a sharp increase in redundant computation since there is no certain answer to how deep the pyramid we should build to match with the certain extent of scale invariance of a trained CNN.

Another solution to face scale invariance is to make full use of the feature maps produced by CNNs. One can easily observe that the layers of standard CNN backbones gradually decrease in size. The subsampling of these layers naturally builds up a pyramid with different strides and receptive fields (RFs). It produces multi-scale feature maps. In general, high-level feature maps produced by later layers with large RFs are encoded with strong semantic information and lead to their robustness to variations such as illumination, rotation, and occlusion. Low-level feature maps produced by early layers with small RFs are less sensitive to semantics but have high resolution and rich details, which are beneficial for localization. To take both advantages, a number of methods are proposed, which can be categorized into **modeling context**, **detecting on a feature pyramid**, and **predicting face scales**.

4.4.2.1 Modeling Context

Additional context is essential for detecting faces, especially for detecting small ones. HR [20] shows that context modeling by fusing feature maps of different scales can dramatically improve the accuracy of detecting small faces. Following a similar fusion strategy as HR, SHF [70] detects in three different dilated CONV branches, aiming to enlarge RF without too much increase in computation. CMS-RCNN [74] downsamples feature maps of strides 4 and 8 to concatenate with those of stride 16 to improve the capability of the RPN to produce proposals for faces at different scales. SSH [41] exploits an approach similar to Inception [50], which concatenates the output from three CONV branches that have 3 × 3, 5 × 5 and 7 × 7 filters, respectively. PyramidBox [51] first adopts an FPN [27] module to build up the context and is further enhanced by deeper and wider SSH modules. DSFD [25] improves the SSH module by replacing CONV layers with dilated CONV layers. CSP [31] upsamples feature maps of strides 8 and 16 to concatenate with those of stride 4, which is fed to an FCN to produce center, scale, and offset heatmaps. The fusion of feature maps encodes rich semantics from high-level feature maps with rich geometric information from low-level feature maps, based on which the detectors can improve their capability of localization and classification toward face scale invariance. Meanwhile, the fusion of feature maps also introduces more layers, such as CONV and POOL to adjust scales and channels, which creates additional computational overhead.

4.4.2.2 Detecting on a Feature Pyramid

Inspired by SSD [30], a majority of recent approaches, such as [4, 9, 25, 41, 51, 69], detect in multiple feature maps of different scales, respectively, and combine detection results. It is considered to be an effective method for weighing between speed and accuracy. SSD [30] puts default boxes on each pixel of the feature maps from 6 detection layers that have strides of 8, 16, 32, 64, and 128. Sharing a similar CNN backbone with SSD, [51, 69] detect on a wider range of layers, which have strides gradually doubling from 4 to 128 pixels. SRN [4] and DSFD [25] introduce the two-stream mechanism, which detects on both the detection layers from the backbone and extra layers applied on the detection layers for feature enhancement. Different from subsampling on more layers, [9, 36, 41] detects only on the last three level feature maps, which are enhanced by their context modeling methods. By detecting on a feature pyramid, detection layers are implicitly trained to be sensitive to different scales, while it also leads to an increase in model size and redundant computation, since the dense sampling may cause some duplicate results from adjacent-level layers.

4.4.2.3 Predicting Face Scales

To eliminate the redundancy from pyramids, several approaches [14, 32, 48] predict the face scales before making a detection. SAFD [14] first generates a global face scale histogram from the input image by the Scale Proposal Network (SPN), which is trained with image-level ground truth histogram vectors and without face location information. A sparse image pyramid is built according to the output histogram, so as to have faces rescaled to the detectable range of the later single-scale RPN. Similarly, RSA [32] detects on a feature pyramid without unnecessary scales, which is built by using the scale histogram to a sequential ResNet [15] blocks that can downsample feature maps recursively. S2AP [48] predicts not only face scales but also face locations by a shallow ResNet18 [15] with scale attention and spatial attention attached, named S^2AP. S^2AP generates a 60-channel feature map, meaning face scales are mapped to 60 bins, each of which is a spatial heatmap that has a high response to its responsible face scale. With the 60-channel feature maps, it is possible to decrease the unnecessary computation with the low-response channels and the low-response spatial areas by a masked convolution.

4.4.3 Proposal Generation

Faces in the wild can be of any possible locations and scales in the image. The general pipeline for most of the early successful face detectors is to first generate proposals in a sliding-window manner, then extract features from the windows using handcrafted descriptors [17, 26, 53, 77] or CNNs [24, 66], and finally apply face classifiers. However, inspired by RPN [45] and SSD [30], modern anchor-based face detectors generate proposals by applying k anchor boxes on each pixel of the extracted CNN features. Specifically, three scales and three aspect ratios are used in Faster R-CNN [45], yielding $k = 9$ anchors on each pixel of the feature maps. Moreover, the detection layer takes the same feature maps as input, yielding $4k$ outputs encoding the coordinates for k anchor boxes from the regressor and $2k$ outputs for face scores from the classifier.

Considering that most of the face boxes are near square, modern face detectors tend to set the aspect ratio of anchors to 1, while the scales are varied. HR [20] defines 25 scales so as to match the cluster results on the WIDER Face [62] training set. S^3FD assigns the anchor scale of 4 times the stride of the current layer to keep anchor sizes smaller than effective receptive fields [37] and ensure the same density of different scale anchors on the image. PyramidBox [51] introduces PyramidAnchors, which generates a group of anchors with larger regions corresponding to a face, such as head and body boxes, to have more context to help detect faces. In [73], extra shifted anchors are added to increase the anchor sample density and significantly increase the average IoU between anchors and small faces. GroupSampling [39] assigns anchors of different scales only on the bottom pyramid layer of FPN [27], but it groups all training samples according to the anchor scales, and randomly samples from groups to ensure the positive and negative sample ratios between groups are the same.

4.5 Datasets and Evaluation

To evaluate different face detection algorithms, datasets are needed. There have been several public datasets, and they are FDDB [21], AFW [77], PASCAL Face [59], MALF [61], WIDER Face [62], MAFA [13], 4K-Face [56], UFDD [40], and DARK Face [3]. Those datasets consist of colored images from real-life scenes. Different datasets may utilize different evaluation criteria. In Sect. 4.5.1, we present an overview of different datasets and cover some statistics such as the number of images and faces, the source of images, the rules of labeling, and challenges brought by the dataset. A detailed analysis of the face detection evaluation criterion is also included in Sect. 4.5.2. Detection results on the datasets are provided and analyzed in Sect. 4.5.3.

4.5.1 Datasets

Some essential statistics of currently accessible datasets are summarized in Table 4.3 including the total number of images and faces, faces per image, how the data was split into different sets, etc. More details are introduced in the following part.

FDDB[4] [21] is short for **F**ace **D**etection **D**ataset and **B**enchmark, which has been one of the most popular datasets for face detector evaluation since its publication in 2010. The images of FDDB were collected from Yahoo! News, 2,845 of which were selected after filtering out duplicate data. Faces were excluded with these factors, (a) height or width less than 20 pixels, (b) the two eyes being non-visible, (c) the angle between the nose and the ray from the camera to the head being less than 90 degrees, (d) failure estimation on position, size or orientation of faces by a human. This led to 5,171 faces left, which were annotated by drawing elliptical face regions covering from the forehead to the chin vertically, and the left cheek to the right cheek horizontally. FDDB helped advance unconstrained face detection in terms of the robustness of expression, pose, scale, and occlusion. However, its images can be heavily biased toward celebrity faces since they were collected from the news. It is also worth noting that although the elliptical style of the face label adopted by FDDB is closer to human cognition, it is not adopted by later datasets and deep learning-based face detectors, which favor the bounding box style with a relatively easier method for defining positive/negative samples by calculating the Intersection over Union (IoU).

Zhu et al. built an annotated faces in-the-wild (**AFW**[5]) dataset [77] by randomly sampling images with at least one large face from Flickr. 468 faces were annotated from 205 images, each of which is labeled with a bounding box and 6 landmarks. **PASCAL Face**[6] [59] was contructed by selecting 851 images from the PASCAL VOC [10] test set with 1,335 faces annotated. Since the two datasets were built to help evaluate the face detectors proposed

[4] http://vis-www.cs.umass.edu/fddb/.

[5] http://www.cs.cmu.edu/~deva/papers/face/index.html.

[6] http://host.robots.ox.ac.uk/pascal/VOC/.

by [77] and [10], they only contain a few hundred images, resulting in limited variations in face appearance and background.

Yang et al. created the **M**ulti-**A**ttribute **L**abeled **F**aces [61] (**MALF**[7]) dataset for fine-grained evaluation on face detection in the wild. The MALF dataset contains 5,250 images from Flickr and Baidu Search with 11,931 faces labeled, which is an evidently larger dataset than FDDB, AFW, and PASCAL Face. The faces in MALF were annotated by drawing axis-aligned square bounding boxes, attempting to contain a complete face with the nose in the center of the bounding box. This may introduce noise for training face detectors since a square bounding box containing 90-degree side faces can have over half of its content being cluttered background. In addition to labeling faces, some attributes were also annotated, such as gender, pose, and occlusion.

In 2016, **WIDER Face**[8] [62] was released, which has been the most popular and widely used face detection benchmark. The images in WIDER Face were collected from popular search engines for predefined event categories following LSCOM [43] and examined manually to filter out similar images and images without faces, resulting in 32,203 images in total for 61 event categories, which were split into 3 subsets for training, validation testing set. To keep large variations in scale, occlusion, and pose, the annotation was performed following two main policies: (a) a bounding box should tightly contain the forehead, chin, and cheek and is drawn for each recognizable face and (b) an estimated bounding box should be drawn for an occluded face, producing 393,703 annotated faces in total. The number of faces per image reaches 12.2 and 50% of the faces are of height between 10 and 50 pixels. WIDER Face outnumbers other datasets in Table 4.3 by a large margin. It means WIDER Face pays never-seen-before attention to small face detection by providing a large number of images with the densest small faces for training, validation, and testing. Furthermore, the authors of WIDER Face defined "easy", "medium", and "hard" levels for the validation and test sets based on the detection rate of EdgeBox [79]. It offers a much more detailed and fine-grained evaluation of face detectors. Hence, the WIDER Face dataset greatly advances the research of CNN-based face detectors, especially the multi-scale CNN designs and utilization of context.

The last four datasets listed in Table 4.3 are less generic than those reviewed above and focus on face detection in specified and different aspects. The **MAFA**[9] [13] dataset focuses on masked face detection, containing 30,811 images with 39,485 masked faces labeled. In addition to the location of eyes and masks, the orientation of the face, the occlusion degree, and the mask type were also annotated for each face. The IJB series[10] [22, 38, 58] were collected for multiple tasks, including face detection, verification, identification, and identity clustering. The IJB-C is the combination of IJB-A and IJB-B with some new face

[7] http://www.cbsr.ia.ac.cn/faceevaluation/.

[8] http://shuoyang1213.me/WIDERFACE/.

[9] http://www.escience.cn/people/geshiming/mafa.html.

[10] https://www.nist.gov/programs-projects/face-challenges.

Table 4.3 Public datasets for face detection. Note that UCCS [1] and WILDEST Face [64] are not included because their data is not currently available. “Blur”, “App.”, “Ill.”, “Occ.”, and “Pose” in the “Variations” columns denote blur, appearance, illumination, occlusion, and pose, respectively

Dataset	#Images	#Faces	#Faces per image	AVG resolution rowspan $(W \times H)$	Split			Variations				
					Train	Val	Test	Blur	App.	Ill.	Occ.	Pose
FDDB [21]	2,845	5,171	1.8	377 × 399	–	–	100%	✓			✓	✓
AFW [77]	205	468	2.3	1491 × 1235	–	–	100%		✓			✓
PASCAL Face [59]	851	1,335	1.5	–	–	–	100%					
MALF [61]	5,250	11,931	2.2	–	–	–	100%		✓		✓	✓
WIDER Face [62]	32,203	393,703	12.2	1024 × 888	40%	10%	50%	✓	✓	✓	✓	✓
MAFA [13]	30,811	39,485	1.2	516 × 512	85%	–	15%		✓		✓	
IJB-A [22]	48,378	497,819	10.2	1796 × 1474	50%	–	50%		✓	✓	✓	✓
IJB-B [58]	76,824	135,518	1.7	894 × 599	–	–	100%		✓	✓	✓	✓
IJB-C [38]	138,836	272,335	1.9	1010 × 671	–	–	100%		✓	✓	✓	✓
4K-Face [56]	5,102	35,217	6.9	3840 × 2160	–	–	100%					
UFDD [40]	6,425	10,897	1.6	1024 × 774	–	–	100%	✓		✓		
DARK Face [3]	6,000	43,849	7.3	1080 × 720	100%	–	–			✓		

data. **4K-Face**[11] [56] was built for the evaluation of large face detection, and contains 5,102 4K-resolution images with 35,217 large faces (>512 pixels). **UFDD**[12] [40] provides a test set with 6,425 images and 10,897 faces in a variety of different weather conditions and degradation such as lens impediments. **DARK Face**[13] [3] concentrates on face detection in low light conditions, and provides 6,000 low-light images for training dark face detector. Since the images are captured in real-world nighttime scenes such as streets, each image in DARK Face contains 7.3 faces on average which is relatively dense.

4.5.2 Accuracy Evaluation Criterion

There are mainly two accuracy evaluation criteria adopted by the datasets reviewed above, one of which is the receiver operating characteristic (ROC) curve obtained by plotting the true positive rate (TPR) against false positives such as those adopted by FDDB [21], MALF [61], UCCS [1], and IJB [38], the other of which is the most popular evaluation criterion from PASCAL VOC [10] by plotting the precision against recall while calculating average precision (AP), such as those adopted by AFW [77], PASCAL Face [59], WIDER Face [62], MAFA [13], 4K-Face [56], UFDD [40], DARK Face [3], and Wildest Face [64]. Since these two kinds of evaluation criterion are two different methods for revealing the performance of detectors under the same calculation of the confusion matrix,[14] we choose the most popular evaluation criteria AP calculated from the precision-again-recall curve in the paper.

To get a precision-again-recall curve, the confusion matrix, which is to define the true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) from the detection and ground truths, should be firstly calculated. A true positive is a detection result matched with a ground truth; otherwise, it is a false positive. The unmatched ground truths are defined as false negatives. True negatives are not applied here since the background can be a large part of the image. To define whether two regions are matched or not, the commonly used intersection over union (IoU), also known as the Jaccard overlap, is applied:

$$IoU = \frac{area(P) \cap area(GT)}{area(P) \cup area(GT)} \tag{4.1}$$

where P is the predicted region, and GT is the ground truth region. In a widely used setting, the IoU threshold is set to 0.5, meaning if the IoU of a predicted region and a ground truth region is greater than or equal to 0.5, the predicted region is marked as matched and thus a true positive, otherwise it is a false positive.

[11] https://github.com/Megvii-BaseDetection/4K-Face.

[12] https://ufdd.info.

[13] https://flyywh.github.io/CVPRW2019LowLight/.

[14] https://en.wikipedia.org/wiki/Confusion_matrix.

After determining true or false positives for each detection, the next step is to calculate the precision and recall from the detection result list sorted by score in descending order to plot the precision-against-recall curve. A granular confidence gap can be defined to sample more precision and recall, but for a simple explanation, we define the gap as a detection result. In nth sampling, we calculate the precision and recall from the top-n detection results:

$$Precision_n = \frac{TP_n}{TP_n + FP_n} \tag{4.2}$$

$$Recall_n = \frac{TP_n}{TP_n + FN_n} \tag{4.3}$$

where TP_n, FP_n and FN_n are true positives, false positives, and false negatives from the top-n results, respectively. Let us say we have 1,000 detection results; then, we have 1,000 pairs of $(recall_i, precision_i)$ which are enough for plotting the curve.

We can compute the area under the precision-against-recall curve, which is AP, to represent the overall performance of a face detector. Under the single IoU threshold setting of 0.5 in WIDER Face evaluation, the top AP for the hard test subset of WIDER reached 0.924. In the WIDER Face Challenge 2019 which uses the same data as the WIDER Face dataset but evaluates face detectors in 10 IoU thresholds of 0.50:0.05:0.95, the top average AP reaches 0.5756.

4.5.3 Results on Accuracy

To understand the progress in recent years on face detection, the results of different datasets are collected from their official homepages. Because of space limitations, only the results from the two most popular datasets are listed. They are Fig. 4.8 for FDDB [21] and Fig. 4.9 for WIDER Face [62]. The FDDB results from 2004 to 2022 are listed. The current ROC curves are much better than those in the past. This means that the detection accuracy is much higher than in the past. The true positive rate is reaching 1.0. If you look into the samples in FDDB, you can find there are some tiny and blurred faces in the ground truth data. Sometimes, it is hard to decide whether they should be faces, even by humans. Therefore, we can say that the current detectors achieve perfect accuracy on FDDB, and almost all faces have been detected.

The WIDER face is newer, larger, and more challenging than FDDB. Most recent face detectors have been tested with it. From Fig. 4.9, it can be found that the accuracy is also very high even on the hard set. The improvement in mAP is not so obvious now. Similar to FDDB, the mAP is almost saturated.

We must note that the current benchmarks, regardless of FDDB, WIDER, or others, only evaluate the accuracy of detection and do not evaluate efficiency. If two detectors achieve similar mAP, but the computational cost of one is just half of another, surely we will think

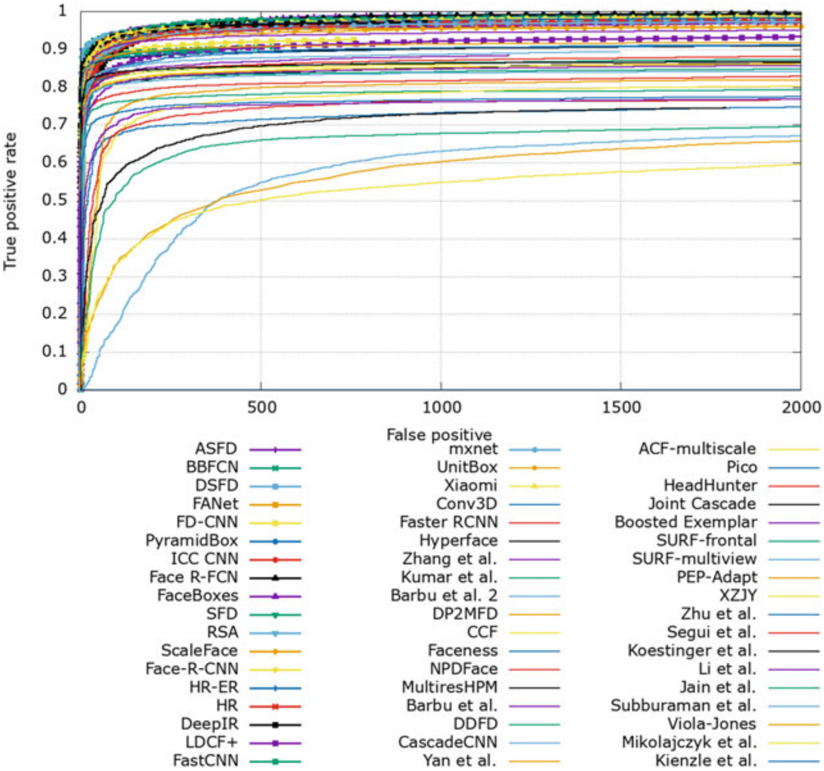
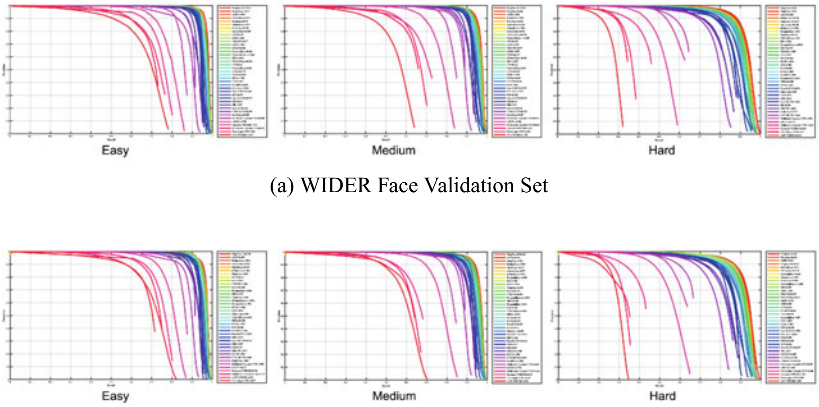


Fig. 4.8 The discrete ROC curves on FDDB for published methods from the result page of FDDB http://vis-www.cs.umass.edu/fddb/results.html in December 2022



(a) WIDER Face Validation Set

(b) WIDER Face Test Set

Fig. 4.9 The results on the validation set and the test sets of WIDER Face. The figures are from the WIDER face homepage http://shuoyang1213.me/WIDERFACE/ in December 2022

the detector with a half computational cost is better than another. Since the accuracy metric is almost saturated, it is time to include efficiency in the evaluation.

4.6 Evaluation of the Computational Cost

Deep learning techniques have brought momentous improvement to face detection and can detect faces more robustly in unconstrained environments. Most of the recent works train and test their models on WIDER Face [62]. As shown in Fig. 4.2, we can find a large AP leap from 2016 to 2017. However, the line has been flat since 2017. If we look deep into the official releasing code of recent works, it can be easily found that newer models tend to use larger scales and a wider range of scales as shown in Table 4.4. These test scales are usually not mentioned in the papers but can lead to a non-negligible great increase in computational cost just for slightly boosting the AP. We may ask a question: Is the AP improved by a better algorithm or the usage of a wider range of test scales?

To evaluate different FLOPs of different face detectors, we implement FLOPs calculator based on PyTorch, which accelerates the calculation of FLOPs by dismissing any calculation related to the value of tensors, while only computing the sizes of tensors and FLOPs. This calculator can also allow us to use the code of defining models from authors with minor changes, which reduces the statistics workload. We released our source code at https://github.com/fengyuentau/PyTorch-FLOPs. The details of the calculation of FLOPs can be found in [11].

Table 4.4 Test scales used by open-source one-stage face detectors [4, 20, 25, 31, 41, 51, 69]. Note that the double-check marks denote the image flipping vertically in addition to the image at the current scale. SSH shrinks and enlarges images to several preset fixed sizes. Since S^{3}FD, two adaptive test scales are used to save GPU memory, one of which is "S" for adaptive shrinking, the other of which is "E" for recursively adaptive enlarging. Scale "F" denotes enlarging the image to the preset largest size

Model	Publication	Test scales (ratio)											
		0.25	0.5	0.75	1	1.25	1.5	1.75	2.0	2.25	S	E	F
HR	CVPR'17	✓	✓		✓								
S^{3}FD	ICCV'17		✓		✓						✓	✓	
PyramidBox	ECCV'18	✓		✓	✓ ✓	✓	✓	✓			✓	✓	
SRN	AAAI'19		✓		✓ ✓		✓			✓			✓
DSFD	CVPR'19		✓		✓ ✓	✓		✓		✓	✓	✓	
CSP	CVPR'19	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓			
		Test scales (resize longer side)											
		100	300	500	600	700	800	900	1000	1100	1200	1400	1600
SSH	ICCV'17			✓			✓				✓		✓
SHF	WACV'20	✓	✓		✓				✓			✓	
RetinaFace	CVPR'20			✓			✓			✓		✓	✓

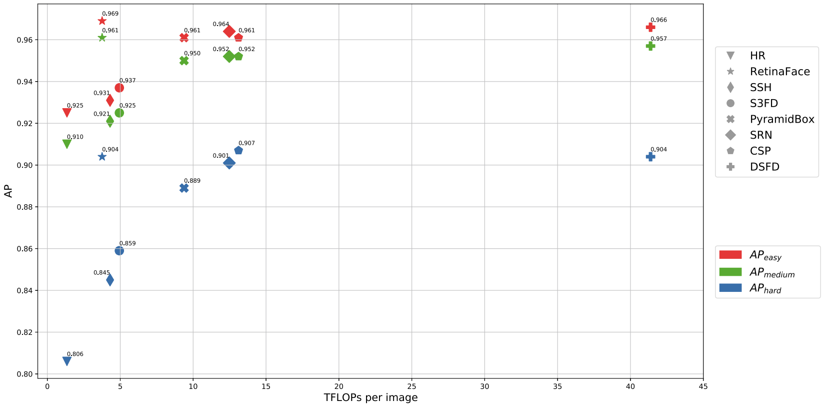


Fig. 4.10 The FLOPs versus multi-scale AP of WIDER Face validation set. Seven models from the WIDER Face result page are listed, which are HR [20], SSH [41], S^{3}FD [69], PyramidBox [51], SRN [4], DSFD [25], and CSP [31]. (*The TFLOPs for some speed-focusing face detectors are listed in Table* 4.9 *because the TFLOPs are in a much smaller scale and cannot fit in this figure.*)

4.6.1 FLOPs Versus AP in Multi-scale Test

The multi-scale test metric is to test a model with a set derived from an image at the original and different scales (with an aspect ratio fixed). The detection results of different scales are then merged and applied with the non-maximum suppression (NMS), so as to suppress the overlapped bounding boxes and reduce false positives. Based on the training data and scheme, a *comfort zone* of a model is determined, which is a range of scales of faces that can be detected. The multi-scale test metric can improve a model's AP by re-scaling out-of-zone faces back into the comfort zone. However, since we cannot determine which of the faces in the test set are out-of-zone, we have to apply re-scaling to every image in the set. It leads to a multiplied increase in FLOPs per image.

Figures 4.10 and 4.11 show the multi-scale test AP and FLOPs of different models on the validation and test sets of the WIDER Face dataset, respectively. We can find a clear trend in the two figures. The FLOPs are increasing and the AP is improving in the sequence of methods HR [20], SSH [41], S^{3}FD [69], PyramidBox [51], SRN [4], and CSP [31]. There are two methods that do not follow the trend. The first one is DSFD [25] which has more than three times of FLOPs than SRN and CSP, but the AP is similar to those of SRN and CSP. It means DSFD has unreasonably high computational costs. The second detector is RetinaFace [9] which gained the best AP but the computational cost is much lower than most other methods.

The two figures (Figs. 4.10 and 4.11) give us a clear view of different face detection models and can guide us to understand different models deeper.

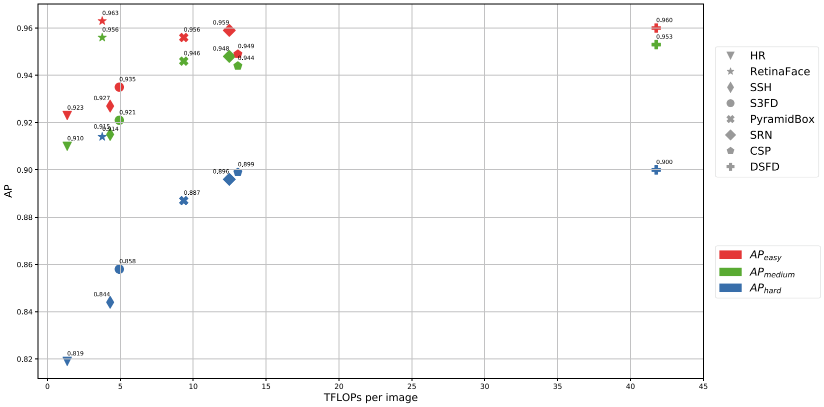


Fig. 4.11 The FLOPs versus multi-scale test AP of WIDER Face test set. Seven models from the WIDER Face result page are listed, which are HR [20], SSH [41], S^{3}FD [69], PyramidBox [51], SRN [4], DSFD [25], and CSP [31]

4.6.2 FLOPs Versus AP in Single-Scale Test

FLOPs can sharply increase in two ways: fundamentally increasing through introducing more complex modules to the network, and through multi-scale testing. As Table 4.4 shows, these models are all tested on various scales. However, why models are tested on these various scales is seldom discussed. How much contribution on AP can one scale bring? Are any scales not necessary?

Single-scale test on a single model. Table 4.5 shows the AP contribution of different scales. The easy subset in WIDER Face [62] contains a large margin of faces of regular size and some large faces, as a result of which shrinking images can help improve the AP. We can observe that AP_{hard} gains the most from scales 1, 1.25 and 1, 1.5 , but not for scale 1, 1.75. Together with FLOPs, we can also observe an increase to the peak at scale 1, 1.25 and then a sharp drop for larger scales. The reason is that a threshold for the largest size of images is set to avoid exceeding the GPU memory. This means that not all 1.75x resized images were sent to a detector in the experiments.

Table 4.6 shows how much the AP and FLOPs will decrease if a model is tested without a scale. As the missing scale becomes larger, the decrease of AP_{easy} decreases. However, this pattern does not apply to AP_{medium} and AP_{hard}. The reason is that the enlarged images will be skipped if their size goes beyond the preset limit, so as to avoid exceeding GPU memory. The larger the scale is, the fewer images will be re-scaled and tested. The drop of FLOPs greatly decreases on a scale of 1.75. This is because the PyramidBox pretrained model is mainly trained on scale 1.

The two Tables 4.5 and 4.6 imply that AP_{easy} is the most sensitive to scales 0.25, AP_{medium} is the most sensitive to scale 0.25 and 1, and AP_{hard} is the most sensitive to

Table 4.5 How different scales impact the AP of PyramidBox [51]. We use Scale = 1 as the baseline and then try adding different scales one by one to test how AP is impacted by different scales

Test scales						AP_{easy}	AP_{medium}	AP_{hard}	TFLOPs
0.25	0.75	1	1.25	1.5	1.75				
		✓				0.947	0.936	0.875	1.37
✓		✓				0.954(+0.007)	0.939(+0.003)	0.872(−0.003)	1.45(+0.008)
	✓	✓				0.952(+0.005)	0.940(+0.004)	0.874(−0.001)	2.14(+0.77)
		✓	✓			0.948(+0.001)	0.938(+0.002)	0.884(+0.009)	2.72(+1.35)
		✓		✓		0.947(+0.000)	0.937(+0.001)	0.881(+0.006)	2.46(+1.09)
		✓			✓	0.946(−0.001)	0.936(+0.000)	0.874(−0.001)	1.63(+0.26)

Table 4.6 How much AP and FLOPs will decrease if a scale is removed. The detector PyramidBox is employed

Test scales						AP_{easy}	AP_{medium}	AP_{hard}	TFLOPs
0.25	0.75	1	1.25	1.5	1.75				
✓	✓	✓	✓	✓	✓	0.957	0.945	0.886	4.94
	✓	✓	✓	✓	✓	0.949(−0.008)	0.940(−0.005)	0.884(−0.002)	4.85(−0.009)
✓		✓	✓	✓	✓	0.954(−0.003)	0.942(−0.003)	0.885(−0.001)	4.16(−0.780)
✓	✓		✓	✓	✓	0.955(−0.002)	0.940(−0.005)	0.850(−0.013)	3.58(−1.360)
✓	✓	✓		✓	✓	0.957(+0.000)	0.944(−0.001)	0.880(−0.006)	3.58(−1.360)
✓	✓	✓	✓		✓	0.958(+0.001)	0.945(+0.000)	0.884(−0.002)	3.84(−1.100)
✓	✓	✓	✓	✓		0.957(+0.000)	0.945(+0.000)	0.886(+0.000)	4.67(−0.270)

scale 1. Note that this is highly related to the training scale. If the model is trained differently, the conclusion may change accordingly.

Single-scale test on multiple models.

Table 4.7 shows the AP and FLOPs of different models on scale 1. The large overall leap is brought by PyramidBox [51], which mainly introduces the FPN [27] module to fuse features from two adjacent scales and the context enhancing module from SSH [41]. The computational cost of PyramidBox is 2X compared with SSH but less than 1/2 of DSFD. However, the APs achieved by PyramidBox and DSFD are comparable.

If some benchmarks can evaluate FLOPs or some other similar efficiency measurements, different face detectors can compare more fairly. It will also promote face detection research to a better stage.

Table 4.7 AP and FLOPs of different models on scale 1

Model	AP_{easy}	AP_{medium}	AP_{hard}	TFLOPs
RetinaFace	0.952	0.942	0.776	0.198
S3FD	0.924	0.906	0.816	0.571
CSP	0.948	0.942	0.774	0.571
SSH	0.925	0.909	0.731	0.587
PyramidBox	0.947	0.936	0.875	1.387
DSFD	0.949	0.936	0.845	1.532

Table 4.8 The results of some state-of-the-art open-source detectors tested with a 720P image containing several faces at scale = 1.0 only. We average the FLOPs (AVG TFLOPs) and latency (AVG latency) by running the test for each model 100 times. Note that "Post-Proc" denotes post-processing stages, such as decoding from anchors, NMS, and so on. For this stage, we adopt the original processing code of each model

Model	AVG TFLOPs	AVG latency (ms)		
		Forward (GPU)	Forward (CPU)	Post-Proc
RetinaFace	0.201	131.60	809.24	8.74 (GPU)
CSP	0.579	154.55	1955.20	27.74 (CPU)
SRN	1.138	204.77	2933.16	8.71 (GPU)
DSFD	1.559	219.63	3671.46	76.32 (CPU)

4.6.3 FLOPs Versus Latency

To compare the two measurements, we convert existing models to the Open Neural Network Exchange (ONNX) format and run them using ONNX Runtime[15] for fair comparisons. Due to the different model formats, we only managed to convert RetinaFace [9], SRN [4], DSFD [25], and CSP [31] to ONNX format. The results are listed in Table 4.8. Those models are evaluated using an NVIDIA QUADRO RTX 6000 with CUDA 10.2, and an INTEL Xeon Gold 6132 CPU @ 2.60 GHz. The GPU contains 4,609 CUDA parallel-processing cores and 24GB of memory.

We can observe that both FLOPs and forward latency increase from RetinaFace [9] to DSFD [25]. Note that although the average FLOPs of RetinaFace are just one-fifth of SRN's, the forward latency of RetinaFace is almost half of SRN's. It implies that FLOPs are not linearly correlated to latency due to the differences in implementation, hardware settings,

[15] https://github.com/microsoft/onnxruntime.

memory efficiency, and others. The reason why the post-processing latency of DSFD and CSP sharply increases is that they do not use GPU-accelerated NMS as others do.

4.7 Speed-Focusing Face Detectors

For the face detectors introduced in the previous sections, the main target is to reach a better AP. Their computational costs are heavy and normally in the magnitude of TFLOPs. It is unrealistic to deploy those heavy models to a face-related system. There are some other open-source face detectors whose target is to make face detection run in real time for practical applications. Their computational costs are in the magnitude of those of GFLOPs or 10 GFLOPs. Their computational cost is much lighter than the heavy ones. Here, we categorize them as speed-focusing face detectors. We collect some popular face detectors and evaluate them in terms of network architectures, AP, FLOPs, and efficiency. They are FaceBoxes [68], YuNet [19], LFFD [16], and ULFG [29].

FaceBoxes [68] is one of the first one-stage deep learning-based models to achieve real-time face detection. FaceBoxes rapidly downsamples feature maps to a stride 32 with two convolution layers with large kernels. Inception blocks [50] are introduced to enhanced feature maps at a stride of 32. Following the multi-scale mechanism from SSD [30], FaceBoes detects on layers `inception3`, `conv3_2`, and `conv4_2` for faces at different scales, resulting in an AP of 0.960 on FDDB [21] and 20 FPS on an INTEL E5-2660v3 CPU at 2.60 GHz.

YuNet [19] adopts a light MobileNet [18] as the backbone. Compared to FaceBoxes, YuNet has more convolution layers on each stride to have fine-grained features and detects on the extra layer of stride 16, which improves the recall of small faces. The evaluation results of the model on the WIDER Face [62] validation set are 0.892 (Easy), 0.883 (Medium), and 0.811 (Hard). The main and well-known repository, libfacedetection [46], takes YuNet as the detection model and offers pure C++ implementation without dependence on DL frameworks, resulting from 156.47 FPS for 640 × 480 images to 3198.63 FPS for 128 × 96 images on an Intel i7-7820X CPU @ 3.60GHz according to the information at https://github.com/ShiqiYu/libfacedetection in December 2022.

LFFD [16] introduces residual blocks for feature extraction, and proposes receptive fields as the natural anchors. The faster version LFFD-v2 managed to achieve 0.875 (Easy), 0.863 (Medium), and 0.754 (Hard) on the WIDER Face validation set while running at 472 FPS using CUDA 10.0 and an NVIDIA RTX 2080Ti GPU.

ULFG [29] adds even more convolution layers on each stride, taking the advantage of depth-wise convolution, which is friendly to edge devices in terms of FLOPs and forward latency. As reported, the slim version of ULFG has an AP of 0.770 (Easy), 0.671 (Medium), and 0.395 (Hard) on the WIDER Face validation set, and can run at 105 FPS with an input resolution of 320 × 240 on an ARM A72 at 1.5 GHz.

Table 4.9 Some popular speed-focusing open-source face detectors. Note that "AVG GFLOPs" are computed on WIDER Face validation set in a single-scale test where only scale = 1.0. The latencies are measured on a CPU

Model	#CONV layers	#Params ($\times 10^6$)	AVG GFLOPs	WIDER face val set			Latency (ms)	
				AP_{easy}	AP_{medium}	AP_{hard}	Forward	Post-Proc
FaceBoxes [68]	33	1.013	1.541	0.845	0.777	0.404	16.52	7.16
ULFG-slim-320 [29]	42	0.390	2.000	0.652	0.646	0.520	19.03	2.37
ULFG-slim-640 [29]				0.810	0.794	0.630		
ULFG-RFB-320 [29]	52	0.401	2.426	0.683	0.678	0.571	21.27	1.90
ULFG-RFB-640 [29]				0.816	0.802	0.663		
YuNet [19]	41	0.055	2.790	0.892	0.883	0.811	11.7	4.6
LFFD-v2 [16]	45	1.520	37.805	0.875	0.863	0.752	178.47	6.70
LFFD-v1 [16]	65	2.282	55.555	0.910	0.880	0.778	229.35	10.08

These lightweight models are developed using various frameworks and tested on different hardware. For a fair comparison, we export these models from their original frameworks to ONNX and test using ONNX Runtime on an INTEL i7-5930K CPU at 3.50 GHz. Results are shown in Table 4.9. We can observe that more CONV layers do not lead to more parameters (FacesBoxes and ULFG series) and more FLOPs (YuNet and ULFG series). This is mainly because of the extensive usage of depth-wise convolution in ULFG. Additionally, note that more FLOPs do not lead to more forward latency due to depth-wise convolution. The post-processing latency across different face detectors seems inconsistent with the forward latency, and we verified that this is caused by different numbers of bounding boxes sent to NMS and the different implementations of NMS (Python-based or Cython-based).

4.8 Conclusions and Discussions

Face detection is one of the most important and popular topics yet still challenging in computer vision. Deep learning has brought remarkable breakthroughs for face detectors. Face detection is more robust and accurate even in unconstrained real-world environments. In this chapter, recent deep learning-based face detectors and benchmarks are introduced. From the evaluations of accuracy and efficiency on different deep face detectors, we can reach a very high accuracy if we do not consider the computational cost. However, there should be a simple and beautiful solution for face detection since it is simpler than generic object detection. The research on face detection can focus on the topics introduced in the following topics in the future.

Superfast Face Detection. There is no definition for superfast face detection. Ideally, a superfast face detector should be able to run in real time on low-cost edge devices even when the input image is 1080P. Empirically speaking, we would like to expect it to be less than 100M FLOPs with a 1080P image as input. For real-world applications, efficiency is one of the key issues. Efficient face detectors can help to save both energy and the cost of hardware. They can also improve the responsiveness of edge devices, such as CCTV cameras and mobile phones.

Detecting Faces in the Long-tailed Distribution. Face samples can be regarded as a long-tailed distribution. Most face detectors are trained for the dominant part of the distribution. We have already had enough samples for faces with variances in illumination, pose, scale, occlusion, blur, and distortion in the WIDER Face dataset. But what about other faces like the old and damaged ones? As people get old, there are many wrinkles on their faces. People who suffer from illnesses or accidents may have damaged faces, such as burn scars on the faces. Face detection is not only a technical problem but also a humanitarian problem. This technology should serve all the people, not only the dominant part of the population. Ideally, face detectors should be able to detect all kinds of faces. However, in most face datasets and benchmarks, most faces are from young people and do not cover all people.

The final goal of face detection is to detect faces with very high accuracy and high efficiency. Therefore, the algorithms can be deployed to many kinds of edge devices and centralized servers to improve the perception capability of computers. There still is a considerable gap to that goal. Face detectors can achieve good accuracy but still require considerable computation. Improving efficiency should be the next step.

References

1. Boult, E.T., Gunther, M., Dhamija, R.A.: 2nd unconstrained face detection and open set recognition challenge. https://vast.uccs.edu/Opensetface/. Access at 2019-10-31
2. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
3. Wei, C., Wang, W., Yang, W., Liu, J.: Deep retinex decomposition for low-light enhancement. In: British Machine Vision Conference (BMVC) (2018)
4. Chi, C., Zhang, S., Xing, J., Lei, Z., Li, S.Z., Zou, X.: Selective refinement network for high performance face detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2019)
5. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems (NIPS) (2016)
6. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2005)

8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
9. Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: RetinaFace: single-shot multi-level face localisation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
10. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
11. Feng, Y., Yu, S., Peng, H., Li, Y.R., Zhang, J.: Detect faces efficiently: a survey and evaluations. IEEE Trans. Biometrics Behav. Identity Sci. **4**(1), 1–18 (2022). https://doi.org/10.1109/TBIOM.2021.3120412
12. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD: deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
13. Ge, S., Li, J., Ye, Q., Luo, Z.: Detecting masked faces in the wild with lle-cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
14. Hao, Z., Liu, Y., Qin, H., Yan, J., Li, X., Hu, X.: Scale-aware face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
16. He, Y., Xu, D., Wu, L., Jian, M., Xiang, S., Pan, C.: Lffd: a light and fast face detector for edge devices. arXiv preprint arXiv:1904.10633 (2019)
17. Jin, H., Liu, Q., Lu, H., Tong, X.: Face detection using improved lbp under bayesian framework. In: International Conference on Image and Graphics (ICIG) (2004)
18. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
19. Wu, W., Peng, H., Yu, S.: YuNet: a tiny millisecond-level face detector. Mach. Intell. Res. (2023)
20. Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
21. Jain, V., Learned-Miller, E.: FDDB: a benchmark for face detection in unconstrained settings. Tech. rep., Technical Report UM-CS-2010-009, University of Massachusetts, Amherst (2010)
22. Klare, B.F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Burge, M., Jain, A.K.: Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
23. Law, H., Deng, J.: CornerNet: detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
24. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
25. Li, J., Wang, Y., Wang, C., Tai, Y., Qian, J., Yang, J., Wang, C., Li, J., Huang, F.: DSFD: dual shot face detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
26. Li, J., Zhang, Y.: Learning surf cascade for fast and accurate object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
27. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

28. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
29. Linzaer: Ultra-light-fast-generic-face-detector-1mb. https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB (2020)
30. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
31. Liu, W., Liao, S., Ren, W., Hu, W., Yu, Y.: High-level semantic feature detection: a new perspective for pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
32. Liu, Y., Li, H., Yan, J., Wei, F., Wang, X., Tang, X.: Recurrent scale approximation for object detection in CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
33. Liu, Y., Tang, X.: BFBox: Searching face-appropriate backbone and feature pyramid network for face detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
34. Liu, Y., Tang, X., Han, J., Liu, J., Rui, D., Wu, X.: HAMBox: delving into mining high-quality anchors on face detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
35. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (1999)
36. Luo, S., Li, X., Zhu, R., Zhang, X.: Sfa: small faces attention face detector. IEEE Access **7**, 171609–171620 (2019)
37. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS) (2016)
38. Maze, B., Adams, J., Duncan, J.A., Kalka, N., Miller, T., Otto, C., Jain, A.K., Niggel, W.T., Anderson, J., Cheney, J., Grother, P.: Iarpa janus benchmark - c: face dataset and protocol. In: International Conference on Biometrics (ICB) (2018)
39. Ming, X., Wei, F., Zhang, T., Chen, D., Wen, F.: Group sampling for scale invariant face detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
40. Nada, H., Sindagi, V.A., Zhang, H., Patel, V.M.: Pushing the limits of unconstrained face detection: a challenge dataset and baseline results. In: IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS) (2018)
41. Najibi, M., Samangouei, P., Chellappa, R., Davis, L.S.: SSH: single stage headless face detector. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
42. Najibi, M., Singh, B., Davis, L.S.: FA-RPN: floating region proposals for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
43. Naphade, M., Smith, J.R., Tesic, J., Chang, S.F., Hsu, W., Kennedy, L., Hauptmann, A., Curtis, J.: Large-scale concept ontology for multimedia. IEEE Multimedia **13**(3), 86–91 (2006)
44. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
45. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS) (2015)
46. S. Yu, et al.: libfacedetection. https://github.com/ShiqiYu/libfacedetection (2021)

47. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
48. Song, G., Liu, Y., Jiang, M., Wang, Y., Yan, J., Leng, B.: Beyond trade-off: accelerate FCN-based face detector with higher accuracy. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
49. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2017)
50. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
51. Tang, X., Du, D.K., He, Z., Liu, J.: Pyramidbox: a context-assisted single shot face detector. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
52. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully convolutional one-stage object detection. In: Proceedings of the IEEE International Conference on Computer Vision (CVPR) (2019)
53. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2001)
54. Wan, S., Chen, Z., Zhang, T., Zhang, B., Wong, K.K.: Bootstrapping face detection with hard negative examples. arXiv preprint arXiv:1608.02236 (2016)
55. Wang, H., Li, Z., Ji, X., Wang, Y.: Face R-CNN. arXiv preprint arXiv:1706.01061 (2017)
56. Wang, J., Yuan, Y., Li, B., Yu, G., Jian, S.: Sface: an efficient network for face detection in large scale variations. arXiv preprint arXiv:1804.06559 (2018)
57. Wang, Y., Ji, X., Zhou, Z., Wang, H., Li, Z.: Detecting faces using region-based fully convolutional networks. arXiv preprint arXiv:1709.05256 (2017)
58. Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A.K., Duncan, J.A., Allen, K., Cheney, J., Grother, P.: Iarpa janus benchmark-b face dataset. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017)
59. Yan, J., Zhang, X., Lei, Z., Li, S.Z.: Face detection by structural models. Image Vis. Comput. **32**(10), 790–799 (2014)
60. Yang, B., Yan, J., Lei, Z., Li, S.Z.: Aggregate channel features for multi-view face detection. In: IEEE International Joint Conference on Biometrics (IJCB) (2014)
61. Yang, B., Yan, J., Lei, Z., Li, S.Z.: Fine-grained evaluation on face detection in the wild. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (FG) (2015)
62. Yang, S., Luo, P., Loy, C.C., Tang, X.: WIDER face: a face detection benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
63. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: point set representation for object detection. In: Proceedings of the IEEE International Conference on Computer Vision (CVPR) (2019)
64. Yucel, M.K., Bilge, Y.C., Oguz, O., Ikizler-Cinbis, N., Duygulu, P., Cinbis, R.G.: Wildest faces: face detection and recognition in violent settings. arXiv preprint arXiv:1805.07566 (2018)
65. Zhang, C., Xu, X., Tu, D.: Face detection using improved faster rcnn. arXiv preprint arXiv:1802.02142 (2018)
66. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. **23**(10), 1499–1503 (2016)
67. Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z.: Face detection based on multi-block lbp representation. In: International Conference on Biometrics (2007)

68. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z.: FaceBoxes: a cpu real-time face detector with high accuracy. In: Proceedings of IEEE International Joint Conference on Biometrics (IJCB) (2017)
69. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z.: S3FD: single shot scale-invariant face detector. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
70. Zhang, Z., Shen, W., Qiao, S., Wang, Y., Wang, B., Yuille, A.: Robust face detection via learning small faces on hard images. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV) (2020)
71. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. In: arXiv preprint arXiv:1904.07850 (2019)
72. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
73. Zhu, C., Tao, R., Luu, K., Savvides, M.: Seeing small faces from robust anchor's perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
74. Zhu, C., Zheng, Y., Luu, K., Savvides, M.: Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. Deep Learning for Biometrics, pp. 57–79 (2017)
75. Zhu, J., Li, D., Han, T., Tian, L., Shan, Y.: ProgressFace: scale-aware progressive learning for face detection. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
76. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: more deformable, better results. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
77. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
78. Zhu, Y., Cai, H., Zhang, S., Wang, C., Xiong, Y.: Tinaface: strong but simple baseline for face detection. arXiv preprint arXiv:2011.13183 (2020)
79. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)

Facial Landmark Localization

5

Xiangyu Zhu, Zhenhua Feng, and Hailin Shi

5.1 Introduction

Facial landmark localization aims to detect a sparse set of facial fiducial points on a human face, some of which include “eye corner”, “nose tip”, and “chin center”. In the pipeline of face analysis, landmark detectors take the input of a face image and the bounding box provided by face detection, and output a set of coordinates of the predefined landmarks, which is illustrated in Fig. 5.1. It provides a fine-grained description of the face topology, such as facial features locations and face region contours, which is essential for many face analysis tasks, e.g., recognition [32], animation [33], attributes classification [34], and face editing [35]. These applications usually run on lightweight devices in uncontrolled environments, requiring landmark detectors to be accurate, robust, and computationally efficient, all at the same time.

Over the past few decades, there have been significant developments in facial landmark detection. The early works consider landmark localization as the process of moving and deforming a face model to an image, and they construct a statistical facial model to model the shape and albedo variations of human faces. The most prominent algorithms include Active Shape Model (ASM) [42], Active Appearance Model (AAM) [43], and Constrained

X. Zhu (✉)
Institute of Automation, Chinese Academy of Sciences, Beijing, China
e-mail: xiangyu.zhu@nlpr.ia.ac.cn

Z. Feng
School of Computer Science and Electronic Engineering, University of Surrey, Guildford, UK
e-mail: z.feng@surrey.ac.uk

H. Shi
Nio Inc., Beijing, China
e-mail: hailinshi.work@outlook.com

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_5

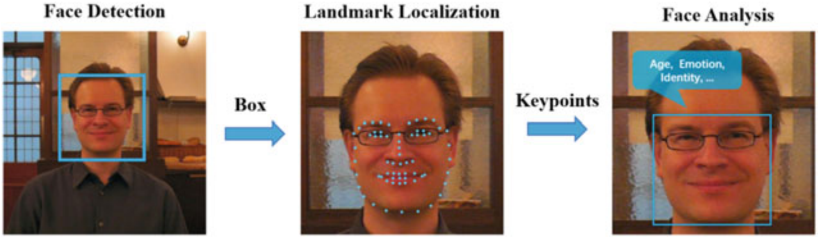


Fig. 5.1 Facial landmark localization

Local Model (CLM) [44], by which the faces in controlled environments (normal lighting and frontal poses) can be well handled. However, these methods deteriorate greatly when facing enormous challenges in the wild, such as large poses, extreme illuminations, low resolution, and partial occlusions. The next wave of methods is based on cascaded regression [45, 88, 89], which cascades a list of weak regressors to reduce the alignment error progressively. For example, the Supervised Descent Method (SDM) [88] updates the landmark locations by several iterations of regressions. In each iteration, a regressor takes the input of the appearance features (e.g., SIFT) around landmarks, and estimates a landmark update to approach the ground-truth locations. The Ensemble of Regression Trees (ERT) [45] learns an ensemble of regression trees to regress the landmarks from a sparse subset of intensity values, so as to handle partial or uncertain labels. One of the most popular landmark detectors Dlib [46] implements ERT as its landmark detector due to its high speed of 1 millisecond per face.

Following the great success of deep learning in computer vision [47], researchers started to predict facial landmarks by deep convolutional neural networks. In general, deep learning-based landmark detectors can be divided into coordinate-based and heatmap-based, illustrated in Fig. 5.2, depending on the detection head of network architecture. Coordinate-based methods output a vector consisting of 2D coordinates of landmarks. On the contrary, heatmap-based methods output one heatmap for each landmark, where the intensity value of the heatmap indicates the probability that this landmark locates in this position. It is commonly agreed [38, 39] that heatmap-based methods detect more accurate landmarks, but are computationally expensive and sensitive to outliers. In contrast, coordinate-based methods are fast and robust, but have sub-optimal accuracy.

In recent years, 3D landmark localization has attracted increasing attention due to its additional geometry information and superiority in handling large poses [40]. However, localizing 3D landmarks is more challenging than 2D landmarks because recovering depth from a monocular image is an ill-posed problem. This requires the model to build a strong 3D face prior from large-scale 3D data in order to accurately detect and locate the facial landmarks in 3D space. Unfortunately, acquiring 3D faces is expensive, and labeling 3D landmarks is also tedious. A feasible solution is to fit a 3D Morphable Model (3DMM) [41] by a neural network [40] and sample the 3D landmarks from the fitted 3D model. Another

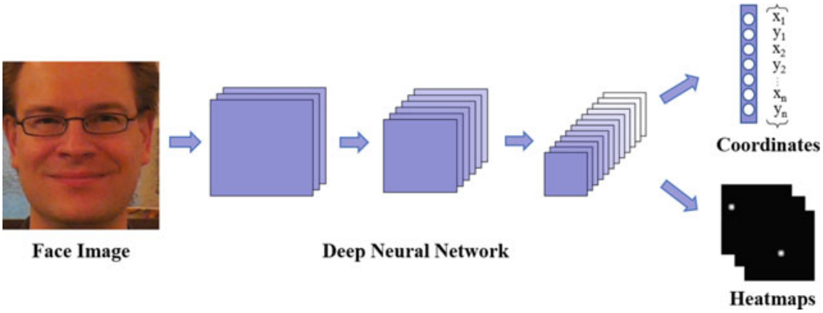


Fig. 5.2 Coordinate-based methods and heatmap-based methods

one is utilizing a fully convolutional network to regress the 3D heatmaps, on which the coordinates of the largest probabilities are sampled as 3D landmarks [51, 52].

5.2 Coordinate Regression

As deep learning has become the mainstream method for facial landmark localization, this section focuses on recent advances in deep learning-based coordinate regression approaches. Given an input face image, coordinate regression-based methods predict the 2D coordinates of a set of predefined facial landmarks directly from the deep features extracted by a backbone network, as shown in Fig. 5.3.

5.2.1 Coordinate Regression Framework

The task of coordinate regression-based facial landmark localization is to find a nonlinear mapping function (usually a deep CNN model):

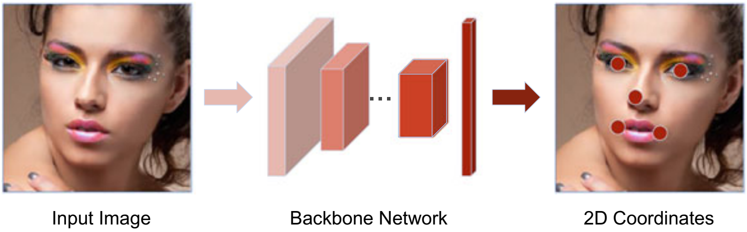


Fig. 5.3 Coordinate regression-based facial landmark localization. The input is an RGB face image, and the output is a vector consisting of the 2D coordinates of all the facial landmarks

$$\Phi : \mathcal{I} \rightarrow \mathbf{s}, \tag{5.1}$$

that outputs the 2D coordinates vector $\mathbf{s} \in \mathbb{R}^{2L}$ of L landmarks for a given facial image $\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$. In general, the input image is cropped by using a bounding box obtained by a face detector in a full-stack facial image/video analysis pipeline. The 2D coordinate vector $\mathbf{s} = [x_1, ..., x_L, y_1, ..., y_L]^T$ consists of the coordinates of L predefined landmarks, where (x_l, y_l) are the X- and Y-coordinates of the lth landmark.

To obtain the above mapping function, a deep neural network can be used, which is formulated as a compositional function:

$$\Phi = (\phi_1 \circ ... \circ \phi_M)(\mathcal{I}), \tag{5.2}$$

with M sub-functions, and each sub-function (ϕ) represents a specific network layer, e.g., convolutional layer and nonlinear activation layer. Most existing deep learning-based facial landmark localization approaches use CNN as the backbone with a regression output layer [24–26].

Given a set of labeled training samples $\Omega = \{\mathcal{I}_i, \mathbf{s}_i\}_{i=1}^{N}$, the network training aims to find the best set of the parameters Φ so that to minimize:

$$\sum_{i=1}^{N} loss(\Phi(\mathcal{I}_i), \mathbf{s}_i), \tag{5.3}$$

where $loss()$ is a predefined loss function that measures the difference between the predicted and ground-truth coordinates over all the training samples. To optimize the above objective function, a variety of optimization methods, such as Stochastic Gradient Descent (SGD) and AdamW, can be used for network training.

5.2.2 Network Architectures

As shown in Fig. 5.3, the input for a coordinate regression-based facial landmark localization model is usually an image enclosing the whole face region. Then a backbone CNN network can be used for feature extraction and fully connected layers are used for regressing the landmark coordinates. With the development of deep learning, different backbone networks have been explored and evaluated for accurate and robust landmark localization. For example, Feng et al. [38] evaluated different backbone networks, including VGG, ResNet, MobileNet, etc., for efficient and high-performance facial landmark localization. As face landmarking is a key element in a full-stack facial image/video analysis system, the design of a lightweight network is crucial for real-time applications. For instance, Guo et al. [18] developed a light framework that is only 2.1 MB and runs at 140 fps on a mobile device. Gao et al. [19] proposed EfficientFAN that applies deep knowledge transfer via a teacher-student network for efficient and effective network training. Feng et al. [38] compared different designs of network architectures and evaluated their inference speed on different devices, including GPU, CPU, and portable devices.

Instead of the whole face image, shape- or landmark-related local patches have also been widely used as the input of neural networks [24, 83]. To use local patches, one can apply CNN-based feature extraction to the local patches centered at each landmark and for fine-grained landmark prediction or update [83]. The advantage of using the whole face region, in which the only input of the network is a cropped face image, is that it does not require the initialization of facial landmarks. In contrast, to use local patches, a system usually requires initial estimates of facial landmarks for a given image. This can be achieved by either using the mean-shape landmarks [83] or the output of another network that predicts coarse landmarks [24, 27, 61].

The accuracy of landmark localization can be degraded by in-plane face rotations and inaccurate bounding boxes output by a face detector. To address these issues, a widely used strategy is to cascade multiple networks to form a coarse-to-fine structure. For example, Huang et al. [28] proposed to use a global network to obtain coarse facial landmarks for transforming a face to the canonical view and then applied multiple networks trained on different facial parts for landmark refinement. Similarly, both Yang et al. [29] and Deng et al. [30] proposed to train a network that predicts a small number of facial landmarks (5 or 19) to transform the face to a canonical view. It should be noted that the first network can be trained on a large-scale dataset so it performs well for unconstrained faces with in-plane head rotation, scale, and translation. With the first stage, the subsequent networks that predict all the landmarks can be trained with the input of normalized faces.

Feng et al. [38] also proposed a two-stage network for facial landmark localization, as shown in Fig. 5.4. The coarse network is trained on a dataset with very heavy data augmentation by randomly rotating an original training image between [−180°, 180°] and perturbing the bounding box with 20% of the original bounding box size. Such a trained network is able to perform well for faces with large in-plane head rotations and low-quality

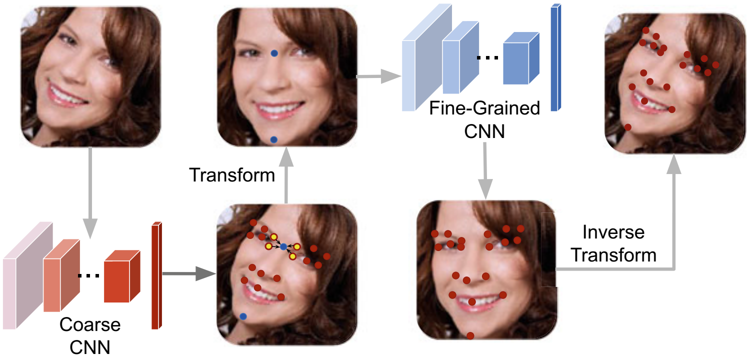


Fig. 5.4 A two-stage coarse-to-fine facial landmark localization framework

bounding boxes. For training the second network, each training sample is fed to the first network to obtain its coarse facial landmarks for geometric normalization. To be specific, two anchor points (blue points in Fig. 5.4) are computed to perform the rigid transformation, where one anchor is the mean of the four inner eye and eyebrow corners and the other one is the chin landmark. Afterward, the normalized training data is lightly augmented by randomly rotating the image between [−10°, 10°] and perturbing the bounding box with 10% of the bounding box size. The aim is to address the issues caused by inaccurate landmark localization of the first network. Finally, a second network is trained on the normalized-and-lightly-augmented dataset for further performance boosting in localization accuracy. The joint use of these two networks in a coarse-to-fine fashion is instrumental in enhancing the generalization capacity and accuracy.

5.2.3 Loss Functions

Another important element for high-performance coordinates regression is the design of a proper loss function. Most existing regression-based facial landmark localization approaches with deep neural networks are based on the L2 loss function. Given a training image $\mathcal{I}$ and a network Φ, we can predict the facial landmarks as a vector $\mathbf{s}' = \Phi(\mathcal{I})$. The loss function is defined as:

$$loss(\mathbf{s}, \mathbf{s}') = \frac{1}{2L} \sum_{i=1}^{2L} f(s_i - s_i'), \tag{5.4}$$

where $\mathbf{s}$ is the ground-truth facial landmark coordinates and s_i is its ith element. For $f(x)$ in the above equation, the L2 loss is defined as:

$$f_{L2}(x) = \frac{1}{2}x^2. \tag{5.5}$$

However, it is well known that the L2 loss function is sensitive to outliers, which has been noted in connection with many existing studies, such as the bounding box regression problem in face detection [31]. To address this issue, L1 and smooth L1 loss functions are widely used for robust regression. The L1 loss is defined as:

$$f_{L1}(x) = |x|. \tag{5.6}$$

The smooth L1 loss is defined piecewise as:

$$f_{smL1}(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < 1 \\ |x| - \frac{1}{2} & \text{otherwise} \end{cases}, \tag{5.7}$$

which is quadratic for small values and linear for large values [31]. More specifically, smooth L1 uses $f_{L2}(x)$ for $x \in (-1, 1)$ and shifts to $f_{L1}(x)$ elsewhere. Figure 5.5 depicts the plots of these three loss functions.

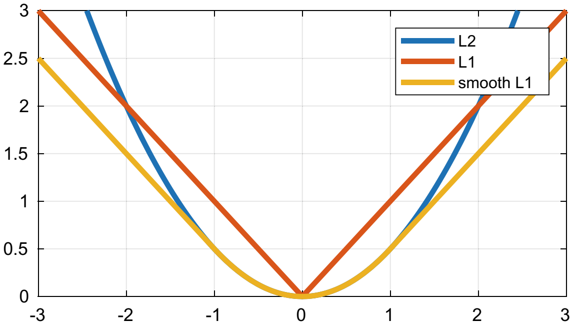


Fig. 5.5 Plots of the L2, L1 and smooth L1 loss functions

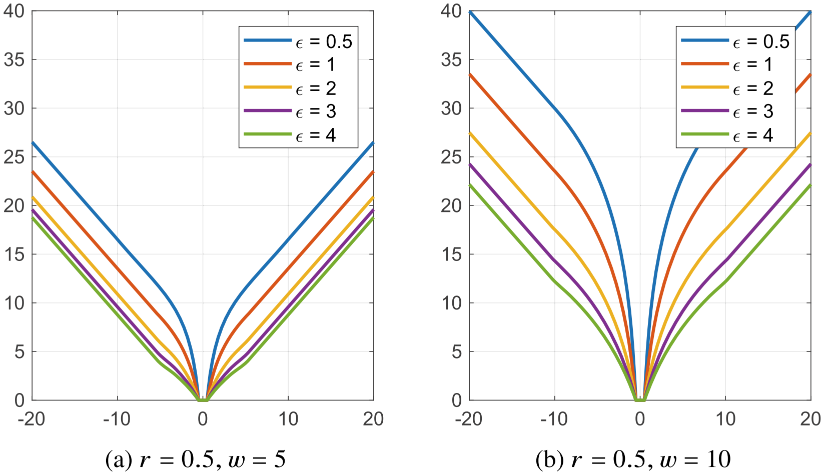
However, outliers are not the only subset of points which deserve special consideration. Feng et al. [38] argued that the behavior of the loss function at points exhibiting small-medium errors is just as crucial to finding a good solution to the landmark localization task. Based on a more detailed analysis, they proposed a new loss function, namely Rectified Wing (RWing) loss, for coordinate regression-based landmark localization. Similar to the original Wing loss function, RWing is also defined piecewise:

$$RWing(x) = \begin{cases} 0 & \text{if } |x| < r \\ w \ln(1 + (|x| - r)/\epsilon) & \text{if } r \leq |x| < w \\ |x| - C & \text{otherwise} \end{cases}, \tag{5.8}$$

where the non-negative parameter r sets the range of rectified region to $(-r, r)$ for very small values. The aim is to remove the impact of noise labels on network convergence. For a training sample with small-medium range errors in $[r, w)$, RWing uses a modified logarithm function, where ϵ limits the curvature of the nonlinear region and $C = w - w \ln(1 + (w - r)/\epsilon)$ is a constant that smoothly links the linear and nonlinear parts. Note that one should not set ϵ to a very small value because this would make the training of a network very unstable and cause the exploding gradient problem for small errors. In fact, the nonlinear part of the RWing loss function just simply takes a part of the curve of $\ln(x)$ and scales it along both the X-axis and Y-axis. Also, RWing applies translation along the Y-axis to allow $RWing(\pm r) = 0$ and to impose continuity on the loss function at $\pm w$. In Fig. 5.6, some examples of the RWing loss with different hyper parameters are demonstrated.

5.3 Heatmap Regression

Another main category of the state-of-the-art facial landmark localization methods is heatmap regression. Different from coordinate regression, heatmap regression outputs a heatmap for each facial landmark. In the heatmap, the intensity value of a pixel in a heatmap indicates the probability that its location is the predicted position of the corresponding



(a) $r = 0.5, w = 5$ (b) $r = 0.5, w = 10$

Fig. 5.6 The Rectified Wing loss function plotted with different hyper parameters, where r and w limit the range of the nonlinear part and ϵ controls the curvature. By design, the impact of the samples with small- and medium-range errors is amplified, and the impact of the samples with very small errors is ignored

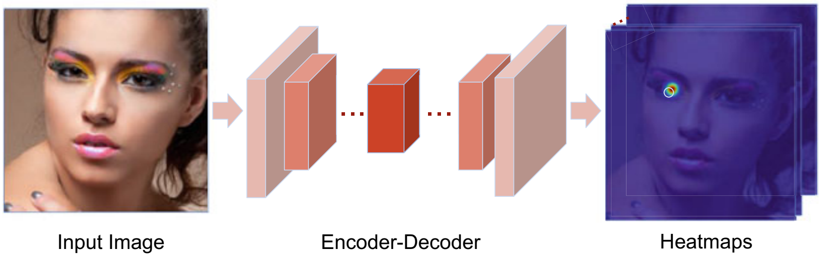


Fig. 5.7 Heatmap regression-based facial landmark localization. The input is a face image and the output are L 2D heatmaps, each for one predefined facial landmark. The backbone network usually has an encoder-decoder architecture

landmark. The task of heatmap regression-based facial landmark localization is to find a nonlinear mapping function:

$$\Phi : \mathcal{I} \to \mathcal{H}, \tag{5.9}$$

that outputs L 2D heatmaps $\mathcal{H} \in \mathbb{R}^{H \times W \times L}$ for a given image $\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$. As shown in Fig. 5.7, heatmap regression usually uses an encoder-decoder architecture for heatmap generation. For network training, typical loss functions used for heatmap generation include MSE and L1.

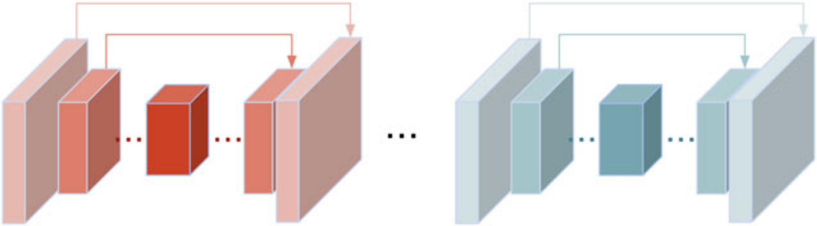


Fig. 5.8 A typical architecture of a stacked hourglass network

5.3.1 Network Architectures

As aforementioned, heatmap regression usually applies an encoder-decoder architecture for high-performance facial landmark localization. The most popular backbone network used for heatmap regression might be the stacked hourglass network [29, 30, 55, 68]. The key to the success of a stacked hourglass network is the use of multiple hourglass networks with residual connections, as shown in Fig. 5.8. On the one hand, the use of residual connections in each hourglass network maintains multi-scale facial features for fine-grained heatmap generation. On the other hand, stacking multiple hourglass networks improves the overall network capacity, so as to improve the quality of a generated heatmap. Besides the stacked hourglass network, another two popular network architectures used for heatmap regression are HRNet [75] and U-Net [77]. Similar to hourglass, both HRNet and U-Net try to find an effective way of using multi-scale features rather than the single use of a deep high-level semantic feature map for heatmap generation.

To reduce false alarms of a generated 2D heatmap, Wu et al. [22] proposed a distance-aware softmax function that facilitates the training of a dual-path network. Lan et al. [79] further investigated the issue of quantization error in heatmap regression, and proposed a heatmap-in-heatmap method for improving the prediction accuracy of facial landmarks. Instead of using a Gaussian map for each facial landmark, Wu et al. [68] proposed to create a boundary heatmap mask for feature map fusion and demonstrated its merits in robust facial landmark localization.

5.3.2 Loss Function

Similar to coordinate regression, the design of a proper loss function is crucial for heatmap regression-based facial landmark localization. Most of the existing heatmap regression methods use MSE or L1 loss for heatmap generation via an encoder-decoder network. However, a model trained with MSE or L1 loss tends to predict blurry and dilated heatmaps with low intensity on foreground pixels compared to the ground-truth ones. To address this issue, Wang et al. [76] proposed an adaptive Wing loss function for heatmap regression. In contrast to the original Wing loss [20], the adaptive Wing loss is a tailored version for heatmap generation. The adaptive Wing loss is able to adapt its shape to different types of ground-truth heatmap pixels. This adaptability penalizes loss more on foreground pixels while less on background pixels, hence improving the quality of a generated heatmap and the performance of the final landmark localization task in terms of accuracy.

To be specific, the adaptive Wing loss function is defined as:

$$AWing(y, \hat{y}) = \begin{cases} w \ln(1 + |\frac{y-\hat{y}}{\epsilon}|^{\alpha - y}) & \text{if } |y - \hat{y}| < \theta \\ A|y - \hat{y}| - C & \text{otherwise} \end{cases}, \tag{5.10}$$

where y and $\hat{y}$ are the intensities of the pixels on the ground truth and predicted heatmaps, respectively. w, θ, ϵ and α are positive values, $A = w(1/(1 + (\theta/\epsilon)^{(\alpha - y)}))(\alpha - y)((\theta/\epsilon)^{(\alpha - y - 1)})(1/\epsilon)$ and $C = (\theta A - w \ln(1 + (\theta/\epsilon)^{\alpha - y}))$ are designed to link different parts of the loss function continuously and smoothly at $|y - \hat{y}| = \theta$. Unlike the Wing loss, which uses w as the threshold, the adaptive Wing loss introduces a new variable θ as the threshold to switch between linear and nonlinear parts. For heatmap regression, a deep network usually regresses a value between 0 and 1, so the adaptive Wing loss sets the threshold in this range. When $|y - \hat{y}| < \theta$, adaptive Wing considers the error to be small and thus needs stronger influence. More importantly, this new loss function adopts an exponential term $\alpha - y$, which is used to adapt the shape of the loss function to y and makes the loss function smooth at the origin.

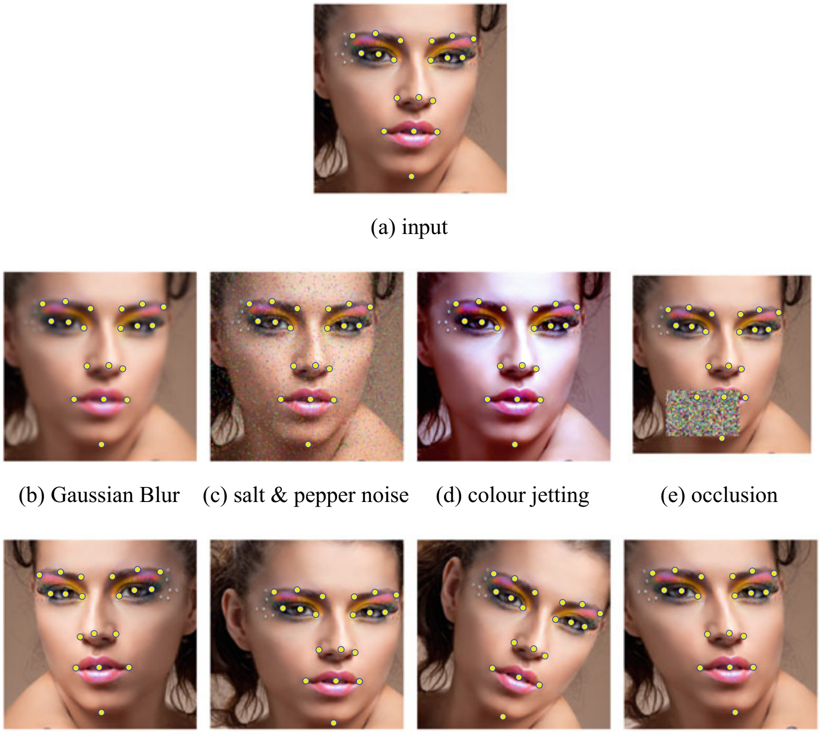
It should be noted that adaptive Wing loss is able to adapt its curvature to the ground-truth pixel values. This adaptive property reduces small errors on foreground pixels for accurate landmark localization, while tolerating small errors on background pixels for better convergence of a network.

5.4 Training Strategies

5.4.1 Data Augmentation

For a deep learning-based facial landmark localization method, a key to the success of network training is big labeled training data. However, it is a difficult and tedious task to manually label a large-scale dataset with facial landmarks. To mitigate this issue, effective data augmentation has become an essential alternative. Existing data augmentation approaches in facial landmark localization usually inject geometric and textural variations into training images. These augmentation approaches are efficient to implement and thus can be easily performed online for network training.

To investigate the impact of these data augmentation methods on the performance of a facial landmark localization model, Feng et al. [26] introduced different data augmentation approaches and performed a systematic analysis of their effectiveness in the context of



(a) input

(b) Gaussian Blur (c) salt & pepper noise (d) colour jetting (e) occlusion

(f) flip (g) bbox perturbation (h) rotation (i) shear

Fig. 5.9 Different geometric and textural data augmentation approaches for facial landmark localization. "bbox" refers to "bounding box"

deep-learning-based facial landmark localization. Feng et al. divided the existing data augmentation techniques into two categories: textural and geometric augmentation, as shown in Fig. 5.9. Textural data augmentation approaches include Gaussian blur, salt and pepper noise, color jetting, and random occlusion. Geometric data augmentation consists of horizontal image flip, bounding box perturbation, rotation and shear transformation. According to the experimental results, all data augmentation approaches improve the accuracy of the baseline model. However, the key finding is that the geometric data augmentation methods are more effective than the textural data augmentation methods for performance boosting. Furthermore, the joint use of all data augmentation approaches performs better than only using a single augmentation method.

In addition, Feng et al. [26] argued that, by applying random textural and geometric variations to the original labeled training images, some augmented samples may be harder and more effective for deep network training. However, some augmented samples are less effective. To select the most effective augmented training samples, they proposed a Hard

Augmented Example Mining (HAEM) method for effective sample mining. In essence, HAEM selects N hard samples from each mini-batch those which exhibit the largest losses but excludes the one of dominant loss. The main reason for this conservative method is that some of the samples generated by a random data augmentation method might be too difficult to train networks. Such samples become "outliers" that could disturb the convergence of the network training. Thus in each mini-batch, HAEM identifies $N + 1$ hardest samples and discards the hardest one to define the hard sample set.

5.4.2 Pose-Based Data Balancing

Existing facial landmark localization methods have achieved good performance for faces in the wild. However, extreme pose variations are still very challenging. To mitigate this problem, Feng et al. [20] proposed a simple but very effective Pose-based Data Balancing (PDB) strategy. PDB argues that the difficulty for accurately localizing faces with large poses is mainly due to data imbalance. This is a well-known problem in many computer vision applications [21].

To perform pose-based data balancing, PDB applies Principal Component Analysis (PCA) to the aligned shapes and projects them to a one dimensional space defined by the shape eigenvector (pose space) controlling pose variations. To be more specific, for a training dataset $\{\mathbf{s}_i\}_{i=1}^{N}$ with N samples, where $\mathbf{s}_i \in \mathbb{R}^{2L}$ is the ith training shape vector consisting of the 2D coordinates of all the L landmarks, the use of Procrustes Analysis aligns all the training shapes to a reference shape, i.e. the mean shape, using rigid transformations. Then PDB approximates any training shape or a new shape, $\mathbf{s}$, using a statistical linear shape model:

$$\mathbf{s} \approx \bar{\mathbf{s}} + \sum_{j=1}^{N_s} p_j \mathbf{s}_j^*, \tag{5.11}$$

where $\bar{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i$ is the mean shape over all the training samples, $\mathbf{s}_j^*$ is the jth eigenvector obtained by applying PCA to all the aligned training shapes and p_j is the coefficient of the jth shape eigenvector. Among those shape eigenvectors, we can find an eigenvector, usually the first one, that controls the yaw rotation of a face. We denote this eigenvector as $\hat{\mathbf{s}}$. Then we can obtain the pose coefficient of each training sample $\mathbf{s}_i$ as:

$$\hat{p}_i = \hat{\mathbf{s}}^T (\mathbf{s}_i - \bar{\mathbf{s}}). \tag{5.12}$$

The distribution of the pose coefficients of all the AFLW training samples is shown in Fig. 5.10. According to the Fig. 5.10, it can be seen that the AFLW dataset is not well-balanced in terms of pose variation.

With the pose coefficients of all the training samples, PDB first categorizes the training dataset into K subsets. Then it balances the training data by duplicating the samples falling into the subsets of lower cardinality. To be more specific, the number of training samples in

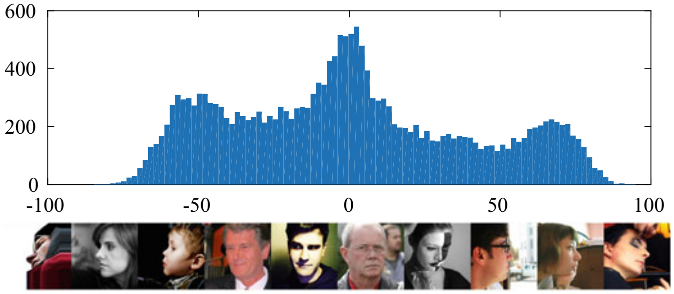


Fig. 5.10 Distribution of the head poses of the AFLW training set

the kth subset is denoted as B_k, and the maximum size of the K subsets is denoted as B^*. To balance the whole training dataset in terms of pose variation, PDB adds more training samples to the kth subset by randomly sampling $B^* - B_k$ samples from the original kth subset. Then all the subsets have the size of B^* and the total number of training samples is increased from $\sum_{k=1}^{K} B_k$ to KB^*. It should be noted that pose-based data balancing is performed before network training by randomly duplicating some training samples of each subset of lower occupancy. After pose-based data balancing, the training samples of each mini-batch are randomly sampled from the balanced training dataset for network training. As the samples with different poses have the same probability to be sampled for a mini-batch, the network training is pose-balanced.

5.5 Landmark Localization in Specific Scenarios

5.5.1 3D Landmark Localization

3D landmark localization aims to locate the 3D coordinates, including 2D positions and depth, of landmarks. The 2D landmark setting assumes that each landmark can be detected by its visual patterns. However, when faces deviate from the frontal view, the contour landmarks become invisible due to self-occlusion. In medium poses, this problem can be addressed by changing the semantic positions of contour landmarks to the silhouette, which is termed landmark marching [62]. However, in large poses where half of the face is occluded, some landmarks are inevitably invisible. In this case, the 3D landmark setting is employed to make the semantic meanings of landmarks consistent, and the face shape can be robustly recovered. As shown in Fig. 5.11, 3D landmarks are always located in their semantic positions, and they should be detected even if they are self-occluded.

In recent years, 3D face alignment has achieved satisfying performance. The methods can be divided into two categories: model-based methods and non-model-based methods. The former performs the 3D face alignment by fitting a 3D Morphable Model (3DMM),

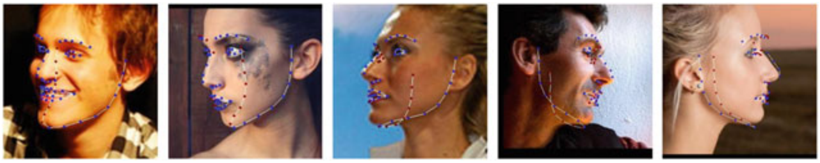


Fig. 5.11 Examples of 3D landmark localization. The blue/red ones indicate visible/invisible landmarks

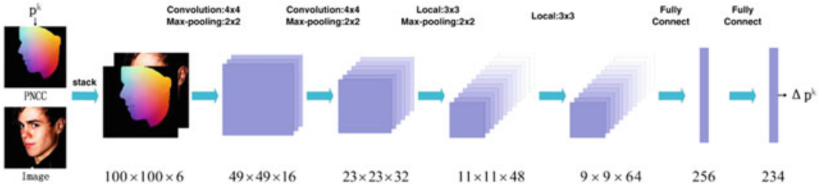


Fig. 5.12 The overview of 3DDFA. At kth iteration, 3DDFA takes the images and the projected normalized coordinate code (PNCC) generated by $\mathbf{p}^k$ as inputs and uses a convolutional neural network to predict the parameter update $\Delta\mathbf{p}^k$

which provides a strong prior of face topology. The latter extracts features from the image and directly regresses that to the 3D landmarks by deep neural networks.

5.5.1.1 3D Dense Face Alignment (3DDFA)

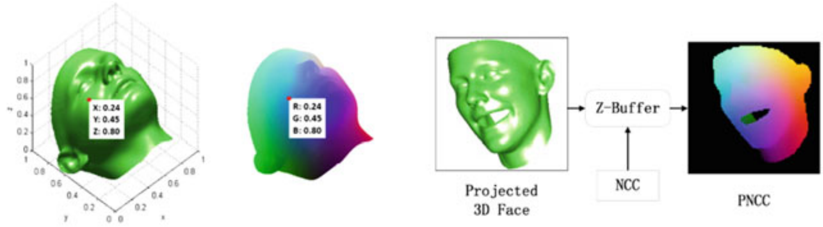
Estimating depth information from a monocular image is an ill-posed problem, and a feasible solution to realize 3D face alignment is introducing a strong 3D face prior. The 3D Dense Face Alignment (3DDFA) is a typical model-based method, which fits a 3DMM by a cascaded convolutional neural network to recover the 3D dense shape. Since the 3DMM is topology-unified, the 3D landmarks can be easily indexed after 3D shape recovery. An overview of 3DDFA is shown in Fig. 5.12. Specifically, the 3D face shape is described as:

$$\mathbf{S} = \overline{\mathbf{S}} + \mathbf{A}_{id}\alpha_{id} + \mathbf{A}_{exp}\alpha_{exp}, \tag{5.13}$$

where $\mathbf{S}$ is the 3D face shape, $\overline{\mathbf{S}}$ is the mean shape, $\mathbf{A}_{id}$ is the principle axes for identity, and $\mathbf{A}_{exp}$ is the principle axes for expression, α_{id} and α_{exp} are the identity and expression parameters that need to be estimated. To obtain the 2D positions of the 3D vertices, the 3D face is projected to the image plane by the weak perspective projection:

$$V(\mathbf{p}) = f * \mathbf{Pr} * \mathbf{R} * (\overline{\mathbf{S}} + \mathbf{A}_{id}\alpha_{id} + \mathbf{A}_{exp}\alpha_{exp}) + \mathbf{t}_{2d}, \tag{5.14}$$

where f is the scalar parameter, $\mathbf{Pr}$ is the orthographic projection matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $\mathbf{R}$ is the rotation matrix derived from the rotation angles $pitch$, yaw, $roll$, and $\mathbf{t}_{2d}$ is the translation



(a) NCC

(b) PNCC

Fig. 5.13 The illustration of the Normalized Coordinate Code (NCC) and the Projected Normalized Coordinate Code (PNCC). NCC denotes the position as its texture ($\text{NCC}_x = R, \text{NCC}_y = G, \text{NCC}_z = B$) and PNCC is generated by rendering the 3D face with NCC as its colormap

vector. Parameters for shape recovery are collected as $\mathbf{p} = [f, pitch, yaw, roll, \mathbf{t}_{2d}, \alpha_{id}, \alpha_{exp}]^T$, and the purpose of 3DDFA is to estimate $\mathbf{p}$ from the input image.

3DDFA is a cascaded-regression-based method that employs several networks to update the parameters step by step. A specially designed feature Projected Normalized Coordinate Code (PNCC) is proposed to reflect the fitting accuracy, which is formulated as:

$$\begin{aligned} \text{NCC}_d &= \frac{\overline{\mathbf{S}}_d - \min(\overline{\mathbf{S}}_d)}{\max(\overline{\mathbf{S}}_d) - \min(\overline{\mathbf{S}}_d)} \quad (d = x, y, z), \\ \text{PNCC} &= \textit{Z-Buffer}(V(\mathbf{p}), \text{NCC}), \end{aligned} \tag{5.15}$$

where $\overline{\mathbf{S}}$ is the mean shape of 3DMM, $\textit{Z-Buffer}(\nu, \tau)$ is the render operation that renders 3D mesh ν colored by τ to an image. PNCC represents the 2D locations of the visible 3D vertices on the image plane. Note that both NCC and PNCC have three channels for x, y, z, which is similar to RGB, and they can be shown in color as in Fig. 5.13.

At the kth iteration, 3DDFA constructs PNCC by the current parameter $\mathbf{p}^k$ and concatenates it with the image as input. Then, a neural network is adopted to predict the parameter update $\Delta\mathbf{p}^k$:

$$\Delta\mathbf{p}^k = Net^k(\mathbf{I}, \text{PNCC}(\mathbf{p}^k)). \tag{5.16}$$

Afterward, the parameter for the $k + 1$ iteration is updated: $\mathbf{p}^{k+1} = \mathbf{p}^k + \Delta\mathbf{p}^k$, and another network is adopted to further update the parameters until convergence. By incorporating 3D prior, 3DDFA localizes the invisible landmarks in large poses, achieving the-state-of-the-art performance. However, it is limited by the computation cost since it cascades several networks to progressively update the fitting result. To deploy 3DDFA on lightweight devices, 3DDFAv2 [63] employs a mobilenet [64] to directly regress the target parameters and also achieves satisfactory performance.

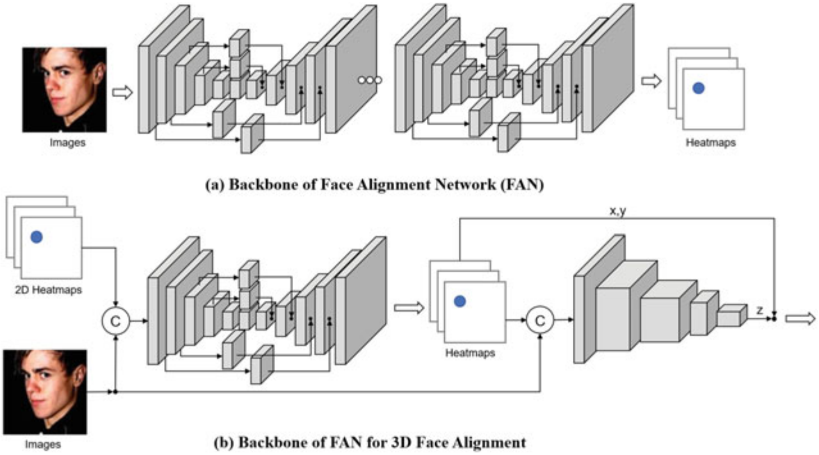

Images
Heatmaps
(a) Backbone of Face Alignment Network (FAN)
2D Heatmaps
C
x,y
Heatmaps
C
z
Images
(b) Backbone of FAN for 3D Face Alignment

Fig. 5.14 **a** The backbone of the Face Alignment Network (FAN). It consists of stacked Hourglass networks [55] in which the bottleneck blocks are replaced with the residual block of [56]. **b** The illustration of FAN for 3D face alignment. The network takes the images and their corresponding 2D landmark heatmaps as input to regress the heatmaps of the projected 3D landmarks, which are then concatenated with the image to regress the depth values of landmarks

5.5.1.2 Face Alignment Network (FAN)

Face Alignment Network (FAN) [52] is a non-model-based method for 3D face alignment, which trains a neural network to regress the landmark heatmaps. FAN constructs a strong backbone to localize 3D landmarks, shown in Fig. 5.14a. Specifically, FAN consists of four stacked hourglass networks [55], and the bottleneck blocks in each hourglass are replaced with the hierarchical, parallel, and multi-scale residual block [56] to further improve the performance. Given an input image, FAN utilizes the network to regress the landmark heatmaps, where each channel of the heatmap is a 2D Gaussian centered at the corresponding landmark's location with a standard deviation of one pixel.

To realize the regression of 3D positions, FAN designs a guided-by-2D-landmarks network to convert 2D landmarks to 3D landmarks, which bridges the performance gap between the saturating 2D landmark localization and the challenging 3D landmark localization. The overview of FAN for 3D landmark localization is shown in Fig. 5.14b. Specifically, given an RGB image and their corresponding 2D landmark heatmaps as input, FAN first regresses the heatmaps of the projected 3D landmarks, obtaining the x, y of 3D landmarks. Then, the projected 3D landmark heatmaps are combined with the input image and sent to a followed network to regress the depth value of each landmark, obtaining the full x, y, z coordinates of 3D landmarks.

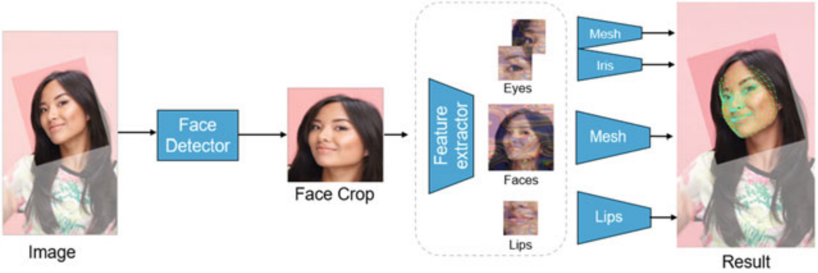


Fig. 5.15 The pipeline of the MediaPipe. Given an input image, the face region is first cropped by the face detector and then sent to the feature extractor. After that, the model is split into several sub-models to predict the global landmarks and important local landmarks including eyes and lips

5.5.1.3 MediaPipe

MediaPipe [60] is a widely used pipeline for 2D and 3D landmark localization. It is proposed to meet the real-time application requirements for face localization such as AR make-up, eye tracking, AR puppeteering, etc. Different from the cascaded framework, MediaPipe uses a single model to achieve comparable performance. The pipeline of MediaPipe is shown in Fig. 5.15. The network first extracts the global feature map from the cropped images, and then the network is split into several sub-networks. One sub-network predicts the 3D face mesh, including 3D landmarks, and outputs the regions of interest (eyes and lips). The remaining two sub-networks are employed to estimate the local landmarks of eyes and lips, respectively. The output of MediaPipe is a sparse mesh composed of 468 points. Through the lightweight architecture [61] and the region-specific heads for meaningful regions, MediaPipe has good efficiency and achieves comparable performance compared with the cascaded methods, realizing the real-time on-device inference.

5.5.1.4 3D Landmark Data

One of the main challenges of 3D landmark localization is the lack of data. Acquiring high-precision 3D face models requires expensive devices and a fully controlled environment, making large-scale data collection infeasible. To overcome this bottleneck, current methods usually label 2D projections of 3D landmarks as an alternative solution. However, it is still laborious since the self-occluded parts have to be guessed by intuition. In recent years, 300W-LP [40, 85], AFLW2000-3D [40, 85], and Menpo-3D [84] have been popular data sets for building 3D landmark localization systems. In addition to hand annotation, training data can be generated by virtual synthesis. Face Profiling [40, 85] proposes to recover a textured 3D mesh from a 2D face image and rotate the 3D mesh to given rotation angles, which can be rendered to generate virtual data, shown in Fig. 5.16. Through face profiling, not only the face samples in large poses (yaw angle up to 90°) can be obtained, but also the dataset can be augmented to any desired scale.

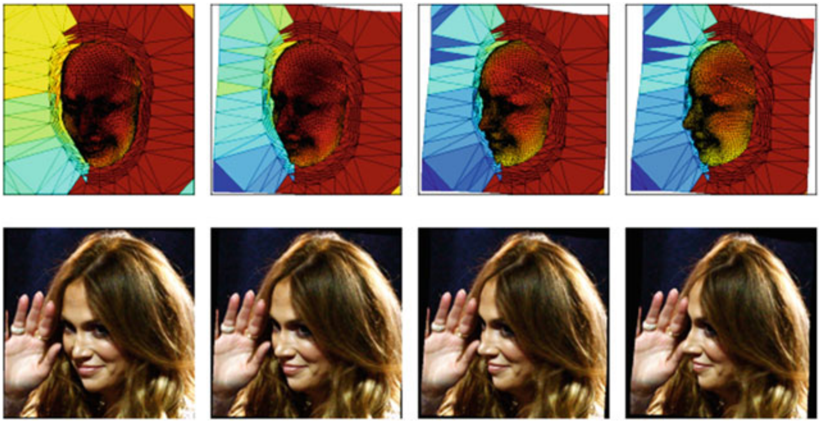


Fig. 5.16 The face profiling process

5.5.2 Landmark Localization on Masked Face

Since the outbreak of the worldwide pandemic COVID-19, facial landmark localization has encountered the great challenge of mask occlusion. First, the collection of masked face data is costly and difficult, especially during the spread of COVID-19. Second, the masked facial image suffers from severe occlusion, making the landmarks more difficult to detect. Taking the 106-point landmark setting as an example, there are around 27 nose and mouth points occluded by the facial mask (Fig. 5.18), which brings not only additional difficulty to landmark detection, but also adverse uncertainty to the ground-truth labeling. These issues cause serious harm to the deep-learning-based landmark localization that relies on labeled data.

It can be perceived that most of the issues lie in the masked face data. Therefore, a feasible and straightforward solution is synthesizing photo-realistic masked face images from mask-free ones, so as to overcome the problems of data collection and labeling. One popular approach [14] , as shown in Fig. 5.17, is composed of three steps, i.e., 3D reconstruction, mask segmentation, and re-rendering of the blended result. Given the source masked face and the target mask-free face, their 3D shapes are first recovered by a 3D face reconstruction method (such as PRNet [53]) to warp the image pixels to the UV space to generate the UV texture. Second, the mask area in the source image is detected by a facial segmentation method [90], which is also warped to the UV space to get a UV mask. Finally, the target UV texture is covered by the UV mask, and the synthesized target texture is re-rendered to the original 2D plane.

There are two benefits of this practice. First, a large number of masked face images can be efficiently produced with geometrically-reasonable and photo-realistic masks, and the mask styles are fully controlled. Second, once the target image has annotated landmarks, the synthesized one does not have to be labeled again. It can directly inherit the ground-truth

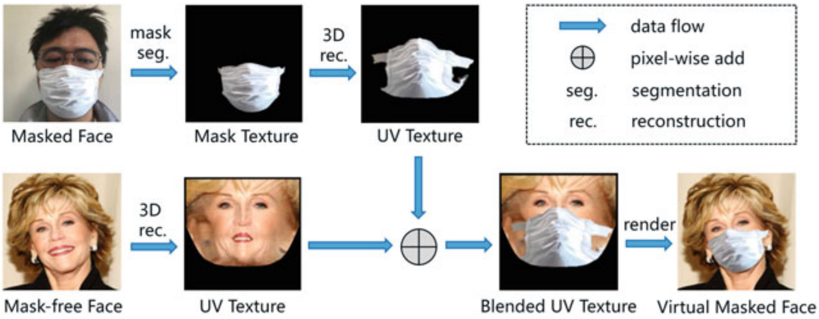


Fig. 5.17 Adding virtual mask to face images by 3D reconstruction and face segmentation

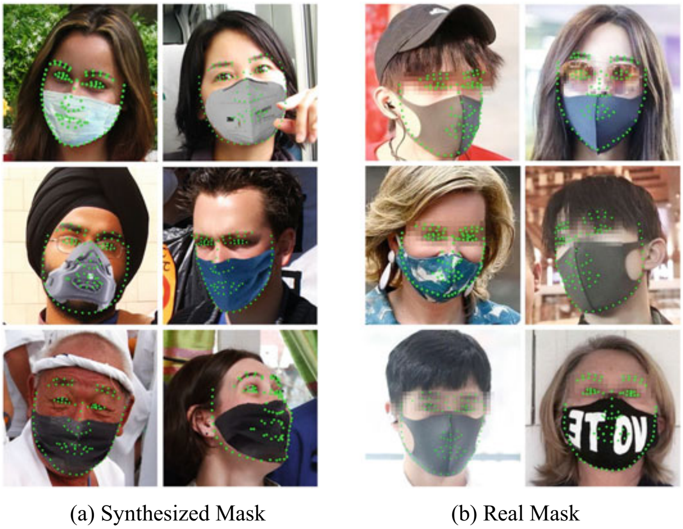


Fig. 5.18 Examples of synthesized and real masked face images [1]

landmarks for training and testing (Fig. 5.18a). With the synthesized masked face images, the mask-robust landmark detection model can be built in the similar manner as in the mask-free condition.

5.5.3 Joint Face Detection and Landmark Localization

The joint detection of face boxes and landmarks has been studied since the early ages when deep learning begins to thrive in biometrics. The initial motivation of joint detection is to boost face detection itself by incorporating landmarks to handle certain hard cases, e.g., large pose, severe occlusion, and heavy cosmetics [5, 6]. Afterward, the community

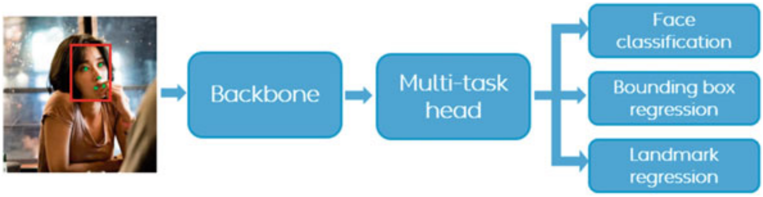


Fig. 5.19 The typical framework of joint detection of face and landmark

pays increasing attention to merging the two tasks as one. The advantages are three-fold: First, the two highly correlated tasks benefit each other when the detector is trained by the annotations from both sides. Second, the unified style brings better efficiency to the whole pipeline of face-related applications, as the two detection tasks can be accomplished by a single lightweight model. Finally, the joint model can be conveniently applied in many tasks, including face recognition, simplifying the implementation in practice. Despite the obvious advantages of the multi-task framework, building such a system requires more expensive training data with labels of multiple face attributes, improving the cost of data annotations.

Networks. The typical framework of joint face and landmark detection is shown in Fig. 5.19. The input image contains human faces that occur with arbitrary pose, occlusion, illumination, cosmetics, resolution, etc. The backbone extracts an effective feature from the input image and feeds it into the multi-task head. The multi-task head outputs the joint detection results, including at least three items, i.e., face classification, face bounding box coordinates, and landmark coordinates. Beyond typical tasks, some methods also predict the head pose, gender [8], and 3D reconstruction [11] simultaneously. The major backbones include FPN [10], Cascaded-CNN [7], multi-scale fusion within rapidly digested CNN [9], YOLO-vX style [3], etc. The former two make full use of hierarchical features and predict fine results, and the latter two have excellent efficiency for CPU-real-time applications.

Learning objectives. The framework should be trained with multiple objectives to perform joint predictions. Equation (5.17) is the typical loss formulation for multiple objective training. $\mathcal{L}_{face-cls}$ is the cross-entropy loss for face classification, which predicts the confidence of whether the candidate is a human face. $\mathcal{L}_{bbox-reg}$ is defined as the L2 or smooth L1 distance between the coordinates of the predicted bounding box and the ground truth, supervising the model to learn the bounding box locations. Similarly, $\mathcal{L}_{lm-reg}$ supervises the model to predict the landmark coordinates in the same way.

$$\mathcal{L} = \alpha_1 \beta_1 \mathcal{L}_{face-cls} + \alpha_2 \beta_2 \lambda \mathcal{L}_{bbox-reg} + \alpha_3 \beta_3 \lambda \mathcal{L}_{lm-reg}, \tag{5.17}$$

where $\{\alpha_1, \alpha_2, \alpha_3\} \in \mathbb{R}$ are the weights for balancing the training toward three objectives, $\{\beta_1, \beta_2, \beta_3\} \in \{0, 1\}$ are binary indicators that activate the supervision if the corresponding annotation presents in the training sample, and $\lambda \in \{0, 1\}$ is applied to activate the supervision of bounding box and landmark if the candidate's ground truth is human face [9]. It is worth noting that the incorporation of β enables the training on partially annotated datasets.

Datasets. The dataset most commonly used for joint detection is the WIDER FACE [13] dataset with the supplementary annotations [11]. The initial purpose of WIDER FACE is to train and evaluate face detection models. The supplementary annotation provides five-point landmarks on each face, enabling the usage for the joint detection task. Owing to the wide utilization of this dataset, most joint detection models predict five-point landmarks, which are sufficient for face alignment in most cases. Besides, some models [8, 30] trained by the 300W [57] dataset predict 68 landmarks for joint detection.

5.6 Evaluations of the State of the Arts

In this section, we introduce how to evaluate the performance of a landmark localization method, including various datasets and evaluation metrics. The evaluation results of representative methods on different datasets are also collected and demonstrated.

5.6.1 Datasets

In recent years, many datasets have been collected for training and testing of 2D facial landmark localization, including COFW [67], COFW-68 [72], 300W [65], 300W-LP [85], WFLW [68], Menpo-2D [83], AFLW [66], AFLW-19 [86], AFLW-68 [87], MERL-RAV [77] and WFLW-68 [39], which are listed in Table 5.1. We introduce some representative datasets as follows:

Table 5.1 An overview of 2D facial landmark datasets. "Train" and "Test" are the number of samples in the training set and the test set, respectively. "Landmark Num." represents the number of annotated landmarks

Dataset	Year	Train	Test	Landmark Num.
AFLW [66]	2011	20, 000	4, 386	21
300W [65]	2013	3, 148	689	68
COFW [67]	2013	1, 345	507	29
COFW-68 [72]	2014	–	507	68
300W-LP [85]	2016	61, 225	–	68
Menpo-2D [83]	2016	7, 564	7, 281	68/39
AFLW-19 [86]	2016	20, 000	4, 386	19
WFLW [68]	2018	7, 500	2, 500	98
AFLW-68 [87]	2019	20, 000	4, 386	68
MERL-RAV [77]	2020	15, 449	3, 865	68
WFLW-68 [39]	2021	7, 500	2, 500	68

300W contains 3, 837 images, some images may have more than one face. Each face is annotated with 68 facial landmarks. The 3, 148 training images are from the full set of AFW [69] (337 images), the training part of LFPW [70] (811 images), and HELEN [71] (2, 000 images). The test set is divided into a common test set and a challenging set. The common set with 554 images comes from the testing part of LFPW (224 images) and HELEN (330 images). The challenging set with 135 images is from the full set of IBUG [65]. 300W-LP [85] augments the pose variations of 300W by the face profiling technique and generates a large data set with 61, 225 samples, much of which are in profile.

COFW contains 1, 007 images with 29 annotated landmarks. The training set with 1, 345 samples is the combination of 845 LFPW samples and 500 COFW samples. The test set with 507 samples has two cases. They are annotated with 29 landmarks (the same as the training set) or 68 landmarks, and the latter is called COFW-68 [72]. Most faces in COFW have large variations in occlusion.

AFLW contains 25, 993 faces with at most 21 visible facial landmarks annotated, but excludes the annotations of invisible landmarks. A protocol [86] is built on the original AFLW and divides the dataset into 20, 000 training samples and 4, 386 test samples. The dataset has large pose variations, especially has thousands of faces in profile. AFLW-19 [86] builds a 19-landmark annotation by removing the 2 ear landmarks. AFLW-68 [87] follows the configuration in 300 W and re-annotates the images with 68 facial landmarks.

Menpo-2D has a training set with 7, 564 images, including 5, 658 front faces and 1, 906 profile faces, and a test set with 7, 281 images, including 5, 335 front faces and 1, 946 profile faces. There are two settings for different poses. The front faces are annotated by 68 landmarks, and the profile faces are annotated by 39 landmarks.

WFLW contains 7, 500 images for training and 2, 500 images for testing. Each face in WFLW is annotated with 98 landmarks and some attributes such as occlusion, make-up, expression and blur. WFLW-68 [39] converts the original 98 landmarks to 68 landmarks for convenient evaluation.

5.6.2 Evaluation Metric

There are three commonly utilized metrics to evaluate the precision of landmark localization, including Normalized Mean Error (NME), Failure Rate (FR) and Cumulative Error Distribution (CED).

Normalized Mean Error (NME) is one of the most widely used metrics in face alignment, which is defined as:

$$\mathrm{NME} = \frac{1}{\mathrm{M}} \sum_{\mathrm{i}=1}^{\mathrm{M}} \frac{||\mathbf{P}_{\mathrm{i}} - \mathbf{P}_{\mathrm{i}}{}^{*}||_{2}}{\mathrm{d}}, \tag{5.18}$$

where $\{\mathbf{P}_i\}$ is the predicted landmark coordinates, $\{\mathbf{P}_i{}^*\}$ is the ground-truth coordinates, M is the total number of landmarks, and d is the distance between outer eye corners (inter-ocular) [39, 68, 75, 79, 82]) or pupil centers (inter-pupils [76, 80]). It can be seen that the error is

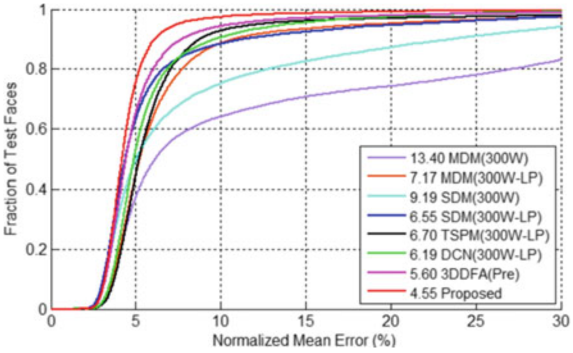


Fig. 5.20 An example of CED curve from [40]. In the curve, x is NME and y is the proportion of samples in the test set whose NMEs are less than x

normalized by d to reduce the deviation caused by face scale and image size. In some cases, the image size [39] or face box size [77] is also used as the normalization factor d. A smaller NME indicates better performance.

Failure Rate (FR) is the percentage of samples whose NMEs are higher than a certain threshold f, denoted as FR_f (f is usually set to 0.1) [57, 68, 92]. A smaller FR means better performance.

Cumulative Error Distribution (CED) is defined as a curve (x, y), where x indicates NME and y is the proportion of samples in the test set whose NMEs are less than x. Figure 5.20 shows an example of CED curve, which provides a more detailed summary of landmark localization performance. Based on CED, the **Area Under the Curve (AUC)** can be obtained by the area enclosed between the CED curve and the x-axis, whose integral interval is $x = 0$ to a threshold $x = f$, denoted as AUC_f. A larger AUC means better performance.

5.6.3 Comparison of the State of the Arts

We demonstrate the performance of some state-of-the-art methods from 2018 to 2022 on commonly used datasets, including LAB [68], SAN [73], HG-HSLE [74], AWing [76], DeCaFA [78], RWing [38], HRNet [75], LUVLi [77], SDL [81], PIPNet [39], HIH [79], ADNet [80], and SLPT [82]. It is worth noting that the reported results should not be compared directly because the model sizes and training data are different.

300W: Table 5.2 summarizes the results on the most commonly used dataset 300W, with three test subsets of "common", "challenging", and "full". The NME of the 68 facial landmarks is calculated to measure the performance. All the results are collected from the corresponding papers.

COFW: Table 5.3 summarizes the results on the COFW and COFW-68, which mainly measure the robustness to occlusion. There are two protocols, the within-dataset protocol (COFW) and cross-dataset protocol (COFW-68). For the within-dataset protocol, the model is trained with $1,345$ images and validated with 507 images on COFW. The NME and $\text{FR}_{0.1}$

Table 5.2 Performance comparison on 300 W, “Common”, “Challenge”, and “Full” represent common set, challenging set, and full set of 300W, respectively. “Backbone” represents the model architecture used by each method

Method	Year	Backbone	NME(%, inter-ocular)		
			Full	Common	Challenge
LAB [68]	2018	ResNet-18	3.49	2.98	5.19
SAN [73]	2018	ITN-CPM	3.98	3.34	6.60
HG-HSLE [74]	2019	Hourglass	3.28	2.85	5.03
AWing [76]	2019	Hourglass	3.07	2.72	4.52
DeCaFA [78]	2019	Cascaded U-net	3.39	2.93	5.26
HRNet [75]	2020	HRNetV2-W18	3.32	2.87	5.15
LUVLi [77]	2020	DU-Net	3.23	2.76	5.16
SDL [81]	2020	DA-Graph	3.04	2.62	4.77
PIPNet [39]	2021	ResNet-101	3.19	2.78	4.89
HIH [79]	2021	2 Stacked HGs	3.33	2.93	5.00
ADNet [80]	2021	Hourglass	2.93	2.53	4.58
SLPT [82]	2022	HRNetW18C-lite	3.17	2.75	4.90

of the 29 landmarks are utilized for comparison. For the cross-dataset protocol, the training set includes the complete 300W dataset (3, 837 images), and the test set is COFW-68 (507 images). The NME and $\text{FR}_{0.1}$ of the 68 landmarks are reported. All the results are collected from the corresponding papers.

WFLW: Table 5.4 summarizes the results on WFLW. The test set is divided into six subsets to evaluate the models in various specific scenarios, which are pose (326 images), expression (314 images), illumination (698 images), make-up (206 images), occlusion (736 images), and blur (773 images). The three metrics of NME, $\text{FR}_{0.1}$ and $\text{AUC}_{0.1}$ of the 98 landmarks are employed to demonstrate the stability of landmark localization. The results of SAN are from the supplemental material of [82]. The results of LUVLi are from the supplemental materials of [77]. The results of SLPT are from the supplemental materials of [82]. For HRNet, the NME is from [75], and the $\text{FR}_{0.1}$ and $\text{AUC}_{0.1}$ are from [81]. The other results are from the corresponding papers.

Table 5.3 Performance comparison on COFW and COFW-68. The threshold of Failure Rate (FR) and Area Under the Curve (AUC) are set to 0.1

Method	Year	Backbone	COFW				COFW-68	
			Inter-Ocular		Inter-Pupil		Inter-Ocular	
			NME(%)	$\mathrm{FR}_{0.1}$(%)	NME(%)	$\mathrm{FR}_{0.1}$(%)	NME(%)	$\mathrm{FR}_{0.1}$(%)
LAB [68]	2018	ResNet-18	3.92	0.39	–	–	4.62	2.17
AWing [76]	2019	Hourglass	–	–	4.94	0.99		
RWing [38]	2020	CNN-6&8	–	–	4.80	–	–	–
HRNet [75]	2020	HRNetV2-W18	3.45	0.19	–	–	–	–
SDL [81]	2020	DA-Graph	–	–	–	–	4.22	0.39
PIPNet [39]	2021	ResNet-101	3.08	–	–	–	4.23	–
HIH [79]	2021	2 Stacked HGs	3.28	0.00	–	–	–	–
ADNet [80]	2021	Hourglass	–	–	4.68	0.59	–	–
SLPT [82]	2022	HRNetW18C-lite	3.32	0.00	4.79	1.18	4.10	0.59

5.7 Conclusion

Landmark localization has been the cornerstone of many widely used applications. For example, face recognition utilizes landmarks to align faces, face AR applications use landmarks to enclose eyes and lips, and face animation fits 3D face models by landmarks. In this chapter, we have discussed typical methods of landmark localization, including coordinate regression and heatmap regression, and some special landmark localization scenarios. Although these strategies have made great progress and enabled robust localization in most cases, there are still many challenging problems remaining to be addressed in advanced applications, including faces in profile, large-region occlusion, temporal consistency, and pixel-level accuracy. With the development of face applications, the benchmark of landmarks on accuracy, robustness, and computation cost becomes higher and higher and more sophisticated landmark localization strategies are needed.

Table 5.4 Performance comparison on WFLW. All, Pose, Expr., Illu., M.u., Occ. and Blur represent full set, pose set, expression set, illumination set, make-up set, occlusion set, and blur set of WFLW, respectively. All results used inter-ocular distance for normalization. The threshold of Failure Rate (FR) and Area Under the Curve (AUC) are set to 0.1

Metric	Method	Year	Backbone	Pose	Expr.	Illu.	M.u.	Occ.	Blur	All
NME (%)	LAB [68]	2018	ResNet-18	10.24	5.51	5.23	5.15	6.79	6.32	5.27
	SAN [73]	2018	ITN-CPM	10.39	5.71	5.19	5.49	6.83	5.80	5.22
	AWing [76]	2019	Hourglass	7.38	4.58	4.32	4.27	5.19	4.96	4.36
	DeCaFA [78]	2019	Cascaded U-net	8.11	4.65	4.41	4.63	5.74	5.38	4.62
	RWing [38]	2020	CNN-6&8	9.79	6.16	5.54	6.65	7.05	6.41	5.60
	HRNet [75]	2020	HRNetV2-W18	7.94	4.85	4.55	4.29	5.44	5.42	4.60
	LUVLi [77]	2020	DU-Net	7.56	4.77	4.30	4.33	5.29	4.94	4.37
	SDL [81]	2020	DA-Graph	7.36	4.49	4.12	4.05	4.98	4.82	4.21
	PIPNet [39]	2021	ResNet-101	7.51	4.44	4.19	4.02	5.36	5.02	4.31
	HIH [79]	2021	2 Stacked HGs	7.20	4.28	4.42	4.03	5.00	4.79	4.21
	ADNet [80]	2021	Hourglass	6.96	4.38	4.09	4.05	5.06	4.79	4.14
	SLPT [82]	2022	HRNetW18C-lite	6.96	4.45	4.05	4.00	5.06	4.79	4.14
$FR_{0.1}$ (%)	LAB [68]	2018	ResNet-18	28.83	6.37	6.73	7.77	13.72	10.74	7.56
	SAN [73]	2018	ITN-CPM	27.91	7.01	4.87	6.31	11.28	6.60	6.32
	AWing [76]	2019	Hourglass	13.50	2.23	2.58	2.91	5.98	3.75	2.84
	DeCaFA [78]	2019	Cascaded U-net	21.40	3.73	3.22	6.15	9.26	6.61	4.84

(continued)

Table 5.4 (continued)

Metric	Method	Year	Backbone	Pose	Expr.	Illu.	M.u.	Occ.	Blur	All
	RWing [38]	2020	CNN-6&8	34.36	9.87	7.16	9.71	15.22	10.61	8.24
	HRNet [75]	2020	HRNetV2-W18	23.01	3.50	4.72	2.43	8.29	6.34	4.64
	LUVLi [77]	2020	DU-Net	15.95	3.18	2.15	3.40	6.39	3.23	3.12
	SDL [81]	2020	DA-Graph	15.95	2.86	2.72	1.45	5.29	4.01	3.04
	HIH [79]	2021	2 Stacked HGs	14.41	2.55	2.15	1.46	5.71	3.49	2.84
	ADNet [80]	2021	Hourglass	12.72	2.15	2.44	1.94	5.79	3.54	2.72
	SLPT [82]	2022	HRNetW18C-lite	12.27	2.23	1.86	3.40	5.98	3.88	2.76
$AUC_{0.1}$	LAB [68]	2018	ResNet-18	0.235	0.495	0.543	0.539	0.449	0.463	0.532
	SAN [73]	2018	ITN-CPM	0.236	0.462	0.555	0.522	0.456	0.493	0.536
	AWing [76]	2019	Hourglass	0.312	0.515	0.578	0.572	0.502	0.512	0.572
	DeCaFA [78]	2019	Cascaded U-net	0.292	0.546	0.579	0.575	0.485	0.494	0.563
	HRNet [75]	2020	HRNetV2-W18	0.251	0.510	0.533	0.545	0.459	0.452	0.524
	RWing [38]	2020	CNN-6&8	0.290	0.465	0.518	0.510	0.456	0.456	0.518
	LUVLi [77]	2020	DU-Net	0.310	0.549	0.584	0.588	0.505	0.525	0.557
	SDL [81]	2020	DA-Graph	0.315	0.566	0.595	0.604	0.524	0.533	0.589
	HIH [79]	2021	2 Stacked HGs	0.332	0.583	0.605	0.601	0.525	0.546	0.593
	ADNet [80]	2021	Hourglass	0.344	0.523	0.580	0.601	0.530	0.548	0.602
	SLPT [82]	2022	HRNetW18C-lite	0.348	0.574	0.601	0.605	0.515	0.535	0.595

Acknowledgements We acknowledge Xiaomei Zhang, Chang Yu, Zidu Wang for helpful discussions and contributions related to this chapter. This work is supported in part by Chinese National Natural Science Foundation Projects under Grants 62176256, and in part by the Youth Innovation Promotion Association CAS under Grant Y2021131.

References

1. Xiang, M., Liu, Y., Liao, T., Zhu, X., Yang, C., Liu, W., Shi, H.: The 3rd grand challenge of lightweight 106-point facial landmark localization on masked faces. In: International Conference on Multimedia & Expo Workshops, pp. 1–6. IEEE (2021)
2. Sha, Y., Zhang, J., Liu, X., Wu, Z., Shan, S.: Efficient face alignment network for masked face. In: International Conference on Multimedia & Expo Workshops, pp. 1–6. IEEE (2021)
3. Liu, Y., Chen, C., Zhang, M., Li, J., Xu, W.: Joint face detection and landmark localization based on an extremely lightweight network. In: International Conference on Image and Graphics, pp. 351–361. Springer, Cham (2021)
4. Sha, Y.: Towards occlusion robust facial landmark detector. In: International Conference on Automatic Face and Gesture Recognition, pp. 1–8. IEEE (2021)
5. Li, Y., Sun, B., Wu, T., Wang, Y.: Face detection with end-to-end integration of a convnet and a 3d model. In: European Conference on Computer Vision, pp. 420–436. Springer, Cham (2016)
6. Chen, D., Hua, G., Wen, F., Sun, J.: Supervised transformer network for efficient face detection. In: European Conference on Computer Vision, pp. 122–138. Springer, Cham (2016)
7. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. Signal Process. Lett. IEEE 1499–1503 (2016)
8. Ranjan, R., Patel, V. M., Chellappa, R.: Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. Trans. Pattern Anal. Mach. Intell. IEEE. 121–135 (2017)
9. Zhuang, C., Zhang, S., Zhu, X., Lei, Z., Wang, J., Li, S. Z.: Fldet: a cpu real-time joint face and landmark detector. In: International Conference on Biometrics, pp. 1–8. IEEE (2019)
10. Xu, Y., Yan, W., Yang, G., Luo, J., Li, T., He, J.: CenterFace: joint face detection and alignment using face as point. Scientific Programming (2020)
11. Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: Retinaface: single-shot multi-level face localisation in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5203–5212. IEEE (2020)
12. Deng, J., Guo, J., Zafeiriou, S.: Single-stage joint face detection and alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. IEEE (2019)
13. Yang, S., Luo, P., Loy, C. C., Tang, X.: Wider face: a face detection benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5525–5533. IEEE (2016)
14. Wang, J., Liu, Y., Hu, Y., Shi, H., Mei, T.: Facex-zoo: a pytorch toolbox for face recognition. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 3779–3782 (2021)
15. Wen, T., Ding, Z., Yao, Y., Ge, Y., Qian, X.: Towards efficient masked-face alignment via cascaded regression. In: International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1–5. IEEE (2021)
16. Hu, H., Wang, C., Jiang, T., Guo, Z., Han, Y., Qian, X.: Robust and efficient facial landmark localization. In: International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1–7. IEEE (2021)

17. Lai, S., Liu, L., Chai, Z., Wei, X.: Light weight facial landmark detection with weakly supervised learning. In: International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1–6. IEEE (2021)
18. Guo, X., Li, S., Yu, J., Zhang, J., Ma, J., Ma, L., Ling, H.: PFLD: a practical facial landmark detector. arXiv preprint arXiv:1902.10859 (2019)
19. Gao, P., Lu, K., Xue, J., Lyu, J., Shao, L.: A facial landmark detection method based on deep knowledge transfer. IEEE Trans. Neural Netw. Learn. Syst. (2021)
20. Feng, Z.H., Kittler, J., Awais, M., Huber, P., Wu, X.J.: Wing loss for robust facial landmark localisation with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2235–2245 (2018)
21. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 761–769 (2016)
22. Wu, Y., Shah, S.K., Kakadiaris, I.A.: GoDP: globally Optimized Dual Pathway deep network architecture for facial landmark localization in-the-wild. Image Vis. Comput. **73**, 1–16 (2018)
23. Cootes, T.F., Walker, K., Taylor, C.J.: View-based active appearance models. In: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 227–232 (2000)
24. Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3476–3483 (2013)
25. Rashid, M., Gu, X., Jae Lee, Y.: Interspecies knowledge transfer for facial keypoint detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6894–6903 (2017)
26. Feng, Z.H., Kittler, J., Wu, X.J.: Mining hard augmented samples for robust facial landmark localization with CNNs. IEEE Signal Process. Lett. **26**(3), 450–454 (2019)
27. Xiao, S., Feng, J., Liu, L., Nie, X., Wang, W., Yan, S., Kassim, A.: Recurrent 3d-2d dual learning for large-pose facial landmark detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1633–1642 (2017)
28. Huang, Z., Zhou, E., Cao, Z.: Coarse-to-fine face alignment with multi-scale local patch regression. arXiv preprint arXiv:1511.04901 (2015)
29. Yang, J., Liu, Q., Zhang, K.: Stacked hourglass network for robust facial landmark localisation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 79–87 (2017)
30. Deng, J., Trigeorgis, G., Zhou, Y., Zafeiriou, S.: Joint multi-view face alignment in the wild. IEEE Trans. Image Process. **28**(7), 3636–3648 (2019)
31. Girshick, R.: Fast R-Cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
32. Taigman, Y., Yang, M., Ranzato, M.A., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014)
33. Cao, C., Weng, Y., Lin, S., Zhou, K.: 3D shape regression for real-time facial animation. ACM Trans. Graph. (TOG) **32**(4), 1–10 (2013)
34. Bettadapura, V.: Face expression recognition and analysis: the state of the art. arXiv preprint arXiv:1203.6722 (2012)
35. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Niebner, M.: Face2face: real-time face capture and reenactment of rgb videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)

36. Trigeorgis, G., Snape, P., Nicolaou, M.A., Antonakos, E., Zafeiriou, S.: Mnemonic descent method: a recurrent process applied for end-to-end face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4177–4187 (2016)
37. Broy, M.: Software engineering — from auxiliary to key technologies. In: Broy, M., Dener, E. (eds.) Software Pioneers, pp. 10–13. Springer, Heidelberg (2002)
38. Feng, Z.H., Kittler, J., Awais, M., Wu, X.J.: Rectified wing loss for efficient and robust facial landmark localisation with convolutional neural networks. Int. J. Comput. Vis. **128**(8), 2126–2145 (2020)
39. Jin, H., Liao, S., Shao, L.: Pixel-in-pixel net: towards efficient facial landmark detection in the wild. Int. J. Comput. Vis. **129**(12), 3174–3194 (2021)
40. Zhu, X., Liu, X., Lei, Z., Li, S.Z.: Face alignment in full pose range: a 3D total solution. IEEE Trans. Pattern Anal. Mach. Intell. **41**(1), 78–92 (2017)
41. Blanz, V., Vetter, T.: Face recognition based on fitting a 3D morphable model. IEEE Trans. Pattern Anal. Mach. Intell. **25**(9), 1063–1074 (2003)
42. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Training models of shape from sets of examples. In: BMVC92, pp. 9–18. Springer, London (1992)
43. Cootes, T.F., Taylor, C.J.: Combining elastic and statistical models of appearance variation. In: European Conference on Computer Vision, pp. 149–163 (2000)
44. Wang, N., Gao, X., Tao, D., Yang, H., Li, X.: Facial feature point detection: a comprehensive survey. Neurocomputing **275**, 50–65 (2018)
45. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867–1874 (2014)
46. King, D.E.: Dlib-ml: a machine learning toolkit. J. Mach. Learn. Res. **10**, 1755–1758 (2009)
47. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
48. Yu, R., Saito, S., Li, H., Ceylan, D., Li, H.: Learning dense facial correspondences in unconstrained images. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4723–4732 (2017)
49. Jourabloo, A., Liu, X.: Pose-invariant 3D face alignment. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3694–3702 (2015)
50. Liu, Y., Jourabloo, A., Ren, W., Liu, X.: Dense face alignment. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1619–1628 (2017)
51. Bulat, A., Tzimiropoulos, G.: Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge. In: European Conference on Computer Vision, pp. 616–624. Springer, Cham (2016)
52. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1021–1030 (2017)
53. Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3d face reconstruction and dense alignment with position map regression network. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 534–551 (2018)
54. Gu, L., Kanade, T.: 3D alignment of face in a single image. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1305–1312 (2006)
55. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision, pp. 483–499 (2016)
56. Bulat, A., Tzimiropoulos, G.: Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3706–3714 (2017)

57. Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: database and results. Image Vis. Comput. **47**, 3–18 (2016)
58. Shen, J., Zafeiriou, S., Chrysos, G.G., Kossaifi, J., Tzimiropoulos, G., Pantic, M.: The first facial landmark tracking in-the-wild challenge: benchmark and results. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 50–58 (2015)
59. Zafeiriou, S., Trigeorgis, G., Chrysos, G., Deng, J., Shen, J.: The menpo facial landmark localisation challenge: a step towards the solution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 170–179 (2017)
60. Grishchenko, I., Ablavatski, A., Kartynnik, Y., Raveendran, K., Grundmann, M.: Attention mesh: high-fidelity face mesh prediction in real-time. arXiv preprint arXiv:2006.10962 (2020)
61. Lv, J., Shao, X., Xing, J., Cheng, C., Zhou, X.: A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3317–3326 (2017)
62. Zhu, X., Lei, Z., Yan, J., Yi, D., Li, S.Z.: High-fidelity pose and expression normalization for face recognition in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 787–796 (2015)
63. Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z., Li, S.Z.: Towards fast, accurate and stable 3d dense face alignment. In: European Conference on Computer Vision, pp. 152–168 (2020)
64. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
65. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: the first facial landmark localization challenge. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 397–403 (2013)
66. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 2144–2151 (2011)
67. Burgos-Artizzu, X.P., Perona, P., Dollár, P.: Robust face landmark estimation under occlusion. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1513–1520 (2013)
68. Wu, W., Qian, C., Yang, S., Wang, Q., Cai, Y., Zhou, Q.: Look at boundary: a boundary-aware face alignment algorithm. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2129–2138 (2018)
69. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2879–2886 (2012)
70. Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.: Localizing parts of faces using a consensus of exemplars. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2930–2940 (2013)
71. Le, V., Brandt, J., Lin, Z., Bourdev, L., Huang, T.S.: Interactive facial feature localization. In: European Conference on Computer Vision, pp. 679–692 (2012)
72. Ghiasi, G., Fowlkes, C.C.: Occlusion coherence: localizing occluded faces with a hierarchical deformable part model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2385–2392 (2014)
73. Dong, X., Yan, Y., Ouyang, W., Yang, Y.: Style aggregated network for facial landmark detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 379–388 (2018)
74. Zou, X., Zhong, S., Yan, L., Zhao, X., Zhou, J., Wu, Y.: Learning robust facial landmark detection via hierarchical structured ensemble. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 141–150 (2019)

75. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W.: Deep high-resolution representation learning for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3349–3364 (2020)
76. Wang, X., Bo, L., Fuxin, L.: Adaptive wing loss for robust face alignment via heatmap regression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6971–6981 (2019)
77. Kumar, A., Marks, T.K., Mou, W., Wang, Y., Jones, M., Cherian, A., Koike-Akino, T., Liu, X., Feng, C.: Luvli face alignment: estimating landmarks' location, uncertainty, and visibility likelihood. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8236–8246 (2020)
78. Dapogny, A., Bailly, K., Cord, M.: Decafa: deep convolutional cascade for face alignment in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6893–6901 (2019)
79. Lan, X., Hu, Q., Cheng, J.: Revisting quantization error in face alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1521–1530 (2021)
80. Huang, Y., Yang, H., Li, C., Kim, J., Wei, F.: Adnet: leveraging error-bias towards normal direction in face alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3080–3090 (2021)
81. Li, W., Lu, Y., Zheng, K., Liao, H., Lin, C., Luo, J., Cheng, C.T., Xiao, J., Lu, L., Kuo, C.F., Miao, S.: Structured landmark detection via topology-adapting deep graph learning. In: European Conference on Computer Vision, pp. 266–283 (2020)
82. Xia, J., Qu, W., Huang, W., Zhang, J., Wang, X., Xu, M.: Sparse local patch transformer for robust face alignment and landmarks inherent relation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4052–4061 (2022)
83. Trigeorgis, G., Snape, P., Nicolaou, M.A., Antonakos, E., Zafeiriou, S.: Mnemonic descent method: a recurrent process applied for end-to-end face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4177–4187 (2016)
84. Deng, J., Roussos, A., Chrysos, G., Ververas, E., Kotsia, I., Shen, J., Zafeiriou, S.: The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking. Int. J. Comput. Vis. **127**(6), 599–624 (2019)
85. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: a 3d solution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 146–155 (2016)
86. Zhu, S., Li, C., Loy, C.C., Tang, X.: Unconstrained face alignment via cascaded compositional learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3409–3417 (2016)
87. Qian, S., Sun, K., Wu, W., Qian, C., Jia, J.: Aggregation via separation: boosting facial landmark detector with semi-supervised style translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10153–10163 (2019)
88. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 532–539 (2013)
89. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. Int. J. Comput. Vis. **107**(2), 177–190 (2014)
90. Zheng, Q., Deng, J., Zhu, Z., Li, Y., Zafeiriou, S.: Decoupled multi-task learning with cyclical self-regulation for face parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4156–4165 (2022)

91. Martyniuk, T., Kupyn, O., Kurlyak, Y., Krashenyi, I., Matas, J., Sharmanska, V.: DAD-3DHeads: a Large-scale dense, accurate and diverse dataset for 3D head alignment from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20942–20952 (2022)
92. Shao, X., Xing, J., Lyu, J., Zhou, X., Shi, Y., Maybank, S.J.: Robust face alignment via deep progressive reinitialization and adaptive error-driven learning. IEEE Trans. Pattern Anal. Mach. Intell. (2021)

Facial Attribute Analysis

6

Jun Wan, Zichang Tan, and Ajian Liu

6.1 Introduction

Facial attributes indicate the intuitive semantic descriptions of a human face like gender, race, expression, and so on. In the past few years, automated facial attribute analysis has become an active field in the area of biometric recognition due to its wide range of possible applications, such as face verification [5, 59], face identification [63, 80], or surveillance [110], just to mention a few. According to the tasks and applications, faces can be described by different *attributes* (e.g., age, gender, ethnicity, and eyeglass) [75, 94], which can be recognized by automated techniques. We can categorize these attributes into two aspects:

- **Binary versus Fine-grained/Non-binary Types.** According to the number of values each attribute can take, facial attributes can be divided into binary and fine-grained (or non-binary) types. The latter indicates that the facial attributes need to be described by more than two values. For instance, binary facial attributes generally include gender, wearing eyeglasses/hats, etc., whereas age, ethnicity, and facial expression (or emotion) are typical fine-grained attributes.
- **Local versus Global Types.** An attribute can also be categorized in terms of the region of the face in which the attribute can be found. Local attributes can thus be inferred *locally*

J. Wan (✉) · A. Liu
Institute of Automation, Chinese Academy of Sciences, Beijing, China
e-mail: jun.wan@ia.ac.cn

A. Liu
e-mail: ajian.liu@ia.ac.cn

Z. Tan
Baidu Research, Beijing, China
e-mail: tanzichang@baidu.com

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_6

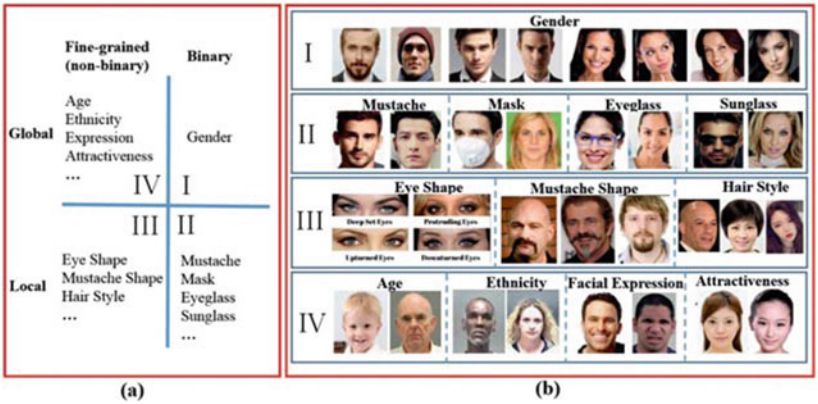


Fig. 6.1 Face attribute analysis with four basic categories. **a** Global & Binary, Local & Binary, Local & Fine-grained, and Global & Fine-grained, **b** the corresponding samples of these four types

from parts of faces (e.g., eyeglass, mustache, big nose, etc.), while global attributes regard the face as a whole (e.g., age, gender, ethnicity, etc.).

Therefore, as shown in Fig. 6.1, we can distinguish four basic types of facial attributes according to the above definition. A global face attribute of the binary type is gender, whereas attributes like mustache, mask, eyeglass, sunglasses, and big nose belong to the second type (locally binary). Attributes such as eye, mustache shape, and hair style, from the local faces can be grouped into local fine-grained types. Meanwhile, age and ethnicity pertain to globally fine-gained types.

Generally, automatic facial attribute recognition is an important task in facial attribute analysis. We can roughly divide the facial attribute recognition into the following tasks: facial age estimation, gender and ethnicity recognition, facial expression recognition, and joint learning of multiple attributes (e.g., smiling, eyeglass, long hair, and so on). For facial age estimation, it aims to estimate a person's age by a given face. Although the topic has been studied for dozens of years, accurate age estimation is still a challenging task due to the complex randomness in facial aging (e.g., various genes and work and living environment). Besides, facial expression can be regarded as an intuitive signal to reflect people's psychological activities, and therefore, facial expression recognition has attracted increasing attention in recent years. Facial expression recognition is also a challenging task due to the ambiguity between different expressions. Moreover, gender and ethnicity recognition plays an important role in video surveillance. Compared with facial age estimation and facial expression recognition, gender and ethnicity recognition are simpler because the categories of gender, and ethnicity are clearly defined (e.g., male vs. female). Moreover, in real-world applications, there are usually dozens of attributes needed to be analyzed at the same time. Many studies focus on how to formulate a multi-task learning framework to jointly learn multiple attributes together. Joint learning of multiple attributes allows knowledge transfer among different attributes. On the other hand, the multi-task learning network could predict all attributes at once, where the calculation is saved to some extent.

In addition to facial attribute recognition, facial attribute manipulation is an important topic in facial attribute analysis. Facial attribute manipulation aims to synthesize a new facial image by modifying some facial attributes, including changing, adding or removing the desired attributes (e.g., changing the hair style, adding the eyeglass, and removing one's beard). Current manipulation methods are generally based on generative models like Generative Adversarial Networks (GANs) [31], where a generator and a discriminator play a mini-max game against each other.

The success of deep learning mainly depends on three factors: model, data, and computing ability. In addition to reviewing the model algorithms of face attribute recognition and manipulation, the development of data is also a focus in this survey. In the era of deep learning, data is the guarantee of model performance. In recent years, a large number of databases have emerged in the field of face attribute analysis, and with the development of deep learning, the database has become larger and larger, like CACD [11] and MIVIA [32]. Moreover, due to the wide attention paid to face attribute analysis in recent years, many competitions have been held to promote the development of the field. For example, Chalearn held two consecutive age estimation competitions in 2015 and 2016. These competitions have greatly promoted the development of relevant fields on facial attribute analysis.

In this chapter, we present a comprehensive survey on facial attribute analysis as shown in Fig. 6.2. The distinguishing insights of this survey are as follows:

- We present a comprehensive review on facial attribute recognition, including facial age estimation, gender and ethnicity recognition, and the works of multi-task learning for facial attribute recognition.
- We summarize the existing datasets for facial attribute analysis. The characteristics of each dataset and the differences among them are analyzed and discussed.
- We introduce the recent competition in facial attribute analysis, and further analyze its challenges and future development trends.

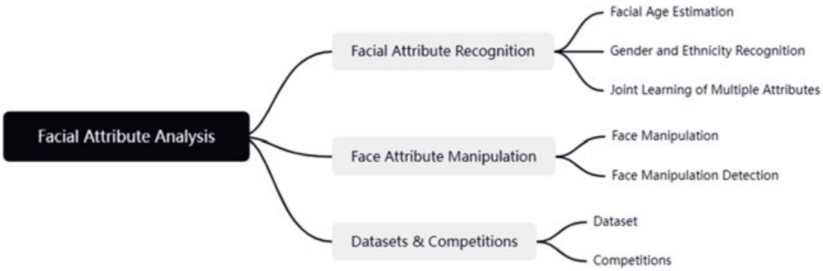

Facial Attribute Analysis
Facial Attribute Recognition
Facial Age Estimation
Gender and Ethnicity Recognition
Joint Learning of Multiple Attributes
Face Attribute Manipulation
Face Manipulation
Face Manipulation Detection
Datasets & Competitions
Dataset
Competitions

Fig. 6.2 We present a comprehensive survey in some hot subtopics of facial attribute analysis

6.2 Facial Age Estimation

Facial age estimation (FAE) refers to predicting a person's age (accumulated years after birth) from his/her face image. It has received a lot of attention due to its wide range of applications, such as video surveillance, image retrieval, and human–computer interaction. On the one hand, the accuracy of FAE is typically limited by the factors of other facial attribute analysis tasks, including pose, facial expression, illumination, occlusion, makeup, and hairstyle. On the other hand, the FAE task also faces three unique challenges:

- Facial aging is uncontrollable. No one can accurately predict the aging process.
- Facial aging patterns are personalized. Each person has a different aging process.
- Facial aging patterns are temporal. Facial changes at a particular time only affect future appearance and not before.

These unique challenges make FAE a difficult and challenging task. In recent decades, FAE has been extensively studied to find out the aging process and patterns. The initial methodologies for age estimation from face images were based on hand-crafted features of facial geometry or skin wrinkles [60]. Later, some papers were published for accurate age prediction. Geng et al. [30] proposed the AGing pattErn Subspace (AGES) approach to model the aging pattern and achieved mean absolute error (MAE) of about 6.22 years on FG-NET database [61]. Methods based on manifold learning [24, 25, 35] for age estimation were proposed as well. These methods learned low-dimensional feature representations via manifold learning to fit a regression function for age prediction. For instance, Guo et al. [35] proposed an age manifold learning scheme to extract facial features, then used a locally adjusted robust regressor (LARR) to estimate the age accurately, reducing the MAE to 5.07 years on FG-NET. Meanwhile, some local features have also become popular for age estimation, such as Gabor [28], Local Binary Patterns (LBP) [34], or Biologically-Inspired Features (BIF) [39]. After features were extracted with the previous local image descriptors, classification or regression methods were used for age estimation, including BIF+SVM [39] and BIF+Canonical Correlation Analysis (CCA) [38]. The use of hand-crafted methods for representing faces and extracting features has the advantage of shaping systems with less complexity. However, as mentioned earlier, the existence of FAE's unique challenges predisposes these hand-crafted approaches to inevitably lead to the loss of critical information and a dramatic increase in labor costs.

Later, the hand-crafted-based approaches were replaced by deep learning techniques. The process of feature extraction and selection is the primary distinction between them. The process is implemented manually in hand-crafted approaches, while deep learning techniques automate it and eliminate the need for human intervention. The summary of published methods in facial age estimation is shown in Table 6.1 where we group published literature into four basic categories: regression-based, classification-based, ranking-based, and label distribution learning-based methods.

Table 6.1 A summary of published methods on facial age estimation from a face image

Publication	Age estimation algorithm	Age database #images(training; testing)	Accuracy
Cai et al. [6]	Regression	Morph II (public) 80%/20%; training/testing	**Morph II** MAE: 3.63 years;
Agustsson et al. [2]	Regression	Morph II (public) 80%/20%; training/testing	**Morph II** MAE: 3.00 years;
Li et al. [66]	Regression	Morph II (public) S1, S2, S3 2 for training, 1 for testing	**Morph II** MAE: 3.15 years;
Wan et al. [111]	Regression	Morph II (public) 80%/20%; training/testing CACD (public) 150 celebrities for training 50 celebrities for testing	**Morph II** MAE: 3.30 years; **CACD** MAE: 5.24 years;
Zhang et al. [127]	Regression	Morph II (public) 80%/20%; training/testing ChaLearn LAP 2015 (public) 2,476 for training 1,136 for validation 1,079 for testing	**Morph II** MAE: 2.36 years; **ChaLearn LAP 2015** MAE: 3.137 years;
Feng et al. [23]	Ranking	Morph II (public) 80%/20%; training/testing	**Morph II** MAE: 4.59 years;
Niu et al. [87]	Ranking	Morph II (public) 80%/20%; training/testing AFAD (public) 80%/20%; training/testing	**Morph II** MAE: 3.27 years; **AFAD** MAE: 3.34 years;
Chen et al. [12]	Ranking	Morph II (public) 80%/20%; training/testing	**Morph II** MAE: 2.96 years;
Zeng et al. [124]	Ranking	Morph II (public) 80%/20%; training/testing ChaLearn LAP 2015 (public) 2,476 for training 1,136 for validation 1,079 for testing	**Morph II** MAE: 1.74 years; **ChaLearn LAP 2015** ϵ-error: 0.232;
Rodríguez et al. [97]	Classification	Morph II (public) 80%/20%; training/testing	**Morph II** MAE: 2.56 years;

(continued)

Table 6.1 (continued)

Publication	Age estimation algorithm	Age database #images(training; testing)	Accuracy
Tan et al. [103]	Classification	Morph II (public) 80%/20%; training/testing FG-NET (public) LOPO	**Morph II** MAE: 2.52 years; **FG-NET** MAE: 2.96 years;
Rothe et al. [99]	Classification	Morph II (public) 80%/20%; training/testing	**Morph II** MAE: 2.68 years;
Pan et al. [88]	Classification	Morph II (public) 80%/20%; training/testing FG-NET (public) LOPO	**Morph II** MAE: 2.16 years; **FG-NET** MAE: 2.68 years;
Gao et al. [26]	LDL	Morph II (public) 80%/20%; training/testing ChaLearn LAP 2015 (public) 2,476 for training 1,136 for validation 1,079 for testing	**Morph II** MAE: 2.42 years; **ChaLearn LAP 2015** MAE: 3.51 years ϵ-error: 0.31;
Gao et al. [27]	LDL	Morph II (public) 80%/20%; training/testing ChaLearn LAP 2015 (public) 2,476 for training 1,136 for validation 1,079 for testing	**Morph II** MAE: 1.97 years; **ChaLearn LAP 2015** MAE: 3.13 years ϵ-error: 0.272;
Akbari et al. [3]	LDL	Morph II (public) 80%/20%; training/testing	**Morph II** MAE: 1.80 years;
Deng et al. [17]	LDL	Morph II (public) 80%/20%; training/testing FG-NET (public) LOPO ChaLearn LAP 2015 (public) 2,476 for training 1,136 for validation 1,079 for testing	**Morph II** MAE: 2.15 years; **FG-NET** MAE: 2.16 years; **ChaLearn LAP 2015** MAE: 2.915 years ϵ-error: 0.243;

6.2.1 Regression-Based Methods

The most intuitive way of estimating facial age is to regard it as a regression problem. Specifically, regression-based methods treat age as a continuous value and solve the age estimation problem by finding a regressor that maps the face representation space to the age value space. The first attempt to use CNN for age estimation was proposed by Yang et al. [120]. They built an age regressor and trained it from scratch. By performing feature extraction and regression output on the faces in the images, the model outputs the predicted age and maintains the consistency of the prediction. However, the performance of facial age estimation is inferior to that obtained by BIF [39]. To further reduce the distance between the regression output and the age labels, Yi et al. [122] first used a mean square error as the loss function, and the regression output will be directly regarded as the predicted age values in the testing phase. Compared with BIF [39], the proposed method has a deeper structure, and the parameters are learned instead of hand-crafted, which leads to a performance improvement.

Cai et al. [6] used another regression model based on the Gaussian process and exploited the possibility of using low-dimensional representations combined with manifold learning to represent age patterns. Moreover, Guo et al. [37] obtained the age regressor by kernel partial least squares. Zhang et al. [29] proposed a multi-task warped Gaussian model for personalized aging patterns. Yan et al. put forward a patch-based regression framework for addressing the human age problem based on Gaussian mixture models.

However, only using a single regressor is susceptible to interference from noisy samples. In order to reduce the impact of noisy data, ARN [2] suggests utilizing multiple regression networks and obtaining the final predicted age by weighted summation in Fig. 6.3, where the series of weights comes from the similarity between the sample and multiple anchor points. Then, Wan et al. [111] designed five cascading structural frameworks to improve the performance of age estimation. Guided by the auxiliary demographic information, their frameworks are capable of extracting discriminative features for ages, which are then combined with the Gaussian process regression to further boost the performance.

In addition, Li et al. [66] proposed a new method to indirectly learn the faces of adjacent ages by using the cumulative hidden layer of AlexNet for feature extraction, which alleviates the problem of sample imbalance to some extent. Inspired by the fine-grained classification,

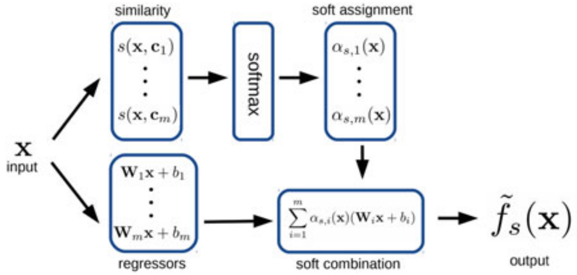

similarity
$s(\mathbf{x}, \mathbf{c}_1)$
$s(\mathbf{x}, \mathbf{c}_m)$
softmax
soft assignment
$\alpha_{s,1}(\mathbf{x})$
$\alpha_{s,m}(\mathbf{x})$
$\mathbf{x}$
input
$\mathbf{W}_1\mathbf{x} + b_1$
$\mathbf{W}_m\mathbf{x} + b_m$
regressors
$\sum_{i=1}^{m} \alpha_{s,i}(\mathbf{x})(\mathbf{W}_i\mathbf{x} + b_i)$
soft combination
$\tilde{f}_s(\mathbf{x})$
output

Fig. 6.3 Process of anchored regression network. (Image comes from [2])

Zhao et al. [127] proposed to incorporate the LSTM networks to focus on the local age-sensitive regions. Compared with the original convolutional networks, the proposed LSTM network is light-weighted and easy to train. Besides, thanks to the combination of global and local features, their method performs well on in-the-wild images. In summary, the practice of treating age as a continuous value can explore the continuity of facial aging to some extent, but it is still difficult to fully explore the continuity of facial aging because most current databases only have integer values for age labels. This limitation is also reflected in the fact that only a very few regression-based methods can achieve comparable performance with other facial age estimation approaches.

6.2.2 Classification-Based Methods

The classification-based methods formulate the FAE as a multi-class classification problem and treat different ages or age groups as independent classes. During the training stage, these approaches try to learn discriminative features using the well-known cross-entropy (CE) loss function. After extracting the aging features, the person's age is inferred by learning the classifier followed by the feature extractor.

Levi and Hassner [65] used a shallow CNN architecture, which contains three convolutional layers and two fully connected layers, to classify the Adience dataset into eight age groups. They compromised between the complexity and performance of the network to reduce the chance of over-fitting. Malli et al. [7] estimated apparent ages with age grouping to account for multiple labels per image. However, this work needs an ensemble of models to further predict the exact age, which seems relatively tedious. Zhu et al. [132] first used an age group classifier to obtain a coarse age range of the face image and then multiple local age estimators to predict the exact age. Later, Tan et al. [103] proposed to transform the age estimation problem into a series of binary classification problems, where each classifier determines whether the face image belongs to the corresponding group or not.

In 2015, Rothe et al. [99] proposed Deep EXpectation (DEX) and became the winner of the Chalearn LAP 2015 age estimation contest. The pipeline of DEX is shown in Fig. 6.4. Specifically, DEX first performs face detection and face cropping on the input faces and then inputs them to the model for classification. Finally, the predicted scores from the classifier output are calculated as the predicted age expectation and used as the final age prediction. Subsequently, Pan et al. [88] proposed a mean-variance loss function based on which to reduce the variance of the predicted scores and fit them to the true age labels. To obtain more details about specific regions of the face, Rodríguez et al. [97] introduced the attention mechanism to extract more detailed features from the face, thus reducing the complexity of the task and discarding irrelevant information.

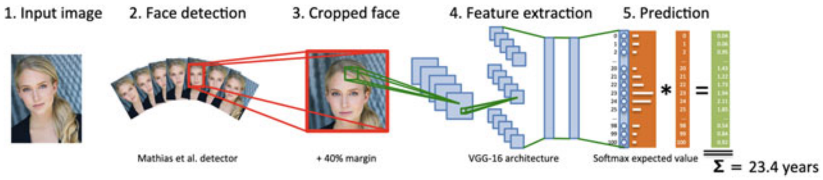


Fig. 6.4 Pipeline of DEX method for age estimation (Image comes from [99])

6.2.3 Ranking-Based Methods

Classification-based methods treated the problem of age estimation as a multi-class classification, and assumed that the labels of each category are uncorrelated and independent. However, the age labels are strongly correlated as an ordered set with a strong correlation for facial aging. To capture the relative correlation among the neighboring age labels, ranking-based methods suggest formulating facial age estimation as an ordinal problem by ranking faces from young to old.

Instead of using a regressor to output a scalar as the predicted age, OR-CNN [87] proposed to employ multiple binary classifiers after training samples through the network, converting age estimation into a binary classification for determining whether it is greater than a particular age. Specifically, it assumed that there is an encoding vector v_n, where each element represents whether the age of a face is older than a specific age value. Given a age y_n, the k-th element in v_n can be denoted as:

$$v_n^k = \begin{cases} 1 & if(y_n > k) \\ 0 & otherwise, \end{cases} \tag{6.1}$$

where the symbol $k \in [0, ..., K]$ indicates an age index and K is the maximum age. Then a softmax layer is applied to obtain the output for each binary classifier and the cross-entropy loss is adopted as the loss function. During the testing stage, the predicted age is obtained by aggregating the predictions from all classifiers:

$$\hat{y}_n = 1 + \sum_{k=1}^{K} \hat{v}_n^k, \tag{6.2}$$

where $\hat{v}_n^k$ stands for the prediction made by the k-th binary classifier. The architecture of the OR-CNN is shown in Fig. 6.5. Based on [87], Ranking-CNN [12] provides a tighter error bound for age ranking and leads to lower estimate errors. To better improve the discrimination of the different ages, ODFL [71] introduces a weighted ranking method to exploit the ranking-preserving age difference information in the learned feature space to further reinforce the model.

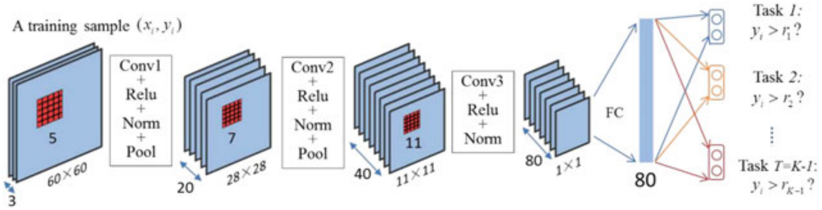


Fig. 6.5 Architecture of the OR-CNN. (Image comes from [87])

Moreover, soft-ranking [124] also treats age estimation as a ranking problem, which incorporates the two important properties of ages, i.e., the ordinal property and the correlation between adjacent ages to provide richer supervision to the networks. In soft-ranking, the ground-truth age encoding vector p_n for n-th image at age y_n can be denoted as:

$$p_n^k = \frac{1}{2} + \frac{1}{2} erf\left(\frac{k - y_n}{\sqrt{2}\sigma}\right), \tag{6.3}$$

where

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \tag{6.4}$$

It is obvious that when k is bigger than y_n, p_n^k is bigger than 0.5, and vice versa. Different from other ranking-base methods, the way of obtaining the prediction age in soft-ranking can be denoted as:

$$\hat{y}_n = \arg\min_k abs(\hat{p}_n^{k0} - \hat{p}_n^{k1}), \tag{6.5}$$

where $abs(\cdot)$ returns the absolute value and the obtained k is used as the final prediction. However, although these ranking-based approaches can capture the sequential information of the aging process to some extent and enhance the age estimation accuracy, they suffer from a lack of scalability. Therefore, label distribution learning-based methods are proposed.

6.2.4 Label Distribution Learning-Based Methods

Several works [17, 26, 27] extend the classification-based approach and suggest the use of a learning framework in which the semantic similarity among age labels is taken into account during the training phase to address the aforementioned problem. These methods convert each age label into a vector, i.e., label distribution, in which each value expresses how similar a face image is to the corresponding age label. Specifically, the corresponding label distribution will be set as a typical Gaussian distribution with a mean of the ground-truth label. Assume the sample is denoted as (x, y) where x denotes the image and y denotes the corresponding age label. The k-th element of the label distribution is denoted as:

$$z^k = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(k-y)^2}{2\sigma^2}\right), \tag{6.6}$$

where σ is the standard deviation and is usually set to 1 or 2. The symbol $k \in [0, ..., K]$ indicates an age index and K is the maximum age. Generally, K is set to 100 for considering the ages from 0 to 100 years.

Given an input image x, the prediction distribution produced by deep neural networks is defined as p, where the k-th element p^k represents the probability of classifying the input image to age k. In the training stage, the Kullback–Leibler (KL) divergence is the common choice of the loss function in the label distribution learning (LDL) framework to measure the discrepancy between the model's output p and ground-truth label distribution z [26]. The formula can be represented as:

$$\ell_{kl}(z, p) = \sum_{k=0}^{K} z^k log \frac{z^k}{p^k}. \tag{6.7}$$

To obtain a specific age prediction, an expectation refinement is employed according to the work [99]. Specifically, the predicted age is denoted as: $\hat{y} = \sum_{k=0}^{K} k \cdot p^k$. The expectation refinement takes the expectation of the output distribution as the final predicted age, which could enhance the stability and reliability of the prediction. A follow-up work [27] propose to regularize the KL divergence with an ℓ_1 distance to further adopted to narrow the gap between the predicted age $\hat{y}$ and the ground-truth label y, which is defined as follows:

$$\ell_{er}(y, \hat{y}) = |y - \hat{y}|, \tag{6.8}$$

where $| \cdot |$ denotes ℓ_1 distance. In the end, the whole loss of training an age estimator with a label distribution and a ℓ_1 regularization can be denoted as:

$$\ell_{all} = \ell_{kl}(z, p) + \ell_{er}(y, \hat{y}). \tag{6.9}$$

Based on this, Akbari et al. [3] proposed the use of a flatter loss function in label distribution learning, which improves the generalization and robustness of the age estimation task in cross-dataset scenarios. However, most previous approaches strongly assume that each category has enough instances to outline its data distribution, which does not correspond to the reality of the dataset. This assumption leads to models that tend to bias the predictions of age categories in the presence of sparse training samples. To mitigate the impact of data imbalance, [17] proposed a progressive margin loss to improve the discrepancy between the intra-class variance and the inter-class variance. The core of PML contains two components: the ordinal margin for exploration of the correlated relationship of the age labels and the variational margin for minimizing the adverse effect on the tailed classes caused by the head classes.

6.3 Gender and Ethnicity Recognition

Gender and ethnicity recognition is an old research subject, which has been studied for many years. In the 1990s, Brunelli and Poggio extracted a set of geometrical features automatically from frontal faces and trained two competing HyperBF networks to perform gender recognition [91]. Owing to the limited number of face images (20 females, 20 males), the proposed HypeerBF networks only obtained an average accuracy of 79% at that time. Then, Erno and Roope [78] evaluated gender classification with automatically detected and aligned faces, showing that the best classification results were achieved with a SVM classifier. Besides, they provided a comprehensive study and comparisons under the same experimental setups using state-of-the-art gender classification methods [79]. The common problem of the above studies is that face images were acquired under controlled conditions (e.g., FERET dataset [90]). However, in real applications, gender classification needs to be performed in unconstrained scenarios, where significant appearance variations make the problem harder (i.e., facial expressions, illumination changes, head pose variations, makeup, etc.).

For the above reasons, gender recognition in the wild is much more challenging compared with constrained environments, and some works [29, 58, 59, 100] attempted to address this problem. Shakhnarovich et al. [100] collected over 3,500 face images from the web and obtained an accuracy of 79% using Harr-like features and Adaboost. Then, Gao et al. [29] proposed the probabilistic boosting tree with Harr-like features for gender recognition on 10,100 real-life faces, and achieved 95.5% accuracy. Later, Kumar et al. [58, 59] investigated face verification using many binary attribute classifiers which included gender information. Zhang et al. [128] proposed a CNN method for facial gender and smile classification in an unconstrained environment. Compared with the previous works, the proposed method [128] considered more attributes (namely age, gender, ethnicity, and facial expression) in a unified framework. Likewise, the proposed method [128] is robust to the presence of unlabeled observations. Inspired by VGGNet [102], Dhomne et al. [18] designed successive convolutional blocks to automatically extract features from faces and perform gender classification. Then, Zhou et al. [130] suggested considering face recognition and gender classification in a unified framework. However, their proposed approach is indeed two decoupled networks, lacking a mechanism to promote each other. Although the above methods take advantage of deep learning, they all use pre-trained CNN without considering the difference between domains, which cannot entirely excavate the capacity of CNN. Consequently, Mittal et al. [81] introduced transfer learning by utilizing two-stage training to progressively modify the weights of the backbone and classification head. The results prove that their approach [81] outperforms other previous methods on four benchmarks.

Similar to gender recognition, traditional facial ethnicity classification methods mainly include feature extraction and recognition. In the work of Hosoi et al. [49], Gabor wavelet transformation and retina sampling were combined to extract key facial features, and then SVM was used for ethnicity classification. Yang et al. [121] presented a method that used local binary pattern histogram (LBPH) features in the ordinary ethnicity classification problem. Then, Guo and Mu [36] studied large-scale ethnicity estimation under variations of age and gender on MORPH II dataset [94]. The above studies have achieved impressive accuracy in ethnicity classification. However, they are still hard to use in real applications. More recently, due to the progress in deep learning techniques, some researchers [112] applied deep CNNs to classify ethnicity from facial images and achieved promising performance. Acien et al. [1] employed VGGFace [89] and Resnet50 [46] to perform race recognition and reached an accuracy over 95%. Further on, [82] studied gender and ethnicity recognition in night-time images from diverse viewing distances. Similar to [112], they proposed a CNN architecture for the task and attained high accuracy. Meanwhile, [1, 82] both proved that using soft biometric traits like gender, ethnicity, and skin color is conducive to face recognition.

6.4 Multi-task Learning for Facial Attribute Estimation

Apart from the above studies, which focus on estimating a single facial attribute, there are some works formulating the multi-task learning (MTL) framework for facial attribute analysis. In a departure from single-task learning (STL), MTL-based methods aim to learn one model to predict multiple face attributes simultaneously, which was first proposed in the 1990s [16]. The authors used part of the knowledge learned from using one task to facilitate the learning of the other related work. Since then, a series of approaches have been successively proposed in the literature.

Recently, biologically inspired deep learning has shown great potential for learning a compact and discriminative feature, which perfectly matches the needs of MTL. In [122], Yi et al. used CNN to extract features from multi-scale face patches, and then the features were concatenated together as the shared feature for different facial attributes (age, gender, and race). To deal with the highly skewed class distribution in the large-scale attribute datasets, Huang et al. [50] suggest maintaining both inter-cluster and inter-class margins to reduce the class imbalance in the neighboring class. While in [43], Hand et al. proposed an alternative idea to cope with the data imbalance problem in multi-task scenarios, which selectively learning with domain adaptive batch resample methods for multi-label attribute prediction. In [20], A Multi-Task Restricted Boltzmann Machine (MT-RBM) was proposed to learn the shared features for multiple binary attribute classification. In [129], the authors proposed to use features extracted by FaceNet and VGG-16 as input to each SVM classifier for each attribute, thus estimating 40 facial attributes in the CelebA and LFWA databases. In [41], Han et al. proposed to learn both attribute correlation and attribute heterogeneity in a CNN, which allows shared feature learning for all the attributes and category-specific feature learning for heterogeneous attributes. Considering the fact that many facial attributes describe local properties, Kalayeh et al. [53] suggested using semantic segmentation to guide the attention of the attribute prediction to the regions where different attributes naturally show up. In [77], a similar idea of introducing segmentation to assist facial attributes was applied for detecting the attributes in partially occluded faces. By utilizing GAN, He et al. [45] generated abstraction images as complementary features and used them for facial part localization. To fully exploit the high correlation between face attributes and label information, [8] proposed to construct a Partially Shared Multi-task Convolutional Neural Network for multi-task facial attribute prediction (Table 6.2).

In summary, multi-task learning for facial attribute estimation takes the entire facial image as input and focuses on exploring the relationships between different attributes, thus improving the recognition performance of a single attribute. Existing approaches model the correlation between different attributes by designing various elaborate network structures. The key to this idea is to learn shared features at the low level and attribute-specific features at the high level. Because of this, FAE methods usually face two main problems: the first is how to assign different layers to learn corresponding attribute features with different characteristics, and the second is how to customize the network to learn more discriminative features by mining the correlations between attributes. It is evident from contemporary research that manual attribute grouping has become a common scheme in FAE. We hope that in future work, automatic attribute grouping strategies will attract more attention, which can adaptively learn the appropriate grouping classification criteria and adjust to the performance of the model during training (Table 6.3).

6.5 Face Attribute Editing

Face attribute manipulation is also called facial attribute editing, which includes modifying single or multiple attributes of a face image while the other attributes of the image are not affected, i.e., to generate a new face with desired attributes while preserving other details. With the development of deep convolutional neural networks and generative models, this technology has been widely used in many fields, including data augmentation, facial beauty, and cartoon animated faces. Face attribute manipulation technology is a double-edged sword. On the one hand, it is widely deployed in FaceApp products. Customers can use it to try a wider range of content changes, such as makeup, wearing glasses, changing hair styles, etc.; On the other hand, it has become a sharp weapon for lawbreakers to steal users' privacy and property.

Facial attribute editing is shown in Fig. 6.6a. Given a face, its purpose is to change gender, wear glasses, change hair color, remove hair, add a beard, etc., while keeping other facial details unchanged. This process is usually completed in Fig. 6.6b by the trained face attribute editing model and the specified attribute vector parameters. According to [10], the facial attributes are generally classified from the following aspects: local or global attributes, such as glasses or age; categorical or binary attributes, such as hair color or gender; continuous or discrete, such as smile or age; identity-relevant or identity-irrelevant, such as skin color or expression. No matter what classification, previous work [85] roughly classifies face attributes into seven categories: Face Parts, Global Attributes, Makeup, Expression, Pose,

Table 6.2 .

Publication	Approach	Face database #images(training; testing)	Accuracy
Yi et al. [122]	Multi-scale CNN 3-layer network multi-label loss	Morph II (public) (10,530; 44,602)	**Morph II** Age: 3.63 years MAE; Gender: 98.0%; Race: 99.1% (Black vs. White)
Huang et al. [50]	CNN features by DeepID2 large margin local embedding; kNN classifier	CelebA (public) (180K, 20K)	**CelebA** 84% (Avg. of 40 attributes)
Ehrlich et al. [20]	Multi-task Restricted Boltzmann Machines with PCA and keypoint features; Multi-task classifier	CelebA (public) (180K, 20K) ChaLearn FotW (6,171; 3,087)	**CelebA** 87% (Avg. of 40 attributes) **FotW** Smile and gender: 76.3% (Avg.)
Hand et al. [44]	Multi-task CNN features 3 Conv. and 2 FC layers); Joint regression of multiple binary attributes	CelebA (public) (180K, 20K) LFWA (public) (6,263; 6,970)	**CelebA** 91% (Avg. of 40 attributes) **LFWA** 86% (Avg. of 40 attributes)
Zhong et al. [129]	Off-the-shelf CNN features by FaceNet and VGG-16 OneSVM classifier per attribute	CelebA (public) (180K, 20K) LFWA (public) (6,263; 6,970)	**CelebA** 86.6% (Avg. of 40 attributes) **LFWA** 84.7% (Avg. of 40 attributes)
Han et al. [41]	Deep multi-task feature learning shared feature learning category-specific feature learning Joint estimation of multiple heterogeneous attributes	Morph II (public) (62,566; 15,641) LFW+ (proposed) (12,559; 3,140) CelebA (public) (180K, 20K) LFWA (public) (6,263; 6,970) CLAP2015 (public) (2,476; 1,136) ChaLearn FotW (6,171; 3,087)	**Morph II** Age: 85.3% CS(5), 3.0 years MAE; Gender: 98% Race: 96.6% (Black, White, Other) **LFW+** Age: 75% CS(5), 4.5 years MAE; Gender: 96.7%; Race: 94.9% **CelebA** 92.1% (Avg. of 40 attributes); **LFWA** 86% (Avg. of 40 attributes) **CLAP2015** Age: 5.2 years MAE **FotW** Accessory: 94.0% (Avg. of 7 attributes); Smile and gender: 86.1% (Avg.)

(continued)

Table 6.2 (continued)

Publication	Approach	Face database #images(training; testing)	Accuracy
Kalayeh et al. [53]	Multi-task CNN features; Semantic Segmentation based Pooling and Gating	CelebA (public) (180K, 20K)	**CelebA** 91.8% (Avg. of 40 attributes)
Mahbub et al. [77]	Segmentwise, Partial,Localized Inference in Training Facial Attribute Classification Ensembles Network	CelebA (public) (180K, 20K)	**CelebA** 90.61% (Avg. of 40 attributes)
He et al. [45]	GAN and a dual-path facial attribute recognition network	CelebA (public) (180K, 20K) LFWA (public) (6,263; 6,970)	**CelebA** 91.81% (Avg. of 40 attributes) **LFWA** 85.2% (Avg. of 40 attributes)
Hand et al. [43]	Domain adaptive batch resample AttCNN Network	CelebA (public) (180K, 20K) LFWA (public) (6,263; 6,970)	**CelebA** 85.05% (Avg. of 40 attributes) **LFWA** 73.03% (Avg. of 40 attributes)
Cao et al. [8]	Partially Shared Multi-task Convolutional Neural Network	CelebA (public) (180K, 20K) LFWA (public) (6,263; 6,970)	**CelebA** 92.98% (Avg. of 40 attributes) **LFWA** 87.36% (Avg. of 40 attributes)

Table 6.3 Description and summary of face attribute editing methods based on generation countermeasure network

Year	Con./Jour.	Name	Method	Key points of the method
2018	ECCV	ELEGANT [116]	Manipulation in Latent Space	All the attributes are encoded in a disentangled manner in the latent space
2018	ACM	SG-GAN [125]	Manipulation in Latent Space	Adopts a one-input multi-output architecture to reduce the dependence on training labels
2020	ECCV	LEED [114]	Manipulation in Latent Space	Its core idea is to decouple the identity and expression attributes in the face to the expression manifold
2021	CVPR	ISF-GAN [73]	Manipulation in Latent Space	ISF encodes the style code w as w^* in the latent space
2021	Graphics and Visual Computing	FaceShapeGene [117]	Manipulation in Latent Space	It encodes the shape information of the face part into separate chunks in the latent space, which can realize the free combination of part-wise latent chunks of different faces
2021	CVPR	L2M-GAN [119]	Manipulation in Latent Space	It is trained end-to-end with GAN and for editing both local and global attributes is latent space
2017	TIP	AttGAN [48]	Conditional Decoder	It imposes an attribute classification constraint to accurately achieve "change what you want", and a reconstruction learning to dimensionally achieve "only change what you want"
2019	CVPR	STGAN [72]	Conditional Decoder	It selectively takes the difference between the target and source attribute vectors as input
2019	ICCV	RelGAN [113]	Conditional Decoder	It contains a G conditions on an input image and relative attributes and performs facial attribute transfer
2020	ECCV	Ling et al. [70]	Conditional Decoder	Its generator (G) conditions on an input image and relative action units to generate an image with target expression

(continued)

Table 6.3 (continued)

Year	Con./Jour.	Name	Method	Key points of the method
2020	ICME	EGGAN [104]	Conditional Decoder	It utilizes latent codes and continuous expression labels as input to generate fine-grained expressions
2021	CVPR	PIRenderer [93]	Conditional Decoder	It can fine-grained mimic accurate movements according to intuitive modifications which are semantically meaningful and fully disentangled parameters
2017	ICCV	CycleGAN [131]	Image-to-image translation	It translates an image from a source domain to a target domain in the absence of paired training examples
2018	CVPR	StarGAN [14]	Image-to-image translation	It allows simultaneous training of multiple datasets with different domains within a single model
2019	CVPR	StyleGAN [54]	Image-to-image translation	It is an alternative architecture for GANs, borrowing from style transfer
2020	ECCV	StyleGANv2 [123]	Image-to-image translation	It unites unconditional image generation and paired image-to-image GANs to distill a particular image manipulation in latent code
2020	CVPR	StarGANv2 [15]	Image-to-image translation	It further increases the diversity of images translated to the target domain, and supports multiple target domains
2021	CVPR	pSp [96]	Image-to-image translation	It introduces a new encoder architecture that can directly generates a series of style vectors, forming the extended $W+$ latent space
2021	CVPR	Lipstick [84]	Image-to-image translation	It extends the definition of makeup, and combines color-matching and pattern addition
2022	CVPR	TransEditor [118]	Image-to-image translation	It highlights the importance of interaction in a dual-space GAN for more controllable editing

(continued)

Table 6.3 (continued)

Year	Con./Jour.	Name	Method	Key points of the method
2018	ECCV	GANimation [92]	Mask-guide	It contains Action Units (AU) annotations, which generates photo-realistic conditioned color masks
2019	CVPR	FReeNet [126]	Landmarks-guided	It adopts a Unified Landmark Converter (ULC) to convert expression in a latent land mark space
2020	CVPR	EF-GAN [115]	Mask-guide	It aims to remove artifacts and blurs by performing progressive facial expression editing with local expression focuses
2020	CVPR	MaskGAN [62]	Mask-guide	In designs the semantic masks serve as a suitable intermediate representation for flexible face manipulation
2021	TIFS	A3GAN [74]	Mask-guide	It introduces a face parsing maps to help the generator distinguish image regions
2021	WACV	FACEGAN [107]	Landmarks-guided	It contains an Action Unit (AU) to transfer the facial motion from the driving face, which are independent of the facial structure preventing the identity leak

Accessories, and Image Attributes. In this section, we review the mainstream face attribute editing methods, which can be divided into encoder-decoder structures, image-to-image translation, and photo-guided architectures. In the following, we will introduce facial editing methods from these three aspects.

6.5.1 Encoder-Decoder Structures

In the encoder-decoder structure, the input image is first mapped into a latent space by the encoder; then, the original attributes are modified on the latent space with a manipulation method (such as in Fig. 6.7a) or a conditional decoder (such as in Fig. 6.7b); Finally, the desired attributes are generated from each point of the latent space based on the decoder.

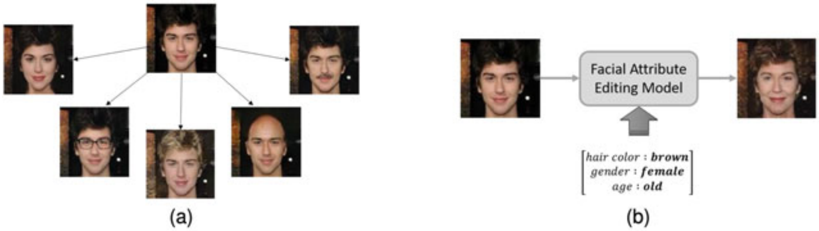
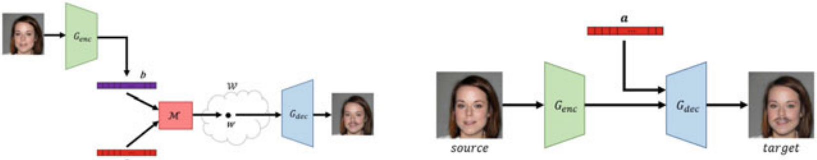


Fig. 6.6 Facial attribute editing. **a** Examples of attribute editings applied to a face image. **b** The overall structure of a face attribute editing model. This figure is from [85]



(a) Manipulation in Latent Space. (b) Manipulation with Conditional Decoder

Fig. 6.7 A general framework based on encoder-decoder structures. This figure is from [85]

Manipulation in Latent Space. There are three challenging difficulties in face attribute editing: (1) Generating images by the given exemplars; (2) Editing multiple face attributes at the same time; and (3) Generating high-quality edited images. ELEGANT [116] can receive **two images with opposite attributes** as input, and implement attribute editing through given **exemplars**; encode all attributes to the latent space in a disentangled way to realize the function of editing multiple attributes at the same time; employ residual image and multi-scale discriminator to improve the resolution and quality of the generated image. The method of editing attributes in latent space usually first uses a pair of images with opposite attributes as network input, and then maps attribute-related information to the predefined region of the latent space. Finally, the transfer of face attributes from exemplar to input sample is completed by exchanging their attribute- related feature regions. Instead of using input image restrictions with opposite attributes, MulGAN [40] directly uses **attribute label constraints** in the predefined latent space. It contains three main components: a generator, a discriminator, and an attribute classifier. In which the attribute classification loss ensures that the model extracts attribute-related information into predefined attribute areas. This allows multiple attributes to be transferred at the same time.

Nitzan et al. [86] decouple identity attributes via latent space mapping. Their approach aims to represent data in a disentangled manner using available pre-trained models with minimal supervision. Specifically, the identity image I_{id} and attribute image I_{attr} are first encoded by E_{id} and E_{attr}, respectively; then the concatenated features are mapped from the mapping network M to W; and finally, the edited image is generated by a pre-trained generator G. This process is completed under the constraints of a series of losses, including adversarial loss L_{adv}, identity loss L_{id}, reconstruction loss L_{rec}, and landmark loss L_{lnd}. The advantages of this method are as follows: (1) The disentangled data representation can be combined with the advanced generator, such as StyleGAN; (2) High quality edited images can be generated by using rich and expressive latent space; (3) The available pre-trained models can reduce the training burden. However, there are several shortcomings: (1) The domain-invariant part cannot be well preserved; (2) The multiple domains cannot be well handled; (3) The multi-modal translations cannot be performed at the same time. Also, using available pre-trained unconditional generators, ISF-GAN [73] achieves multi-modal and multi-domain image-to-image translation. In detail, given the attribute label d and a randomly sampled noise z, the Implicit Style Function (ISF) first encodes the style code w as w^*. In this way, the image generated by generator G can be specified by attribute label d, while other attributes remain unchanged. Similar to [86], G is a pre-trained model (such as StyleGAN). At the same time, a new discriminator is introduced to distinguish the real and fake samples and classifies the image attributes. Previous methods usually focus on editing predefined facial attributes, while ignoring the control of the geometric shape of the facial part. This is due to the previous method using discrete attribute labels instead of continuous geometric parameters.

In order to alleviate the dependence on labeled data, SG-GAN [125] adopts a one-input multi-output architecture, and sparsely group learning strategy reduces the dependence on training labels. Further, LEED [114] edits the facial expressions of the front and profile faces in a disentangled way and does not rely on expression labels at all. Specifically, its core idea is to decouple the identity and expression attributes in the face to the expression manifold, where the neutral face captures the identity attributes, and the displacement between the neutral image and expressive image captures the expression attributes. FaceShapeGene [117] encodes the shape information of the face part into separate chunks in the latent space, which can realize the free combination of part-wise latent chunks of different faces to transfer the specified facial part shape. Due to the well-defined local support regions of attributes, the existing models are better at handling a local attribute than a global one. At the same time, a fixed pre-trained GAN cannot be trained end-to-end with the attribute editing network. L2M-GAN [119] is a latent space factorization model, which is trained end-to-end with GAN and is effective for editing both local and global attributes.

There are lots of other models adopting a similar idea in latent space to manipulate facial attributes. PuppetGAN [109] is a cross-domain image manipulation model; RSGAN [83] is a system integrating face swapping, attribute editing and random face parts synthesis; LADN [33] is an adversarial disentangling network for facial makeup and de-makeup; S^{2}GAN [47] is an effective method to simulate the natural aging process. Other works are to find a direction vector in latent space, which applies the desired attribute changes to the corresponding image, such as DFI [108], Facelet-Bank [13], StyleRig [105], and InterFaceGAN [101].

Manipulation with Conditional Decoder The early methods try to establish a latent representation independent of attributes, and then further edit the attributes. However, this excessive attribute independent constraint not only limits the capacity of the latent representation, but also leads to the generation of over-smooth and distorted images. Instead of imposing constraints on the latent representation, AttGAN [48] imposes an attribute classification constraint to accurately achieve "change what you want" by just ensuring the correct change of desired attributes, and a reconstruction learning to dimensionally achieve "only change what you want" by preserving attribute-excluding details. Considering that a specific editing task is definitely only related to the changed attributes, not all the target attributes, STGAN [72] selectively takes the difference between the target and source attribute vectors as input.

The previous methods edit face images under discrete emotional labels or absolute conditions, while the editing effect is poor for changing condition-irrelevant regions or fine-grained editing. Ling et al. [70] replace continuous absolute conditions with relative conditions, such as relative action units. Then, a generator is built based on U-Net to generate high-quality attribute editing images through multi-scale fusion mechanism. Especially different from the latent editing method, generator (G) conditions on an input image and relative action units to generate images with target expression. Further, RelGAN [113] uses relative attributes to describe the desired change on selected attributes, which can modify interested attributes in a fine-grained way, while leaving other attributes unchanged. RelGAN [113] consists of a single generator G and three discriminators D. In which G conditions on an input image and relative attributes, and performs facial attribute transfer or interpolation. PIRenderer [93] can fine-grained mimic accurate movements according to intuitive modifications which are semantically meaningful and fully disentangled parameters. EGGAN [104] utilizes latent codes and continuous expression labels as input to generate fine-grained expressions. When given the source latent code and the target expression label, EGGAN [104] generates a new image with the target expression in a conditional manner.

6.5.2 Image-to-Image Translation

Some work completes the attribute editing task in the way of image domain translation, where the specific attributes of the source image are edited as the attributes of the target image. The image-to-image translation methods regard different values of specific attributes as different domains (modalities) and do not require the condition (attribute) vector as an input of the generator.

CycleGAN [131] aims to solve the problem of paired training data will not be available, which translates an image from a source domain to a target domain in the absence of paired training examples. This process is completed by an inverse mapping and a cycle consistency loss. StarGAN [14] aims to solve the problem of poor scalability of existing methods when dealing with more than two domains, which allows simultaneous training of multiple datasets with different domains within a single model. StarGAN generated images of higher visual quality, owing to the generalization capability behind the multi-task learning setting and the utilization of all available labels from multiple datasets with mask vector setting. On this basis, StarGANv2 [15] further increases the diversity of images translated to the target domain, and supports multiple target domains.

StyleGAN [54, 55] cannot obtain plausible editing results with high controllability, especially for complex attributes. StyleGANv2 [123] unites unconditional image generation and paired image-to-image GANs to distill a particular image manipulation in latent code, which results in both fast inference and impressive quality than StyleGAN [54]. Based on the Transformer framework, TransEditor [118] highlights the importance of interaction in a dual-space GAN for more controllable editing. Previous methods suffer from three limitations: (1) incapability of generating image by exemplars; (2) being unable to transfer multiple face attributes simultaneously; (3) low quality of generated images, such as low resolution or artifacts. Pixel2style2pixel (pSp) [96] introduce a new encoder architecture that can directly generate a series of style vectors, forming the extended $W+$ latent space. In which the styles are fed into a pre-trained StyleGAN [54] generator.

In addition, there are other image translation frameworks with similar ideas to implement attribute editing. ByeGlassesGAN [64] is a multi-task framework to automatically detect eyeglass areas and remove them from a face image. DosGAN [69] utilizes domain information as explicit supervision for unconditional or conditional image-to-image translation. Lipstick [84] extends the definition of makeup and combines color matching and pattern addition through color transfer branch and pattern transfer branch.

6.5.3 Mask/Landmarks-Guided Architectures

In recent years, face attribute editing methods based on mask and face landmarks have attracted more and more researchers' attention. GANimation [92] is a GAN conditioning scheme based on Action Units (AU) annotations, which generates photo-realistic conditioned color masks. A3GAN [74] is an Attribute-Aware Attentive face aging model to improve the quality of low-level image content and naturalize high-level semantic information. In particular, a face parsing maps is designed to help the generator distinguish image regions. Cascade EF-GAN [115] aims to remove artifacts and blurs around expression-intensive regions by performing progressive facial expression editing with local expression focuses. MaskGAN [62] aims to improve the degree of freedom for users to interactively manipulate images. In which the semantic masks serve as a suitable intermediate representation for flexible face manipulation.

Recent works have demonstrated high-quality results by combining the facial landmark. FReeNet [126] is a multi-identity face reenactment framework to edit facial expressions, which adopts a Unified Landmark Converter (ULC) to convert expression in a latent landmark space. The face reenactment is task of one person's identity is taken from the source image and the facial motion from the driving image. However, different identities will cause the driver facial structure to leak into the output, affecting the reenactment result. FACEGAN [107] contains an Action Unit (AU) to transfer the facial motion from the driving face, which is independent of the facial structure preventing the identity leak.

6.6 Recent Competitions

In recent years, academia and industry have cooperated closely and launched a series of competitions related to face attributes, which have made great contributions to the promotion of face attribute recognition, face attribute forgery detection, and other fields. Next, we will introduce some recent competitions.

Deepfake Detection Challenge@CVPR 2020[1] Deepfake is a technology that uses deep neural networks to digitally edit facial attributes. This facial attribute editing technique could have a significant impact on people's confirmation of the authenticity of online information. These content generation and attribute editing techniques may affect the quality of online information and the protection of human rights, especially given that deepfakes can be maliciously used for misinformation, manipulation, or misleading others.

The partnership of AWS, Facebook, Microsoft, AI Media Integrity Steering Committee, and academia came together to create the Deepfake Detection Challenge (DFDC) [19]. The goal of the challenge is to inspire researchers around the world to build innovative new technologies to help detect deepfake images and videos of faces (Fig. 6.8).

Challenge participants must submit their code to an online black box environment for testing. Open proposals are eligible for challenge prizes as long as they adhere to the terms of the open-source license. All submissions will be evaluated in the same manner. The results will be displayed on the leaderboard.

Guess The Age@CAIP 2021[2] In Guess the Age 2021, the goal of the contestants is to train a deep neural network for age estimation from face images, resulting in the best performance in terms of accuracy and regularity on a private test set. Contestants must adhere to two main restrictions: (1) only samples from the training set provided by the sponsor, namely the MIVIA Age dataset; (2) only a single neural network can be used, and the method of network ensemble cannot be used. The MIVIA dataset consists of 575,073 images of more than 9,000 identities, from different ages; in particular, they are drawn from the VGGFace2 [9] dataset and annotated with age through knowledge distillation techniques, making the dataset very heterogeneous in terms of face size, lighting conditions, facial pose, gender, and ethnicity. Each image of the dataset contains a cropped face. An example image is shown in Fig. 6.9.

[1] https://www.kaggle.com/c/deepfake-detection-challenge.

[2] https://gta2021.unisa.it/#instructions.

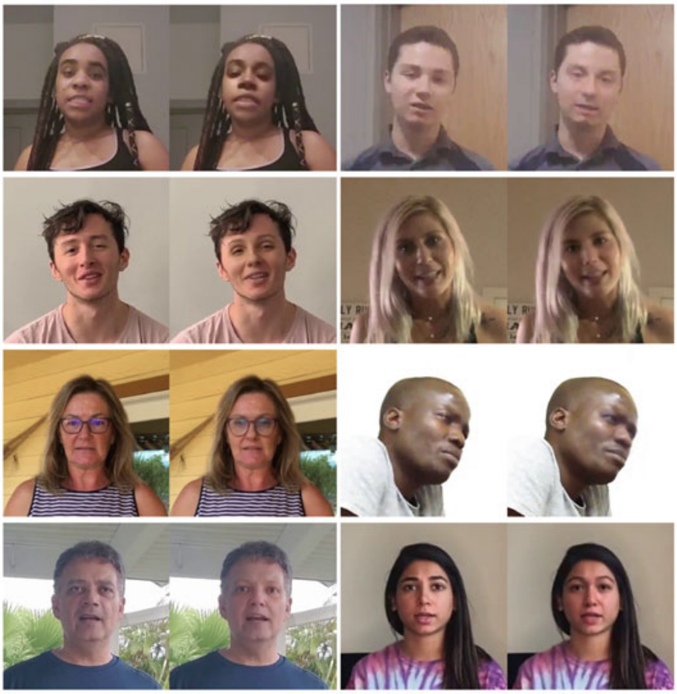


Fig. 6.8 Some example face swaps from the DFDC. The left is a real face image, and the right is a virtual generated fake image. Participants need to use an algorithm to accurately identify the fake image. Dataset [19]

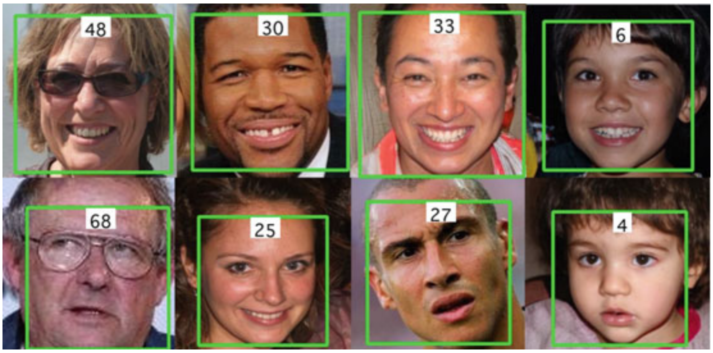


Fig. 6.9 Given a picture in the GTA competition, the contestant needs to submit a model to estimate the corresponding age

The competition restricts the use of the MIVIA Age Dataset and limits the competition to methods based on a single neural network. Participants are free to propose new DCNN architectures or define innovative training procedures for standard DCNN architectures.

Before the registration deadline, 20 teams from all over the world officially sent their applications: 9 teams from Europe, 5 teams from Asia, 4 teams from North America, 1 team from South America, and 1 team from Africa. All requests came from academia, except for one from a company. In the end, the top three teams are: BTWG [4], CIVALab [106], and GoF.

BTWG used EfficientNetV2-M as the backbone network, performed face recognition pre-training on MS-celeb-1M, and divided the learning process into two steps: feature representation learning and classifier learning. In the data preprocessing stage, the team performed data augmentation on the training set by applying RandAugment and represented each age label as a Gaussian distribution. Then, in the representation learning step, they achieve good performance by using a custom loss function that combines KL divergence (label distribution loss) and L1 loss for regularization. Finally, in the classification stage, the team fine-tunes the fully connected layers using only the balanced version of the training set by employing a modified MSE loss function.

Image Forgery Detection Challenge@CVPR 2022[3] Image forgery technology refers to a collection of image processing techniques based on deep learning, such as image editing, image synthesis, image generation, etc. In recent years, the misuse of image forgery techniques has attracted a great deal of public attention. In real-world face authentication scenarios, hackers use image forgery techniques to edit face attributes to attack digital identity accounts. Therefore, with the widespread application of face recognition today, image forgery detection is an important technology for the security of artificial intelligence systems.

Unlike existing published image forgery detection datasets downloaded directly from YouTube or produced only by a few popular image forgery software, the dataset named Multi-Forgery Detection Challenge (Multi-FDC) [19] covers more real Scene and extensive image forgery techniques such as face swapping, reenactment, attribute editing, full compositing, artificial PS, etc. This challenge aims to encourage researchers around the world to develop innovative and general techniques to protect a wider variety of real-world image forgery from simultaneous attacks.

This competition attracted 674 teams from all over the world with around 2000 valid submission counts. There are 106 teams on the final validation set leaderboard and 68 teams on the final test set leaderboard. The top 10 teams were invited to the final stage, and in stage 3, 6 teams actually participated in the final stage.

In this multi-forgery detection challenge, the sponsor formulates the problem of multi-forgery detection. During the challenge, some successful attempts can be seen, such as unreasonable data validity, simulation of unseen types of forgery, multiple models as inductive biases, etc. On the other hand, it can be noticed that the explicit modeling of unseen forgery types and the architectural design of the forgery detection task are more or less miss-

[3] https://tianchi.aliyun.com/competition/entrance/531954/introduction.

Table 6.4 Summary of face attribute datasets

Datasets	Images	Number of attributes	Binary	Fine-gained	Global	Local
CelebA [76]	202,599	40	✓		✓	✓
LFWA [51]	13,233	73	✓		✓	✓
MORPH II [95]	55,132	3		✓	✓	✓
Adience [21]	16,300	2		✓	✓	✓
PCSO [42]	181,545	3		✓	✓	✓
ChaLearn LAP 2016 [22]	5,713	3		✓	✓	✓
CACD [11]	163,446	1	✓		✓	
VGGFace2 [9]	3.31 M	2	✓		✓	

ing. From the competition, some promising future directions in the field of image forgery detection can also be seen. Large-scale and diverse datasets, robustness to unseen forgery types, and fast adaptation to certain forgery types are promising directions that can be further explored.

6.7 Datasets

In this chapter, we summarize the commonly used face attribute datasets in recent years, and mark the dataset images in Binary, Fine-grained, Global, Local, as shown in Table 6.4. In addition, the digital editing of face attributes has also become a research hotspot in recent years. Today, with the widespread popularity of face recognition, the digital editing technology of face attributes has brought great challenges to account security. Therefore, we also summarize some currently commonly used deepfake detection datasets as shown in Table 6.5.

Next, we will select some representative datasets according to the following categories: facial attribute analysis and deepfake detection.

6.7.1 Face Attribute Datesets

MORPH [95] is a longitudinal face database, which can be used for face modeling, photo-realistic animation, face recognition, etc. It provides a huge amount of publicly available longitudinal images. The dataset is divided into commercial and academic versions. The academic version includes 55,134 pictures of 13,000 people. The photo collection spans from 2003 to 2007. The age of the characters is 16–77 years old, and the average age is 33 years old (Fig. 6.10).

Table 6.5 Summary of deepfake detection datasets

Datasets	Real videos	Fake videos	Total videos	Year
UADFV [67]	49	49	98	2018
DeepfakeTIMIT [57]	320	320	640	2018
FF-DF [98]	1,000	1,000	2,000	2019
DFDC [19]	1,131	4,113	5,244	2019
Celeb-DF [68]	590	5,639	6,229	2020
DeepFake MNIST+ [52]	10000	10000	20000	2021
FakeAVCeleb [56]	500	19,500	20,000	2022

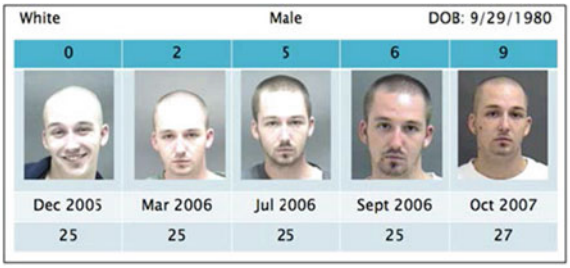


Fig. 6.10 Example images of MORPH dataset. The samples are all collected in the prison, and they are characterized by standardization and accurate labeling. However, the environment is relatively closed, and there is no rich diversity of data collected from wild

The MORPH II dataset is currently one of the most popular age estimation datasets. MORPH II is also a cross-temporal dataset that includes pictures of the same person at different ages. In addition to age, the MORPH II dataset also records other information about people, such as gender, race, whether they wear glasses, etc.

CACD (Cross-Age Celebrity Dataset) [11] is a large-scale dataset for face recognition and retrieval across ages, containing 163,446 images of 2,000 celebrities from the Internet.

It used celebrity names and years (2004–2013) to collect images from search engines as keywords, by simply subtracting the year of birth from the year the photo was taken, as an annotated age of the images, ranging from 14 to 62 years old (Fig. 6.11).

LFW (Labeled Faces in the Wild) [51] is a database organized by the Computer Vision Laboratory of the University of Massachusetts, Amherst, and is mainly used to study face recognition problems in unrestricted situations. The LFW database mainly collects images from the Internet, not closed labs. The face pictures provided are all derived from natural scenes in life, so the recognition difficulty will increase, especially due to the influence of multiple poses, lighting, expressions, age, occlusion, and other factors, even the photos of the same person are very different. And some photos may show more than one face. It contains a total of more than 13,233 face images, a total of 5,749 people, each image is identified with the name of the corresponding person, of which 1,680 people correspond to more than one image, that is, about 1,680 people contain more than two faces. For these

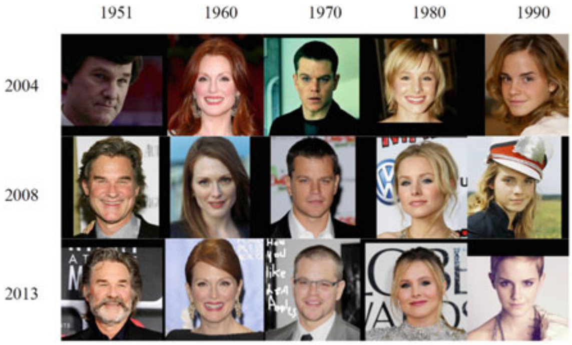


Fig. 6.11 Examples of face images across age in CACD. The top row numbers are the birth years of the celebrities, and left column numbers indicate the years in which the images were taken. Images in the same column are of the same celebrity. [11]

multi-face images, only the face of the center coordinate is selected as the target, and other areas are regarded as background interference. The size of each picture is 250 × 250, most of which are color images, but there are also a few black and white face pictures. In the LFW database, some faces have poor lighting conditions, extreme poses, severe occlusions, and low resolution, so it is difficult to identify them. And many groups are not well represented, such as the elderly and children over the age of 80 are very few, there are no babies, the proportion of women is low, and there are many ethnic samples with rare or no samples.

Celeb A (CelebFaces Attribute) [76] is opened by the Chinese University of Hong Kong, which contains 202,599 face pictures of 10,177 celebrities, including 118,165 female face pictures and 138,704 male face pictures. Each picture is marked with features, including the face bbox, 5 face feature point coordinates, and 40 attribute markers, such as whether to wear glasses, long or short hair, nose, lips, hair color, gender, and other characteristics. The images in this dataset cover large pose variations and background clutter. With a lot of diversity, a lot of volume and rich annotations (Fig. 6.12).

6.7.2 Deepfake Detection Datasets

DFDC's data volume is as high as 472GB, including 119,197 videos, each video is 10 s long, but the frame rate varies from 15 to 30 fps, and the resolution also varies from 320 × 240 to 3840 × 2160. Among the training videos, 19,197 videos are real footage of about 430 actors, and the remaining 100,000 videos are fake face videos generated from real videos. Fake face generation uses DeepFakes, GAN-based, and some non-learned methods, so that the dataset contains as many fake face videos as possible. The video in this dataset contains sound, which is not available in most datasets at present, but there is no annotation information for sound. At present, the SOTA score loss is around 0.42, the AUC in the domain is 91.33%, and there is still some room for improvement.

Celeb-DF is currently widely used, and the v2 version is an extension of the v1 version. The v1 version contains 408 raw videos captured from YouTube and 795 DeepFake videos

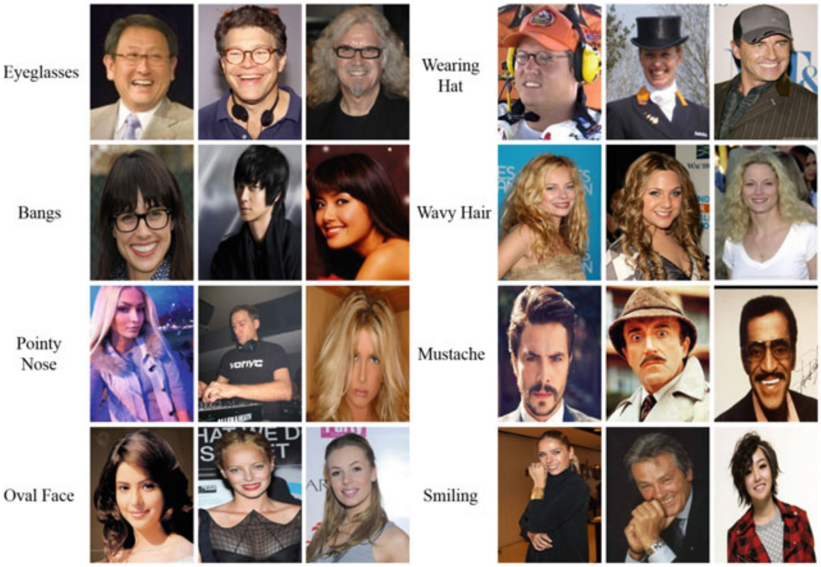


Fig. 6.12 Example images of CelebA dataset [76]. CelebA has rich face attributes, such as eyeglass, bangs, smiling, and mustache

generated from real videos. The v2 version expands the video to 590 and 563 respectively. According to literature research, the current SOTA has not exceeded 0.7.

6.8 Conclusions

In the past years, facial attribute analysis has made great progress, especially after the emergence of deep learning technologies, where many new technologies and databases have been proposed, and the performance has been greatly improved. In this survey, we have carried out a detailed review on facial attribute analysis. At first, we have reviewed the advances and challenges in facial attribute recognition, including facial age estimation, gender and ethnicity recognition, facial expression recognition, and multi-task learning for the recognition of multiple attributes. Then, we have reviewed the works in facial attribute manipulation, including manipulation methods and manipulation detection methods. We also have reviewed the great developments of facial attribute datasets and the competitions of facial attribute analysis in the past years. Although significant progress has been made in facial attribute analysis, there are still some challenges, which will be the future research directions:

- Accurate facial age estimation is still challenging although its accuracy has been largely improved in the past years. In the future, age estimation will be a long-term research issue, especially studying how to train a high accuracy model for age estimation.
- Multi-task learning is a trend for facial attribute recognition and will continue drawing attention in the coming years. The existing facial attribute datasets usually only contain a few attributes. How to collect a large attribute database with comprehensive attribute annotations will be a problem worthy of attention.
- Joint learning of digital (i.e., deepfake) and physical attacks (video-replay, 3D facial mask) detection is a new future direction, which is meaningful in real applications.

References

1. Acien, A., Morales, A., Vera-Rodriguez, R., Bartolome, I., Fierrez, J.: Measuring the gender and ethnicity bias in deep models for face recognition. In: Iberoamerican Congress on Pattern Recognition, pp. 584–593. Springer (2018)
2. Agustsson, E., Timofte, R., Van Gool, L.: Anchored regression networks applied to age estimation and super resolution. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1643–1652 (2017)
3. Akbari, A., Awais, M., Feng, Z., Farooq, A., Kittler, J.: Distribution cognisant loss for cross-database facial age estimation with sensitivity analysis. IEEE Trans. Pattern Anal. Mach. Intell. (2020)
4. Bao, Z., Tan, Z., Zhu, Y., Wan, J., Ma, X., Lei, Z., Guo, G.: Lae: long-tailed age estimation. In: International Conference on Computer Analysis of Images and Patterns, pp. 308–316. Springer (2021)
5. Berg, T., Belhumeur, P.: Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 955–962 (2013)
6. Cai, L., Huang, L., Liu, C.: Age estimation based on improved discriminative gaussian process latent variable model. Multimedia Tools Appl. **75**(19), 11977–11994 (2016)
7. Can Malli, R., Aygun, M., Kemal Ekenel, H.: Apparent age estimation using ensemble of deep learning models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 9–16 (2016)
8. Cao, J., Li, Y., Zhang, Z.: Partially shared multi-task convolutional neural network with local constraint for face attribute learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4290–4299 (2018)
9. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: a dataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 67–74. IEEE (2018)
10. Chandaliya, P.K., Kumar, V., Harjani, M., Nain, N.: SCDAE: ethnicity and gender alteration on CLF and UTKFace dataset. Comput. Vis. Image Process. (2020)
11. Chen, B.C., Chen, C.S., Hsu, W.H.: Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. IEEE Trans. Multimedia **17**(6), 804–815 (2015)
12. Chen, S., Zhang, C., Dong, M.: Deep age estimation: from classification to ranking. IEEE Trans. Multimedia **20**(8), 2209–2222 (2017)

13. Chen, Y.C., Lin, H., Shu, M., Li, R., Tao, X., Ye, Y., Shen, X., Jia, J.: Facelet-bank for fast portrait manipulation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)
14. Choi, Y., Choi, M., Kim, M., Ha, J.W., Choo, J.: Stargan: unified generative adversarial networks for multi-domain image-to-image translation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
15. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: diverse image synthesis for multiple domains. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
16. Cottrell, G., Metcalfe, J.: Empath: face, emotion, and gender recognition using holons. Adv. Neural Inf. Process. Syst. **3** (1990)
17. Deng, Z., Liu, H., Wang, Y., Wang, C., Yu, Z., Sun, X.: Pml: progressive margin loss for long-tailed age classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10503–10512 (2021)
18. Dhomne, A., Kumar, R., Bhan, V.: Gender recognition through face using deep learning. Procedia Comput. Sci. (2018)
19. Dolhansky, B., Howes, R., Pflaum, B., Baram, N., Ferrer, C.C.: The deepfake detection challenge (dfdc) preview dataset. arXiv preprint arXiv:1910.08854 (2019)
20. Ehrlich, M., Shields, T.J., Almaev, T., Amer, M.R.: Facial attributes classification using multi-task representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 47–55 (2016)
21. Eidinger, E., Enbar, R., Hassner, T.: Age and gender estimation of unfiltered faces. IEEE Trans. Inf. Forensics Secur. **9**(12), 2170–2179 (2014)
22. Escalera, S., Torres Torres, M., Martinez, B., Baró, X., Jair Escalante, H., Guyon, I., Tzimiropoulos, G., Corneou, C., Oliu, M., Ali Bagheri, M., et al.: Chalearn looking at people and faces of the world: face analysis workshop and challenge 2016. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8 (2016)
23. Feng, S., Lang, C., Feng, J., Wang, T., Luo, J.: Human facial age estimation by cost-sensitive label ranking and trace norm regularization. IEEE Trans. Multimedia **19**(1), 136–148 (2016)
24. Fu, Y., Huang, T.S.: Human age estimation with regression on discriminative aging manifold. IEEE Trans. Multimedia **10**(4), 578–584 (2008)
25. Fu, Y., Xu, Y., Huang, T.S.: Estimating human age by manifold analysis of face pictures and regression on aging features. In: 2007 IEEE International Conference on Multimedia and Expo, pp. 1383–1386. IEEE (2007)
26. Gao, B.B., Xing, C., Xie, C.W., Wu, J., Geng, X.: Deep label distribution learning with label ambiguity. IEEE Trans. Image Process. **26**(6), 2825–2838 (2017)
27. Gao, B.B., Zhou, H.Y., Wu, J., Geng, X.: Age estimation using expectation of label distribution learning. In: IJCAI (2018)
28. Gao, F., Ai, H.: Face age classification on consumer images with gabor feature and fuzzy lda method. In: International Conference on Biometrics, pp. 132–141. Springer (2009)
29. Gao, W., Ai, H.: Face gender classification on consumer images in a multiethnic environment. In: International Conference on Biometrics, pp. 169–178 (2009)
30. Geng, X., Zhou, Z.H., Smith-Miles, K.: Automatic age estimation based on facial aging patterns. IEEE Trans. Pattern Anal. Mach. Intell. **29**(12), 2234–2240 (2007)
31. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
32. Greco, A.: Guess the age 2021: age estimation from facial images with deep convolutional neural networks. In: Tsapatsoulis, N., Panayides, A., Theocharides, T., Lanitis, A., Pattichis, C., Vento, M. (eds.) Computer Analysis of Images and Patterns, pp. 265–274. Springer, Cham (2021)

33. Gu, Q., Wang, G., Chiu, M.T., Tai, Y.W., Tang, C.K.: Ladn: local adversarial disentangling network for facial makeup and de-makeup. In: ICCV (2019)
34. Gunay, A., Nabiyev, V.V.: Automatic age classification with lbp. In: 2008 23rd International Symposium on Computer and Information Sciences, pp. 1–4. IEEE (2008)
35. Guo, G., Fu, Y., Dyer, C.R., Huang, T.S.: Image-based human age estimation by manifold learning and locally adjusted robust regression. IEEE Trans. Image Process. **17**(7), 1178–1188 (2008)
36. Guo, G., Mu, G.: A study of large-scale ethnicity estimation with gender and age variations. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 79–86 (2010)
37. Guo, G., Mu, G.: Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 657–664 (2011)
38. Guo, G., Mu, G.: Joint estimation of age, gender and ethnicity: Cca vs. pls. In: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp. 1–6 (2013)
39. Guo, G., Mu, G., Fu, Y., Huang, T.S.: Human age estimation using bio-inspired features. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 112–119. IEEE (2009)
40. Guo, J., Qian, Z., Zhou, Z., Liu, Y.: Mulgan: facial attribute editing by exemplar. arXiv:2109.12492 (2019)
41. Han, H., Jain, A.K., Wang, F., Shan, S., Chen, X.: Heterogeneous face attribute estimation: a deep multi-task learning approach. IEEE Trans. Pattern Anal. Mach. Intell. **40**(11), 2597–2609 (2017)
42. Han, H., Otto, C., Liu, X., Jain, A.K.: Demographic estimation from face images: human vs. machine performance. IEEE Trans. Pattern Anal. Mach. Intell. **37**(6), 1148–1161 (2014)
43. Hand, E., Castillo, C., Chellappa, R.: Doing the best we can with what we have: multi-label balancing with selective learning for attribute prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
44. Hand, E.M., Chellappa, R.: Attributes for improved attributes: a multi-task network utilizing implicit and explicit relationships for facial attribute classification. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
45. He, K., Fu, Y., Zhang, W., Wang, C., Jiang, Y.G., Huang, F., Xue, X.: Harnessing synthesized abstraction images to improve facial attribute recognition. In: IJCAI, pp. 733–740 (2018)
46. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
47. He, Z., Kan, M., Shan, S., Chen, X.: S2gan: share aging factors across ages and share aging trends among individuals. In: ICCV (2020)
48. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: Attgan: facial attribute editing by only changing what you want (2017)
49. Hosoi, S., Takikawa, E., Kawade, M.: Ethnicity estimation with facial images. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings, pp. 195–200 (2004)
50. Huang, C., Li, Y., Loy, C.C., Tang, X.: Learning deep representation for imbalanced classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5375–5384 (2016)
51. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: Workshop on Faces in ‘Real-Life’ Images: Detection, Alignment, and Recognition (2008)

52. Huang, J., Wang, X., Du, B., Du, P., Xu, C.: Deepfake mnist+: a deepfake facial animation dataset. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1973–1982 (2021)
53. Kalayeh, M.M., Gong, B., Shah, M.: Improving facial attribute prediction using semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6942–6950 (2017)
54. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
55. Karras, T., Laine, S., Aittala, M., Hellsten, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
56. Khalid, H., Tariq, S., Kim, M., Woo, S.S.: Fakeavceleb: a novel audio-video multimodal deepfake dataset. arXiv preprint arXiv:2108.05080 (2021)
57. Korshunov, P., Marcel, S.: Deepfakes: a new threat to face recognition? assessment and detection. arXiv preprint arXiv:1812.08685 (2018)
58. Kumar, N., Belhumeur, P., Nayar, S.: Facetracer: a search engine for large collections of images with faces. In: European Conference on Computer Vision, pp. 340–353 (2008)
59. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 365–372 (2009)
60. Kwon, Y.H., da Vitoria Lobo, N.: Age classification from facial images. Comput. Vis. Image Underst. **74**(1), 1–21 (1999)
61. Lanitis, A., Taylor, C.J., Cootes, T.F.: Toward automatic simulation of aging effects on face images. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 442–455 (2002)
62. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: towards diverse and interactive facial image manipulation. CVPR (2020)
63. Lee, R.S., Liu, J.N.: An integrated elastic contour fitting and attribute graph matching model for automatic face coding and recognition. In: Knowledge-Based Intelligent Information Engineering Systems, 1999. Third International Conference, pp. 292–295 (1999)
64. Lee, Y.H., Lai, S.H.: Byeglassesgan: identity preserving eyeglasses removal for face images (2020)
65. Levi, G., Hassner, T.: Age and gender classification using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 34–42 (2015)
66. Li, K., Xing, J., Hu, W., Maybank, S.J.: D2c: deep cumulatively and comparatively learning for human age estimation. Pattern Recogn. **66**, 95–105 (2017)
67. Li, Y., Chang, M.C., Lyu, S.: In ictu oculi: exposing ai created fake videos by detecting eye blinking. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7 (2018). https://doi.org/10.1109/WIFS.2018.8630787
68. Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S.: Celeb-df: a large-scale challenging dataset for deepfake forensics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3207–3216 (2020)
69. Lin, J., Chen, Z., Xia, Y., Liu, S., Qin, T., Luo, J.: Exploring explicit domain supervision for latent space disentanglement in unpaired image-to-image translation. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1–1 (2019)
70. Ling, J., Xue, H., Song, L., Yang, S., Xie, R., Gu, X.: Toward fine-grained facial expression manipulation. In: European Conference on Computer Vision, pp. 37–53. Springer (2020)

71. Liu, H., Lu, J., Feng, J., Zhou, J.: Ordinal deep feature learning for facial age estimation. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), pp. 157–164. IEEE (2017)
72. Liu, M., Ding, Y., Xia, M., Liu, X., Ding, E., Zuo, W., Wen, S.: Stgan: a unified selective transfer network for arbitrary image attribute editing (2019)
73. Liu, Y., Chen, Y., Bao, L., Sebe, N., Lepri, B., Nadai, M.D.: Isf-gan: an implicit style function for high-resolution image-to-image translation. arXiv:2109.12492 (2021)
74. Liu, Y., Li, Q., Sun, Z., Tan, T.: A3gan: an attribute-aware attentive generative adversarial network for face aging. IEEE Trans. Inf. Forensics Secur. **PP**(99), 1–1 (2021)
75. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3730–3738 (2015)
76. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3730–3738 (2015)
77. Mahbub, U., Sarkar, S., Chellappa, R.: Segment-based methods for facial attribute detection from partial faces. IEEE Trans. Affect. Comput. **11**(4), 601–613 (2018)
78. Makinen, E., Raisamo, R.: Evaluation of gender classification methods with automatically detected and aligned faces. IEEE Trans. Pattern Anal. Mach. Intell. **30**(3), 541–547 (2008)
79. Mäkinen, E., Raisamo, R.: An experimental comparison of gender classification methods. Pattern Recogn. Lett. **29**(10), 1544–1556 (2008)
80. Manyam, O.K., Kumar, N., Belhumeur, P., Kriegman, D.: Two faces are better than one: face recognition in group photographs. In: 2011 International Joint Conference on Biometrics (IJCB), pp. 1–8 (2011)
81. Mittal, S., Mittal, S.: Gender recognition from facial images using convolutional neural network. In: IEEE International Conference on Image Information Processing (2019)
82. Narang, N., Bourlai, T.: Gender and ethnicity classification using deep learning in heterogeneous face recognition. In: 2016 International Conference on Biometrics (ICB), pp. 1–8. IEEE (2016)
83. Natsume, R., Yatagawa, T., Morishima, S.: Rsgan: face swapping and editing using face and hair representation in latent spaces. In: Acm Siggraph Posters (2018)
84. Nguyen, T., Tran, A., Hoai, M.: Lipstick ain't enough: beyond color matching for in-the-wild makeup transfer (2021)
85. Nickabadi, A., Fard, M.S., Farid, N.M., Mohammadbagheri, N.: A comprehensive survey on semantic facial attribute editing using generative adversarial networks. arXiv preprint arXiv:2205.10587 (2022)
86. Nitzan, Y., Bermano, A., Li, Y., Cohen-Or, D.: Face identity disentanglement via latent space mapping. arXiv:2005.07728 (2020)
87. Niu, Z., Zhou, M., Wang, L., Gao, X., Hua, G.: Ordinal regression with multiple output cnn for age estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4920–4928 (2016)
88. Pan, H., Han, H., Shan, S., Chen, X.: Mean-variance loss for deep age estimation from a face. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5285–5294 (2018)
89. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition (2015)
90. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The feret evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10), 1090–1104 (2000)
91. Poggio, B., Brunelli, R., Poggio, T.: Hyberbf networks for gender classification (1992)
92. Pumarola, A., Agudo, A., Martinez, A.M., Sanfeliu, A., Moreno-Noguer, F.: Ganimation: anatomically-aware facial animation from a single image. In: European Conference on Computer Vision (2018)

93. Ren, Y., Li, G., Chen, Y., Li, T.H., Liu, S.: Pirenderer: controllable portrait image generation via semantic neural rendering. In: CVPR (2021)
94. Ricanek, K., Tesafaye, T.: Morph: a longitudinal image database of normal adult age-progression. In: 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pp. 341–345 (2006)
95. Ricanek, K., Tesafaye, T.: Morph: a longitudinal image database of normal adult age-progression. In: 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pp. 341–345. IEEE (2006)
96. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
97. Rodríguez, P., Cucurull, G., Gonfaus, J.M., Roca, F.X., Gonzalez, J.: Age and gender recognition in the wild with deep attention. Pattern Recogn. **72**, 563–571 (2017)
98. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics++: learning to detect manipulated facial images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1–11 (2019)
99. Rothe, R., Timofte, R., Van Gool, L.: Deep expectation of real and apparent age from a single image without facial landmarks. Int. J. Comput. Vis. **126**(2), 144–157 (2018)
100. Shakhnarovich, G., Viola, P.A., Moghaddam, B.: A unified learning framework for real time face detection and classification. In: Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 2002. Proceedings, pp. 14–21 (2002)
101. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: interpreting the disentangled face representation learned by gans. IEEE Trans. Pattern Anal. Mach. Intell. (2020)
102. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
103. Tan, Z., Wan, J., Lei, Z., Zhi, R., Guo, G., Li, S.Z.: Efficient group-n encoding and decoding for facial age estimation. IEEE Trans. Pattern Anal. Mach. Intell. **40**(11), 2610–2623 (2017)
104. Tang, J., Shao, Z., Ma, L.: Fine-grained expression manipulation via structured latent space. In: ICME (2020)
105. Tewari, A., Elgharib, M., Bharaj, G., Bernard, F., Theobalt, C.: Stylerig: rigging stylegan for 3d control over portrait images. CVPR (2020)
106. Toubal, I.E., Lyu, L., Lin, D., Palaniappan, K.: Single view facial age estimation using deep learning with cascaded random forests. In: International Conference on Computer Analysis of Images and Patterns, pp. 285–296. Springer (2021)
107. Tripathy, S., Kannala, J., Rahtu, E.: Facegan: Facial attribute controllable reenactment gan. In: Workshop on Applications of Computer Vision (2021)
108. Upchurch, P., Gardner, J., Pleiss, G., Pless, R., Snavely, N., Bala, K., Weinberger, K.: Deep feature interpolation for image content changes. In: CVPR (2017)
109. Usman, B., Dufour, N., Saenko, K., Bregler, C.: Puppetgan: cross-domain image manipulation by demonstration. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
110. Vaquero, D.A., Feris, R.S., Tran, D., Brown, L., Hampapur, A., Turk, M.: Attribute-based people search in surveillance environments. In: 2009 Workshop on Applications of Computer Vision (WACV), pp. 1–8 (2009)
111. Wan, J., Tan, Z., Lei, Z., Guo, G., Li, S.Z.: Auxiliary demographic information assisted age estimation with cascaded structure. IEEE Trans. Cybern. **48**(9), 2531–2541 (2018)
112. Wang, W., He, F., Zhao, Q.: Facial ethnicity classification with deep convolutional neural networks. In: Chinese Conference on Biometric Recognition, pp. 176–185 (2016)

113. Wu, P.W., Lin, Y.J., Chang, C.H., Chang, E.Y., Liao, S.W.: Relgan: multi-domain image-to-image translation via relative attributes. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
114. Wu, R., Lu, S.: Leed: label-free expression editing via disentanglement (2020)
115. Wu, R., Zhang, G., Lu, S., Chen, T.: Cascade ef-gan: progressive facial expression editing with local focuses. CVPR (2020)
116. Xiao, T., Hong, J., Ma, J.: Elegant: exchanging latent encodings with gan for transferring multiple face attributes. In: ECCV (2018)
117. Xu, S.Z., Huang, H.Z., Zhang, F.L., Zhang, S.H.: Faceshapegene: a disentangled shape representation for flexible face image editing. Graph. Visual Comput. (2021)
118. Xu, Y., Yin, Y., Jiang, L., Wu, Q., Zheng, C., Loy, C.C., Dai, B., Wu, W.: TransEditor: transformer-based dual-space GAN for highly controllable facial editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
119. Yang, G., Fei, N., Ding, M., Liu, G., Lu, Z., Xiang, T.: L2m-gan: learning to manipulate latent space semantics for facial attribute editing. In: CVPR (2021)
120. Yang, M., Zhu, S., Lv, F., Yu, K.: Correspondence driven adaptation for human profile recognition. In: CVPR 2011, pp. 505–512. IEEE (2011)
121. Yang, Z., Ai, H.: Demographic classification with local binary patterns. In: International Conference on Biometrics, pp. 464–473 (2007)
122. Yi, D., Lei, Z., Li, S.Z.: Age estimation by multi-scale convolutional network. In: Asian Conference on Computer Vision, pp. 144–158. Springer (2014)
123. Viazovetskyi, Y., Ivashkin, V., Kashin, E.: Stylegan2 distillation for feed-forward image manipulation (2020)
124. Zeng, X., Huang, J., Ding, C.: Soft-ranking label encoding for robust facial age estimation. IEEE Access **8**, 134209–134218 (2020)
125. Zhang, J., Shu, Y., Xu, S., Cao, G., Zhong, F., Qin, X.: Sparsely grouped multi-task generative adversarial networks for facial attribute manipulation. ACM (2018)
126. Zhang, J., Zeng, X., Wang, M., Pan, Y., Liu, L., Liu, Y., Ding, Y., Fan, C.: Freenet: multi-identity face reenactment (2019)
127. Zhang, K., Liu, N., Yuan, X., Guo, X., Gao, C., Zhao, Z., Ma, Z.: Fine-grained age estimation in the wild with attention lstm networks. IEEE Trans. Circuits Syst. Video Technol. **30**(9), 3140–3152 (2019)
128. Zhang, K., Tan, L., Li, Z., Qiao, Y.: Gender and smile classification using deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 34–38 (2016)
129. Zhong, Y., Sullivan, J., Li, H.: Face attribute prediction using off-the-shelf cnn features. In: 2016 International Conference on Biometrics (ICB), pp. 1–7. IEEE (2016)
130. Zhou, Y., Ni, H., Ren, F., Kang, X.: Face and gender recognition system based on convolutional neural networks. In: International Conference on Mechatronics and Automation (2019)
131. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV) (2017)
132. Zhu, Y., Li, Y., Mu, G., Guo, G.: A study on apparent age estimation. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 25–31 (2015)

Face Presentation Attack Detection

7

Zitong Yu, Chenxu Zhao, and Zhen Lei

7.1 Introduction

Face recognition technology has been widely used in daily interactive applications such as checking-in and mobile payment due to its convenience and high accuracy. However, its vulnerability to presentation attacks (PAs) limits its reliable use in ultra-secure application scenarios. A presentation attack defined in ISO standard [60] is as a presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system. Specifically, PAs range from simple 2D prints, replays and more sophisticated 3D masks and partial masks. To defend the face recognition systems against PAs, both academia and industry have paid extensive attention to developing face presentation attack detection (PAD) [81] technology (or namely "face anti-spoofing (FAS)").

Z. Yu
School of Computing and Information Technology, Great Bay University, Dongguan 523000, China
e-mail: yuzitong@gbu.edu.cn

C. Zhao
MiningLamp Technology, Beijing 100102, China
e-mail: Zhaochenxu@mininglamp.com

Z. Lei (✉)
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
e-mail: zlei@nlpr.ia.ac.cn

School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China

Centre for Artificial Intelligence and Robotics, Hong Kong Institute of Science & Innovation, Chinese Academy of Sciences, Hong Kong, China

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_7

During the initial phase, commercial PAD systems are usually designed based on strong prior knowledge of obvious and macro liveness cues such as eye-blinking [34], face and head movement [67] (e.g., nodding and smiling), and gaze tracking [2]. They assume that the 2D print attacks are static and lack of *interactive* dynamic cues. Despite easy development and deployment, these methods suffer from high false acceptance errors when presented with replayed face videos or partial wearable 3D masks that mimic the interactive liveness cues. To eliminate the requirement of interactive dynamics and explore more intrinsic and micro features for face PAD, plenty of traditional handcrafted feature-based methods [9, 30, 37] are proposed for face PAD. On the one hand, according to the evidence that PAs degraded static/dynamic texture details and spoof artifacts (e.g., moiré pattern), classical handcrafted texture descriptors (e.g., LBP [9] and HOG [30]), image quality assessment metrics [21], and micro motion [63] features are designed for extracting effective spoofing patterns from various color spaces (e.g., RGB, HSV, and YCbCr). On the other hand, considering that 3D mask attacks might contain realistic textural appearance but no quasi-periodic live physiological cues, facial video-based remote physiological signals (e.g., rPPG [75, 78]) measurement technique is introduced for 3D high-fidelity mask attack detection.

Subsequently, with the development of deep learning for computer vision and release of large-scale and diverse face PAD datasets [25, 47, 89], plenty of end-to-end deep learning-based methods [3, 46, 74, 76, 80, 84, 87] are proposed for face PAD. Similar to many binary classification tasks (e.g., gender classification), many works [23, 27, 29, 35, 36, 73] treat face PAD as a binary bonafide/PA classification problem, and thus utilize a simple binary cross-entropy loss for model learning. Recently, researchers found that binary cross-entropy loss cannot provide explicit task-aware supervision signals and the models supervised by this loss easily learn unfaithful patterns [46] (e.g., bezel and background). To alleviate this issue, more and more recent works focus on leveraging auxiliary pixel-wise supervision [3, 23, 46, 76, 83] to provide more fine-grained context-aware supervision signals. For example, according to the geometric discrepancy between bonafide with facial depth and flat 2D PAs, pseudo-depth labels [3, 46] are designed to force the model to learn local geometric cues for bonafide/PA discrimination. Similarly, pixel-wise auxiliary supervisions with reflection maps [76] and binary mask label [23, 47] benefit models by describing the local physical cues on the pixel/patch level.

In real-world scenarios, different domain conditions (e.g., illumination, face resolution, and sensor noise) influence the face PAD system a lot. For instance, a well-designed and -trained deep PAD model on images with normal illumination and high face resolutions might perform poorly under low-light and low-resolution scenarios due to their large distribution gaps. Therefore, learning more generalizable and robust features against unseen domain shifts is vital for practical applications. To this end, domain adaptation [33] and generalization [28, 57, 69] methods are introduced to enhance the generalization capability of deep face PAD models. The former leverages the knowledge from target domain to bridge the gap between source and target domains, while the latter helps PAD models learn generalized feature representation from multiple source domains directly without any access to target data.

The structure of this chapter is as follows. Section 7.2 introduces the research background, including presentation attacks, pipeline for face PAD, and existing datasets. Section 7.3 reviews the handcrafted feature-based and deep learning-based methods for face PAD. Section 7.4 provides experimental results and analysis. Section 7.5 discusses the practical face PAD-based industrial applications. Finally, Section 7.6 summarizes the conclusions and lists some future challenges.

7.2 Background

In this section, we will first introduce the common face PAs and general face PAD pipeline in face recognition system. Then, mainstream camera sensors for face PAD are presented. Finally, existing face PAD datasets are summarized.

7.2.1 Face Presentation Attacks

Face presentation attacks usually mislead the real-world automatic face recognition (AFR) systems via presenting face upon physical mediums (e.g., a photograph, a video, or a 3D mask) of a targeted or obfuscated person in front of the imaging sensors. Some representative PA types are illustrated in Fig. 7.1. According to the intention of whether the attackers would mimic targeted identities or hide their own identities, face PAs [49] can be divided into two categories: (1) *impersonation*, which spoofs the AFR systems to be recognized as someone else via copying a genuine user's facial attributes to presentation instruments such as photo, electronic screen, and 3D mask; and (2) *obfuscation*, which decorates the face to hide or remove the attacker's own identity via wearing glasses/wig or with makeup/tattoo.

PAs are broadly classified into *2D* and *3D* attacks according to the geometric depth. Common 2D PAs usually contain print and replay attacks such as flat/wrapped printed photos, eye/mouth-cut photos, and replay of face videos on electronic screens. Compared with traditional 2D PAs, 3D presentation attacks such as 3D face masks and mannequins

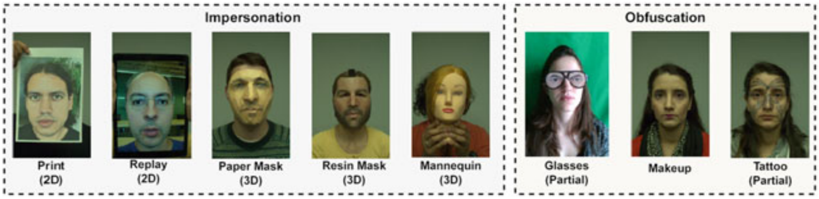


Fig. 7.1 Visualization of different types of face presentation attacks [26]

are more realistic in terms of color, texture, and geometry structure, which can be made of different materials, e.g., paper, resin, plaster, plastic, silicon, and latex. According to the proportion of covered facial region, PAs can be also divided into *whole* and *partial* attacks. Compared with common PAs covering the whole face, a few partial attacks are only presented on partial facial regions. For example, attackers would cut out the eye regions from the print face photo to spoof the eye blinking-based PAD system while funny eyeglasses with adversarial patterns would be worn over the eye region to attack the face PAD algorithms. Compared with attacks on whole face, partial attacks are more obscure and challenging to defend.

7.2.2 Face PAD Pipeline in Face Recognition Systems

The pipeline of face PAD in automatic face recognition systems (AFR) is illustrated in Fig. 7.2. There are *parallel* and *serial* schemes for integrating face PAD with AFR systems. For the *parallel* scheme, the detected faces can be passed over the face PAD and AFR modules to obtain the respective predicted scores, which are then used for parallel fusion. The combined new final score is used to determine whether the sample comes from a genuine user or not. The parallel scheme is suitable to be deployed in multi-core or multi-thread systems with good parallel computation specifications to perform PAD and face recognition simultaneously. Besides, the parallel scheme leaves the space for robust fusion strategies design with PAD and AFR outputs. As for *serial* scheme, detected faces are first forwarded to PAD module to reject the PAs, and only the filtered bonafide faces can be forwarded into the face recognition phase. Despite delayed PAD time for the subsequent AFR of bonafide access attempts, the serial scheme avoids extra work in the AFR module for the case of PAs, since the PAs have already been detected in an early stage.

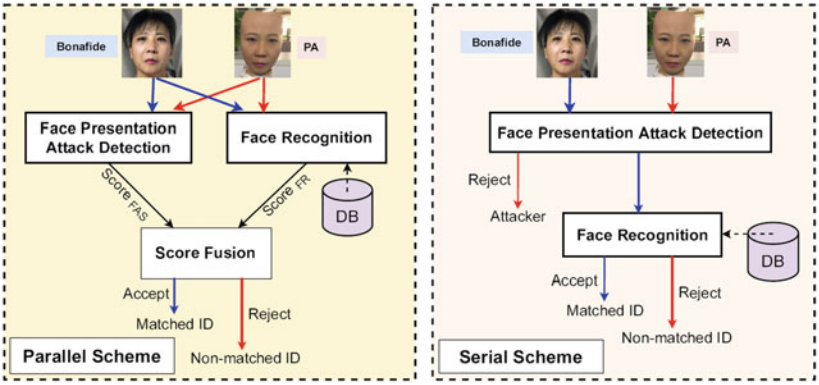


Fig. 7.2 Typical face PAD pipeline. Face PAD could be integrated with face recognition systems in (left) parallel or (right) serial scheme for reliable face ID matching

7.2.3 Camera Sensors for Face PAD

Commercial RGB camera-based face PAD has been widely used in daily face recognition applications like mobile unlocking due to its satisfactory security and low hardware cost. Besides visible RGB modality, depth and near-infrared (NIR) modalities are also widely used in practical PAD deployment with acceptable costs. As for the depth camera, accurate 3D depth geometric surface of the captured face can be measured, which is very appropriate for flat 2D PAD without rich 3D facial cues. Two representative types of depth sensors are Time of Flight (TOF) and 3D Structured Light (SL). Compared with SL, TOF is not only more robust to environmental conditions such as distance and outdoor lighting but also more expensive. Depth cameras with TOF or SL are usually embedded in mainstream mobile phone platforms (e.g., iPhone, Samsung, OPPO, and Huawei) to benefit the RGB–Depth-based face PAD. And, for NIR cameras [61], they contain complementary spectrums besides RGB, which explicitly capture material-aware reflection discrepancy between bonafide faces and PAs but are sensitive to long distance. In addition, RGB–NIR integration hardware modules are also popular in access control systems due to their high performance–price ratio. In real-world deployment with high-security needs, integrating with all three modalities (RGB–NIR–Depth) usually provides the most robust performance in terms of environmental conditions (lighting and distance) and attack types (print, replay, and 3D mask). The characteristics of different sensors for face PAD are compared in Table 7.1. Visualization of typical bonafide and PA samples with RGB, depth, and NIR modalities are given in Fig. 7.9.

Table 7.1 Comparison with camera sensors for face PAD under two environments (lighting condition and distance) and three PA types (print, replay, and 3D mask). "NIR", "TOF", "SL" are short for "Near Infrared", "Time of Flight", "Structured Light", respectively

Sensor	Cost	Environment		Attack type		
		Lighting	Distance	Print	Replay	3D Mask
RGB	Cheap	Poor	Good	Medium	Medium	Medium
Depth (TOF)	Medium	Good	Good	Good	Good	Poor
Depth (SL)	Cheap	Medium	Poor	Good	Good	Poor
NIR	Cheap	Good	Poor	Good	Good	Medium
RGB-NIR	Medium	Good	Medium	Good	Good	Good
RGB-Depth	Medium	Medium	Medium	Good	Good	Medium
RGB-NIR-Depth	Expensive	Good	Good	Good	Good	Good

7.2.4 Face PAD Datasets

In the past decade, a few face PAD datasets have been established for training new PAD techniques and evaluating their performance against domain shifts and PA types. Detailed statistics and descriptions of publicly available unimodal and multimodal face PAD datasets are summarized in Tables 7.2 and 7.3, respectively.

In terms of RGB-based unimodal face PAD datasets shown in Table 7.2, there are only five public datasets [13, 51, 62, 71, 90] at the early stage from years 2010 to 2015. Due to the immature 3D mask manufacturing process with high cost at that time, these datasets only contain 2D PAs (i.e., print and replay attacks) and limited subjects (no more than 50), which have insufficient data scale and attack diversity for generalizable face PAD training and evaluation. Subsequently, there are two main trends for unimodal dataset development: (1) *larger-scale subjects and data amount*. For example, the recently released datasets CelebA-Spoof [89] and HiFiMask [40] contain more than 600000 images and 50000 videos, respectively. Besides, MSU USSA [50] and CelebA-Spoof [89] record more than 1000 and 10000 subjects, respectively. (2) *diverse attack types*. In addition to common 2D print and replay attacks, more sophisticated 3D attacks and novel partial attacks are considered in recent face PAD datasets. For instance, there are high-fidelity 3D mask attacks made of different kinds of materials (e.g., 3D print, plaster, resin) in HKBU-MARs V2 [43] and HiFiMask [40]. As shown in shown in Table 7.3, similar trends of larger-scale subjects and data amount as well as attack types can be found in the development of multimodal face PAD datasets. Moreover, it can be observed that *more kinds of modalities* are collected in recent face PAD datasets. For example, HQ-WMCA [26] and PADISI-Face [56] contain five modalities (RGB, Depth, NIR, short-wave infrared (SWIR), and Thermal).

7.3 Methodology

To determine the liveness of user's faces during the identity verification procedure, *interactive* face PAD methods are usually adopted. However, such interactive instructions (e.g., eye-blinking, facial expression, head movement, and vocal repeating) require users' long-term participation, which is unfriendly and inconvenient. Thanks to the recent software-based methods designed with rich face PAD cues, the *silent* face PAD system could automatically and quickly detect the PAs without any user interactions. In this section, we summarize the classical handcrafted PAD feature-based and recent deep learning-based methods for silent face PAD.

Table 7.2 A summary of unimodal face PAD datasets. "#Sub.", "I", and "V" are short for "Subjects", "images", and "videos", respectively

Dataset & Reference	Year	#Bonafide/PA	#Sub.	Attack types
NUAA [62]	2010	5105/7509(I)	15	Print(flat, wrapped)
YALE_Recaptured [51]	2011	640/1920(I)	10	Print(flat)
CASIA-MFSD [90]	2012	150/450(V)	50	Print(flat, wrapped, cut), Replay(tablet)
REPLAY-ATTACK [13]	2012	200/1000(V)	50	Print(flat), Replay(tablet, phone)
MSU-MFSD [71]	2014	70/210(V)	35	Print(flat), Replay(tablet, phone)
REPLAY-Mobile [15]	2016	390/640(V)	40	Print(flat), Replay(monitor)
HKBU-MARs V2 [43]	2016	504/504(V)	12	Mask(hard resin) from Thatsmyface and REAL-f
MSU USSA [50]	2016	1140/9120(I)	1140	Print(flat), Replay(laptop, tablet, phone)
OULU-NPU [10]	2017	720/2880(V)	55	Print(flat), Replay(phone)
Rose-Youtu [33]	2018	500/2850(V)	20	Print(flat), Replay(monitor, laptop), Mask(paper, crop-paper)
SiW [46]	2018	1320/3300(V)	165	Print(flat, wrapped), Replay(phone, tablet, monitor)
WFFD [27]	2019	2300/2300(I) 140/145(V)	745	Waxworks(wax)
SiW-M [47]	2019	660/968(V)	493	Print(flat), Replay, Mask(hard resin, plastic, silicone, paper, Mannequin), Makeup(cosmetics, impersonation, Obfuscation), Partial(glasses, cut paper)
Swax [64]	2020	Total 1812(I) 110(V)	55	Waxworks(wax)
CelebA-Spoof [89]	2020	156384/469153(I)	10177	Print(flat, wrapped), Replay(monitor, tablet, phone), Mask(paper)
CASIA-SURF 3DMask [83]	2020	288/864(V)	48	Mask(mannequin with 3D print)
HiFiMask [40]	2021	13650/40950(V)	75	Mask(transparent, plaster, resin)

7.3.1 Handcrafted Feature-Based Face PAD

According to the features properties, we introduce the handcrafted feature-based face PAD approaches based on five main cues, i.e., structural material, image quality, texture, micro

Table 7.3 A summary of multimodal face PAD datasets. “SWIR” is short for short-wave infrared

Dataset & Reference	Year	#Bonafide/PA	#Sub.	Sensor	Attack types
3DMAD [17]	2013	170/85(V)	17	RGB, Depth	Mask(paper, hard resin)
MLFP [1]	2017	150/1200(V)	10	RGB, NIR, Thermal	Mask(latex, paper)
CSMAD [4]	2018	104/159(V+I)	14	RGB, Depth, NIR, Thermal	Mask(custom silicone)
CASIA-SURF [88]	2019	3000/18000(V)	1000	RGB, Depth, NIR	Print(flat, wrapped, cut)
WMCA [25]	2019	347/1332(V)	72	RGB, Depth, NIR, Thermal	Print(flat), Replay(tablet), Partial(glasses), Mask(plastic, silicone, and paper, Mannequin)
CeFA [31]	2020	6300/27900(V)	1607	RGB, Depth, NIR	Print(flat, wrapped), Replay, Mask(3D print, silica gel)
HQ-WMCA [26]	2020	555/2349(V)	51	RGB, Depth, NIR, SWIR, Thermal	Laser or inkjet Print(flat), Replay(tablet, phone), Mask(plastic, silicon, paper, mannequin), Makeup, Partial(glasses, wigs, tattoo)
PADISI-Face [56]	2021	1105/924(V)	360	RGB, Depth, NIR, SWIR, Thermal	Print(flat), Replay(tablet, phone), Partial(glasses, funny eye), Mask(plastic, silicone, transparent, Mannequin)

motion, and physiological signals. The handcrafted features are usually cascaded with a support vector machine (SVM) or a multi-layer perception (MLP) for binary classification to distinguish bonafide faces from PAs.

Structural Material-Based Approaches. In real-world cases, PAs are always broadcasted by physical presentation attack instruments (PAIs) (e.g., paper, glass screen, and resin mask), which have obvious material properties different from human facial skin. Such differences can be explicitly described as meaningful spoofing cues (e.g., structural depth and specular reflection) for face PAD. In order to obtain the 3D structure or material of the face, the most direct way is to use a binocular/depth or SWIR camera. However, as a single RGB camera is the most common hardware configuration in practical applications, lots of face PAD research works still focus on 3D and material cue estimation based on the monocular RGB camera. On one hand, based on the assumption that 2D PAs on paper and screen are usually flat and without depth information, Wang et al. [68] proposed to recover the sparse 3D shape of face images to detect various 2D attacks. On the other hand, the illumination and reflection discrepancy of the structural materials between human facial skin and 2D PAs are used as important spoof cues. Kim et al. [16] utilized the illumination diffusion cues based on the fact that illumination from 2D surfaces of 2D attacks diffuses slower and has a more uniform intensity distribution than 3D surfaces. Besides, Wen et al. [71] proposed to calculate the statistical features based on the percentage of the specular reflection components from face image to detect the screen replay attacks. The methods based on the structural material cues have great rationality to detect the 2D PAs theoretically. However, estimating depth and material information from a monocular RGB camera is an ill-conditioned problem, and the computational complexity of these methods is high.

Image Quality-Based Approaches. As the spoof faces are usually broadcasted of the real face from specific physical PAIs, the corresponding face image quality might be degraded due to the color distortion and instrument artifacts, which can be utilized as a significant cue for face PAD. Galbally et al. [21] adopted 25 (21 full-reference and 4 non-reference) image quality assessment (IQA) metrics for face liveness detection. Wen et al. [71] employed three kinds of different IQA features (blurriness, color moment, and color difference) for face PAD, which can effectively represent the intrinsic distortion of spoof images. Image quality-based methods are effective for screen-replayed faces, low-quality printed faces, and rough 3D mask spoof face detection. However, high-quality printed faces as well as high-fidelity 3D mask faces would result in high false acceptance rates for these methods.

Texture-Based Approaches. Due to the PAI properties, textural details in spoof faces are usually coarse and smoothed. In contrast, bonafide faces captured via cameras directly keep more fine-grained local texture cues. Based on this evidence, many texture-based approaches have been developed for face PAD. Specifically, several classical local texture descriptors such as local binary pattern (LBP) [48] and histogram of oriented gradients (HOG) [30] are used to capture fine-grained texture features from face images. Based on the observation that texture features in the HSV color space are more invariant across different environments, Boulkenafet et al. [9] proposed to extract LBP-based color texture features from HSV space, which is efficient and generalizable. However, the texture-based methods rely on high-resolution input to distinguish subtle texture differences between bonafide and spoofing faces. If the image quality is not good enough, it will result in a high false rejection rate. In addition, due to the diversity of image acquisition conditions and spoofing instruments, extracted texture patterns are also variant, which degrades its generalizability under complex real-world scenarios.

Micro Motion-Based Approaches. Liveness detection by capturing the user's short-term micro motion characteristics without interaction is feasible as facial dynamics (e.g., expression and head movement) or dynamic textures from live and spoof samples are distinguishable. Tirunagari et al. [63] proposed to temporally magnify the facial motion first, and then extract two kinds of dynamic features including the histogram of oriented optical flow (HOOF) and Local Binary Pattern histograms from Three Orthogonal Planes (LBP-TOP) for face PAD. However, motion magnification usually brings external noises, which influences the robustness of the subsequent feature representation. Instead of motion magnification, Siddiqui et al. [59] employed dynamic mode decomposition to select the most reliable dynamic mode for facial motion feature extraction. However, the micro motion-based methods are not effective for wrapped/shaking paper attack and video replay attacks due to interference of undesired dynamics. These methods assume that there is a clear non-rigid motion discrepancy between bonafide and PAs, but in fact such micro motion is quite difficult to describe and represent explicitly.

Remote Photoplethysmograph-Based Approaches. Physiological signal is another important living body signal, and it is also an intrinsic cue for distinguishing live faces from artificial materials. In recent years, remote photoplethysmograph (rPPG) technology [79] has developed quickly, which aims at measuring blood pulse flow by modeling the subtle skin color changes caused by the heartbeat. Due to the low transmittance characteristics of artificial materials, rPPG signals from the live faces are usually periodic, but more noisy on the PAs such as 3D mask and printed paper. Therefore, rPPG signals are suitable for face liveness detection. Li et al. [37] analyzed the live/spoof rPPG cues via calculating the statistics of the rPPG frequency responses. Different from the method of spectrum analysis using long-term observation of rPPG signals in the frequency domain, Liu et al. [41] proposed to leverage the temporal similarity of facial rPPG signals for fast 3D mask attack detection, which can be within one second by analyzing the time-domain waveform of the rPPG signal. However, rPPG cues are sensitive to the head motion, light condition, and video quality. Another disadvantage is that the replayed video attack on electronic screen might still contain weak periodic rPPG signals.

7.3.2 Deep Learning-Based Face PAD

With the data-driven deep learning fashion in computer vision, deep neural networks have also been widely used in face PAD. Here we highlight some traditional deep learning approaches with cross-entropy and pixel-wise supervision first, and then introduce domain generalized deep learning methods.

Traditional Deep Learning Approaches with Cross-Entropy Supervision. As face PAD can be intuitively treated as a binary (bonafide vs. PA) or multi-class (e.g., bonafide, print, replay, mask) classification task, numerous deep learning methods are directly supervised with cross-entropy (CE) loss. Given an extracted face input X, deep PAD features F can be represented via forwarding the deep models Φ, and then the cascaded classification heads make the binary predictions Y, which are supervised by the binary cross-entropy (BCE) loss

$$L_{BCE} = -(Y_{gt} log(\Phi(X)) + (1 - Y_{gt}) log(1 - \Phi(X))), \tag{7.1}$$

where Y_{gt} is the ground truth ($Y_{gt} = 0$ for PAs and $Y_{GT} = 1$ for bonafide. Supervised with BCE loss, Yang et al. [73] proposed the first end-to-end deep face PAD method using shallow convolutional neural networks (CNN) for bonafide/PA feature representation. Through the stacked convolution layers, CNN is able to capture the semantic spoof cues (e.g., hand-hold contour of the printed paper). However, training CNN from scratch easily leads to overfitting in the face PAD task due to the limited data amount and coarse supervision signals from BCE loss. To alleviate these issues, on the one hand, some recent researches [12, 24] usually finetune the ImageNet-pretrained models (e.g., ResNet18 and vision transformer) with BCE loss for face PAD. Transferring the well-trained model parameters on large-scale generic object classification task to downstream face PAD data is relatively easier and more efficient. On the other hand, a few works modify BCE loss into a multi-class CE version to provide CNNs with more fine-grained and discriminative supervision signals. Xu et al. [72] rephrased face PAD as a fine-grained classification problem and propose to supervise deep model with multi-class (e.g., bonafide, print, and replay) CE loss. In this way, the intrinsic properties from bonafide as well as particular attack types could be explicitly represented. However, models supervised with multi-class CE loss still suffer from unsatisfactory convergence due to the class imbalance. Another issue is that these supervision signals with only global constraints might cause face PAD models to easily overfit to unfaithful patterns but neglect vital local spoof patterns.

Traditional Deep Learning Approaches with Pixel-wise Supervision. Compared with bonafide faces, PAs usually have discrepant physical properties in local responses. For example, 2D PAs such as plain printed paper and electronic screen are without local geometric facial depth information while the bonafide is in reverse. Motivated by this evidence, some recent works [3, 70, 84] adopt pixel-wise *pseudo-depth* labels (see the fourth column in Fig. 7.4) to guide the deep models, enforcing them to predict the genuine depth for the bonafide samples while zero maps for the PAs. To leverage the multi-level features for

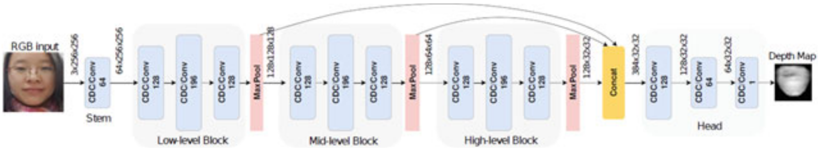


Fig. 7.3 The multi-scale architecture of DepthNet [46] with vanilla convolutions ("Conv" for short) and CDCN [84] with CDC. Inside the blue block are the convolutional filters with 3 × 3 kernel size and their feature dimensionalities

accurate facial depth estimation, Atoum et al. [3] proposed the multi-scale fully convolutional network, namely "DepthNet". With supervision using pseudo-depth labels, the DepthNet is able to predict holistic depth maps for bonafide faces while coarse zero maps for 2D PAs as explainable decision evidence. To further improve the fine-grained intrinsic feature representation capacity, Yu et al. [84] proposed a novel deep operator called central difference convolution (CDC), which can replace vanilla convolutions in DepthNet without extra learnable parameters to form the CDCN architecture (see Fig. 7.3 for detailed structures). Specifically, the CDC operator can be formulated as:

$$y(p_0) = \underbrace{\theta \cdot \sum_{p_n \in R} w(p_n) \cdot (x(p_0 + p_n) - x(p_0))}_{Central\ Difference\ Convolution} + \underbrace{(1-\theta) \cdot \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n)}_{Vanilla\ Convolution}, \tag{7.2}$$

where x, y, and w denote the input features, output features, and learnable convolutional weights, respectively. p_0 denotes the current location on both input and output feature maps while p_n enumerates the locations in neighbor region R. The hyperparameter $\theta \in [0, 1]$ is the trade-off of contributions between intensity-level and gradient-level information. DepthNet with vanilla convolution is a special case of CDCN with CDC when $\theta = 0$, i.e., aggregating local intensity information without gradient message. CDCN is favored in pixel-wise supervision framework and widely used in the deep face PAD community due to its excellent representation capacities of both low-level detailed and high-level semantic cues.

Considering the costly generation of the pseudo-depth maps as well as the meaningless use for 3D face PAs with realistic depth, binary mask label [23] (see the second column in Fig. 7.4) is easier to obtain and more generalizable to all PAs. Specifically, binary supervision would be provided for the deep embedding features at each spatial position corresponding to the bonafide/PA distributions in each original patch (e.g., 16 × 16). With binary mask supervision, the models are able to localize the PAs in the corresponding patches, which is attack-type-agnostic and spatially interpretable. There are also other auxiliary pixel-wise supervisions such as pseudo-reflection map [76] and 3D point cloud map [38] (see the third and last columns of Fig. 7.4, respectively). The former provides physical material reflection cues while the latter contains dense 3D geometric cues. To further learn more intrinsic material-related features, multi-head supervision is developed in [76] to supervise

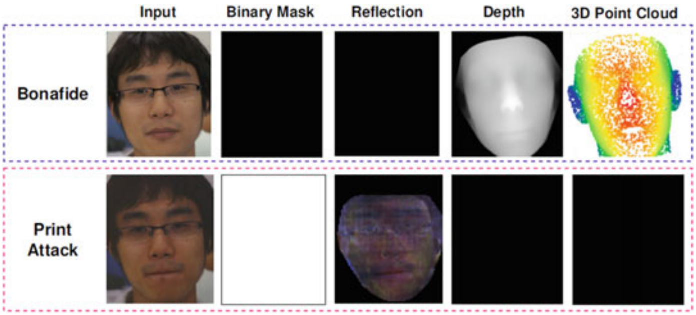


Fig. 7.4 Visualization of pixel-wise supervision signals [77] including binary mask label [23], pseudo-reflection maps [76], pseudo-depth labels [84] and 3D point cloud maps [38] for face PAD

PAD models with multiple pixel-wise labels (i.e., pseudo-depth, binary mask, and pseudo-reflection) simultaneously. The corresponding pixel-wise loss functions can be formulated as

$$L_{depth} = \frac{1}{H \times W} \sum_{i \in H, j \in W} \left\| D_{(i,j)} - D_{gt(i,j)} \right\|_2^2, \tag{7.3}$$

$$L_{reflection} = \frac{1}{H \times W \times C} \sum_{i \in H, j \in W, c \in C} \left\| R_{(i,j,c)} - R_{gt(i,j,c)} \right\|_2^2, \tag{7.4}$$

$$L_{binarymask} = \frac{1}{H \times W} \sum_{i \in H, j \in W} -(B_{gt(i,j)} log(B_{(i,j)}) + (1 - B_{gt(i,j)}) log(1 - B_{(i,j)})), \tag{7.5}$$

where D_{gt}, R_{gt}, and B_{gt} denote ground truth depth map, reflection map, and binary mask map, respectively. H, W, and C mean the height, width, and channels of the maps, respectively. Overall, pixel-wise auxiliary supervision benefits the physically meaningful and explainable representation learning. However, the pseudo-auxiliary labels are usually generated coarsely without human annotations, which are sometimes inaccurate and noisy for partial attacks. For example, the binary mask for FunnyEye glasses attacks should cover the eye regions instead of the whole face).

Generalized Deep Learning Approaches to Unseen Domains. There might be undesired external conditional changes (e.g., in illumination and sensor quality) in real-world deployment. Traditional end-to-end deep learning-based face PAD methods easily overfit to the feature distribution of training data from seen domains and are sensitive to the domain shifts between unseen target domains and seen source domains. In the field of face PAD, "domain shifts" usually indicate the PA-irrelated external environmental changes and actually influence the appearance quality. To alleviate this issue, more recent works focus on enhancing the domain generalization capacity of the face PAD models. On the one hand, some works [28, 57] design domain-aware adversarial constraints to force the PAD models to learn domain-irrelative features from multiple source domains. They assume that the domain-irrelative features contain intrinsic bonafide/PA cues across all seen domains thus might generalize well on unseen domains. On the other hand, a few works [54, 83] utilize domain-aware meta-learning to learn the domain generalized feature space. Specifically, faces from partial source domains are used as query set while those from remained non-overlap domains as support set, which mimics the unseen domains and minimizes their risks at the training phase. To alternatively force the meta-learner to perform well on support sets (domains), the learned models have robust generalization capacity. Domain generalization helps the FAS model learn generalized feature representation from multiple source domains directly without any access to target data, which is more practical for real-world deployment. Despite generalization capacity enhancement for unseen domains, it would deteriorate the discrimination capability for PAD under the seen scenarios to some extent.

7.4 Experimental Results

Here, evaluation results of handcrafted feature-based and deep learning-based approaches on four face PAD datasets (i.e., OULU-NPU [10], CASIA-MFSD [90], Replay-Attack [13], and MSU-MFSD [71]) are compared and analyzed. Specifically, OULU-NPU is used for intra-dataset testings while all four datasets (see Fig. 7.5 for typical examples) are used for cross-dataset testings under serious domain shifts.

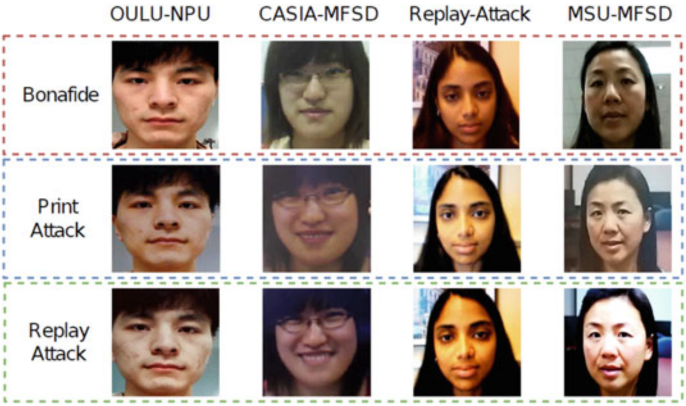


Fig. 7.5 Visualization of the bonafide and PAs from four face PAD datasets. It can be seen that serious domain shifts (e.g., face resolution and illumination) occur among these datasets

7.4.1 Evaluation Metrics

As face PAD systems usually focus on the concept of bonafide and PA acceptance and rejection, two basic metrics False Rejection Rate (FRR) and False Acceptance Rate (FAR) [20] are widely used. The ratio of incorrectly accepted spoofing attacks defines FAR, whereas FRR stands for the ratio of incorrectly rejected live accesses [14]. The most commonly used metrics in both intra- and cross-testing scenarios are Half Total Error Rate (*HTER*) [14], Equal Error Rate (*EER*), and Area Under the Curve (*AUC*). HTER is found out by calculating the average of FRR (ratio of incorrectly rejected bonafide score) and FAR (ratio of incorrectly accepted PA). EER is a specific value of HTER at which FAR and FRR have equal values. AUC represents the degree of separability between bonafide and spoofings. Recently, Attack Presentation Classification Error Rate (*APCER*), Bonafide Presentation Classification Error Rate (*BPCER*), and Average Classification Error Rate (*ACER*) suggested in ISO/IEC DIS 30107- 3:2017 standard [6] are also used for intra-dataset testings [10, 46]. BPCER and APCER measure bonafide and attack classification error rates, respectively. ACER is calculated as the mean of BPCER and APCER, evaluating the reliability of intra-dataset performance.

7.4.2 Intra-dataset Testings

Intra-dataset testing protocol has been widely used in most face PAD datasets to evaluate the model's discrimination ability for PA detection under scenarios with slight domain shift. As the training and testing data are sampled from the same datasets, they share similar domain distribution in terms of the recording environment, subject behavior, etc. The most classical intra-dataset testing protocols are the four sub-protocols of OULU-NPU dataset [10]. Protocol 1 is used to evaluate the generalization performance of the face PAD algorithms under different lighting and background scenarios. Protocol 2 evaluates the PAD performance under unseen PAIs. In Protocol 3, the models are alternatively trained on videos recorded by five smartphones while videos by the remaining smartphone are used for evaluation. Protocol 4 mixes the scenarios of the first three protocols to simulate real-world scenarios, and aims to evaluate the performance of face PAD methods in the integrated scenarios. The performance comparison of recent face PAD methods is shown in Table 7.4. Benefitted from the powerful representation capacity of neural networks with a data-driven fashion, most deep learning methods (except DeepPixBiS [23]) outperform the handcrafted features-based method GRADIANT [23]. With the task-aware pixel-wise supervisions, some deep models such as CDCN [84], FAS-SGTD [70], Disentangled [86], MT-FAS [53], DC-CDN [82], and NAS-FAS [83] have reached satisfied performance ($<5\%$ ACER) on the most challenging Protocol 4 with mixed domain shifts in terms of external environment, attack mediums and recording cameras.

Table 7.4 The results of intra-dataset testing with four sub-protocols on the OULU-NPU [10] dataset. The lower APCER/BPCER/ACER, the better performance. Best results are in **bold**

Prot.	Method	APCER(%) ↓	BPCER(%) ↓	ACER(%) ↓	Prot.	Method	APCER(%) ↓	BPCER(%) ↓	ACER(%) ↓
1	GRADIANT [7]	1.3	12.5	6.9	2	DeepPixBiS [23]	11.4	0.6	6.0
	DRL-FAS [11]	5.4	4.0	4.7		De-Spoof [29]	4.2	4.4	4.3
	STASN [74]	1.2	2.5	1.9		Auxiliary [46]	2.7	2.7	2.7
	Auxiliary [46]	1.6	1.6	1.6		GRADIANT [7]	3.1	1.9	2.5
	De-Spoof [29]	1.2	1.7	1.5		Disentangled [86]	1.1	3.6	2.4
	Disentangled [86]	1.7	0.8	1.3		STASN [74]	4.2	0.3	2.2
	FAS-SGTD [70]	2.0	0.0	1.0		FAS-SGTD [70]	2.5	1.3	1.9
	CDCN [84]	0.4	1.7	1.0		DRL-FAS [11]	3.7	**0.1**	1.9
	BCN [76]	**0.0**	1.6	0.8		BCN [76]	2.6	0.8	1.7
	DeepPixBiS [23]	0.8	**0.0**	0.4		CDCN [84]	1.5	1.4	1.5
	DC-CDN [82]	0.5	0.3	0.4		MT-FAS [53]	1.3	1.4	1.4
	MT-FAS [53]	**0.0**	0.8	0.4		DC-CDN [82]	**0.7**	1.9	1.3
	NAS-FAS [83]	0.4	**0.0**	**0.2**		NAS-FAS [83]	1.5	0.8	**1.2**
3	DeepPixBiS [23]	11.7 ± 19.6	10.6 ± 14.1	11.1 ± 9.4	4	DeepPixBiS [23]	36.7 ± 29.7	13.3 ± 14.1	25.0 ± 12.7
	GRADIANT [7]	2.6 ± 3.9	5.0 ± 5.3	3.8 ± 2.4		GRADIANT [7]	5.0 ± 4.5	15.0 ± 7.1	10.0 ± 5.0
	De-Spoof [29]	4.0 ± 1.8	3.8 ± 1.2	3.6 ± 1.6		Auxiliary [46]	9.3 ± 5.6	10.4 ± 6.0	9.5 ± 6.0
	DRL-FAS [11]	4.6 ± 3.6	1.3 ± 1.8	3.0 ± 1.5		STASN [74]	6.7 ± 10.6	8.3 ± 8.4	7.5 ± 4.7
	Auxiliary [46]	2.7 ± 1.3	3.1 ± 1.7	2.9 ± 1.5		DRL-FAS [11]	8.1 ± 2.7	6.9 ± 5.8	7.2 ± 3.9
	STASN [74]	4.7 ± 3.9	**0.9 ± 1.2**	2.8 ± 1.6		CDCN [84]	4.6 ± 4.6	9.2 ± 8.0	6.9 ± 2.9
	FAS-SGTD [70]	3.2 ± 2.0	2.2 ± 1.4	2.7 ± 0.6		De-Spoof [29]	1.2 ± 6.3	6.1 ± 5.1	5.6 ± 5.7
	BCN [76]	2.8 ± 2.4	2.3 ± 2.8	2.5 ± 1.1		BCN [76]	2.9 ± 4.0	7.5 ± 6.9	5.2 ± 3.7
	CDCN [84]	2.4 ± 1.3	2.2 ± 2.0	2.3 ± 1.4		FAS-SGTD [70]	6.7 ± 7.5	3.3 ± 4.1	5.0 ± 2.2
	Disentangled [86]	2.8 ± 2.2	1.7 ± 2.6	2.2 ± 2.2		Disentangled [86]	5.4 ± 2.9	3.3 ± 6.0	4.4 ± 3.0
	MT-FAS [53]	2.3 ± 1.5	1.9 ± 1.8	2.1 ± 1.7		DC-CDN [82]	5.4 ± 3.3	2.5 ± 4.2	4.0 ± 3.1
	DC-CDN [82]	2.2 ± 2.8	1.6 ± 2.1	1.9 ± 1.1		MT-FAS [53]	**0.9 ± 2.0**	6.4 ± 4.9	3.7 ± 2.9
	NAS-FAS [83]	**2.1 ± 1.3**	1.4 ± 1.1	**1.7 ± 0.6**		NAS-FAS [83]	4.2 ± 5.3	**1.7 ± 2.6**	**2.9 ± 2.8**

7.4.3 Cross-Dataset Testings

This protocol focuses on cross-dataset level domain generalization ability measurement, which usually trains models on one or several datasets (source domains) and then tests on unseen datasets (shifted target domain). We summarize recent deep face PAD approaches on two favorite cross-dataset testings [57, 84] on four benchmark datasets (i.e., OULU-NPU (O) [10], CASIA-MFSD (C) [90], Replay-Attack (I) [13], and MSU-MFSD (M) [71]) in Table 7.5. As illustrated in Fig. 7.5, there are serious domain shifts among these four datasets in terms of resolution, illumination, sensor noise, etc. When trained on Replay-Attack and tested on CASIA-MFSD, most handcrafted feature-based methods as well as traditional deep models perform poorly (>20% HTER) due to the serious lighting and camera resolution variations. In contrast, when trained on multiple source datasets (i.e., OULU-NPU, MSU-MFSD, and Replay-Attack), domain generalization-based methods achieve acceptable performance on CASIA-MFSD (especially SSDG [28] and SSAN [69] with 10.44% and 10.00% HTER, respectively). Overall, introducing more training data from diverse domains might benefit and stabilize the generalized feature learning.

7.5 Applications

In this section, we will concentrate on describing face presentation attack detection in different industry applications, including the attributes of the scenarios, sensors, protocols, and approaches.

7.5.1 Online Identity Verification Scenario

As illustrated in Fig. 7.6, this scenario refers to the online face recognition authentication process by customers through their mobile devices or PCs. Face PAD in the online identity verification scenario aims to force the algorithm to discriminate spoofing faces from the criminals. Criminals attempt to obtain the authentication result of the attacked individual via spoofing faces. After obtaining the authentication result, they can steal the money or information from the accessed account. The architectures of this scenario are:

- The system requires a high level of security indicating the face PAD algorithm is required to reach a higher performance.
- The application runs on the client side, and most of the devices are mobile phones. Thus, the algorithm needs to reach strong hardware compatibility.
- The criminals' attack conditions are more relaxed due to the relatively private application environment. Alternatively, the attack cost is cheap, and repeated attempts can be made.

Table 7.5 Results of the cross-dataset testings among OULU-NPU (O), CASIA-MFSD (C), Replay-Attack (I), and MSU-MFSD (M) datasets with different numbers of source domains for training. For example, "C to I" means training on CASIA-MFSD and then testing on Replay-Attack

Method		C to I	I to C	C&O&M to I		I&O&M to C	
		HTER(%) ↓	HTER(%) ↓	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
Handcrafted Feature	LBP [19]	55.9	57.6	–	–	–	–
	Motion [19]	50.2	47.9	–	–	–	–
	LBP-TOP [19]	49.7	60.6	–	–	–	–
	Motion-Mag [63]	50.1	47.0	–	–	–	–
	Spectral Cubes [52]	34.4	50.0	–	–	–	–
	Color Texture [8]	30.3	37.7	40.40	62.78	30.58	76.89
Traditional Deep Learning	De-Spoof [29]	28.5	41.1	–	–	–	–
	STASN [74]	31.5	30.9	–	–	–	–
	Auxiliary [46]	27.6	28.4	29.14	71.69	33.52	73.15
	Disentangled [86]	22.4	30.3	–	–	–	–
	FAS-SGTD [70]	17.0	**22.8**	–	–	–	–
	BCN [76]	16.6	36.4	–	–	–	–
	PS [77]	13.8	31.3	19.55	86.38	18.25	86.76
	CDCN [84]	15.5	32.6	–	–	–	–
	DC-CDN [82]	**6.0**	30.1	15.88	91.61	15.00	92.80
	MT-FAS [53]	–	–	11.93	94.95	18.44	89.67
	NAS-FAS [83]	–	–	11.63	96.98	15.21	92.64

(continued)

Table 7.5 (continued)

Method		C to I	I to C	C&O&M to I		I&O&M to C	
		HTER(%) ↓	HTER(%) ↓	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
Generalized Deep Learning	MADDG [57]	–	–	22.19	84.99	24.50	84.51
	PAD-GAN [65]	–	–	20.87	86.72	19.68	87.43
	RF-Meta [58]	–	–	17.30	90.48	20.27	88.16
	DRDG [45]	–	–	15.56	91.79	19.05	88.79
	FGHV [42]	–	–	16.29	90.11	12.47	93.47
	SSDG [28]	–	–	11.71	96.59	10.44	95.94
	SSAN [69]	–	–	**8.88**	**96.79**	**10.00**	**96.67**

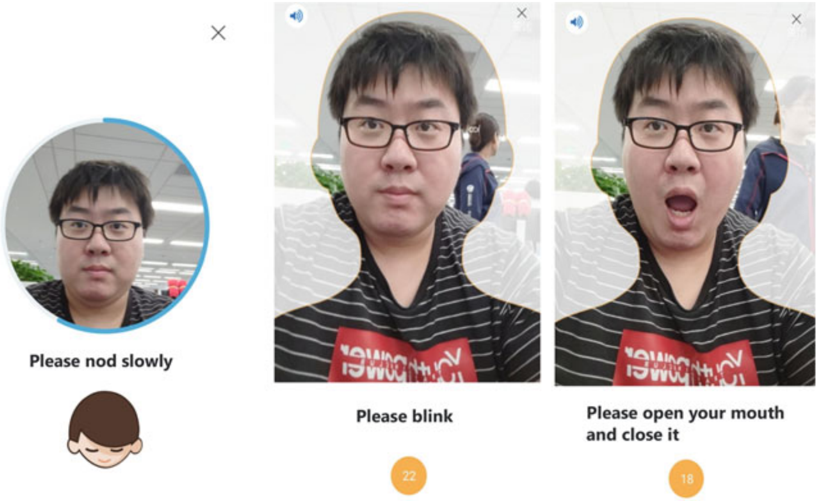


Fig. 7.6 The online identity verification scenario. The customer completes an identity authentication process online through the mobile APP. During this process, it is usually required to make cooperative actions according to the system prompts

- A variety of devices and unpredictable novel PAs keep evolving and unknown PAs may be presented to them. Data-driven models may give unpredictable results when faced with out-of-distribution samples.
- Customers could cooperate to a certain extent.
- Only one face is allowed in one operation process, and multiple faces will be regarded as illegal operations.

Sensors in this scenario are usually diverse due to the customers' diverse mobile devices. In order to achieve satisfactory compatibility for the diverse hardware, we prefer RGB modality because most hardware devices support this modality. However, approaches designed with the single RGB modality usually have weak feature representation compared with multimodal inputs. To bridge this gap, large-scale training data would be collected to cover as many PAs and domains as possible.

Approaches in this scenario always treat the face PAD problem as a binary classification task [35], and utilize binary cross-entropy loss to optimize the model. Domain adaptation and generalization approaches [28, 32, 65] can also be applied in this scenario. For example, meta-learning [44, 45, 66]-based methods can be adopted to improve the model's generalization capacity on unseen attacks and domains. To enhance model robustness, in this scenario, the face PAD system usually receives additional dynamic information by requiring the customers to cooperate to complete the facial actions [34, 67] or by changing the color of the screen [85] (see Fig. 7.7 for visualization). This interactive instructions is also called *Liveness Detection.*

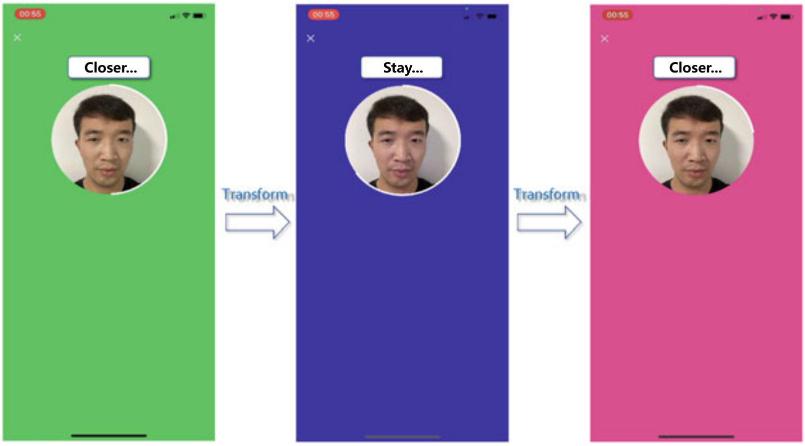


Fig. 7.7 Color verification code. During one verification process, verifying whether the reflective color matches the color verification code, in which the background presents high-brightness images of different colors, provides additional dimensional knowledge for the face PAD algorithm

Protocols and Evaluations. For better evaluation under this scenario, we need to build a sufficiently large dataset and a sufficiently rigorous testing protocol. Considering that the attacker in this scenario can repeatedly attack the system in a private environment, we need to ensure that the PAD algorithm can reach a high true positive rate for PAs and decrease the false negative rate based on all bonafide samples being correctly detected. To further verify this goal, the Receiver Operating Characteristic (ROC) curve [5] is proposed to evaluate the face PAD method on the large-scale dataset. In the ROC, we pursue a lower false positive rate and higher true negative rate under the same false positive rate.

Related Applications. In this scenario, there are some typical applications with similar characteristics, such as the FaceID of mobile devices. In contrast, mobile phone manufacturers can select more sensors, some of which have multiple modalities and could improve the security level.

7.5.2 Offline Payment Scenario

As illustrated in Fig. 7.8, this scenario refers to the process in which the customer utilizes the fixed face recognition instrument for offline identity authentication or payment. Face PAD approach aims to secure the face recognition system from malicious PAs including 3D masks. This scenario has the following diverse characteristics:

- Offline payment scenarios will directly involve money transactions, which require the system to dedicate a very high-security level that also needs the face anti-spoofing algo-

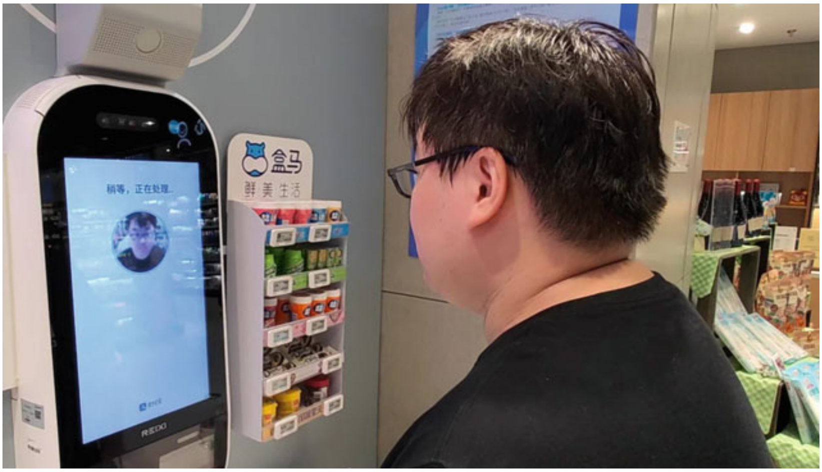


Fig. 7.8 The offline payment scenario. The customer completes a face recognition payment process through an industry-specific machine with a fixed offline location. Such special machines are generally equipped with multi-modalities sensors

rithm performance to reach a higher level. With respect to this, a single RGB modality is almost incapable.

- The application also runs on the client side. However, the carrier equipment of the system is generally a standardized industry-specific machine, and multimodal cameras can be equipped. The domain in this scenario is relatively simple due to the device's fixed sensor and fixed location.
- The criminals' attack conditions are constrained due to the relatively public application environment and fixed device location. Generally, there will be staff or other customers on site. For attackers, the attack cost increases as the numbers of repeated attempts are reduced.
- In this scenario, the most significant challenge to the system comes from the 3D high-fidelity masks and head models, because equipping multimodal cameras effectively defends the common planar PAs such as print, replay, and paper mask.
- Customers are only required to do limited cooperation.
- Only one face is allowed in one operation process, and multiple faces will be regarded as illegal operations.

Sensors in this scenario prefer to choose multimodal cameras, such as RGB and NIR binocular camera or RGB, NIR and Depth structured light camera [22] or TOF camera [18]. The combination of NIR and Depth modalities aims to effectively defend against planar attacks such as print and replay. As illustrated in Fig. 7.9, the 2D planar PAs cannot perform

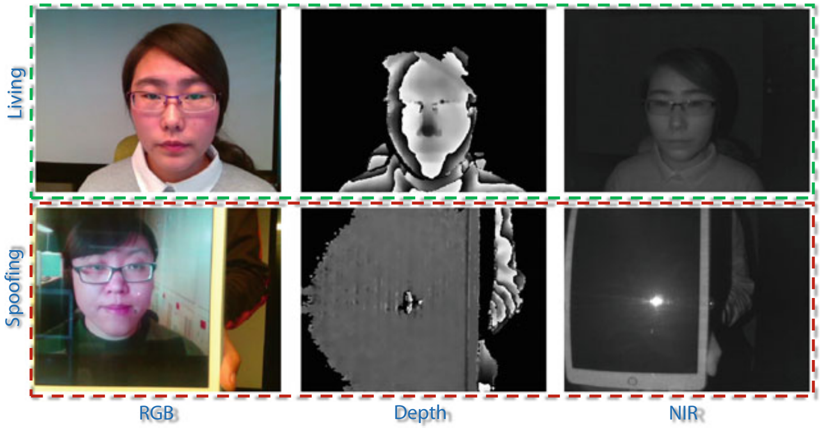


Fig. 7.9 Imaging of bonafide and spoofing faces in RGB, NIR, and Depth multimodal sensors. In contrast, the 2D planar PAs show significantly different patterns in imaging NIR and Depth

face imaging in these two modalities. Combined with the RGB modality, it can defend against some 3D forms of attack, such as 3D high-fidelity masks.

Approaches in this scenario treated the face PAD problem as a typical multimodal fusion task. With multi-modal inputs, mainstream methods extract complementary multimodal features using feature-level fusion strategies. As there is redundancy across multimodal features, direct feature concatenation easily results in high-dimensional features and overfitting. To alleviate this, Zhang et al. [88] proposed the SD-Net to utilize a feature re-weighting mechanism to select the informative and discard the redundant channel features among RGB, depth, and NIR modalities. However, even if the features of the three modalities are combined, some spoofing faces are still challenging to discriminate, such as 3D high-fidelity masks. To further boost the multi-modal performance, Liu et al. [39] proposed a large-scale 3D high-fidelity mask dataset and the contrastive context-aware learning, which is able to leverage rich contexts accurately among pairs of bonafide and high-fidelity mask attack.

Protocols and Evaluations. A sufficiently large-scale dataset as well as rigorous testing protocols should be established to evaluate the algorithm for this scenario. Considering that the system in this scenario requires a very high-security level, we need to ensure that the algorithm can reach a high true positive rate for the spoofing faces and decrease the false negative rate based on all positive samples being correctly detected. ROC curve [5] is utilized to evaluate the face PAD method on the large-scale dataset. In the ROC, we pursue a lower false positive rate and higher true negative rate under the same false positive rate.

Related Applications. In this scenario, there are some typical applications with similar characteristics, such as face access control in the buildings. Face access control has relatively low requirements on the system's security level, and different sensors can be selected according to the actual condition and cost.

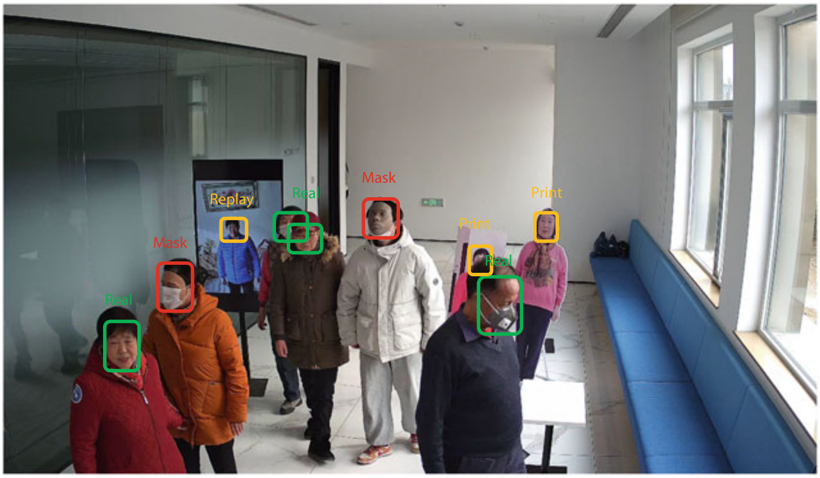


Fig. 7.10 The surveillance scenario. There will be multiple faces involved in the surveillance camera. In addition to regular faces, there will be criminals wearing high-fidelity masks and faces that belong to noise in the background of posters, advertisements, etc

7.5.3 Surveillance Scenario

As illustrated in Fig. 7.10, this scenario refers to the process of the customer unconsciously passing through the surveillance camera framing area. Compared with the above two scenarios, the function of the face anti-spoofing module in the surveillance scenario is quite different. The PAs in this scenario are mainly divided into two categories. One is the criminal who wears a mask and mixes in the crowd trying to escape the surveillance. The other is the PAs shown on the screen or demonstrated on the billboard in the background of the surveillance. For the monitoring system, the first category is an attack behavior, and the second category is a noise. This scenario has the following diverse characteristics:

- According to the properties of the above-mentioned two categories of face PAs, the face PAD approach in this scenario aims to caution against abnormal behavior and remove background noise, so it means nothing to the system that requires a very high-security level.
- The application can run on the cloud side. Because this is surveillance or a similar scenario, the sensors deployed are mostly surveillance cameras.
- The imaging in this scenario is a kind of long-range monitoring. In the long-range monitoring, the features of each low-resolution face are relatively sparse, which increases the difficulty for the face PAD algorithm. Alternatively, in order to be caught by the surveillance cameras as less as possible, the criminals will pass through the acquisition area

of the cameras as quickly as possible. In other words, the number of repeated attacks is reduced.

- In long-range camera monitoring, the most challenging problem in this scenario is whether the face PAD algorithm could effectively discriminate bonafide faces, 3D high-fidelity masks, and print/replay in the background.
- The customers are completely passive and not required to cooperate.
- The number of faces is no longer a limitation toward the system.

Sensors deployed in this scenario are mostly surveillance cameras [55]. The images are captured by the cameras with long imaging distance and wide imaging range. Multiple faces emerge in the viewfinder at the same time and at different distances. It also includes faces on screens and posters. Some of these sensors also contain multimodal modules, and these multimodal cameras (such as NIR) can mainly deal with dark light environments. In fact, due to the long distance, it is challenging to capture rich spoofing cues. Alternatively, in this scenario, only the RGB modality can provide adequate embedding knowledge.

Approaches in this scenario formulate a long-distance face PAD problem. To the best of our knowledge, this issue is less studied in the current face PAD community, and we still do not have a standardized, well-defined benchmark.

Protocols and Evaluations. To better evaluate the algorithm for this scenario, we need to establish a sufficiently large-scale dataset as well as generalized protocols. Considering that the system in this scenario does not require a very high-security level, we firstly ensure that the algorithm can reach a high true negative rate and bonafide faces will not be misidentified, in case it makes the system repeatedly alarm. Based on this foundation, we will continue to decrease the false negative rate. To further verify this goal, the ROC curve [5] is utilized to evaluate the face PAD method on the large-scale dataset. In the ROC, we pursue a lower false negative rate and higher true positive rate under the same false negative rate.

Related Applications. In this scenario, there are some typical applications with similar characteristics, such as the passenger flow monitoring in the market. In contrast, face PAD approaches in this type of system only require excluding the face on the poster or screen.

7.6 Conclusion and Future Challenge

In this chapter, a comprehensive review of the research and practical applications related to face PAD are carried out. On the one hand, handcrafted feature- and deep learning-based approaches based on unimodal or multimodal cameras can all detect face PAs to a certain extent without rigorous interaction. It is still hard to say which feature performs better in different application scenarios. For example, traditional rPPG-based methods are good at 3D mask attack detection in unimodal RGB scenarios while appearance-based deep learning methods perform well on 2D face PAD. Extensive experiments and practical applications have proved that (1) combining multiple features can achieve better face PAD results; and (2) task-aware prior knowledge (e.g., pixel-wise supervision), advanced learning paradigms (e.g., domain generalization) and multimodal inputs (e.g., RGB–NIR–Depth) can benefit the discrimination, generalization, and interpretability of face PAD algorithms. On the other hand, hardware cost of multimodal sensors (e.g., RGB+Depth+NIR is more expensive and costs more spatial spaces than RGB+Depth or RGB+NIR) and practical application requirements (e.g., distance and real-time efficiency) need to be comprehensively considered for real-world deployment under different application scenarios.

Face PAD has achieved rapid improvement over the past few years due to advanced camera sensors and well-designed algorithms. However, face PAD is still an unsolved problem in terms of task-oriented challenges such as subtle spoof pattern representation, complex real-world domain gaps, and rapidly iterative novel attacks as well as novel application scenarios like long-range monitoring for surveillance. Here, we list the two limitations of the current development. On one side, the evaluation under saturating and unpractical testing benchmarks/protocols cannot really reflect the effectiveness of the methods for real-world applications. For example, most datasets are recorded in controlled lab environments but rarely considering the real-world offline payment and surveillance scenarios. Thus, it is urgent to establish more large-scale practical application-oriented benchmarks to bridge the gaps between academia and industry. On the other side, most existing works train and adapt deep face PAD models with huge stored source face data in fixed scenarios, and neglect (1) the privacy and biometric sensitivity issue; and (2) the continuous adaptation for emerging domains and attacks. To design source-free continuous learning face PAD algorithms might be potential for domain-robust real-world deployment and dynamic updating against novel attacks.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 62276254, 62176256, 62206280, 62106264, and 62306061) and the InnoHK program.

References

1. Agarwal, A., Yadav, D., Kohli, N., Singh, R., Vatsa, M., Noore, A.: Face presentation attack with latex masks in multispectral videos. In: CVPRW (2017)
2. Ali, A., Deravi, F., Hoque, S.: Liveness detection using gaze collinearity. In: ICEST. IEEE (2012)
3. Atoum, Y., Liu, Y., Jourabloo, A., Liu, X.: Face anti-spoofing using patch and depth-based cnns. In: IJCB (2017)
4. Bhattacharjee, S., Mohammadi, A., Marcel, S.: Spoofing deep face recognition with custom silicone masks. In: BTAS (2018)
5. Bi, J., Bennett, K.P.: Regression error characteristic curves. In: ICML (2003)
6. Biometrics., I.J.S.: Information technology–biometric presentation attack detection–part 3: testing and reporting (2017)
7. Boulkenafet, Z., Komulainen, J., Akhtar, Z., Benlamoudi, A., Samai, D., Bekhouche, S.E., Ouafi, A., Dornaika, F., Taleb-Ahmed, A., Qin, L., et al.: A competition on generalized software-based face presentation attack detection in mobile scenarios. In: IJCB. IEEE (2017)

8. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face anti-spoofing based on color texture analysis. In: ICIP (2015)
9. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face spoofing detection using colour texture analysis. TIFS (2016)
10. Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A.: Oulu-npu: a mobile face presentation attack database with real-world variations. In: FGR (2017)
11. Cai, R., Li, H., Wang, S., Chen, C., Kot, A.C.: Drl-fas: a novel framework based on deep reinforcement learning for face anti-spoofing. IEEE TIFS (2020)
12. Chen, H., Hu, G., Lei, Z., Chen, Y., Robertson, N.M., Li, S.Z.: Attention-based two-stream convolutional networks for face spoofing detection. TIFS (2019)
13. Chingovska, I., Anjos, A., Marcel, S.: On the effectiveness of local binary patterns in face anti-spoofing. In: Biometrics Special Interest Group (2012)
14. Chingovska, I., Dos Anjos, A.R., Marcel, S.: Biometrics evaluation under spoofing attacks. IEEE TIFS (2014)
15. Costa-Pazo, A., Bhattacharjee, S., Vazquez-Fernandez, E., Marcel, S.: The replay-mobile face presentation-attack database. In: BIOSIG. IEEE (2016)
16. De Marsico, M., Nappi, M., Riccio, D., Dugelay, J.L.: Moving face spoofing detection via 3d projective invariants. In: ICB, pp. 73–78. IEEE (2012)
17. Erdogmus, N., Marcel, S.: Spoofing face recognition with 3d masks. TIFS (2014)
18. Foix, S., Alenya, G., Torras, C.: Lock-in time-of-flight (tof) cameras: a survey. IEEE Sens. J. **11**(9), 1917–1926 (2011)
19. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: Can face anti-spoofing countermeasures work in a real world scenario? In: ICB (2013)
20. Galbally, J., Alonso-Fernandez, F., Fierrez, J., Ortega-Garcia, J.: A high performance fingerprint liveness detection method based on quality related features. Futur. Gener. Comput. Syst. **28**(1), 311–321 (2012)
21. Galbally, J., Marcel, S., Fierrez, J.: Image quality assessment for fake biometric detection: application to iris, fingerprint, and face recognition. IEEE TIP (2013)
22. Geng, J.: Structured-light 3d surface imaging: a tutorial. Adv. Optics Photonics **3**(2), 128–160 (2011)
23. George, A., Marcel, S.: Deep pixel-wise binary supervision for face presentation attack detection. In: ICB (2019)
24. George, A., Marcel, S.: On the effectiveness of vision transformers for zero-shot face anti-spoofing. arXiv preprint arXiv:2011.08019 (2020)
25. George, A., Mostaani, Z., Geissenbuhler, D., Nikisins, O., Anjos, A., Marcel, S.: Biometric face presentation attack detection with multi-channel convolutional neural network. TIFS (2019)
26. Heusch, G., George, A., Geissbühler, D., Mostaani, Z., Marcel, S.: Deep models and shortwave infrared information to detect face presentation attacks. TBIOM (2020)
27. Jia, S., Li, X., Hu, C., Guo, G., Xu, Z.: 3d face anti-spoofing with factorized bilinear coding. arXiv preprint arXiv:2005.06514 (2020)
28. Jia, Y., Zhang, J., Shan, S., Chen, X.: Single-side domain generalization for face anti-spoofing. In: CVPR (2020)
29. Jourabloo, A., Liu, Y., Liu, X.: Face de-spoofing: anti-spoofing via noise modeling. In: ECCV (2018)
30. Komulainen, J., Hadid, A., Pietikainen, M.: Context based face anti-spoofing. In: BTAS (2013)
31. Li, A., Tan, Z., Li, X., Wan, J., Escalera, S., Guo, G., Li, S.Z.: Casia-surf cefa: a benchmark for multi-modal cross-ethnicity face anti-spoofing. WACV (2021)
32. Li, H., Jialin Pan, S., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: CVPR (2018)

33. Li, H., Li, W., Cao, H., Wang, S., Huang, F., Kot, A.C.: Unsupervised domain adaptation for face anti-spoofing. IEEE TIFS (2018)
34. Li, J.W.: Eye blink detection based on multiple gabor response waves. In: ICMLC, vol. 5, pp. 2852–2856. IEEE (2008)
35. Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M., Hadid, A.: An original face anti-spoofing approach using partial convolutional neural network. In: IPTA (2016)
36. Li, L., Xia, Z., Jiang, X., Roli, F., Feng, X.: Compactnet: learning a compact space for face presentation attack detection. Neurocomputing (2020)
37. Li, X., Komulainen, J., Zhao, G., Yuen, P.C., Pietikäinen, M.: Generalized face anti-spoofing by detecting pulse from face videos. In: ICPR (2016)
38. Li, X., Wan, J., Jin, Y., Liu, A., Guo, G., Li, S.Z.: 3dpc-net: 3d point cloud network for face anti-spoofing (2020)
39. Liu, A., Zhao, C., Yu, Z., Wan, J., Su, A., Liu, X., Tan, Z., Escalera, S., Xing, J., Liang, Y., Guo, G., Lei, Z., Li, S.Z., Zhang, D.: Contrastive context-aware learning for 3d high-fidelity mask face presentation attack detection. IEEE TIFS (2022)
40. Liu, A., Zhao, C., Yu, Z., Wan, J., Su, A., Liu, X., Tan, Z., Escalera, S., Xing, J., Liang, Y., et al.: Contrastive context-aware learning for 3d high-fidelity mask face presentation attack detection. IEEE TIFS (2022)
41. Liu, S., Lan, X., Yuen, P.: Temporal similarity analysis of remote photoplethysmography for fast 3d mask face presentation attack detection. In: WACV (2020)
42. Liu, S., Lu, S., Xu, H., Yang, J., Ding, S., Ma, L.: Feature generation and hypothesis verification for reliable face anti-spoofing. In: AAAI (2022)
43. Liu, S., Yuen, P.C., Zhang, S., Zhao, G.: 3d mask face anti-spoofing with remote photoplethysmography. In: ECCV. Springer (2016)
44. Liu, S., Zhang, K.Y., Yao, T., Bi, M., Ding, S., Li, J., Huang, F., Ma, L.: Adaptive normalized representation learning for generalizable face anti-spoofing. In: ACM MM (2021)
45. Liu, S., Zhang, K.Y., Yao, T., Sheng, K., Ding, S., Tai, Y., Li, J., Xie, Y., Ma, L.: Dual reweighting domain generalization for face presentation attack detection. In: IJCAI (2021)
46. Liu, Y., Jourabloo, A., Liu, X.: Learning deep models for face anti-spoofing: binary or auxiliary supervision. In: CVPR (2018)
47. Liu, Y., Stehouwer, J., Jourabloo, A., Liu, X.: Deep tree learning for zero-shot face anti-spoofing. In: CVPR (2019)
48. Määttä, J., Hadid, A., Pietikäinen, M.: Face spoofing detection from single images using micro-texture analysis. In: IJCB (2011)
49. Marcel, S., Nixon, M.S., Fierrez, J., Evans, N.: Handbook of biometric anti-spoofing: Presentation attack detection. Springer (2019)
50. Patel, K., Han, H., Jain, A.K.: Secure face unlock: spoof detection on smartphones. TIFS (2016)
51. Peixoto, B., Michelassi, C., Rocha, A.: Face liveness detection under bad illumination conditions. In: ICIP. IEEE (2011)
52. Pinto, A., Pedrini, H., Schwartz, W.R., Rocha, A.: Face spoofing detection through visual codebooks of spectral temporal cubes. IEEE TIP (2015)
53. Qin, Y., Yu, Z., Yan, L., Wang, Z., Zhao, C., Lei, Z.: Meta-teacher for face anti-spoofing. IEEE TPAMI (2021)
54. Qin, Y., Zhao, C., Zhu, X., Wang, Z., Yu, Z., Fu, T., Zhou, F., Shi, J., Lei, Z.: Learning meta model for zero- and few-shot face anti-spoofing. AAAI (2020)
55. Qureshi, F.Z., Terzopoulos, D.: Surveillance camera scheduling: a virtual vision approach. Multimedia Syst. **12**(3), 269–283 (2006)
56. Rostami, M., Spinoulas, L., Hussein, M., Mathai, J., Abd-Almageed, W.: Detection and continual learning of novel face presentation attacks. In: ICCV (2021)

57. Shao, R., Lan, X., Li, J., Yuen, P.C.: Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In: CVPR (2019)
58. Shao, R., Lan, X., Yuen, P.C.: Regularized fine-grained meta face anti-spoofing. In: AAAI (2020)
59. Siddiqui, T.A., Bharadwaj, S., Dhamecha, T.I., Agarwal, A., Vatsa, M., Singh, R., Ratha, N.: Face anti-spoofing with multifeature videolet aggregation. In: ICPR (2016)
60. international organization for standardization: Iso/iec jtc 1/sc 37 biometrics: information technology biometric presentation attack detection part 1: framework. In: https://www.iso.org/obp/ui/iso (2016)
61. Sun, X., Huang, L., Liu, C.: Context based face spoofing detection using active near-infrared images. In: ICPR (2016)
62. Tan, X., Li, Y., Liu, J., Jiang, L.: Face liveness detection from a single image with sparse low rank bilinear discriminative model. In: ECCV. Springer (2010)
63. Tirunagari, S., Poh, N., Windridge, D., Iorliam, A., Suki, N., Ho, A.T.: Detection of face spoofing using visual dynamics. IEEE TIFS (2015)
64. Vareto, R.H., Saldanha, A.M., Schwartz, W.R.: The swax benchmark: attacking biometric systems with wax figures. In: ICASSP (2020)
65. Wang, G., Han, H., Shan, S., Chen, X.: Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In: CVPR (2020)
66. Wang, J., Zhang, J., Bian, Y., Cai, Y., Wang, C., Pu, S.: Self-domain adaptation for face anti-spoofing. In: AAAI (2021)
67. Wang, L., Ding, X., Fang, C.: Face live detection method based on physiological motion analysis. Tsinghua Sci. Technol. **14**(6), 685–690 (2009)
68. Wang, T., Yang, J., Lei, Z., Liao, S., Li, S.Z.: Face liveness detection using 3d structure recovered from a single camera. In: ICB, pp. 1–6. IEEE (2013)
69. Wang, Z., Wang, Z., Yu, Z., Deng, W., Li, J., Li, S., Wang, Z.: Domain generalization via shuffled style assembly for face anti-spoofing. In: CVPR (2022)
70. Wang, Z., Yu, Z., Zhao, C., Zhu, X., Qin, Y., Zhou, Q., Zhou, F., Lei, Z.: Deep spatial gradient and temporal depth learning for face anti-spoofing. In: CVPR (2020)
71. Wen, D., Han, H., Jain, A.K.: Face spoof detection with image distortion analysis. IEEE TIFS (2015)
72. Xu, X., Xiong, Y., Xia, W.: On improving temporal consistency for online face liveness detection. In: ICCVW (2021)
73. Yang, J., Lei, Z., Li, S.Z.: Learn convolutional neural network for face anti-spoofing. arXiv preprint arXiv:1408.5601 (2014)
74. Yang, X., Luo, W., Bao, L., Gao, Y., Gong, D., Zheng, S., Li, Z., Liu, W.: Face anti-spoofing: model matters, so does data. In: CVPR (2019)
75. Yu, Z., Cai, R., Li, Z., Yang, W., Shi, J., Kot, A.C.: Benchmarking joint face spoofing and forgery detection with visual and physiological cues. arXiv preprint arXiv:2208.05401 (2022)
76. Yu, Z., Li, X., Niu, X., Shi, J., Zhao, G.: Face anti-spoofing with human material perception. In: ECCV (2020)
77. Yu, Z., Li, X., Shi, J., Xia, Z., Zhao, G.: Revisiting pixel-wise supervision for face anti-spoofing. IEEE TBIOM (2021)
78. Yu, Z., Li, X., Wang, P., Zhao, G.: Transrppg: remote photoplethysmography transformer for 3d mask face presentation attack detection. IEEE SPL (2021)
79. Yu, Z., Li, X., Zhao, G.: Facial-video-based physiological signal measurement: recent advances and affective applications. IEEE Signal Process. Mag. (2021)
80. Yu, Z., Qin, Y., Li, X., Wang, Z., Zhao, C., Lei, Z., Zhao, G.: Multi-modal face anti-spoofing based on central difference networks. In: CVPRW (2020)

81. Yu, Z., Qin, Y., Li, X., Zhao, C., Lei, Z., Zhao, G.: Deep learning for face anti-spoofing: a survey. IEEE TPAMI (2022)
82. Yu, Z., Qin, Y., Zhao, H., Li, X., Zhao, G.: Dual-cross central difference network for face anti-spoofing. In: IJCAI (2021)
83. Yu, Z., Wan, J., Qin, Y., Li, X., Li, S.Z., Zhao, G.: Nas-fas: Static-dynamic central difference network search for face anti-spoofing. IEEE TPAMI (2020)
84. Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., Zhou, F., Zhao, G.: Searching central difference convolutional networks for face anti-spoofing. In: CVPR (2020)
85. Zhang, J., Tai, Y., Yao, T., Meng, J., Ding, S., Wang, C., Li, J., Huang, F., Ji, R.: Aurora guard: reliable face anti-spoofing via mobile lighting system. arXiv preprint arXiv:2102.00713 (2021)
86. Zhang, K.Y., Yao, T., Zhang, J., Tai, Y., Ding, S., Li, J., Huang, F., Song, H., Ma, L.: Face anti-spoofing via disentangled representation learning. In: ECCV (2020)
87. Zhang, S., Liu, A., Wan, J., Liang, Y., Guo, G., Escalera, S., Escalante, H.J., Li, S.Z.: Casia-surf: a large-scale multi-modal benchmark for face anti-spoofing. TBIOM **2**(2), 182–193 (2020)
88. Zhang, S., Wang, X., Liu, A., Zhao, C., Wan, J., Escalera, S., Shi, H., Wang, Z., Li, S.Z.: A dataset and benchmark for large-scale multi-modal face anti-spoofing. In: CVPR (2019)
89. Zhang, Y., Yin, Z., Li, Y., Yin, G., Yan, J., Shao, J., Liu, Z.: Celeba-spoof: large-scale face anti-spoofing dataset with rich annotations. In: ECCV. Springer (2020)
90. Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., Li, S.Z.: A face antispoofing database with diverse attacks. In: ICB (2012)

Face Feature Embedding

8

Yuge Huang, Jianqing Xu, and Shouhong Ding

8.1 Introduction

A facial feature typically represents the original image in an embedding space, where the distance between embeddings is used to measure face similarity. To implement face verification and recognition at scale, it is indispensable to have a discriminative face embedding in which faces of the same person have small distances and faces of different people have large distances. Once such embedding is obtained, face verification involves thresholding the distance between the two embeddings, and face recognition becomes a k-Nearest Neighbor (k-NN) classification problem that takes feature embeddings as inputs rather than original images.

Traditional methods attempt to utilize handcrafted features as facial embeddings. The previous works (i.e., Kanade's work [30] in the early 70s) focused on the investigation of geometrical relationships, including distances and angles between facial landmarks (such as eyes, mouths, etc.). Later, numerous methods have been proposed to explore a variety of features for face recognition, such as Gabor [58], SIFT [5], and Local Binary Patterns [1, 2]. However, such features are susceptible to variants of pose and illumination, failing to produce satisfactory performance in practical applications. In recent years, the revival of Convolutional Neural Networks (CNN) has achieved remarkable results in deep face recognition, owing to vast amounts of training data [6, 87], efficient network architectures

Y. Huang (✉) · J. Xu · S. Ding
Tencent, Shanghai 200233, P.R. China
e-mail: yugehuang@tencent.com

J. Xu
e-mail: joejqxu@tencent.com

S. Ding
e-mail: ericshding@tencent.com

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_8

[8, 37], and the application of an appropriate loss functions [9, 72]. Besides, CNNs learn the embeddings of deep facial features through the training process, and these facial embeddings have a great discriminative power.

This chapter aims to review previous works on these three aspects in obtaining a discriminative face feature embedding. The remaining of the chapter is structured as follows. Section 8.2 reviews various loss functions for general deep face recognition. Sections 8.3 and 8.4 cover the commonly-used neural network architectures and large-scale training datasets, respectively. Section 8.5 provides an overview of several prevalent topics in deep face recognition, including long tail, noise-tolerant, uncertainty learning, and cross-variation face recognition. Section 8.6 presents three distinct loss functions, each designed for a specific purpose in deep face recognition: general face embedding learning, cross-variation face recognition, and uncertainty estimation. Finally, Sect. 8.7 draws a conclusion of this chapter.

8.2 Loss Function

A primary challenge for deep face recognition is to design an appropriate loss function that enhances the discriminative power of the face feature embedding. The initial deep learning approaches for face recognition like DeepFace [66] and DeepID [59] train the CNNs using a cross-entropy-based softmax loss function, which is commonly employed in object recognition [18, 34]. This loss function can be formulated as follows:

$$\mathcal{L} = -\log \sum_{i=1}^{N} \frac{e^{W_{y_i} x_i + b_{y_i}}}{\sum_{j=1}^{c} e^{W_j x_i + b_j}}, \tag{8.1}$$

where N is the batch size, $x_i \in R^d$ is the feature embedding of i-th sample that belongs to the y_i class, $W_j \in R^d$ is the j-th column of the weight $W \in R^{d \times c}$ and b_j is the bias term. c and d represent the number of classes (identities) and the size of feature embedding, respectively. Nonetheless, researchers [42, 62] find that the learned features with the original softmax are not discriminative enough for the practical face recognition problem, because the testing identities are usually disjoint from the training set. Thereby, various extensions of loss functions have been proposed to increase the discriminative power of face feature embedding. These loss functions can be generally divided into two groups: softmax-based classification losses and metric learning losses. Softmax-based classification techniques typically enhance the discriminative power of the original softmax by incorporating a margin penalty or mining strategy into the softmax formulation. Metric learning methods aim to directly learn a mapping from face images to a compact Euclidean space, where inter-person distance is increased and intra-person distance is decreased. In practice, many studies usually combine the advantages of the aforementioned learning procedures to train the network.

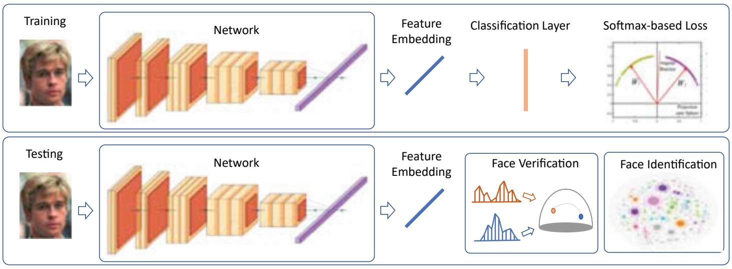


Fig. 8.1 **General pipeline of softmax-based classification methods**. At the training stage, the embedding layer is followed by a fully connected classification layer. The whole network is trained by a softmax-based classification loss function in an end-to-end manner. At the testing stage, the classification layer is discarded and only the face embeddings are used to perform distance calculations or KNN classification for face verification or identification

8.2.1 Softmax-Based Classification Loss Function

The general pipeline of softmax-based methods is depicted in Fig. 8.1. In the training phase, these methods add a fully connected classification layer after the embedding layer and then train the entire network end-to-end to obtain face feature embeddings. In the testing phase, the classification layer is discarded and only the face embeddings are used to perform distance calculations or KNN classification for face verification or identification.

The initial efforts to improve the original softmax function start with introducing the norm constraints. As demonstrated in [48], face feature embeddings trained with a softmax loss have a tendency to overfit high-quality data and fail to correctly classify faces acquired under challenging conditions. To overcome this disadvantage, L_2-softmax [48] adds an L2-constraint to the feature embeddings, which restricts them to lie on a hypersphere with a fixed radius. DeepVisage [17] shares a similar philosophy but uses a particular case of batch normalization to normalize the feature descriptor. Instead of directly utilizing the hard normalization operation, Ring loss [86] employs soft normalization, where it gradually learns to constrain the norm to the scaled unit circle. NormFace [71] and CoCo loss [43] further apply L_2 normalization constraint on both face embeddings and classification layer weights. Consequently, the original softmax can be modified as follows:

$$\mathcal{L} = -\log \frac{e^{s(\cos\theta_{y_i})}}{e^{s(\cos\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{s(\cos\theta_j)}}. \tag{8.2}$$

where the $cos\theta_j$ is derived from the inner product $W_j x_i$ when the individual weight is normalized to $\|W_j\| = 1$ and the face feature is normalized and re-scaled to s. Despite the softmax loss function reformulated by normalization improves face recognition performance, there is still a performance gap for practical applications. Thus, in addition to feature or weight normalization, margin-based or mining-based loss functions became increasingly popular. These variants share the following general loss formulation:

$$\mathcal{L} = -\log \frac{e^{T(\cos\theta_{y_i})}}{e^{T(\cos\theta_{y_i}) + \sum_{j=1, j\neq y_i}^{n} e^{N(t,\cos\theta_j)}}}, \tag{8.3}$$

where the functions $T(\cos\theta_{y_i})$ and $N(t, \cos\theta_j)$ define the positive and negative cosine similarities, respectively.

8.2.1.1 Margin-Based Loss Function

L-softmax [41] first introduces a multiplicative margin penalty into the original softmax loss by letting $T(\cos\theta_{y_i}) = \|W_{y_i}\|\|x_i\|cos(m\theta_{y_i})$, where m is the multiplicative margin. Sphereface [42] simplifies this function further by normalizing the weights; specifically, $T(\cos\theta_{y_i}) = \|x_i\|cos(m\theta_{y_i})$. However, the multiplicative angular margin in $cos(m\theta_{y_i})$ must be computed using a series of approximations, resulting in an unstable convergence during training. Thus, they propose a hybrid loss function that incorporates the original softmax loss to alleviate the convergence issue in practice. CosFace [72] and AM-Softmax [70] introduce an additive margin penalty on the cosine space $T(\cos\theta_{y_i}) = scos(\theta_{y_i}) + m$, improving the convergence stability and discriminative power of face feature embeddings. Then, ArcFace [9] introduces an additive angular margin penalty $T(\cos\theta_{y_i}) = scos(\theta_{y_i} + m)$ that corresponds to the geodesic distance margin penalty on a hypersphere manifold.

Although these margin-based loss functions are simple to implement and achieve better performance than the original softmax, several problems persist. For example, two crucial hyperparameters, s and m, which are essential for training stability and final recognition performance, must be manually selected. Face samples of varying image quality share the same fixed hyperparameter values. To address the first issue, AdaCos [81] deeply analyzes the effects of the hyperparameters s and m in the margin-based softmax loss functions from the perspective of classification probability and proposes a hyperparameter-free cosine-based loss function. P2SGrad [82] investigates cosine softmax losses by analyzing their gradients and proposes a probability-to-similarity gradient that uses cosine similarity rather than classification probability for updating neural network parameters. To address the second issue, certain studies [32, 45] incorporate face image quality into margin-based loss functions, thereby preventing the learned face feature embedding from overfit to low-quality samples. Specifically, MagFace [45] employs a magnitude-aware margin to pull easy samples to class centers while pushing difficult samples away. The positive cosine similarity is defined as $T(\cos\theta_{y_i}) = scos(\theta_{y_i} + m(a_i))$, where a_i represents the magnitude of each feature and $m(a_i)$ is a strictly increasing convex function. AdaFace [32] assigns different importance to different samples based on their image quality, thereby avoiding the overemphasis of unidentifiable images while concentrating on challenging yet recognizable samples. The positive cosine similarity is defined as $T(\cos\theta_{y_i}) = scos(\theta_{y_i} + g_{angle}) - g_{add}$, where g_{angle} and g_{add} are the functions of $\|\hat{z_i}\|$ and $\hat{z_i}$ is a normalized image quality in a batch.

Table 8.1 The decision boundaries of softmax-based loss functions under the binary classification case

Loss	Decision boundary
L-softmax	$\|W_{y_i}\| cos(m\theta_{y_i}) = \|W_j\| cos(m\theta_j)$
NormFace	$\cos\theta_{y_i} = \cos\theta_j$
SphereFace	$\cos(m\theta_{y_i}) = \cos\theta_j$
CosFace	$\cos\theta_{y_i} - m = \cos\theta_j$
ArcFace	$\cos(\theta_{y_i} + m) = \cos\theta_j$
MagFace	$\cos(\theta_{y_i} + m(a_i)) = \cos\theta_j$
AdaFace	$\cos(\theta_{y_i} - m \cdot \|\hat{z_i}\|) - (m \cdot \|\hat{z_i}\| + m) = \cos\theta_j$
MV-Arc-Softmax	$\cos(\theta_{y_i} + m) = \cos\theta_j$ (easy) $\cos(\theta_{y_i} + m) = t\cos\theta_j + t - 1$ (hard)
CurricularFace	$\cos(\theta_{y_i} + m) = \cos\theta_j$ (easy) $\cos(\theta_{y_i} + m) = (t + \cos\theta_j)\cos\theta_j$(hard)

8.2.1.2 Mining-Based Loss Function

A hard sample mining strategy is essential for enhancing the discriminative ability of face feature embeddings, as hard samples are more informative and thus more discriminatory than easy samples. Although mining-based loss functions such as Focal loss [38] and Online Hard Sample Mining [56] are widely used in object detection, they are rarely employed in face recognition. In face recognition, hard mining can be subdivided into hard-positive and hard-negative mining. Hard-positive refers to faces that are visually dissimilar to the same individual, whereas hard negative refers to faces that are visually similar to different identities [19].

Recent studies [26, 74] introduce the hard-negative mining strategy into the margin-based loss functions to enhance face feature embedding from the negative view. They both define misclassified samples as hard samples. To emphasize hard samples, MV-Arc-Softmax [74] re-weights the negative logits with an extra margin penalty. CurricularFace [26] employs the Curriculum Learning (CL) strategy to concentrate on simple samples during the initial training phase and hard samples later on. Table 8.1 provides a summary of the decision boundaries of the softmax-based loss functions in the case of binary classification.

8.2.2 Metric Learning Loss Function

The objective of metric learning is to learn a distance function. As depicted in Fig. 8.2, the core concept of metric learning is bringing together similar samples and pushing apart dissimilar samples in a feature space. The contrastive loss [23, 61, 62] and the triplet loss [52] are the commonly used metric learning-based loss functions. The contrastive function can be formulated as follows:

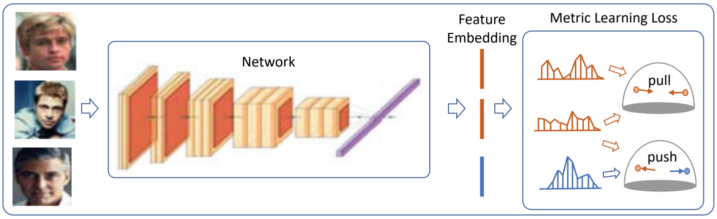


Fig. 8.2 General pipeline of metric learning methods. The face embedding is learned by bringing together similar samples and pushing apart dissimilar samples in a feature space

$$\mathcal{L} = \begin{cases} \max(0, \|f(x_i) - f(x_j)\| - \epsilon^+), & \mathrm{y}_i = y_j \\ \max(0, \epsilon^- - \|f(x_i) - f(x_j)\|), & \mathrm{y}_i \neq y_j \end{cases} \tag{8.4}$$

where $y_i = y_j$ denotes $f(x_i)$ and $f(x_j)$ is a positive pair, $y_i \neq y_j$ denotes a negative pair, $f(\cdot)$ is the feature embedding function, and ϵ^+ and ϵ^- are the positive and negative margins, respectively. DDML [23] only uses the contrastive loss, while the DeepID series of works [61, 62] combine the contrastive and softmax loss to learn a discriminative representation. In contrast to the contrastive loss that considers the absolute distances of the matching pairs and non-matching pairs, the triplet loss considers the relative difference of the distances between them. The formula of the triplet loss is as follows:

$$\mathcal{L} = \sum_{i=1}^{N} \left[\|f(x_i^a) - f(x_i^p)\| - \|f(x_i^a) - f(x_i^n)\| + \epsilon\right]_+ \tag{8.5}$$

where ϵ is the margin, x_i^a denotes the anchor sample, x_i^p and x_i^n refer to the positive sample and negative sample, respectively. FaceNet [52] is the first work to directly optimize the embedding only using the triplet loss. Following FaceNet, several works [50, 51] also adopt the triplet loss as the training supervision. Due to the fact that contrastive loss and triplet loss only consider one negative sample at a time, they are susceptible to combinatorial explosion when dealing with massive amounts of training data containing numerous negative pairs. Even though equipped with a hard sample mining strategy, they occasionally encounter training instability due to the selection of training samples. Center loss [76] and its variant [77] are simple alternatives to improve the original softmax loss. Center loss learns a center for each class and penalizes the distances between the deep features and the corresponding class centers. It is formulated as follows:

$$\mathcal{L} = \|f(x_i) - c_{y_i}\| \tag{8.6}$$

where $f(x_i)$ is the face feature embedding belonging to the y_i-th class and c_{y_i} is the learned class center for class y_i.

By observing that the class center learned using the center loss is identical to the classification weight of the last layer using the softmax loss function, we may gain a unified understanding of the softmax classification loss function and the metric learning function. Softmax-based loss functions can be viewed as prototype learning and use sample-to-prototype comparisons between training samples and class-wise prototypes stored in the final classification layer. In contrast, metric learning loss functions compare samples to one another. Softmax-based and metric learning-based loss functions can be connected if the prototype is likewise considered a sample. Several loss functions that unify these two core paradigms are proposed based on this observation. Circle loss [63] presents a unified formula for learning pair-wise and class-wise deep features simultaneously. VPL [11] represents each class as a distribution to simulate sample-to-sample comparison within the classification framework.

8.3 Network Architectures

The advanced feature learning capability of deep convolutional networks contributes in part to the rapid development of deep face recognition. This section summarizes the prevalent network architectures for deep face recognition. These networks can be split into three categories based on their design intent: general, specialized, and mobile networks.

8.3.1 General Networks

Face recognition is typically modeled as a visual classification task. Therefore, the state-of-the-art classification networks can be easily used as feature extraction backbones for face recognition. In the ImageNet2012 challenge, the solution utilizing AlexNet [34] as its backbone outperforms the conventional approach by a large margin, allowing researchers to glimpse the potential of convolutional networks for the first time. VGGNet [57] improves AlexNet with a deeper network structure and a smaller 3×3 convolution kernel, demonstrating that a deeper structure can enhance a network's expressive ability. GoogLeNet [64] proposes the structure of Inception as the basic unit of the network to further improve the performance. VGGNet and GoogLeNet are afterward employed as the backbone for the face recognition works, i.e., VGGface [47] and FaceNet [52] respectively, achieving remarkable performance on the standard face benchmarks. Afterward, ResNet [18] with the residual structure as the basic unit is proposed. The network solves training difficulty when the convolutional network's depth increases. As a general-purpose backbone network, ResNet is widely used in various tasks of computer vision. Consequently, works such as ArcFace [9] and CurricularFace [26] employ a variant of ResNet as the backbone network for feature extraction. These works have achieved remarkable results on various face recognition testing sets, greatly promoting the research progress in the field of face recognition. To obtain more discriminative face embeddings, DenseNet [25] presents a network structure with Dense Block as the basic unit. This structure integrates and mixes the characteristics of different layers, further mitigating the influence of gradient dispersion during network training. Meanwhile, SENet [22] and AttentionNet [46] are proposed to extract key features. The resulting features can be used as fundamental network components for face recognition.

8.3.2 Specialized Networks

Several network structures are specially designed to accommodate the characteristics of face recognition tasks. The primary objective of these networks is to ensure that the extracted features have to contain pertinent semantic information. GroupFace [33] introduces a face recognition architecture consisting of K fully connected layers and proposes a self-distributed grouping method to effectively supervise multiple latent group-aware representations. AFRN [31] utilizes a feature-pair relational network to capture the relations between two local appearance patches. Certain works [12, 60, 62] divide face images into multiple patches and employ separate networks to extract patch embeddings. The features extracted by each network are then combined into the face's final feature embedding. FANFace [78] achieves feature integration through a new proposed layer that combines the face landmark network and the face feature extraction network to guide the face feature extraction with face landmark information.

8.3.3 Mobile Networks

Although networks with hundreds of layers and millions of parameters achieve great recognition accuracy, it is impossible to deploy these networks on a large number of mobile and embedded devices. To solve the issue, many mobile networks are provided in order to strike a compromise between network inference speed and recognition precision. SqeezeNet [28] proposes to replace 3×3 convolution with 1×1 convolution, whereas the MobileNet series [20, 21, 49] replace the conventional convolution with a depth-wise separable convolution. MobileFaceNet [8] applies the MobileNet idea to the development of a mobile network for face recognition. Later, alternative networks for image recognition tasks have emerged. For instance, ShuffleNet [80] proposes channel shuffle, while EfficientNet [67] investigates the expansion of depth, breadth, and resolution during the network architecture design. RepVGG [13] proposes a re-parameterization technique to produce a simple architecture that is friendly to GPUs and dedicated inference chips. These networks have demonstrated their efficacy in the object recognition task, and it is worthwhile to investigate their applicability in deep face recognition.

8.4 Large-Scale Training Datasets

Large-scale training datasets are essential for learning discriminative enough face embeddings. This section discusses the publicly available large-scale training datasets for face recognition, summarized in Table 8.2.

CASIA-WebFace [79]: CASIA-WebFace is the first public dataset commonly used for deep face recognition training. The dataset contains 500K photos of 10K celebrities collected from the IMDb website. It is semi-automatically cleaned via tag-constrained similarity clustering. Specifically, the authors start with each celebrity's main photo and those photos that contain only one face. Then faces are gradually added to the dataset constrained by feature similarities and name tags.

CelebA [44]: CelebA is a large-scale face attributes dataset with over 200K photos of celebrities, and each is annotated with 40 attributes. Images in this dataset cover large pose variations and background clutter. Besides, CelebA has large diversities, large quantities, and rich annotations, including $10,177$ identities, $202,599$ number of face images, and 5 landmark locations, 40 binary attributes annotations per image. The dataset can be employed as the training and test sets for the following computer vision tasks: face attribute recognition, face recognition, face detection, landmark (or facial part) localization, face editing, and face synthesis.

UMDFace [4]: UMDFace is a face dataset containing two parts: static images and video frames. The static image part contains $367,888$ face annotations for $8,277$ subjects divided into 3 batches. The annotations contain human-curated bounding boxes for faces and estimated pose (yaw, pitch, and roll), locations of twenty-one keypoints, and gender information generated by a pre-trained neural network. The video frame part contains $3,735,476$ annotated video frames extracted from a total of $22,075$ for $3,107$ subjects. The annotations

Table 8.2 Public large-scale training datasets for face recognition

Dataset	Year	#Identities	#Images	Source	Description
CASIA	2014	10575	494414	IMDb	First published large-scale face dataset
CelebA	2015	10,177	202,599	Search engine	Rich annotations of attributes and identities
UMDFace	2015	8277	367K	Search engine	Still images and video frames; abundant variation of facial pose
MS1M	2016	100 K	10M	Search engine	Large-scale public dataset of celebrity faces; noisy
VGGFace2	2017	9131	3.31M	Search engine	Head part of long tail; cross pose, age and ethnicity
IMDB	2018	57K	1.7M	IMDB	Large-scale noise-controlled dataset
Glint360K	2021	360 K	17M	Search engine	Large-scale and cleaned dataset
WebFace260M	2021	4M	260M	Search engine	Largest public dataset of celebrity faces

contain the estimated pose (yaw, pitch, and roll), locations of twenty-one keypoints, and gender information generated by a pre-trained neural network.

MS1M [16]: MS-Celeb-1M is a large-scale face recognition dataset consisting of 100K identities with around 100 facial photos per identity. The initial identification labels are automatically extracted from web sites. Consequently, the original dataset contains a substantial amount of noise. Deng et al. [9] improve this dataset semi-automatically and offer MS1MV2 and MS1MV3 as noise-cleaned variants. This original MS1M dataset has been withdrawn and should no longer be used.

VGGFace2 [6]: VGGFace2 is made of around 3.31 million images divided into 9131 classes, each representing a different identity. There are two divides of the dataset: one for training and another for testing. The latter contains around 170000 images divided into 500 identities and all the other images belong to the remaining 8631 classes available for training. To construct the datasets, the authors focused their efforts on reaching a very low label noise and a high pose and age diversity.

IMDb [69]: IMDb is a large-scale noise-controlled dataset for face recognition research. The dataset contains about 1.7 million faces, 59 K identities, which are manually cleaned from 2.0 million raw images. All images are derived from the IMDb website.

Glint360K [3]: An et al. [3] merged CASIA, MS1MV2, and Celeb500K face datasets, and cleaned the mixed dataset to form a new face training dataset called Glint360K. This dataset contains 17 million images of 360 K individuals.

WebFace260M [87]: WebFace260M is a new million-scale face benchmark constructed for the research community to close the data gap behind the industry. It provides a noisy version containing 4M identities and 260M faces and high-quality training data with 42M images of 2M identities cleaned by an automatic cleaning method.

8.5 Specific Face Recognition Topics

In addition to the loss function, network architecture, and training dataset, several problems still need to be solved to obtain a good face embedding in various scenarios. This section discusses several specific research topics in deep face recognition, including long tail, cross-variation, noise-robust, and uncertainty learning.

8.5.1 Long-Tail Learning

Most large-scale face datasets exhibit long-tailed distribution. That is, only a limited number of classes (persons) frequently appear, while most of the other classes have spare examples [83]. A native solution to this issue is simply cutting the tailed data and only keeping identities with enough examples [47]. The flaw of such a disposal strategy, however, is that information within these data may be omitted. Thus, several methods are proposed for incorporating such complementary knowledge into rich classes in order to improve the overall performance of face recognition. Range loss [83] and MML [75] investigate the hybrid approach that combines the softmax and metric learning loss functions in order to utilize long-tailed data. In particular, Range loss minimizes the k largest range's mean values in one class and maximizes the shortest inter-class distance within one batch. MML increases the margin between these overly close class center pairs by establishing a minimum margin for all class center pairs. An alternative solution is to integrate the adaptive margin into the margin-based softmax loss function. AdaptiveFace [40] uses an adaptive margin to make the model learn a particular margin for each class for adaptively squeezing its intra-class variations. Fair loss [39] learns an appropriate adaptive margin by Deep Q-learning for each class.

8.5.2 Cross-Variation Face Recognition

In real-world applications of face recognition, a primary challenge is to handle the diverse variation in pose, resolution, race and illumination, etc. Thus, it is essential to improve the generalization ability of face feature embedding across diverse variation factors. There are usually two schemes: variation-specific and generic methods to recognize these hard samples with large variations. Variation-specific methods are usually designed for a particular task. For instance, to achieve pose-invariant face recognition, either handcrafted or learned features are extracted to enhance robustness against pose while remaining discriminative to the identities [65]. To address resolution-invariant face recognition, a unified feature space is learned in [53] for mapping Low-Resolution (LR) and High-Resolution (HR) images. However, the aforementioned methods were specifically designed for the respective variations, therefore, their ability to generalize from one variation to another is limited. Different from variation-specific methods, generic methods focus on improving the discriminative power of face embeddings under cross variations. URFace [55] presents a framework for learning universal representations with significant variations unseen in the training data. It divides the feature embedding into multiple sub-embeddings corresponding to a number of semantically significant variations, such as low resolution, occlusion, and head pose, and then trains each sub-embedding with synthetic data. DDL [27] utilizes a distillation loss to improve the discriminative ability of face embedding on samples with large variations.

8.5.3 Noise-Robust Learning

Due to the data source and collection methods, label noise is inevitable in large-scale datasets for face recognition. This work [69] sheds light on the influences of data noise for the face recognition task and demonstrates that margin-based methods are more susceptible to massive label noise. To solve the problem, certain studies [9, 69, 85] aim to clean the noise data manually or algorithmically. For example, Deng et al. [9] refine the original MS1M, and Wang et al. [69] manually construct a noise-controlled IMDb-Face dataset. Zhang et al. [85] propose an automatic noise cleansing framework using two cascaded graph convolutional networks, which perform the global-to-local discrimination to select valuable data in a noisy environment.

Due to the expense of accurate manual annotations, an embedding learning of noise-robust face has garnered considerable interest. Hu et al. [24] present a noise-tolerant loss function by computing time-varying weights for samples based on the angle distribution. Wang et al. [73] introduced a co-mining strategy that uses the loss values as the cue to detect noisy labels and re-weights the predicted clean faces to make them dominate the model training. AMC [84] is a meta learning-based noise-cleaning algorithm that learns the data distribution to be cleaned and makes automatic adjustments based on class differences. Sub-center ArcFace [10] relaxes the intra-class constraint of ArcFace to improve the robustness against label noise. Specifically, it adopts K sub-centers for each class, and the training sample only needs to be close to any of the K positive sub-centers as opposed to a single positive center. The recent work PartialFC [3] alleviates the influence of the noise by random-sampling a subset of negative class centers to compute the margin-based softmax loss. Even though some progress has been made, it remains a challenging research topic to learn a discriminative feature embedding under noise.

8.5.4 Uncertainty Learning

In deep face recognition, embedding techniques with margin-based softmax loss functions have proven to be highly effective. The manifold spaces exploited by these approaches are spherical. However, these models rely on deterministic embeddings and hence suffer from the feature ambiguity dilemma. Whereby, low-quality or noisy images are mapped into poorly learned regions of representation space, leading to inaccuracies. To overcome this challenge, PFE [54] and DUL [7] leverage probabilistic modeling to model face images with a multi-variate independent Gaussian distribution instead of a point mass. As a result, one anticipates that the related Gaussian covariance will indicate the degree to find the ambiguous facial feature, serving as a measure of uncertainty. These methods are modeled in Gaussian statistics and simultaneously output multiple Gaussian statistical variables. SCF [36] presents a framework for face confidence learning in spherical space. Mathematically, it extends von Mises-Fisher density to its r-radius counterpart and derives a new closed-form optimization objective. Theoretically, the suggested probabilistic approach improves interpretability, leading to principled feature pooling.

8.6 Specific Loss Functions for Deep Face Recognition

This section introduces three specific loss functions for deep face recognition. Specifically, Sect. 8.6.1 presents a softmax-based classification loss function termed CurricularFace for discriminative face feature learning. Section 8.6.2 presents a general loss function termed DDL to handle hard face samples with diverse variations in real-world face recognition applications. Section 8.6.3 presents a von Mises-Fisher density-based loss function termed SCF for face confidence learning in spherical space.

8.6.1 Adaptive Curricular Learning Loss (CurricularFace)

In the past margin-based softmax loss functions, the mining strategy was disregarded. Consequently, the difficulty of each sample is not exploited, leading to convergence issues when employing a large margin on small backbones, e.g., MobileFaceNet [8]. Also, previous mining-based loss functions over-emphasize hard samples in the early training stage, hindering the model from convergence. Taking ArcFace and MV-Arc-Softmax as examples, ArcFace introduces a fixed margin $T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m)$ from the perspective of the positive cosine similarity. MV-Arc-Softmax introduces an additional margin from the perspective of negative cosine similarity for hard samples. As illustrated in Fig. 8.3, the decision condition of ArcFace shifts from $\cos\theta_{y_i} = \cos\theta_j$ (i.e., blue line) to $\cos(\theta_{y_i} + m) = \cos\theta_j$ (red line) for each sample, and the decision boundary of MV-Arc-Softmax becomes $\cos(\theta_{y_i} + m) = t\cos\theta_j + t - 1$ (green line). Conversely, the adaptive curriculum learning loss, the first attempt to introduce adaptive curriculum learning into deep face recognition, adjusts the weights of hard samples in different training stages. The decision condition becomes $\cos(\theta_{y_i} + m) = (t + \cos\theta_j)\cos\theta_j$ (purple line). During training, the decision boundary for hard samples changes from one purple line (**early stage**) to another (**later stage**), which emphasizes easy samples first and hard samples later.

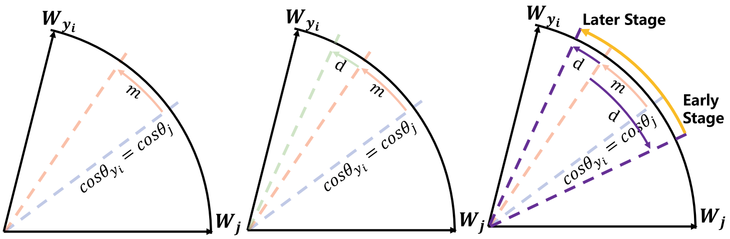

W_{y_i}
m
$\cos\theta_{y_i} = \cos\theta_j$
W_j
W_{y_i}
d
m
$\cos\theta_{y_i} = \cos\theta_j$
W_j
W_{y_i}
Later Stage
d
m
d
$\cos\theta_{y_i} = \cos\theta_j$
Early Stage
W_j
W_j

Fig. 8.3 **Blue line**, **red line**, **green line** and **purple line** denote the decision boundary of Softmax, ArcFace, MV-Arc-Softmax, and CurricularFace, respectively. m denotes the angular margin added by ArcFace. d denotes the additional margin of MV-Arc-Softmax and CurricularFace. In MV-Arc-Softmax, $d = (t-1)\cos\theta_j + t - 1$. In CurricularFace, $d = (t + \cos\theta_j - 1)\cos\theta_j$

8.6.1.1 Loss Function Formulation

The formulation of this loss function is contained in the general form 8.3, where positive and negative cosine similarity functions are defined as follows:

$$T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m), \tag{8.7}$$

$$N(t, \cos_{\theta_j}) = \begin{cases} \cos\theta_j, & T(\cos\theta_{y_i}) - \cos\theta_j \geq 0 \\ \cos\theta_j(t + \cos\theta_j), & T(\cos\theta_{y_i}) - \cos\theta_j < 0. \end{cases} \tag{8.8}$$

It should be noted that the positive cosine similarity can adopt any margin-based loss functions and ArcFace is adopted as an example. As shown in Fig. 8.4, the modulation coefficient $I(t, \theta_j)$ of hard sample negative cosine similarity depends on both the values of t and θ_j. At the early training stage, learning from easy samples is beneficial for model convergence. Thus, t should be close to zero and $I(\cdot) = t + \cos\theta_j$ is smaller than 1. Moreover, the weights of hard samples are reduced and easy samples are emphasized relatively. As training goes on, the model gradually focuses on the hard samples, i.e., the value of t should increase and $I(\cdot)$ is larger than 1. Therefore, the hard samples are emphasized with larger weights. Moreover, within the same training stage, $I(\cdot)$ is monotonically decreasing with θ_j so that a harder sample can be assigned with a larger coefficient according to its difficulty. The

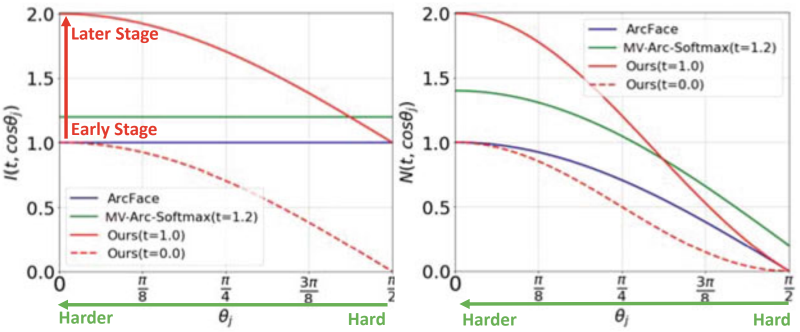


Fig. 8.4 Different training strategies for modulating negative cosine similarities of hard samples (i.e., the misclassified samples) in ArcFace [9], MV-Arc-Softmax [74], and CurricularFace. **Left**: The modulation coefficients $I(t, \cos\theta_j)$ for negative cosine similarities of hard samples in different methods, where t is an adaptively estimated parameter and θ_j denotes the angle between the hard sample and the non-ground truth j-class center. **Right**: The corresponding hard samples' negative cosine similarities $N(t, \cos\theta_j) = I(t, \cos\theta_j)\cos\theta_j + c$ after modulation, where c indicates a constant. On one hand, during early training stages (e.g., t is close to 0), the hard sample's negative cosine similarities are usually reduced and this leads to a smaller loss for the hard sample than the original one. Therefore, easier samples are relatively emphasized; during later training stages (e.g., t is close to 1), the hard sample's negative cosine similarities are enhanced and thus leads to larger hard sample loss. On the other hand, in the same training stage, the hard samples' negative cosine similarities are modulated with $\cos\theta_j$. Specifically, **the smaller the angle θ_j is, the larger the modulation coefficient should be**

value of the parameter t is *automatically* estimated, otherwise it may require lots of effort for manual tuning.

8.6.1.2 Optimization

CurricularFace can be easily optimized by the conventional stochastic gradient descent. Assuming that x_i denotes the deep feature of i-th sample belonging to the y_i class, the function input is the logit f_j, where j denotes the j-th class. In the forwarding process, when $j = y_i$, it is the same as the ArcFace, i.e., $f_j = sT(\cos\theta_{y_i})$, $T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m)$. When $j \neq y_i$, it has two cases. If x_i is an easy sample, it is the same as the original softmax, i.e., $f_j = s\cos\theta_j$. Otherwise, it will be modulated as $f_j = sN(t, \cos\theta_j)$, where $N(t, \cos\theta_j) = (t + \cos\theta_j)\cos\theta_j$. In the backward propagation process, the gradients w.r.t. x_i and W_j can also be divided into three cases and computed as follows:

$$
\begin{aligned}
\frac{\partial L}{\partial x_i} &= \begin{cases} \frac{\partial L}{\partial f_{y_i}}(s\frac{\sin(\theta_{y_i}+m)}{\sin\theta_{y_i}})W_{y_i}, & j = y_i \\ \frac{\partial L}{\partial f_j} sW_j, & j \neq y_i, \text{ easy} \\ \frac{\partial L}{\partial f_j} s(2\cos\theta_j + t)W_j & j \neq y_i, \text{ hard} \end{cases} \\
\frac{\partial L}{\partial W_j} &= \begin{cases} \frac{\partial L}{\partial f_{y_i}}(s\frac{\sin(\theta_{y_i}+m)}{\sin\theta_{y_i}})x_i, & j = y_i \\ \frac{\partial L}{\partial f_j} sx_i, & j \neq y_i, \text{ easy} \\ \frac{\partial L}{\partial f_j} s(2\cos\theta_j + t)x_i & j \neq y_i, \text{ hard} \end{cases}
\end{aligned}
\tag{8.9}
$$

Based on the above formulations, the gradient modulation coefficients of hard samples are determined by $M(\cdot) = 2\cos\theta_j + t$, which consists of two parts, the negative cosine similarity $\cos\theta_j$ and the value of t. As shown in Fig. 8.5, on the one hand, the coefficients increase with the adaptive estimation of t (described in the next subsection) to emphasize hard samples. On the other hand, these coefficients are assigned different importance according to their corresponding difficulty ($\cos\theta_j$). Therefore, the values of M in Fig. 8.5 are plotted as a range at each training iteration. However, the coefficients are fixed to be 1 and a constant t in ArcFace and MV-Arc-Softmax, respectively.

8.6.1.3 Adaptive Estimation of *t*

It is critical to determine appropriate values of t in different training stages. Ideally, the value of t can indicate the model training stages. Empirically, the *average* of positive cosine similarities is a good indicator. However, mini-batch statistic-based methods usually face an issue: when much extreme data is sampled in one mini-batch, the statistics can be vastly noisy and the estimation will be unstable. Exponential Moving Average (EMA) is a common solution to address this issue [35]. Specifically, let $r^{(k)}$ be the *average of the positive cosine similarities* of the k-th batch and be formulated as $r^{(k)} = \sum_i \cos\theta_{y_i}$, the value of t can be defined as follows:

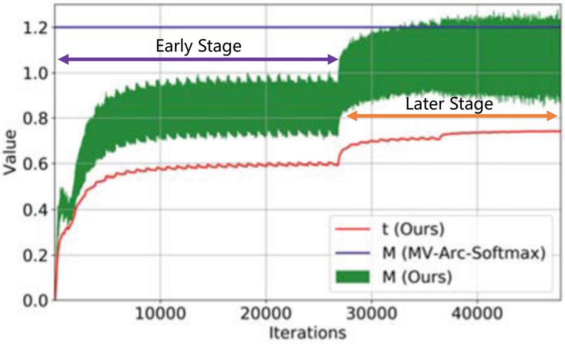


Fig. 8.5 The adaptive parameter t (red line) and gradient modulation coefficients M of CurricularFace (green area) and MV-Arc-Softmax (blue line) in training. Since the number of mined hard samples reduces as training progresses, the green area, *i.e.*, the range of M values, is relatively smooth in the early stage and exhibits burrs in later stage

Algorithm 8.1: CurricularFace

```
Input: The deep feature of $i$-th sample $x_i$ with its label $y_i$, last fully connected layer
       parameters $W$, cosine similarity $\cos\theta_j$ of two vectors, embedding network parameters
       $\Theta$, learning rate $\lambda$, and margin $m$
iteration number $k \leftarrow 0$, parameter $t \leftarrow 0$, $m \leftarrow 0.5$;
while not converged do
    if $\cos(\theta_{y_i} + m) \geq \cos\theta_j$ then
        $N(t, \cos\theta_j) = \cos\theta_j$;
    else
        $N(t, \cos\theta_j) = (t^{(k)} + \cos\theta_j)\cos\theta_j$ ;
    end
    $T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m)$;
    Compute the loss $\mathcal{L}$ by Eq. 8.11;
    Compute the gradients of $x_i$ and $W_j$ by Eq. 8.9;
    Update the parameters $W$ and $\Theta$ by: $W^{(k+1)} = W^{(k)} - \lambda^{(k)} \frac{\partial L}{\partial W}$,
    $\Theta^{(k+1)} = \Theta^{(k)} - \lambda^{(k)} \frac{\partial L}{\partial x_i} \frac{\partial x_i}{\partial \Theta^{(k)}}$;
    $k \leftarrow k + 1$;
    Update the parameter $t$ by Eq. 8.10;
end
Output: $W$, $\Theta$.
```

$$t^{(k)} = \alpha r^{(k)} + (1 - \alpha) t^{(k-1)}, \tag{8.10}$$

where $t^0 = 0$, α is the momentum parameter and set to 0.99. With the EMA, the hyperparameter tuning is avoided and the modulation coefficients of hard sample negative cosine similarities $I(\cdot)$ can be adaptive to the current training stage. To sum up, the loss function of CurricularFace is formulated as follows:

$$\mathcal{L} = -\log \frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{j=1, j\neq y_i}^{n} e^{sN(t^{(k)}, \cos\theta_j)}}, \tag{8.11}$$

where $N(t^{(k)}, \cos\theta_j)$ is defined in Eq. 8.8. The entire training process is summarized in Algorithm 8.1.

8.6.2 Distribution Distillation Loss (DDL)

Face images with significant variations are usually far away from the easy ones in the feature space and are much more challenging to tackle. This section refers to such samples as hard samples. DDL is a loss function proposed to narrow the performance gap between the easy and hard samples. It is generic and can be applied to diverse variations to improve face recognition in hard samples by leveraging the best of both the variation-specific and generic methods. Specifically, it first adopts current SotA face classifiers as the baseline (e.g., Arcface) to construct the initial similarity distributions between teacher and student according to the difficulties of samples, respectively, and then directly optimizes the similarity distributions to improve the performance on hard samples. Figure 8.6 illustrates the framework of DDL. The training set is first separated into two parts, *i.e.*, $\mathcal{E}$ for easy samples and $\mathcal{H}$ for hard samples to form the teacher and student distributions, respectively. To ensure a good teacher distribution, the state-of-the-art face recognition model [9] is used as the initialization. The extracted features are subsequently used to construct the positive and negative pairs (Sect. 8.6.2.1), which are further utilized to estimate the similarity distributions (Sect. 8.6.2.2). Finally, based on the similarity distributions, the DDL is utilized for training the classifier (Sect. 8.6.2.3).

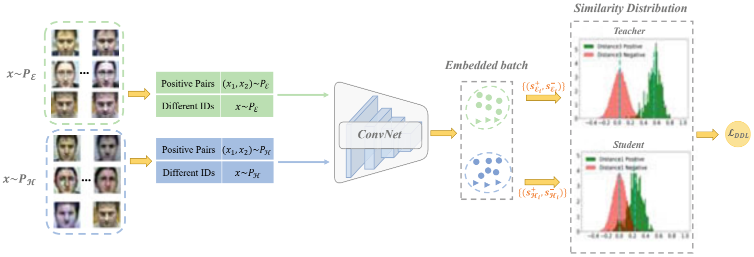


Fig. 8.6 **Illustration of DDL**. The b positive pairs (i.e., $2b$ samples) and b samples with different identities are sampled for both the teacher $P_{\mathcal{E}}$ and student $P_{\mathcal{H}}$ distributions. $\{(s^+_{\mathcal{E}_i}, s^-_{\mathcal{E}_i})|i = 1, ..., b\}$ indicates the b positive and negative pairs from $P_{\mathcal{E}}$ respectively to estimate the teacher distribution. $\{(s^+_{\mathcal{H}_i}, s^-_{\mathcal{H}_i})|i = 1, ..., b\}$ also indicates b positive and negative pairs from $P_{\mathcal{H}}$ to estimate the student distribution

8.6.2.1 Sampling Strategy from $P_{\mathcal{E}}$ and $P_{\mathcal{H}}$

Given two types of input data from both $P_{\mathcal{E}}$ and $P_{\mathcal{H}}$, each mini-batch consists of four parts, two kinds of positive pairs (i.e., $(x_1, x_2) \sim P_{\mathcal{E}}$ and $(x_1, x_2) \sim P_{\mathcal{H}}$), and two kinds of samples with different identities (i.e., $x \sim P_{\mathcal{E}}$ and $x \sim P_{\mathcal{H}}$).

Positive Pairs. The positive pairs are constructed offline in advance, and each pair consists of two samples with the same identity. As shown in Fig. 8.6, samples of each positive pair are arranged in order. After embedding data into a high-dimensional feature space by a deep network $\mathcal{F}$, the similarity of a positive pair s^+ can be obtained as follows:

$$s_i^+ = < \mathcal{F}(x_{pos_{i1}}), \mathcal{F}(x_{pos_{i2}}) >, i = 1, ..., b \tag{8.12}$$

where $x_{pos_{i1}}, x_{pos_{i2}}$ are the samples of one positive pair. Note that positive pairs with similarity less than 0 are usually outliers, which are deleted as a practical setting since the main goal is not to specifically handle noise.

Negative Pairs. Different from the positive pairs, negative pairs are constructed online from the samples with different identities via hard-negative mining. To be specific, the negative pairs with the largest similarities are selected and the similarity of a negative pair s^- is defined as:

$$s_i^- = \max_j \left(\{s_{ij}^- = < \mathcal{F}\left(x_{neg_i}\right), \mathcal{F}(x_{neg_j}) > | j = 1, ..., b\} \right), \tag{8.13}$$

where x_{neg_i}, x_{neg_j} are from different subjects. Once the similarities of positive and negative pairs are constructed, the corresponding distributions can be estimated, which is described in the next subsection.

8.6.2.2 Similarity Distribution Estimation

The process of similarity distribution estimation is similar to [68], which is performed in a simple and piece-wise differentiable manner using 1D histograms with soft assignment. Specifically, two samples x_i, x_j from the same person form a positive pair, and the corresponding label is denoted as $m_{ij} = +1$. In contrast, two samples from different persons form a negative pair, and the label is denoted as $m_{ij} = -1$. Then, two sample sets $\mathcal{S}^+ = \{s^+ = \langle \mathcal{F}(x_i), \mathcal{F}(x_j) \rangle | m_{ij} = +1\}$ and $\mathcal{S}^- = \{s^- = \langle \mathcal{F}(x_i), \mathcal{F}(x_j) \rangle | m_{ij} = -1\}$ corresponding to the similarities of positive and negative pairs are obtained, respectively. Let p^+ and p^- denote the two probability distributions of $\mathcal{S}^+$ and $\mathcal{S}^-$, respectively. As in cosine distance-based methods [9], the similarity of each pair is bounded to $[-1, 1]$, and this type of one-dimensional distribution can be estimated by fitting simple histograms with uniformly spaced bins. Assuming R-dimensional histograms $H^+$and H^-, with the nodes $t_1 = -1, t_2, \cdots, t_R = 1$ uniformly filling $[-1, 1]$ with the step $\triangle = \frac{2}{R-1}$, the value h_r^+ of the histogram H^+ at each bin is as follows:

$$h_r^+ = \frac{1}{|S^+|} \sum_{(i,j):m_{ij=+1}} \delta_{i,j,r}, \tag{8.14}$$

where (i, j) spans all the positive pairs. The weights $\delta_{i,j,r}$ are chosen by an exponential function as:

$$\delta_{i,j,r} = exp(-\gamma(s_{ij} - t_r)^2), \tag{8.15}$$

where γ denotes the spread parameter of Gaussian kernel function, and t_r denotes the rth node of histograms. The estimation of H^- proceeds analogously.

8.6.2.3 Loss Function Formulation

To minimize the performance disparity between easy and hard samples, two loss terms are employed. These terms constrain the similarity distribution of the hard samples, also known as the student distribution, to closely resemble the similarity distribution of the easy samples, referred to as the teacher distribution.

KL Divergence Loss. The teacher distribution consists of two similarity distributions of both positive and negative pairs, denoted as P^+ and P^-, respectively. Similarly, the student distribution also consists of two similarity distributions, denoted as Q^+ and Q^-. The KL divergence is adopted to constrain the similarity between the student and teacher distributions, which is defined as follows:

$$\begin{aligned}\mathcal{L}_{KL} &= \lambda_1 \mathbb{D}_{KL}(P^+||Q^+) + \lambda_2 \mathbb{D}_{KL}(P^-||Q^-)\\ &= \underbrace{\lambda_1 \sum_s P^+(s)\log\frac{P^+(s)}{Q^+(s)}}_{KL\,loss\,on\,pos.\,pairs} + \underbrace{\lambda_2 \sum_s P^-(s)\log\frac{P^-(s)}{Q^-(s)}}_{KL\,loss\,on\,neg.\,pairs},\end{aligned} \tag{8.16}$$

where λ_1, λ_2 are the weight parameters.

Order Loss. Only using KL loss does not guarantee good performance. In fact, the teacher distribution may choose to approach the student distribution, leading to more confusion regions between the distributions of positive and negative pairs. This is the opposite of the objective (see Fig. 8.7). To address this problem, a simple yet effective term named *order loss* is proposed to minimize the distances between the expectations of similarity distributions

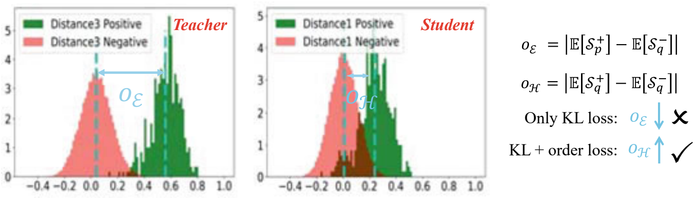


Fig. 8.7 Illustration of the effects of the order loss. Similarity distributions are constructed by Arcface on SCface [15], in which 2 kinds of order distances are formed from both the teacher and student distributions according to Eq. 8.17

from the negative and positive pairs to control the overlap. The order loss can be formulated as follows:

$$\mathcal{L}_{order} = -\lambda_3 \sum_{(i,j)\in(p,q)} (\mathbb{E}[\mathcal{S}_i^+] - \mathbb{E}[\mathcal{S}_j^-]), \tag{8.17}$$

where $\mathcal{S}_p^+$ and S_p^- denote the similarities of positive and negative pairs of the teacher distribution; $\mathcal{S}_q^+$ and S_q^- denote the similarities of positive and negative pairs of the student distribution; and λ_3 is the weight parameter.

8.6.3 Sphere Confidence Face (SCF)

Uncertainty estimation of face recognition features can reduce the impact of low-quality pictures such as exaggerated expressions, large gestures, and blurring on the accuracy of face recognition. PFE uses Gaussian distribution to model face features for estimating the uncertainty. Unlike PFE defined in the Euclidean space, Sphere Confidence Face (SCF) for face confidence learning in an r-radius spherical space captures the most likely feature representation and obtains its local concentration value on spheres.

8.6.3.1 r-Radius *von Mises-Fisher* Distribution

Recent advances in face recognition (e.g., ArcFace and CosFace) suggest that spherical space is more suitable than Euclidean space for feature learning. The SCF adopts and extends this concept to probabilistic confidence modeling. Specifically, given a face image $\mathbf{x}$ from input space $\mathcal{X}$, the conditional latent distribution is modeled as a *von Mises-Fisher* (vMF) distribution [14] defined on a d-dimensional unit sphere $\mathbb{S}^{d-1} \subset \mathbb{R}^d$,

$$p(\mathbf{z}'|\mathbf{x}) = C_d(\kappa_{\mathbf{x}}) \exp\left(\kappa_{\mathbf{x}} \boldsymbol{\mu}_{\mathbf{x}}^T \mathbf{z}'\right), \tag{8.18}$$

$$C_d(\kappa_{\mathbf{x}}) = \frac{\kappa_{\mathbf{x}}^{d/2-1}}{(2\pi)^{d/2} \mathcal{I}_{d/2-1}(\kappa_{\mathbf{x}})}, \tag{8.19}$$

where $\mathbf{z}', \boldsymbol{\mu}_{\mathbf{x}} \in \mathbb{S}^{d-1}$, $\kappa_{\mathbf{x}} \geq 0$ (subscripts indicate statistical dependencies on $\mathbf{x}$) and $\mathcal{I}_\alpha$ denotes the modified Bessel function of the first kind at order α. $\boldsymbol{\mu}_{\mathbf{x}}$ and $\kappa_{\mathbf{x}}$ are the mean direction and concentration parameters, respectively. The greater the value of $\kappa_{\mathbf{x}}$, the higher the concentration around the mean direction $\boldsymbol{\mu}_{\mathbf{x}}$. The distribution is unimodal for $\kappa_{\mathbf{x}} > 0$, and it degenerates to uniform on the sphere for $\kappa_{\mathbf{x}} = 0$.

Then, it is further extended to r-radius vMF, which is defined over the support of an r-radius sphere $r\mathbb{S}^{d-1}$. Formally, for any $\mathbf{z} \in r\mathbb{S}^{d-1}$, there exists a one-to-one correspondence between $\mathbf{z}'$ and $\mathbf{z}$ such that $\mathbf{z} = r\mathbf{z}'$. Then, the r-radius vMF density (denoted as r-vMF($\boldsymbol{\mu}_{\mathbf{x}}, \kappa_{\mathbf{x}}$)) can be obtained using the change-of-variable formula:

$$p(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}'|\mathbf{x}) \left| \det \left(\frac{\partial \mathbf{z}'}{\partial \mathbf{z}} \right) \right| = \frac{C_d(\kappa_{\mathbf{x}})}{r^d} \exp \left(\frac{\kappa_{\mathbf{x}}}{r} \boldsymbol{\mu}_{\mathbf{x}}^T \mathbf{z} \right). \tag{8.20}$$

8.6.3.2 Loss Function Formulation

State-of-the-art deterministic embeddings defined in spherical spaces, such as ArcFace and CosFace, are essentially Dirac delta $p(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - f(\mathbf{x}))$, where $f : \mathcal{X} \mapsto r\mathbb{S}^{d-1}$ is a deterministic mapping. According to the following definition, the Dirac delta can be extended into spherical space.

Definition 8.1 *(Spherical Dirac delta).* A probability density $p(\mathbf{z})$ on the support of an r-radius sphere $r\mathbb{S}^{d-1}$ is spherical Dirac delta $\delta(\mathbf{z} - \mathbf{z}_0)$ (for some fixed $\mathbf{z}_0 \in r\mathbb{S}^{d-1}$), if and only if the following three conditions hold:

$$\delta(\mathbf{z} - \mathbf{z}_0) = \begin{cases} 0 & \mathbf{z} \neq \mathbf{z}_0 \\ \infty & \mathbf{z} = \mathbf{z}_0 \end{cases}; \quad \int_{r\mathbb{S}^{d-1}} \delta(\mathbf{z} - \mathbf{z}_0) d\mathbf{z} = 1;$$

$$\int_{r\mathbb{S}^{d-1}} \delta(\mathbf{z} - \mathbf{z}_0) \phi(\mathbf{z}) d\mathbf{z} = \phi(\mathbf{z}_0).$$

By utilizing the definition, a new training objective can be established. Deep face recognition classifiers typically map the spherical feature space $r\mathbb{S}^{d-1}$ to a label space $\mathbb{L}$ via a linear mapping parameterized by a matrix $\mathbf{W} \in \mathbb{R}^{n \times d}$, where n is the number of face identities. Let $\mathbf{w}_{\mathbf{x}\in c}$ denote the classifier weight given a face image $\mathbf{x}$ belonging to class c, which can be easily obtained from any given *pre-trained* model by extracting the cth row of $\mathbf{W}$. By virtue of these classifier weights, a conventional deterministic embedding as a spherical Dirac delta can act as a desired latent prior over the sphere, and the training objective can be defined as the KL divergence between the spherical Dirac delta and the model distribution $p(\mathbf{z}|\mathbf{x})$. Specifically, the objective is to minimize $\mathbb{E}_{\mathbf{x}}[D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))]$, where $q(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - \mathbf{w}_{\mathbf{x}\in c})$ and $p(\mathbf{z}|\mathbf{x})$ is modeled as r-radius vMF parameterized by $\mu(\mathbf{x})$ and $\kappa(\mathbf{x})$ ($||\mu(\mathbf{x})||_2 = 1$ and $\kappa(\mathbf{x}) > 0$; here dependencies on $\mathbf{x}$ are shown in functional forms in place of subscripts. The formulation is shown as follows:

$$\min_p \mathbb{E}_{\mathbf{x}}\big[D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))\big] = \mathbb{E}_{\mathbf{x}}\left[-\left(\int_{r\mathbb{S}^{d-1}} q(\mathbf{z}|\mathbf{x}) \log p(\mathbf{z}|\mathbf{x}) d\mathbf{z}\right) - \mathbb{H}_{q(\mathbf{z}|\mathbf{x})}(\mathbf{z})\right]. \tag{8.21}$$

Note that minimizing Eq. 8.21 with regard to p is equivalent to minimizing the cross-entropy between q and p with regard to $\boldsymbol{\mu}$ and κ conditional on $\mathbf{x}$. Therefore, the optimization objective can be defined as the minimization of $\mathbb{E}_{\mathbf{x}}[\mathcal{L}(\boldsymbol{\mu}(\mathbf{x}), \kappa(\mathbf{x}))]$ over all $\boldsymbol{\mu}$ and κ, where

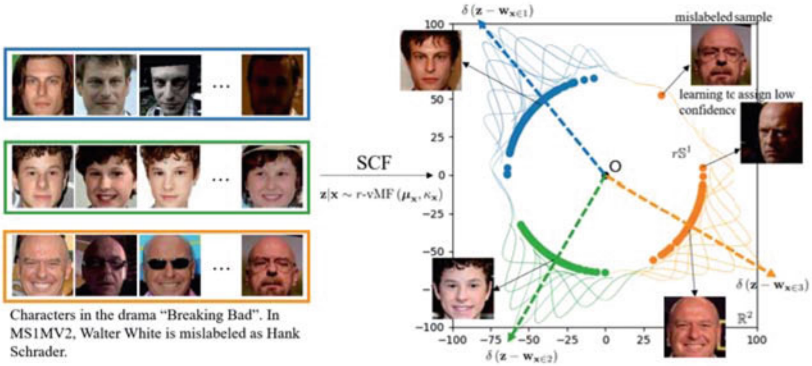

...
Characters in the drama "Breaking Bad". In MS1MV2, Walter White is mislabeled as Hank Schrader.
SCF
$\mathbf{z}|\mathbf{x} \sim r\text{-vMF}(\mu_{\mathbf{x}}, \kappa_{\mathbf{x}})$
$\delta(\mathbf{z} - \mathbf{w}_{\mathbf{x}\in 1})$
mislabeled sample
learning tc assign low confidence
O
$r\mathbb{S}^1$
$\delta(\mathbf{z} - \mathbf{w}_{\mathbf{x}\in 3})$
$\mathbb{R}^2$
$\delta(\mathbf{z} - \mathbf{w}_{\mathbf{x}\in 2})$
100
50
25
0
−25
−50
−100
−100
−75
−50
−25
0
25
100

Fig. 8.8 A 2D toy example of training SCF. SCF learns a mapping from the input space $\mathcal{X}$ to an r-radius spherical space, $r\mathbb{S}^1 \subset \mathbb{R}^2$. The latent code of each image is assumed to obey a conditional distribution, i.e., $\mathbf{z}|\mathbf{x} \sim r\text{-vMF}\left(\boldsymbol{\mu}_{\mathbf{x}}, \kappa_{\mathbf{x}}\right)$, where $\boldsymbol{\mu}_{\mathbf{x}}$ and $\kappa_{\mathbf{x}}$ are parameterized by neural networks. Each identity has a class template $\mathbf{w}_{\mathbf{x}\in c}$ that induces a spherical Dirac delta, for $c = 1, 2, 3$. Optimization proceeds by minimizing $D_{\text{KL}}\left(\delta\left(\mathbf{z} - \mathbf{w}_{\mathbf{x}\in c}\right) \| r\text{-vMF}\left(\boldsymbol{\mu}_{\mathbf{x}}, \kappa_{\mathbf{x}}\right)\right)$. Experiments are carried out using a subset of MS1MV2 containing three identities. There are *mislabeled samples* for the third identity which hamper training otherwise. SCF learns to assign low confidence to such samples in an adaptive manner

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\mu}(\mathbf{x}), \kappa(\mathbf{x})) &= -\int_{r\mathbb{S}^{d-1}} q(\mathbf{z}|\mathbf{x}) \log p(\mathbf{z}|\mathbf{x}) d\mathbf{z} \\
&= -\frac{\kappa(\mathbf{x})}{r} \mu(\mathbf{x})^T \mathbf{w}_{\mathbf{x}\in c} - \left(\frac{d}{2} - 1\right) \log \kappa(\mathbf{x}) \\
&\quad + \log(\mathcal{I}_{d/2-1}(\kappa(\mathbf{x}))) + \frac{d}{2} \log 2\pi r^2.
\end{aligned} \tag{8.22}
$$

Figure 8.8 showcases a 2D toy example of training SCF. The detailed explanations can be found in the figure caption.

8.6.3.3 Theoretical Perspective

In contrast to PFE which maximizes the expectation of the mutual likelihood score of genuine pairs, the SCF framework minimizes the KL divergence between spherical Dirac delta and r-radius vMF by virtue of classifier weights. This is a reasonable choice that can be justified theoretically by Theorem 8.1. Intuitively, regularization to the spherical Dirac delta δ encourages the latents that are closer to their corresponding classifier weights to have larger concentration values (thus higher confidence); and vice versa (see Theorem 8.2).

Theorem 8.1 *An r-radius vMF density r-vMF$(\boldsymbol{\mu}, \kappa)$ tends to a spherical Dirac delta $\delta(\mathbf{z} - r\boldsymbol{\mu})$, as $\kappa \to \infty$.*

Proof By leveraging the asymptotic expansion of the modified Bessel function of the first kind: for any complex number z with large $|z|$ and $|\arg z| < \pi/2$,

$$\mathcal{I}_\alpha(z) \sim \frac{e^z}{\sqrt{2\pi z}} \left(1 + \sum_{N=1}^{\infty} \frac{(-1)^N}{N!(8z)^N} \prod_{n=1}^{N} \left(4\alpha^2 - (2n-1)^2 \right) \right), \tag{8.23}$$

when $\kappa \to \infty$, $\mathcal{I}_{d/2-1}(\kappa) \sim e^\kappa / \sqrt{2\pi\kappa}$. Then, these theoretical results (Theorem 8.1) can be readily shown with this fact given. □

Theorem 8.2 *The quantity* $\cos\langle \mu(\mathbf{x}), \mathbf{w}_{\mathbf{x}\in c} \rangle$ *is a strictly increasing function of* κ^* *in the interval* $(0, +\infty)$*, where* $\kappa^* = \arg\min_\kappa \mathcal{L}(\mu, \kappa)$.

Proof Taking partial derivative of the loss function $\mathcal{L}(\mu, \kappa)$ with regard to κ and setting it to zero yields the equality

$$\frac{\partial \mathcal{L}}{\partial \kappa} := 0 \Longrightarrow \cos\langle \mu(\mathbf{x}), \mathbf{w}_{\mathbf{x}\in c} \rangle = \frac{\mathcal{I}_{d/2}(\kappa^*)}{\mathcal{I}_{d/2-1}(\kappa^*)}, \tag{8.24}$$

where $\kappa^* = \arg\min_\kappa \mathcal{L}(\mu, \kappa)$. Then, for any $u > v \geq 0$ and $\kappa > 0$, define $F_{uv}(\kappa) := \mathcal{I}_u(\kappa)/\mathcal{I}_v(\kappa)$. According to [29], the following properties of $F_{uv}(\kappa)$ can be obtained:

$$\lim_{\kappa \to 0} F_{uv}(\kappa) = 0, \quad \lim_{\kappa \to \infty} F_{uv}(\kappa) = 1. \tag{8.25}$$

Furthermore, $0 < F_{uv}(\kappa) < 1$ and its derivative is always positive in the interval $(0, +\infty)$, i.e., $F'_{uv}(\kappa) > 0$, which concludes the proof. □

Theorem 8.2 suggests that the closer $\boldsymbol{\mu}$ gets to $\mathbf{w}_{\mathbf{x}\in c}$ the higher the value of κ^* is. For models trained with softmax-based loss, the smaller the angle between $\boldsymbol{\mu}$ and its class center $\mathbf{w}_{\mathbf{x}\in c}$ is, the more confident prediction the model has. Given only one face image during the testing phase, predicting its class center for the unknown subject is an ill-posed problem. This framework circumvents this difficulty by predicting its *kappa* confidence, a mathematical measure of how close the test face image is to its unknown class center.

8.6.3.4 Feature Pooling with Confidence

In cases where one subject has multiple face images (observations), it is desirable to obtain one single compact representation from multiple ones before performing face verification using cosine distance. Given two subjects A and B, each with multiple images $\{\mathbf{x}^{\cdot}_{(m)}\}$ ("·" can be either A or B), the proposed model predicts their statistics $\boldsymbol{\mu}^{\cdot}_{(m)}$ and $\boldsymbol{\kappa}^{\cdot}_{(m)}$. Theorem 8.2 suggests that the proposed framework allows a natural interpretation of κ^* to be a measure of confidence (the inverse of uncertainty). This leads to a principled feature pooling:

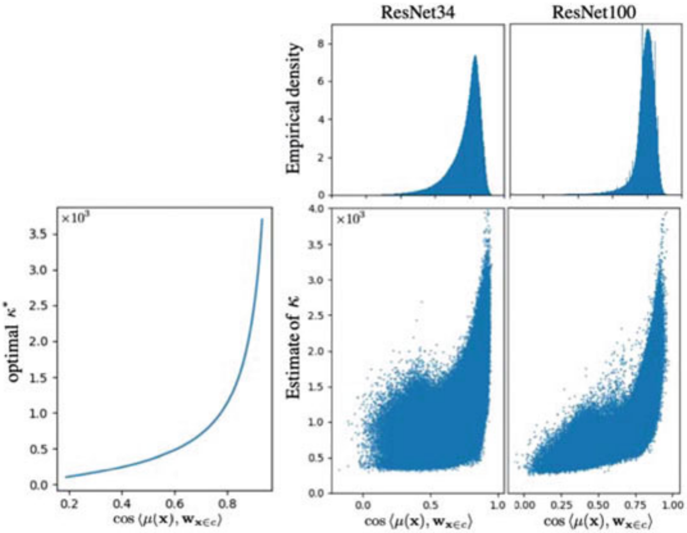


Fig. 8.9 Left: the function plot of the inverse of Eq. 8.24. Bottom right: the empirical correlation between cosine value $\cos\langle\mu(\mathbf{x}), \mathbf{w}_{\mathbf{x}\in c}\rangle$ and concentration value κ. Top right: marginalized empirical densities of cosine value on two backbones

$$\mathbf{z}^A = \frac{\sum_m \kappa^A_{(m)} \boldsymbol{\mu}^A_{(m)}}{\sum_m \kappa^A_{(m)}}, \mathbf{z}^B = \frac{\sum_m \kappa^B_{(m)} \boldsymbol{\mu}^B_{(m)}}{\sum_m \kappa^B_{(m)}}, \tag{8.26}$$

where $\mathbf{z}^A$ and $\mathbf{z}^B$ are pooled features for A and B, respectively. Then, cosine distance is utilized to measure the similarity, i.e., $\cos\langle\mathbf{z}^A, \mathbf{z}^B\rangle$. As illustrated in Fig. 8.9 (right), there is a strong correlation between the cosine value $\cos\langle\mu(\mathbf{x}), \mathbf{w}_{\mathbf{x}\in c}\rangle$ and the concentration parameter κ. The closer the angular distance between $\mu(\mathbf{x})$ and $\mathbf{w}_{\mathbf{x}\in c}$ is, the higher the concentration value (confidence) becomes. This corroborates Theorem 1, indicating that the confidence model indeed learns the latent distribution that is unimodal vMF for each single class and forms a mixture of vMFs overall, which confirms the hypothesis in SCF.

8.7 Conclusions

In this chapter, we provide a complete review of the critical factors for obtaining a discriminative face feature embedding, including loss functions, network structures, and commonly used large public training datasets. In addition, we briefly introduce specific topics related to deep face feature embedding. These include long tail learning, noise-robust learning, uncertainty learning, and cross-variation face recognition. Deep face recognition has dramatically improved SOTA's performance and fostered the development of successful real-world applications by leveraging large-scale annotated data and innovative deep learning techniques. To further improve deep facial feature embeddings, however, several remaining issues should be resolved. For example, it is vital to evaluate and boost the trustworthiness of the recognition system with the wild application of the face recognition system. Thus, research on improving the fairness, interpretability, security, and privacy of face feature embedding is essential. Although a large amount of labeled data is already available, data from actual application scenes is still required for training to improve the recognition performance of the corresponding scenes. Labeling the face data of these scenes is a time-consuming and labor-intensive task. Thus, exploring effective ways to utilize these unlabeled data to enhance face recognition is a promising research direction.

References

1. Ahonen, T., Hadid, A., Pietikäinen, M.: Face description with local binary patterns: Application to face recognition. IEEE Trans. PAMI **28**(12) (2006)
2. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Proc. of the ECCV (2004)
3. An, X., et al.: Killing two birds with one stone: Efficient and robust training of face recognition CNNs by partial FC. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
4. Bansal, A., et al.: Umdfaces: An annotated face dataset for training deep networks. In: 2017 IEEE International Joint Conference on Biometrics (IJCB). IEEE (2017)
5. Bicego, M., Lagorio, A., Grosso, E., Tistarelli, M.: On the use of SIFT features for face authentication. In: Proc. of the CVPR (2006)
6. Cao, Q., et al.: Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2018). IEEE (2018)
7. Chang, J., et al.: Data uncertainty learning in face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
8. Chen, S., et al.: Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In: Chinese Conference on Biometric Recognition. Springer, Cham (2018)
9. Deng, J., et al.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
10. Deng, J., et al.: Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In: European Conference on Computer Vision. Springer, Cham (2020)
11. Deng, J., et al.: Variational prototype learning for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
12. Ding, C., Dacheng, T.: Robust face recognition via multimodal deep face representation. IEEE Trans. Multimed. **17**(11), 2049–2058 (2015)
13. Ding, X., et al.: Repvgg: Making vgg-style convnets great again. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
14. Fisher, N.I., Toby, L., Brian, J.J.E.: Statistical Analysis of Spherical Data. Cambridge University Press (1993)
15. Grgic, M., Kresimir, D., Sonja, G.: SCface-surveillance cameras face database. Multimedia Tools Appl. **51**(3), 863–879 (2011)
16. Guo, Y., et al.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: European Conference on Computer Vision. Springer, Cham (2016)

17. Hasnat, A., et al.: Deepvisage: Making face recognition simple yet with powerful generalization skills. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2017)
18. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
19. Hoo, S.C., Adeshina, S.O., Haidi, I.: Survey on loss function for face recognition. In: Proceedings of the 11th International Conference on Robotics, Vision, Signal Processing and Power Applications. Springer, Singapore (2022)
20. Howard, A., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
21. Howard, A.G., et al.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. ArXiv preprint arXiv:1704.04861 (2017)
22. Hu, J., Li, S., Gang, S.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
23. Hu, J., Jiwen, L., Yap-Peng, T.: Discriminative deep metric learning for face verification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
24. Hu, W., et al.: Noise-tolerant paradigm for training face recognition CNNs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
25. Huang, G., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
26. Huang, Y., et al.: Curricularface: adaptive curriculum learning loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
27. Huang, Y., et al.: Improving face recognition from hard samples via distribution distillation loss. In: European Conference on Computer Vision. Springer, Cham (2020)
28. Iandola, F.N., et al.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. ArXiv preprint arXiv:1602.07360 (2016)
29. Jones, A.L.: An extension of an inequality involving modified Bessel functions. J. Math. Phys. **47**(1–4), 220–221 (1968)
30. Kanade, T.: Picture Processing System by Computer Complex and Recognition of Human Faces. Doctoral Dissertation, Kyoto University (1973)
31. Kang, B.-N., et al.: Attentional feature-pair relation networks for accurate face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
32. Kim, M., Anil, K.J., Xiaoming, L.: AdaFace: Quality adaptive margin for face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
33. Kim, Y., et al.: Groupface: Learning latent groups and constructing group-based representations for face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
34. Krizhevsky, A., Ilya, S., Geoffrey, E.H.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
35. Li, B., Yu, L., Xiaogang, W.: Gradient harmonized single-stage detector. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 1st edn (2019)
36. Li, S., et al.: Spherical confidence learning for face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
37. Li, X., et al.: Airface: Lightweight and efficient model for face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
38. Lin, T.-Y., et al.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2017)

39. Liu, B., et al.: Fair loss: Margin-aware reinforcement learning for deep face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
40. Liu, H., et al.: Adaptiveface: Adaptive margin and sampling for face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
41. Liu, W., et al.: Large-margin softmax loss for convolutional neural networks. ArXiv preprint arXiv:1612.02295 (2016)
42. Liu, W., et al.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
43. Liu, Y., Hongyang, L., Xiaogang, W.: Rethinking feature discrimination and polymerization for large-scale recognition. ArXiv preprint arXiv:1710.00870 (2017)
44. Liu, Z., et al.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision (2015)
45. Meng, Q., et al.: Magface: A universal representation for face recognition and quality assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
46. Oktay, O., et al.: Attention u-net: Learning where to look for the pancreas. ArXiv preprint arXiv:1804.03999 (2018)
47. Parkhi, O.M., Andrea, V., Andrew, Z.: Deep Face Recognition (2015)
48. Ranjan, R., Carlos, D.C., Rama, C.: L2-constrained softmax loss for discriminative face verification. ArXiv preprint arXiv:1703.09507 (2017)
49. Sandler, M., et al.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
50. Sankaranarayanan, S., Azadeh, A., Rama, C.: Triplet similarity embedding for face verification. ArXiv preprint arXiv:1602.03418 (2016)
51. Sankaranarayanan, S., et al.: Triplet probabilistic embedding for face verification and clustering. In: 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS). IEEE (2016)
52. Schroff, F., Dmitry, K., James, P.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
53. Shekhar, S., Vishal, M.P., Rama, C.: Synthesis-based recognition of low resolution faces. In: 2011 International Joint Conference on Biometrics (IJCB). IEEE (2011)
54. Shi, Y., Anil, K.J.: Probabilistic face embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
55. Shi, Y., et al.: Towards universal representation learning for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
56. Shrivastava, A., Abhinav, G., Ross, G.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
57. Simonyan, K., Andrew, Z.: Very deep convolutional networks for large-scale image recognition. ArXiv preprint arXiv:1409.1556 (2014)
58. Su, Y., Shan, S., Chen, X., Gao, W.: Hierarchical ensemble of global and local classifiers for face recognition. IEEE Trans. Image Process. **18**(8) (2009)
59. Sun, Y., Xiaogang, W., Xiaoou, T.: Deep learning face representation from predicting 10,000 classes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
60. Sun, Y., Xiaogang, W., Xiaoou, T.: Hybrid deep learning for face verification. In: Proceedings of the IEEE International Conference on Computer Vision (2013)

61. Sun, Y., et al.: Deepid3: Face recognition with very deep neural networks. ArXiv preprint arXiv:1502.00873 (2015)
62. Sun, Y., et al.: Deep learning face representation by joint identification-verification. In: Advances in Neural Information Processing Systems 27 (2014)
63. Sun, Y., et al.: Circle loss: A unified perspective of pair similarity optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
64. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
65. Tai, Y., et al.: Face recognition with pose variations and misalignment via orthogonal procrustes regression. IEEE Trans. Image Process. **25**(6), 2673–2683 (2016)
66. Taigman, Y., et al.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
67. Tan, M., Quoc, L.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR (2019)
68. Ustinova, E., Victor, L.: Learning deep embeddings with histogram loss. In: Advances in Neural Information Processing Systems 29 (2016)
69. Wang, F., et al.: The devil of face recognition is in the noise. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
70. Wang, F., et al.: Additive margin softmax for face verification. IEEE Sig. Process. Lett. **25**(7), 926–930 (2018)
71. Wang, F., et al.: Normface: L2 hypersphere embedding for face verification. In: Proceedings of the 25th ACM International Conference on Multimedia (2017)
72. Wang, H., et al.: Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
73. Wang, X., et al.: Co-mining: Deep face recognition with noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
74. Wang, X., et al.: Mis-classified vector guided softmax loss for face recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 7th edn (2020)
75. Wei, X., et al.: Minimum margin loss for deep face recognition. Pattern Recognit. **97**, 107012 (2020)
76. Wen, Y., et al.: A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision. Springer, Cham (2016)
77. Wu, Y., et al.: Deep face recognition with center invariant loss. In: Proceedings of the on Thematic Workshops of ACM Multimedia 2017 (2017)
78. Yang, J., Adrian, B., Georgios, T.: Fan-face: a simple orthogonal improvement to deep face recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 7th edn (2020)
79. Yi, D., et al.: Learning face representation from scratch. ArXiv preprint arXiv:1411.7923 (2014)
80. Zhang, X., et al.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
81. Zhang, X., et al.: Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
82. Zhang, X., et al.: P2sgrad: Refined gradients for optimizing deep face models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
83. Zhang, X., et al.: Range loss for deep face recognition with long-tailed training data. In: Proceedings of the IEEE International Conference on Computer Vision (2017)

84. Zhang, Y., et al.: Adaptive label noise cleaning with meta-supervision for deep face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
85. Zhang, Y., et al.: Global-local gcn: Large-scale label noise cleansing for face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
86. Zheng, Y., Dipan, K.P., Marios, S.: Ring loss: Convex feature normalization for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
87. Zhu, Z., et al.: Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)

Video-Based Face Recognition

9

Anirudh Nanduri, Jingxiao Zheng, and Rama Chellappa

9.1 Introduction

Video-based face recognition is an active research topic because of a wide range of applications including visual surveillance, access control, video content analysis, etc. Compared to still face recognition, video-based face recognition is more challenging due to a much larger amount of data to be processed and significant intra/inter-class variations caused by motion blur, low video quality, occlusion, frequent scene changes, and unconstrained acquisition conditions.

To develop the next generation of unconstrained video-based face recognition systems, two datasets have been recently introduced, IARPA Benchmark B (IJB-B) [54] and IARPA Janus Surveillance Video Benchmark (IJB-S) [23], acquired under more challenging scenarios, compared to the Multiple Biometric Grand Challenge (MBGC) dataset [30] and the Face and Ocular Challenge Series (FOCS) dataset [32] which were collected in relatively controlled conditions. IJB-B and IJB-S datasets were captured in unconstrained settings and contain faces with much more intra/inter-class variations on pose, illumination, occlusion, video quality, scale, etc.

A. Nanduri (✉)
University of Maryland, College Park MD 20742, USA
e-mail: snanduri@umd.edu

J. Zheng
Waymo, Mountain View CA 94043, USA
e-mail: jingxiaozheng@waymo.com

R. Chellappa
Johns Hopkins University, Baltimore MD 21218, USA
e-mail: rchella4@jhu.edu

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_9

The IJB-B dataset is a template-based dataset that contains 1845 subjects with 11,754 images, 55,025 frames, and 7,011 videos where a template consists of a varying number of still images and video frames from different sources. These images and videos are totally unconstrained, with large variations in pose, illumination, image quality, etc. Samples from this dataset are shown in Fig. 9.1. In addition, the dataset comes with protocols for 1-to-1 template-based face verification, 1-to-N template-based open-set face identification, and 1-to-N open-set video face identification. For the video face identification protocol, the gallery is a set of still-image templates. The probe is a set of videos (e.g., news videos), each of which contains multiple shots with multiple people and one bounding box annotation to specify the subject of interest. Probes of videos are searched among galleries of still images. Since the videos are composed of multiple shots, it is challenging to detect and associate the faces for the subject of interest across shots due to large appearance changes. In addition, how to efficiently leverage information from multiple frames is another challenge, especially when the frames are noisy.

Similar to the IJB-B dataset, the IJB-S dataset is also an unconstrained video dataset focusing on real-world visual surveillance scenarios. It consists of 202 subjects from 1421 images and 398 surveillance videos, with 15,881,408 bounding box annotations. Samples of frames from IJB-S are shown in Fig. 9.2. Three open-set identification protocols accompany this dataset for surveillance video-based face recognition where each video in these protocols is captured from a static surveillance camera and contains single or multiple subjects: (1) in surveillance-to-single protocol, probes collected from surveillance videos are searched in galleries consisting of one single high-resolution still-image; (2) in surveillance-to-booking protocol, same probes are searched among galleries consisting of seven high-resolution

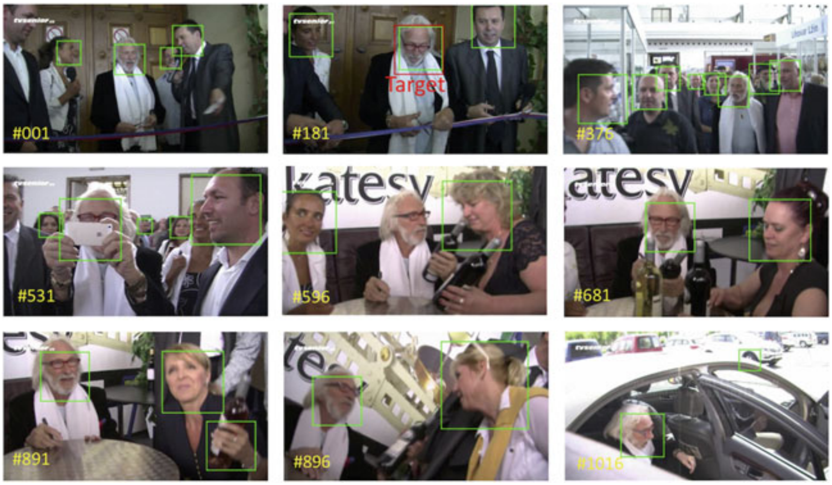


Fig. 9.1 Example frames of a multiple-shot probe video in the IJB-B dataset. The target annotation is in the red box and face detection results from the face detector are in green boxes

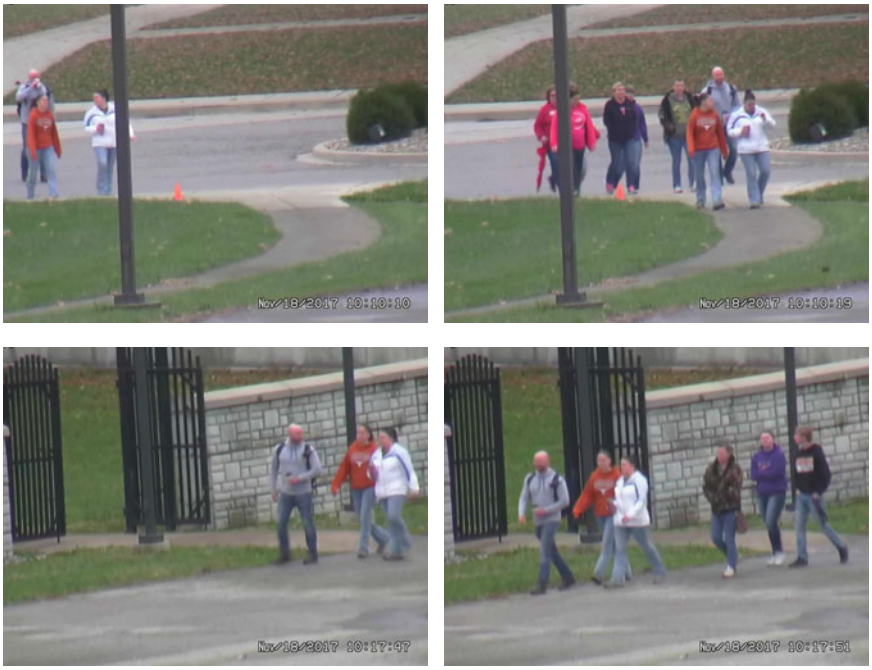


Fig. 9.2 Example frames of two single-shot probe videos in the IJB-S dataset

still face images covering frontal and profile poses. Probe templates in (1) and (2) should be detected and constructed by the recognition system itself; (3) in the most challenging surveillance-to-surveillance protocol, both gallery and probe templates are from videos, which implies that probe templates need to be compared with relatively low-quality gallery templates.

From these datasets, we summarize the four common challenges in video-based face recognition as follows:

1. For video-based face recognition, test data are from videos where each video contains tens of thousands of frames and each frame may have several faces. This makes the scalability of video-based face recognition a challenging problem. In order to make the face recognition system to be operationally effective, each component of the system should be fast, especially face detection, which is often the bottleneck in recognition.
2. Since faces are mostly from unconstrained videos, they have significant variations in pose, expression, illumination, blur, occlusion, and video quality. Thus, any face representations we design must be robust to these variations and to errors in face detection and association steps.

3. Faces with the same identity across different video frames need to be grouped by a reliable face association method. Face recognition performance will degrade if faces with different identities are grouped together. Videos in the IJB-B dataset are acquired from multiple shots involving scene and view changes, while most videos in IJB-S are low-quality remote surveillance videos. These conditions increase the difficulty of face association.
4. Since each video contains a different number of faces for each identity, the next challenge is how to efficiently aggregate a varying-length set of features from the same identity into a fixed-size or unified representation. Exploiting the correlation information in a set of faces generally results in better performance than using only a single face.

In this chapter, we mainly focus on the second and fourth challenges. After face association, video faces from the same identities are associated into sets and the correlation between samples in the same set can be leveraged to improve the face recognition performance. For video-based face recognition, a temporal deep learning model such as Recurrent Neural Network (RNN) can be applied to yield a fixed-size encoded face representation. However, large-scale labeled training data is needed to learn robust representations, which is very expensive to collect in the context of the video-based recognition problem. This is also true for the adaptive pooling method [28, 57] for image set-based face recognition problems. For IJB-B and IJB-S datasets, the lack of large-scale training data makes it challenging to train an RNN-based method. Also, RNN can only work on sequential data, while faces associated from videos are sometimes without a certain order. On the contrary, representative and discriminative models based on manifolds and subspaces have also received attention for image set-based face recognition [50, 52]. These methods model sets of image samples as manifolds or subspaces and use appropriate similarity metrics for set-based identification and verification. One of the main advantages of subspace-based methods is that different from the sample mean, the subspace representation encodes the correlation information between samples. In low-quality videos, faces have significant variations due to blur, extreme poses, and low resolution. Exploiting the correlation between samples by subspaces will help to learn a more robust representation to capture these variations. Also, a fixed-size representation is learned from an arbitrary number of video frames.

To summarize, we describe an automatic system by integrating deep learning components to overcome the challenges in unconstrained video-based face recognition. The proposed system first detects faces and facial landmarks using two state-of-the-art DCNN face detectors, the Single Shot Detector (SSD) for faces [6] and the Deep Pyramid Single Shot Face Detector (DPSSD) [38]. Next, we extract deep features from the detected faces using state-of-the-art DCNNs [38] for face recognition. SORT [4] and TFA [5] are used for face association in single-shot/multiple-shot videos respectively. Finally, in the proposed face recognition system, we learn a subspace representation from each video template and match pairs of templates using principal angles-based subspace-to-subspace similarity metric on the learned subspace representations. An overview of the proposed system is shown in Fig. 9.3.

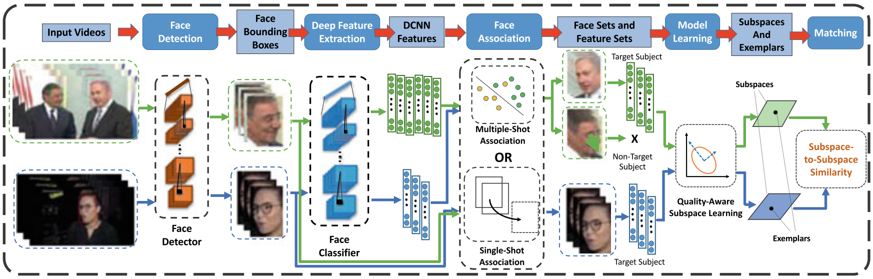


Fig. 9.3 Overview of the proposed system

We present the results of our face recognition system on the challenging IJB-B and IJB-S datasets, as well as MBGC and FOCS datasets. The results demonstrate that the proposed system achieves improved performance over other deep learning-based baselines and state-of-the-art approaches.

9.2 Related Work

9.2.1 Pre Deep Learning Methods

Frame-Based Fusion: An immediate possible utilization of temporal information for video-based face recognition is to fuse the results obtained by a 2D face recognition algorithm on each frame of the sequence. The video sequence can be seen as an unordered set of images to be used for both training and testing phases. During testing one can use the sequence as a set of probes, each of them providing a decision regarding the identity of the person. Appropriate fusion techniques can then be applied to provide the final identity. Perhaps the most frequently used fusion strategy in this case is majority voting [26, 45].

In [35], Park et al. adopt three matchers for frame-level face recognition: FaceVACS, PCA, and correlation. They use the sum rule (with min-max normalization) to fuse results obtained from the three matchers and the maximum rule to fuse results of individual frames. In [25], the concept of identity surface is proposed to represent the hyper-surface formed by projecting face patterns of an individual to the feature vector space parameterized with respect to pose. This surface is learned from gallery videos. In the testing stage, model trajectories are synthesized on the identity surfaces of enrolled subjects after the pose parameters of the probe video have been estimated. Every point on the trajectory corresponds to a frame of the video and trajectory distance is defined as a weighted sum of point-wise distances. The model trajectory that yields the minimum distance to the probe video's trajectory gives the final identification result. Based on the result that images live approximately in a bilinear space of motion and illumination variables, Xu et al. estimate these parameters for each frame of a probe video sequence with a registered 3D generic face model [56]. They then replace the generic model with a person-specific model of each subject in the gallery to synthesize video sequences with the estimated illumination and motion parameters. Frame-wise comparison is conducted between the synthesized videos and the probe video. A synthesized video is considered as a winner if one of its frames yields the smallest distance across all frames and all the subjects in the gallery.

Ensemble Matching: Without recourse to modeling temporal dynamics, one can consider a video as an ensemble of images. Several methods have focused on utilizing image-ensembles for object and face recognition [1, 14, 16, 41]. For example, it was shown by Jacobs et al. that the illumination cone of a convex Lambertian surface can be approximated by a 9-dimensional linear subspace [3]. Motivated by this, the set of face images of the same person under varying illumination conditions is frequently modeled as a linear subspace of 9-dimensions. In such applications, an object 'category' consists of image sets of several 'instances'. A common approach in such applications is to approximate the image space of a single face/object under these variations as a linear subspace. A simplistic model for object appearance variations is then a mixture of subspaces. Zhou and Chellappa study the problem of measuring similarity between two ensembles by projecting the data into a Reproducing Kernel Hilbert Space (RKHS). The ensemble distance is then characterized as the probabilistic distance (Chernoff distance, Bhattacharyya distance, Kullback–Leibler (KL) divergence, etc.) in RKHS.

Appearance Modeling: Most face recognition approaches rely on a model of appearance for each individual subject. The simplest appearance model is a static image of the person. Such appearance models are rather limited in utility in video-based face recognition tasks where subjects may be imaged under varying viewpoints, illuminations, expressions, etc. Thus, instead of using a static image as an appearance model, a sufficiently long video that encompasses several variations in facial appearance can lend itself to building more robust appearance models. Several methods have been proposed for extracting more descriptive appearance models from videos. For example, a facial video is considered as a sequence of images sampled from an "appearance manifold". In principle, the appearance manifold of a subject contains all possible appearances of the subject. In practice, the appearance manifold for each person is estimated from training data of videos. For ease of estimation, the appearance manifold is considered to be a collection of affine subspaces, where each subspace encodes a set of similar appearances of the subject. Temporal variations of appearances in a given video sequence are then modeled as transitions between the appearance subspaces. This method is robust to large appearance changes if sufficient 3D view variations and illumination variations are available in the training set. Further, the tracking problem can be integrated into this framework by searching for a bounding box on the test image that minimizes the distance of the cropped region to the learned appearance manifold.

Basri and Jacobs [3] represent the appearance variations due to shape and illumination on human faces, using the assumption that the 'shape-illumination manifold' of all possible illuminations and head poses is generic for human faces. This means that the shape-illumination manifold can be estimated using a set of subjects exclusive of the test set. They show that the effects of face shape and illumination can be learned using Probabilistic PCA from a small, unlabeled set of video sequences of faces in randomly varying lighting conditions. Given a novel sequence, the learned model is used to decompose the face appearance manifold into albedo and shape-illumination manifolds, producing the classification decision using robust likelihood estimation.

Wang et al. [52] proposed a Manifold-to-Manifold Distance (MMD) for face recognition based on image sets. In [51], the proposed approach models the image set with its second-order statistic for image set classification.

Chen et al. [9] and [10] proposed a video-based face recognition algorithm using sparse representations and dictionary learning. They used the identity information (face, body, and motion) in multiple frames and the accompanying dynamic signature to recognize people in unconstrained videos. Their approach is based on video-dictionaries for face and body. Video-dictionaries are a generalization of sparse representation and dictionaries for still images. They design the video-dictionaries to implicitly encode temporal, pose, and illumination information. In addition, the video-dictionaries are learned for both face and body, which enables the algorithm to encode both identity cues. To increase the ability to learn nonlinearities, they further apply kernel methods for learning dictionaries. Zheng et al. [60] proposed a hybrid dictionary learning and matching approach for video-based face recognition.

9.2.2 Deep Learning Based Methods

Face Recognition: Taigman et al. [49] learned a DCNN model on the frontalized faces generated from 3D shape models built from the face dataset. Sun et al. [46, 47] achieved results surpassing human performance for face verification on the LFW dataset [21]. Schroff et al. [44] adopted the GoogLeNet trained for object recognition to face recognition and trained on a large-scale unaligned face dataset. Parkhi et al. [36] achieved impressive results using a very deep convolutional network based on VGGNet for face verification. Ding et al. [12] proposed a trunk-branch ensemble CNN model for video-based face recognition. Chen et al. [7] trained a 10-layer CNN on CASIAWebFace dataset [59] followed by the JB metric and achieved state-of-the-art performance on the IJB-A [24] dataset. Chen et al. [8] further extended [7] and designed an end-to-end system for unconstrained face recognition and reported a very good performance on IJB-A, JANUS CS2, LFW, and YouTubeFaces [55] datasets. In order to tackle the training bottleneck for the face recognition network, Ranjan et al. [37] proposed the crystal loss to train the network on very large-scale training data. Zheng et al. [61] achieved good performance on video face datasets including IJB-B [54] and IJB-S [23]. Deng et al. [11] introduced sub-center Additive Angular Margin Loss (ArcFace) loss which significantly increases the discriminative power of the model and also makes it less susceptible to label noise by encouraging one dominant sub-class that contains the majority of clean faces and non-dominant sub-classes that include hard/noisy faces.

Video Face Recognition: Most deep-learning-based video face recognition methods extract the features from each frame and take a weighted average of them. [14, 16, 29, 58] use attention weights or quality scores to aggregate the features. Some methods like [27, 31, 42] model the spatio-temporal information with an attention mechanism to find the focus of video frames. [34, 41] propose synthesizing representative face images from a video sequence.

9.3 Method

For each video, we first detect faces from video frames and align them using the detected fiducial points. Deep features are then extracted for each detected face using our DCNN models for face recognition. Based on different scenarios, we use face association or face tracking to construct face templates with unique identities. For videos with multiple shots, we use the face association technique TFA [5] to collect faces from the same identities across shots. For single-shot videos, we use the face tracking algorithm SORT introduced in [4] to generate tracklets of faces. After templates are constructed, in order to aggregate face representations in videos, subspaces are learned using quality-aware principal component analysis. Subspaces along with quality-aware exemplars of templates are used to produce the similarity scores between video pairs by a quality-aware principal angle-based subspace-to-subspace similarity metric. In the following sections, we discuss the proposed video-based face recognition system in detail.

9.3.1 Face/Fiducial Detection

The first step in our face recognition pipeline is to detect faces in images (usually for galleries) and videos. We use two DCNN-based detectors in our pipeline based on different distributions of input.

For regular images and video frames, faces are relatively bigger and with higher resolution. We use SSD trained with the WIDER face dataset as our face detector [6]. For small and remote faces in surveillance videos, we use DPSSD [38] for face detection. DPSSD is fast and capable of detecting tiny faces, which is very suitable for face detection in videos.

After raw face detection bounding boxes are generated using either SSD or DPSSD detectors, we use All-in-One Face [40] for fiducial localization. It is followed by a seven-point face alignment step based on the similarity transform on all the detected faces.

9.3.2 Deep Feature Representation

After faces are detected and aligned, we use the DCNN models to represent each detected face. The models are state-of-the-art networks with different architectures for face recognition. Different architectures provide different error patterns during testing. After fusing the results from different models, we achieve performance better than a single model. Design details of these networks along with their training details are described in Sect. 9.4.2.

9.3.3 Face Association

In previous steps, we obtain raw face detection bounding boxes using our detectors. Features for the detected bounding boxes are extracted using face recognition networks. The next important step in our face recognition pipeline is to combine the detected bounding boxes from the same identity to construct templates for good face recognition results.

For single-shot videos, which means the bounding boxes of a certain identity will probably be contiguous, we rely on SORT [4] to build the tracklets for each identity. For multi-shot videos, it is challenging to continue tracking across different scenes. In the proposed system, we use [5] to adaptively update the face associations through one-shot SVMs.

9.3.4 Model Learning: Deep Subspace Representation

After deep features are extracted for each face template, since each template contains a varying number of faces, these features are further encoded into a fixed-size and unified representation for efficient face recognition.

The simplest representation of a set of samples is the sample mean. However, video templates contain faces with different quality and large variations in illumination, blur, and pose. Since average pooling treats all the samples equally, the outliers may deteriorate the discriminative power of the representation. Different from other feature aggregation approaches that require a large amount of extra training data which are not available for datasets like IJB-B and IJB-S, we propose a subspace representation for video face templates.

9.3.4.1 Subspace Learning from Deep Representations

A d-dimensional subspace S can be uniquely defined by an orthonormal basis $\mathbf{P} \in \mathbb{R}^{D\times d}$, where D is the dimension of features. Given face features from a video sequence $\mathbf{Y} \in \mathbb{R}^{D\times N}$, where N is the sequence length, $\mathbf{P}$ can be found by optimizing:

$$\underset{\mathbf{P},\mathbf{X}}{\text{minimize}} \ \ \|\mathbf{Y} - \mathbf{P}\mathbf{X}\|_F^2 \quad s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \tag{9.1}$$

which is the reconstruction error of features $\mathbf{Y}$ in the subspace S. It is exactly the principal component analysis (PCA) problem and can be easily solved by eigenvalue decomposition. Let $\mathbf{Y}\mathbf{Y}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ be the eigenvalue decomposition, where $\mathbf{U} = \left[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_D\right]$ are eigenvectors and $\mathbf{\Lambda} = diag\{\lambda_1, \lambda_2, \ldots, \lambda_D\}$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$ are the corresponding eigenvalues, we have $\mathbf{P} = \left[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_d\right]$ consisting of the first d basis in $\mathbf{U}$. We use **Sub** to denote this basic subspace learning algorithm (9.1).

9.3.4.2 Quality-Aware Subspace Learning from Deep Representations

In a face template from videos, faces contain large variations in pose, illumination, occlusion, etc. Even in a tracklet, faces have different poses because of head movement, or being occluded in some frames because of the interaction with the environment. When learning the subspace, treating the frames equally is not an optimal solution. In our system, the detection score for each face bounding box provided by the face detector can be used as a good indicator of the face quality, as shown in [37]. Hence, following the quality pooling proposed in [37], we propose quality-aware subspace learning based on detection scores. The learning problem is modified (9.1) as

$$\underset{\mathbf{P},\mathbf{X}}{\text{minimize}} \sum_{i=1}^{N} \tilde{d}_i \|\mathbf{y}_i - \mathbf{P}\mathbf{x}_i\|_2^2 \quad s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \tag{9.2}$$

where $\tilde{d}_i = softmax(ql_i)$ is the normalized detection score of face i, q is the temperature parameter and

$$l_i = \min(\frac{1}{2}\log\frac{d_i}{1-d_i}, t) \tag{9.3}$$

which is upper bounded by threshold t to avoid extreme values when the detection score is close to 1.

Let $\tilde{\mathbf{Y}} = \left[\sqrt{d_1}\mathbf{y}_1, \cdots, \sqrt{d_N}\mathbf{y}_N\right]$ be the normalized feature set, and the corresponding eigenvalue decomposition be $\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^T$. We have

$$\mathbf{P}_D = \left[\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \cdots, \tilde{\mathbf{u}}_d\right] \tag{9.4}$$

which consists of the first d bases in $\tilde{\mathbf{U}}$. The new subspace is therefore learned by treating samples differently according to their quality. This quality-aware learning algorithm is denoted as **QSub**.

9.3.5 Matching: Subspace-to-Subspace Similarity for Videos

After subspace representations are learned for video templates, inspired by a manifold-to-manifold distance [52], we measure the similarity between two video templates of faces using a subspace-to-subspace similarity metric. In this part, we first introduce the widely used metric based on principal angles. Then we propose several weighted subspace-to-subspace metrics which take the importance of basis directions into consideration.

9.3.5.1 Principal Angles and Projection Metric

One of the most used subspace-to-subspace similarities is based on principal angles. The principal angles $0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_r \le \frac{\pi}{2}$ between two linear subspaces S_1 and S_2 can be computed by Singular Value Decomposition (SVD).

Let $\mathbf{P}_1 \in \mathbb{R}^{D\times d_1}$, $\mathbf{P}_2 \in \mathbb{R}^{D\times d_2}$, denoting the orthonormal basis of S_1 and S_2, respectively. The SVD is $\mathbf{P}_1^T\mathbf{P}_2 = \mathbf{Q}_{12}\boldsymbol{\Lambda}\mathbf{Q}_{21}^T$, where $\boldsymbol{\Lambda} = diag\{\sigma_1, \sigma_2, \ldots, \sigma_r\}$. $\mathbf{Q}_{12}$ and $\mathbf{Q}_{21}$ are orthonormal matrices. The singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$ are exactly the cosine of the principal angles as $\cos\theta_k = \sigma_k,\ k = 1, 2, \ldots, r$.

Projection metric [13] is a popular similarity metric based on principal angles:

$$s_{PM}(S_1, S_2) = \sqrt{\frac{1}{r}\sum_{k=1}^{r}\cos^2\theta_k} \tag{9.5}$$

Since $\|\mathbf{P}_1^T\mathbf{P}_2\|_F^2 = \|\mathbf{Q}_{12}\boldsymbol{\Lambda}\mathbf{Q}_{21}^T\|_F^2 = \|\boldsymbol{\Lambda}\|_F^2 = \sum_{k=1}^r \sigma_k^2 = \sum_{k=1}^r \cos^2\theta_k$, we have

$$s_{PM}(S_1, S_2) = s_{PM}(\mathbf{P}_1, \mathbf{P}_2) = \sqrt{\frac{1}{r}\|\mathbf{P}_1^T\mathbf{P}_2\|_F^2} \tag{9.6}$$

and there is no need to explicitly compute the SVD. We use **PM** to denote this similarity metric (9.6).

9.3.5.2 Exemplars and Basic Subspace-to-Subspace Similarity

Existing face recognition systems usually use cosine similarity between exemplars to measure the similarity between templates. The exemplar of a template is defined as its sample mean, as $\mathbf{e} = \frac{1}{L}\sum_{i=1}^{L}\mathbf{y}_i$, where $\mathbf{y}_i$ are samples in the template. Exemplars mainly capture the average and global representation of the template. On the other hand, the projection metric we introduced above measures the similarity between two subspaces, which models the correlation between samples. Hence, in the proposed system, we make use of both of them by fusing their similarity scores as the subspace-to-subspace similarity between two video sequences.

Suppose subspaces $\mathbf{P}_1 \in \mathbb{R}^{D\times d_1}$ and $\mathbf{P}_2 \in \mathbb{R}^{D\times d_2}$ are learned from a pair of video templates $\mathbf{Y}_1 \in \mathbb{R}^{D\times L_1}$ and $\mathbf{Y}_2 \in \mathbb{R}^{D\times L_2}$ in deep features respectively, by either **Sub** or **QSub** methods introduced in Sect. 9.3.4. Their exemplars are $\mathbf{e}_1 = \frac{1}{L_1}\sum_{i=1}^{L_1}\mathbf{y}_{1i}$ and $\mathbf{e}_2 = \frac{1}{L_2}\sum_{i=1}^{L_2}\mathbf{y}_{2i}$ respectively. Combining the orthonormal bases and exemplars, the subspace-to-subspace similarity can be computed as

$$\begin{aligned} s(\mathbf{Y}_1, \mathbf{Y}_2) &= s_{Cos}(\mathbf{Y}_1, \mathbf{Y}_2) + \lambda s_{PM}(\mathbf{P}_1, \mathbf{P}_2) \\ &= \frac{\mathbf{e}_1^T\mathbf{e}_2}{\|\mathbf{e}_1\|_2\|\mathbf{e}_2\|_2} + \lambda\sqrt{\frac{1}{r}\|\mathbf{P}_1^T\mathbf{P}_2\|_F^2} \end{aligned} \tag{9.7}$$

where $s_{Cos}(\mathbf{Y}_1, \mathbf{Y}_2)$ is the cosine similarity between exemplars, denoted as **Cos**, and $s_{PM}(\mathbf{P}_1, \mathbf{P}_2)$ is computed by (9.6). Since the DCNN features are more robust if we keep their signs, instead of using $s_{Cos}^2(\mathbf{Y}_1, \mathbf{Y}_2)$ as in [52] where the sign information is lost, we use $s_{Cos}(\mathbf{Y}_1, \mathbf{Y}_2)$ in our formulation. Accordingly, we also take the square root of the principal angle term to keep the scale consistent. λ here is a hyperparameter that balances the cosine similarity and principal angle similarity. If $\mathbf{P}_i$'s are learned by **Sub**, we denote the whole similarity metric (including exemplars computing and subspace learning) as **Cos+Sub-PM**. If $\mathbf{P}_i$'s are learned by the proposed **QSub**, we denote the similarity as **Cos+QSub-PM**.

9.3.5.3 Quality-Aware Exemplars

In either **Cos+Sub-PM** or **Cos+QSub-PM** we are still using simple average pooling to compute the exemplars. But as discussed in Sect. 9.3.4, templates consist of faces of different quality. Treating them equally in pooling will let low-quality faces deteriorate the global representation of the template. Therefore, we propose to use the same normalized detection score as in Sect. 9.3.4 to compute the quality-aware exemplars by $\mathbf{e}_D = \frac{1}{L}\sum_{i=1}^{L} \tilde{d}_i \mathbf{y}_i$, where $\tilde{d}_i = softmax(ql_i)$ and l_i are computed by (9.3). Then, the cosine similarity between the quality-aware exemplars is

$$s_{QCos}(\mathbf{Y}_1, \mathbf{Y}_2) = \frac{\mathbf{e}_{D1}^T \mathbf{e}_{D2}}{\|\mathbf{e}_{D1}\|_2 \|\mathbf{e}_{D2}\|_2} \tag{9.8}$$

and we denote it as **QCos**. Using the new cosine similarity, the similarity becomes

$$s(\mathbf{Y}_1, \mathbf{Y}_2) = s_{QCos}(\mathbf{Y}_1, \mathbf{Y}_2) + \lambda s_{PM}(\mathbf{P}_1, \mathbf{P}_2) \tag{9.9}$$

If P_i's are learned by **QSub**, the similarity is further denoted by **QCos+QSub-PM**.

9.3.5.4 Variance-Aware Projection Metric

As previously discussed, the projection metric $S_{PM}(S_1, S_2)$ is the square root of the mean square of principle angles between two subspaces and it treats each basis direction in each subspace equally. But these basis vectors are actually eigenvectors of an eigenvalue decomposition problem. Different basis vectors correspond to different eigenvalues, which represent the variance of data in the corresponding direction. Obviously, those basis directions with larger variances contain more information than those with smaller variances. Therefore, based on the variance of each basis direction, we propose a variance-aware projection metric:

$$s_{VPM}(\mathbf{P}_1, \mathbf{P}_2) = \sqrt{\frac{1}{r}\|\tilde{\mathbf{P}}_1^T \tilde{\mathbf{P}}_2\|_F^2} \tag{9.10}$$

where

$$\tilde{\mathbf{P}}_i = \frac{1}{tr(\log(\mathbf{\Lambda}_i))}\mathbf{P}_i \log(\mathbf{\Lambda}_i) \tag{9.11}$$

$\mathbf{\Lambda}_i$ is a diagonal matrix whose diagonals are eigenvalues corresponding to eigenvectors in $\mathbf{P}_i$. $\frac{1}{tr(\log(\mathbf{\Lambda}_i))}$ is the normalization factor. We use the logarithm of variance to weigh different basis directions in a subspace. This similarity metric is inspired by the Log-Euclidean distance used for image set classification in [51]. Empirically, we use $\max(0, \log(\mathbf{\Lambda}_i))$ instead of $\log(\mathbf{\Lambda}_i)$ to avoid negative weights. We use **VPM** to denote this similarity metric (9.10).

9.3.5.5 Quality-Aware Subspace-to-Subspace Similarity

By combining the quality-aware subspace learning, quality-aware exemplars and variance-aware projection metric, we propose the quality-aware subspace-to-subspace similarity between two video templates as

$$s(\mathbf{Y}_1, \mathbf{Y}_2) = s_{QCos}(\mathbf{Y}_1, \mathbf{Y}_2) + \lambda s_{VPM}(\mathbf{P}_{D1}, \mathbf{P}_{D2}) \tag{9.12}$$

where s_{QCos} is defined in (9.8), $\mathbf{P}_{Di}$'s are learned by (9.4) and s_{VPM} is defined in (9.10). This similarity metric is denoted as **QCos+QSub-VPM**. Comparisons of the proposed similarity metrics and other baselines on several challenging datasets are discussed in Sect. 9.4.

9.4 Experiments

In this section, we report video-based face recognition results for the proposed system on two challenging video face datasets, IARPA Janus Benchmark B (IJB-B) and IARPA Janus Surveillance Video Benchmark (IJB-S), and compare them with other baseline methods. We also provide results on Multiple Biometric Grand Challenge (MBGC), and Face and Ocular Challenge Series (FOCS) datasets, to demonstrate the effectiveness of the proposed system. We introduce the details of datasets, protocols, and our training and testing procedures in the following sections.

9.4.1 Datasets

IARPA Janus Benchmark B (IJB-B): IJB-B dataset is an unconstrained face recognition dataset. It contains 1845 subjects with 11,754 images, 55,025 frames, and 7,011 multiple-shot videos. IJB-B is a template-based dataset where a template consists of a varying number of still images or video frames from different sources. A template can be either an image-only, video-frame-only, or mixed-media template. Sample frames from this dataset are shown in Fig. 9.1.

In this work, we only focus on the 1:N video protocol of IJB-B. It is an open-set 1:N identification protocol where each given probe is collected from a video and is searched among all gallery faces. Gallery candidates are ranked according to their similarity scores to the probes. Top-K rank accuracy and True Positive Identification Rate (TPIR) over False Positive Identification Rate(FPIR) are used to evaluate the performance. The gallery templates are separated into two splits, G_1 and G_2, all consisting of still images. For each video, we are given the frame index with a face bounding box of the first occurrence of the target subject, as shown in Fig. 9.1. Based on this anchor, all the faces in that video with the same identity should be collected to construct the probes. The identity of the first occurrence bounding box will be considered as the template identity for evaluation.

IARPA Janus Surveillance Video Benchmark (IJB-S): Similar to IJB-B, the IJB-S dataset is also a template-based, unconstrained video face recognition dataset. It contains faces in two separate domains: high-resolution still images for galleries and low-quality, remotely captured surveillance videos for probes. It consists of 202 subjects from 1421 images and 398 single-shot surveillance videos. The number of subjects is small compared to IJB-B, but it is even more challenging due to the low-quality nature of surveillance videos.

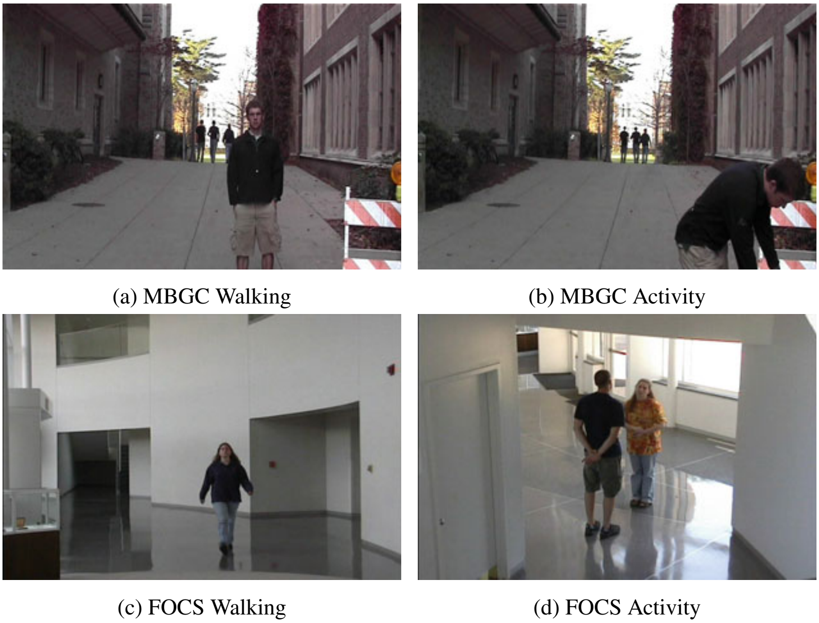
Based on the choices of galleries and probes, we are interested in three different surveillance video-based face recognition protocols: surveillance-to-single protocol, surveillance-to-booking protocol, and surveillance-to-surveillance protocol. These are all open-set 1:N protocols where each probe is searched among the given galleries. Like IJB-B, the probe templates are collected from videos, but no annotations are provided. Thus raw face detections are grouped to construct templates with the same identities.

Galleries consist of only single frontal high-resolution images for surveillance-to-single protocol. Galleries are constructed by both frontal and multiple-pose high-resolution images for surveillance-to-booking protocol. For the most challenging surveillance-to-surveillance protocol, galleries are collected from surveillance videos as well, with given bounding boxes. In all three protocols, gallery templates are split into two splits, G_1 and G_2. During evaluation, the detected faces in videos are first matched to the ground truth bounding boxes to find their corresponding identity information. The majority of identities that appear in each template will be considered as the identity of the template and will be used for further identification evaluation. Example frames are shown in Fig. 9.2. Notice the remote faces are of very low quality.

Multiple Biometric Grand Challenge (MBGC): The MBGC Version 1 dataset contains 399 walking (frontal face) and 371 activity (profile face) video sequences from 146 people. Figure 9.4 shows some sample frames from different walking and activity videos. In the testing protocol, verification is specified by two sets: target and query. The protocol requires the algorithm to match each target sequence with all query sequences. Three verification experiments are defined: walking-vs-walking (WW), activity-vs-activity (AA), and activity-vs-walking (AW).

Face and Ocular Challenge Series (FOCS): The video challenge of FOCS is designed for frontal and non-frontal video sequence matching. The FOCS UT Dallas dataset contains 510 walking (frontal face) and 506 activity (non-frontal face) video sequences of 295 subjects with a frame size of 720×480 pixels. Like MBGC, FOCS specifies three verification protocols: walking-vs-walking, activity-vs-walking, and activity-vs-activity. In these experiments, 481 walking videos and 477 activity videos are chosen as query videos. The size of target sets ranges from 109 to 135 video sequences. Sample video frames from this dataset are shown in Fig. 9.4.

IJB-MDF: The IARPA JANUS Benchmark Multi-domain Face (IJB-MDF) dataset consists of images and videos of 251 subjects captured using a variety of cameras corresponding to visible, short-, mid-, and long-wave infrared and long-range surveillance domains. There



(a) MBGC Walking

(b) MBGC Activity

(c) FOCS Walking

(d) FOCS Activity

Fig. 9.4 Examples of MBGC and FOCS datasets

are 1,757 visible enrollment images, 40,597 short-wave infrared (SWIR) enrollment images, and over 800 videos spanning 161 hours.

9.4.2 Implementation Details

In this section, we discuss the implementation details for each dataset respectively.

9.4.2.1 IJB-B

For the IJB-B dataset, we employ the SSD face detector [6] to extract the face bounding boxes in all images and video frames. We employ the facial landmark branch of All-in-One Face [40] for fiducial detection on every detected bounding box and apply facial alignment based on these fiducials using the seven-point similarity transform.

The aligned faces are further represented using three networks proposed in [39]. We denote them as Network A, Network B, and Network C. Network A modifies the ResNet-101 [20] architecture. It has an input size of dimensions 224 × 224 and adds an extra fully connected layer after the last convolutional layer to reduce the feature dimensionality to 512. Also, it replaces the original softmax loss with the crystal loss [37] for more stable training. Network B uses the Inception-ResNet-v2 [48] model as the base network. Similar to Network A, an additional fully-connected layer is added for dimensionality reduction.

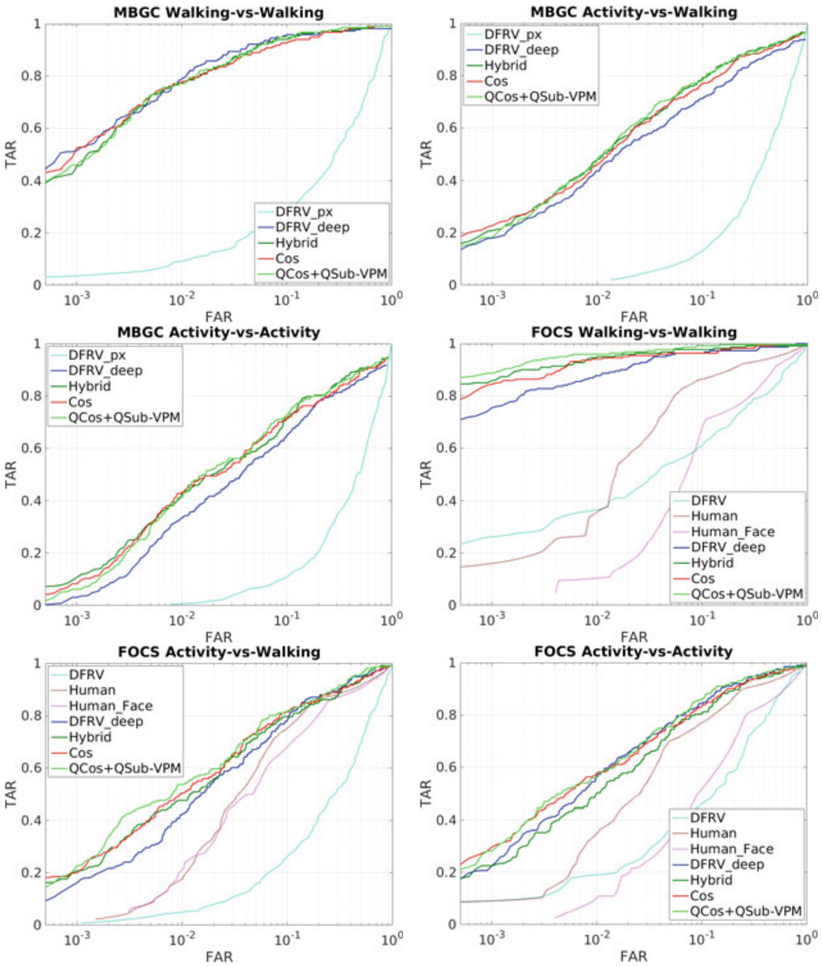


Fig. 9.5 Verification results on MBGC and FOCS datasets

Naive softmax followed by cross-entropy loss is used for this network. Network C is based on the face recognition branch in the All-in-One Face architecture [40]. The branch consists of seven convolutional layers followed by three fully-connected layers.

Network A and Network C are trained on the MSCeleb-1M dataset [19] which contains 3.7 million images from 57,440 subjects. Network B is trained on the union of three datasets called the Universe dataset: 3.7 million still images from the MSCeleb-1M dataset, 300,000 still images from the UMDFaces dataset [2], and about 1.8 million video frames from the UMDFaces Video dataset. For each network, we further reduce its dimensionality to 128 by triplet probabilistic embedding (TPE) [43] trained on the UMDFaces dataset.

For face association, we follow the details in [5]. Then, features from associated bounding boxes are used to construct the probe templates. We use quality-aware pooling for both gallery and probe templates to calculate their exemplars (**QCos**) where $t = 7$ and $q = 0.3$ are used for detection score normalization. Subspaces are built by applying the quality-aware subspace learning method (**QSub**) on each template and taking the top three eigenvectors with the largest corresponding eigenvalues. When fusing the cosine similarity and variance-aware projection similarity metric (**VPM**), we use $\lambda = 1$ so two similarity scores are fused equally. We compute the subspace-to-subspace similarity score for each network independently and combine the similarity scores from three networks by score-level fusion. We also implement baseline methods using combinations of exemplars from vanilla average pooling (**Cos**), subspaces learned by regular PCA (**Sub**), and projection similarity metric (**PM**).

9.4.2.2 IJB-S

For the IJB-S dataset, we employ the multi-scale face detector DPSSD to detect faces in surveillance videos. We only keep face bounding boxes with detection scores greater than 0.4771, to reduce the number of false detections. We use the facial landmark branch of All-in-One Face [40] as the fiducial detector. Face alignment is performed using the seven-point similarity transform.

Different from IJB-B, since IJB-S does not specify the subject of interest, we are required to localize and associate all the faces for different subjects to yield the probe sets. Since IJB-S videos are single-shot, we use SORT [4] to track every face appearing in the videos. Faces in the same tracklet are grouped to create a probe template. Since some faces in surveillance videos are of extreme pose, blur, and low resolution, to improve precision, tracklets consisting of such faces should be rejected during the recognition stage. By observation, we find that most of the short tracklets are of low quality and not reliable. The average of the detection score provided by DPSSD is also used as an indicator of the quality of the tracklet. On the other hand, we also want to take the performance of face detection into consideration to strike a balance between recall and precision. Thus in our experiments, we use two configurations for tracklets filtering: (1) We keep those tracklets with lengths greater than or equal to 25 and an average detection score greater than or equal to 0.9 to reject low-quality tracklets and focus on precision. It is referred to as **with Filtering**. (2) Following the settings in [23], we produce results without any tracklets filtering and focusing on both precision and recall. It is referred to as **without Filtering**.

Because of the remote acquisition scenario and the presence of blurred probes in the IJB-S dataset, we retrain Network A with the same crystal loss but on the Universe dataset used by Network B. We denote it as Network D. We also retrain Network B with the crystal loss [37] on the same training data. We denote it as Network E. As a combination of high-capacity network and large-scale training data, Networks D and E are more powerful than Networks A, B, and C. As before, we reduce feature dimensionality to 128 using the TPE trained on the UMDFaces dataset.

In IJB-S, subspace learning and matching parts are the same as IJB-B except that we combine the similarity score by score-level fusion from Network D and E. Notice that for the surveillance-to-surveillance protocol, we only use single Network D for representation as Network E is ineffective for low-quality gallery faces in this protocol.

9.4.2.3 MBGC and FOCS

For MBGC and FOCS datasets, we use All-in-One Face for both face detection and facial landmark localization. The MBGC and FOCS datasets contain only one person in a video in general. Hence, for each frame, we directly use the face bounding box with the highest detection score as the target face. Similar to IJB-S, bounding boxes are filtered based on detection scores. From the detected faces, deep features are extracted using Network D. Since MBGC and FOCS datasets do not provide training data, we also use the TPE trained on the UMDFaces dataset to reduce feature dimensionality to 128. For MBGC and FOCS, subspace learning and matching parts are the same as IJB-B and IJB-S.

9.4.3 Evaluation Results

In the following section, we first show some face association results on IJB-B and IJB-S datasets. Then we compare the performance of the proposed face recognition system with several baseline methods. For each dataset, all the baseline methods listed below use deep features extracted from the same network and with the same face detector.

- **Cos:** We compute the cosine similarity scores directly from the exemplars with average pooling.
- **QCos:** We compute the cosine similarity scores from the exemplars with quality-aware average pooling.
- **Cos+Sub-PM:** Subspace-to-subspace similarity is computed by fusing the plain cosine similarity and plain projection metric, and subspaces are learned by plain PCA.
- **QCos+Sub-PM:** Subspace-to-subspace similarity is computed by fusing the quality-aware cosine similarity and plain projection metric, and subspaces are learned by plain PCA.
- **QCos+QSub-PM:** Subspace-to-subspace similarity is computed by fusing the quality-aware cosine similarity and plain projection metric, and subspaces are learned by quality-aware subspace learning.
- **QCos+QSub-VPM:** Subspace-to-subspace similarity is computed by fusing the quality-aware cosine similarity and variance-aware projection metric, and subspaces are learned by quality-aware subspace learning.

IJB-B: Figures 9.6 and 9.7 show some examples of our face association results using TFA in IJB-B dataset. Table 9.1 shows the Top-K Accuracy results for IJB-B video protocol. In this dataset, besides the baselines, our method is compared with original results in [5]

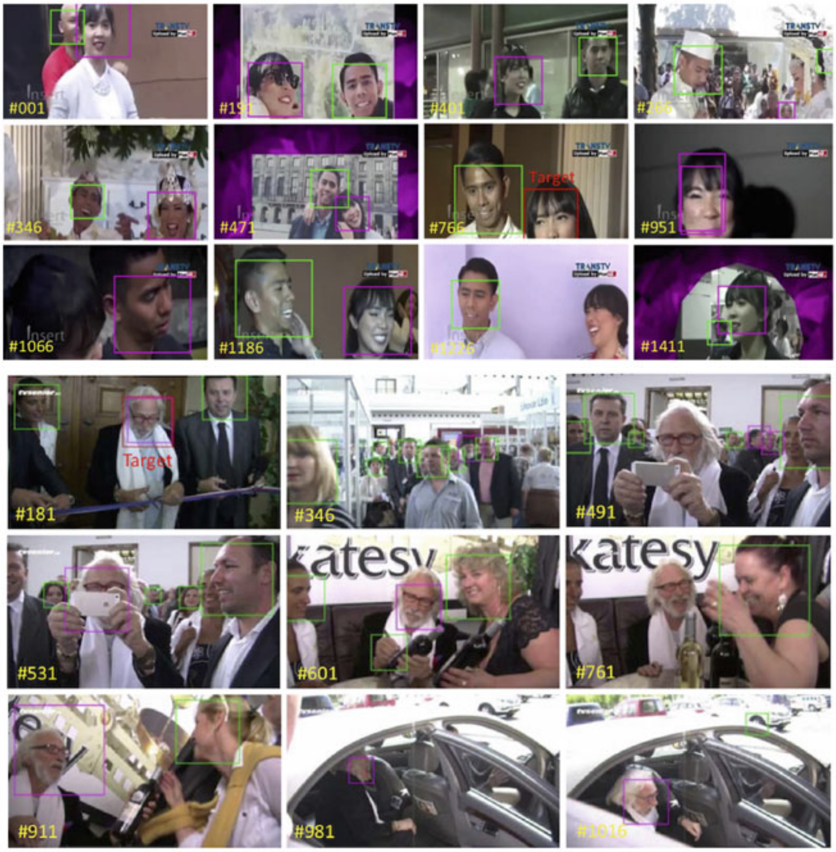


Fig. 9.6 Examples of face association results by TFA on IJB-B. The target annotation is in the red box, and the associated faces of the target subject are in magenta-colored boxes

corresponding to different iteration numbers. The results shown are the average of the two galleries. Notice that our proposed system and [5] use the same face association method, but we have different networks and feature representation techniques.

IJB-S: Figure 9.8 shows some examples of our face association results using SORT in IJB-S dataset. Tables 9.2, 9.3 and 9.4 show the results for IJB-S surveillance-to-single protocol, surveillance-to-booking protocol and surveillance-to-surveillance protocol respectively. Notice that under the **with Filtering** configuration, we use the regular top-K average accuracy for evaluation. Under the **without Filtering** configuration, we use the End-to-End Retrieval Rate (EERR) metric proposed in [23] for evaluation. For surveillance-to-surveillance protocol, we show results for two different network configurations as well. We also implement state-of-the-art network ArcFace [11] on IJB-S and compare it with our method. Results from ArcFace are shown with the prefix **Arc-**.

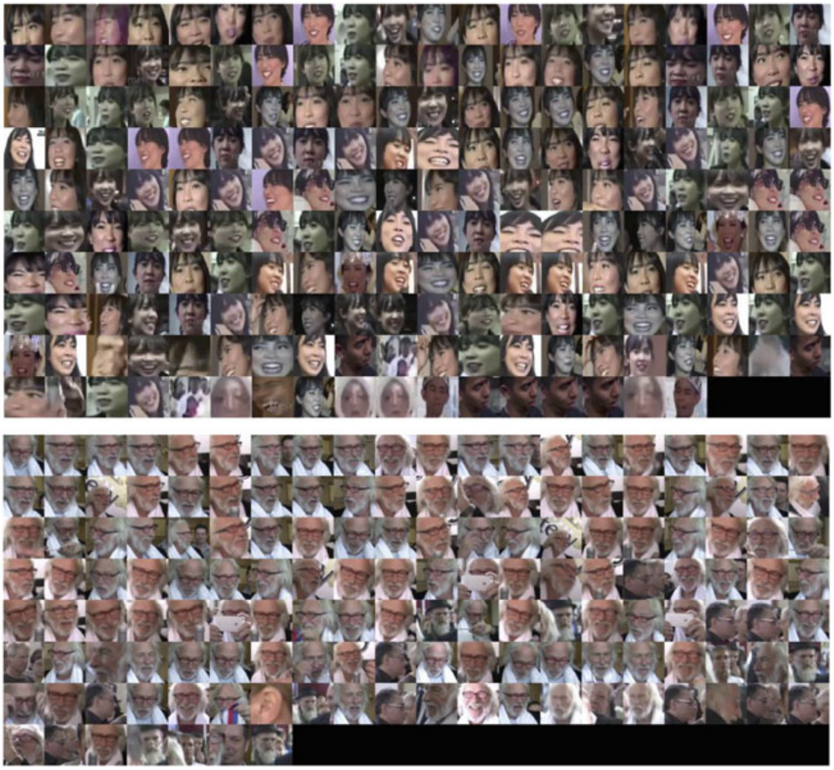


Fig. 9.7 Associated faces by TFA corresponding to examples in Fig. 9.6. Face images are in the order of the confidence of face association

Two recent works [15, 17] have reported results on the IJB-S dataset. These works mainly focused on face recognition and not detection so they built video templates by matching their detections with ground truth bounding boxes provided by the protocols and evaluated their methods using identification accuracy and not EERR metric. Our system focuses on detection, association, and recognition. Therefore after detection, we associate faces across the video frames to build templates without utilizing any ground truth information and evaluate our system using both identification accuracy and EERR metric. Since these two template-building procedures are so different, a direct comparison is not meaningful.

MBGC: The verification results for the MBGC dataset are shown in Table 9.5 and Fig. 9.5. We compare our method with the baseline algorithms, **Hybrid** [60] and [9] using either raw pixels as **DFRV**$_{px}$ (reported in their paper) or deep features as **DFRV**$_{deep}$ (our implementation). We also report the results of the proposed method applied to the ArcFace features with the prefix **Arc-**. Figure 9.5 does not include all the baselines, for a clearer view. The result of [9] is not in the table because the authors did not provide exact numbers in their paper.

Table 9.1 1:N Search Top-K Average Accuracy and TPIR/FPIR of IJB-B video search protocol

Methods	Rank = 1	Rank = 2	Rank = 5	Rank = 10	Rank = 20	Rank = 50	FPIR = 0.1	FPIR = 0.01
[5] with Iteration 0	55.94%	–	68.40%	72.89%	–	83.71%	44.60%	28.73%
[5] with Iteration 3	61.01%	–	73.39%	77.90%	–	87.62%	49.73%	34.11%
[5] with Iteration 5	61.00%	–	73.46%	77.94%	–	87.69%	49.78%	33.93%
Cos	78.37%	81.35%	84.39%	86.29%	88.30%	90.82%	73.15%	52.19%
QCos	78.43%	81.41%	84.40%	86.33%	88.34%	90.88%	**73.19%**	**52.47%**
Cos+Sub-PM	77.99%	81.45%	84.68%	86.75%	88.96%	91.91%	72.31%	38.44%
QCos+Sub-PM	78.02%	81.46%	84.76%	86.72%	88.97%	91.91%	72.38%	38.88%
QCos+QSub-PM	78.04%	81.47%	84.73%	86.72%	88.97%	91.93%	72.39%	38.91%
QCos+QSub-VPM	**78.93%**	**81.99%**	**84.96%**	**87.03%**	**89.24%**	**92.02%**	71.26%	47.35%

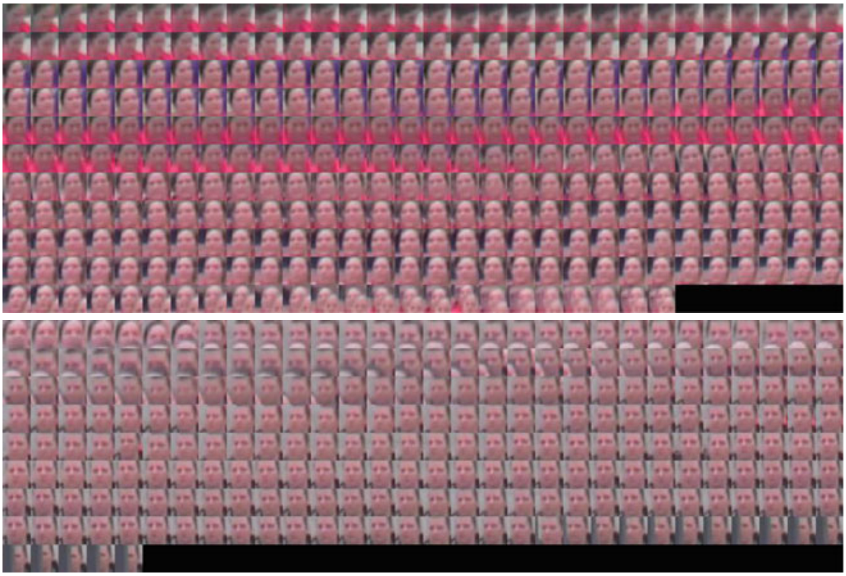


Fig. 9.8 Associated faces using SORT in IJB-S. Face images are in their temporal order. Notice the low-quality faces at the boundaries of tracklets since the tracker cannot reliably track anymore

FOCS: The verification results of FOCS dataset are shown in Table 9.5 and Fig. 9.5. O'Toole et al. [33] evaluated the human performance on this dataset. In the figures, **Human** refers to

Table 9.2 1:N Search results of IJB-S surveillance-to-single protocol. Using both Network D and E for representation

Methods	Top-K average accuracy **with filtering**						EERR metric **without filtering**					
	R=1	R=2	R=5	R=10	R=20	R=50	R=1	R=2	R=5	R=10	R=20	R=50
Arc-Cos [11]	52.03%	56.83%	63.16%	69.05%	76.13%	88.95%	24.45%	26.54%	29.35%	32.33%	36.38%	44.81%
Arc-QCos+QSub-PM	60.92%	65.06%	70.45%	75.19%	80.69%	90.29%	28.73%	30.44%	32.98%	35.40%	38.70%	45.46%
Cos	64.86%	70.87%	77.09%	81.53%	86.11%	93.24%	29.62%	32.34%	35.60%	38.36%	41.53%	46.78%
QCos	65.42%	71.34%	77.37%	81.78%	86.25%	93.29%	29.94%	32.60%	35.85%	38.52%	41.70%	46.78%
Cos+Sub-PM	69.52%	75.15%	80.41%	84.14%	87.83%	94.27%	32.22%	34.70%	37.66%	39.91%	42.65%	47.54%
QCos+Sub-PM	69.65%	75.26%	80.43%	84.22%	87.81%	94.25%	32.27%	34.73%	37.66%	39.91%	42.67%	47.54%
QCos+QSub-PM	**69.82%**	**75.38%**	**80.54%**	**84.36%**	**87.91%**	**94.34%**	**32.43%**	**34.89%**	**37.74%**	**40.01%**	**42.77%**	**47.60%**
QCos+QSub-VPM	69.43%	75.24%	80.34%	84.14%	87.86%	94.28%	32.19%	34.75%	37.68%	39.88%	42.56%	47.50%

Table 9.3 1:N Search results of IJB-S surveillance-to-booking protocol. Using both Network D and E for representation

Methods	Top-K average accuracy **with filtering**						EERR metric **without filtering**					
	R=1	R=2	R=5	R=10	R=20	R=50	R=1	R=2	R=5	R=10	R=20	R=50
Arc-Cos [11]	54.59%	59.12%	65.43%	71.05%	77.84%	89.16%	25.38%	27.58%	30.59%	33.42%	37.60%	45.05%
Arc-QCos+QSub-VPM	60.86%	65.36%	71.30%	76.15%	81.63%	90.70%	28.66%	30.64%	33.43%	36.11%	39.57%	45.70%
Cos	66.48%	71.98%	77.80%	82.25%	86.56%	93.41%	30.38%	32.91%	36.15%	38.77%	41.86%	46.79%
QCos	66.94%	72.41%	78.04%	82.37%	86.63%	93.43%	30.66%	33.17%	36.28%	38.84%	41.88%	46.84%
Cos+Sub-PM	69.39%	74.55%	80.06%	83.91%	87.87%	**94.34%**	32.02%	34.42%	37.59%	39.97%	42.64%	**47.58%**
QCos+Sub-PM	69.57%	74.78%	80.06%	83.89%	87.94%	94.33%	32.16%	34.61%	37.62%	39.99%	42.71%	47.57%
QCos+QSub-PM	69.67%	74.85%	80.25%	84.10%	88.04%	94.22%	32.28%	34.77%	37.76%	40.11%	**42.76%**	47.57%
QCos+QSub-VPM	**69.86%**	**75.07%**	**80.36%**	**84.32%**	**88.07%**	94.33%	**32.44%**	**34.93%**	**37.80%**	**40.14%**	42.72%	**47.58%**

human performance with all bodies of target subjects seen, and **Human_Face** refers to the performance that only faces of the target subjects are seen. Here besides baseline algorithms and **Hybrid** [60], we also compare our method with [9] in either raw pixels as **DFRV**$_{px}$ (reported in their paper) or deep features as **DFRV**$_{deep}$ (our implementation). We also report the results using ArcFace features. Similarly, the results of [9] and human performance are not in the table since they did not provide exact numbers.

Table 9.4 1:N Search results of IJB-S surveillance-to-surveillance protocol. D stands for only using Network D for representation. D+E stands for using both Network D and E for representation

Methods	Top-K average accuracy **with filtering**						EERR metric **without filtering**					
	R=1	R=2	R=5	R=10	R=20	R=50	R=1	R=2	R=5	R=10	R=20	R=50
Arc-Cos [11]	8.68%	12.58%	18.79%	26.66%	39.22%	68.19%	4.98%	7.17%	10.86%	15.42%	22.34%	37.68%
Arc-QCos+QSub-PM	8.64%	12.57%	18.84%	26.86%	39.78%	**68.21%**	**5.26%**	**7.44%**	**11.31%**	15.90%	**22.68%**	**37.83%**
Cos(D+E)	9.24%	12.51%	19.36%	25.99%	32.95%	52.95%	4.74%	6.62%	10.70%	14.88%	19.29%	30.64%
QCos+QSub-VPM(D+E)	**9.56%**	**13.03%**	19.65%	27.15%	35.39%	56.02%	4.77%	6.78%	10.88%	15.52%	20.51%	32.16%
Cos(D)	8.54%	11.99%	19.60%	28.00%	37.71%	59.44%	4.42%	6.15%	10.84%	15.73%	21.14%	33.21%
QCos(D)	8.62%	12.11%	19.62%	28.14%	37.78%	59.21%	4.46%	6.20%	10.80%	15.81%	21.06%	33.17%
Cos+Sub-PM(D)	8.19%	11.79%	19.56%	28.62%	39.77%	63.15%	4.26%	6.25%	10.79%	16.18%	22.48%	34.82%
QCos+Sub-PM(D)	8.24%	11.82%	19.68%	28.68%	39.68%	62.96%	4.27%	6.25%	10.92%	16.18%	22.39%	34.69%
QCos+QSub-PM(D)	8.33%	11.88%	19.82%	28.65%	39.78%	62.79%	4.33%	6.21%	10.96%	16.19%	22.48%	34.69%
QCos+QSub-VPM(D)	8.66%	12.27%	**19.91%**	**29.03%**	**40.20%**	63.20%	4.30%	6.30%	10.99%	**16.23%**	22.50%	34.76%

Table 9.5 Verification results on MBGC and FOCS datasets

Methods	MBGC						FOCS					
	WW		AW		AA		WW		AW		AA	
	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1
Arc-Cos [11]	84.40%	92.20%	53.88%	75.00%	32.47%	66.49%	98.18%	**99.09%**	48.61%	69.44%	48.36%	78.87%
Arc-QCos+QSub-PM	**85.32%**	92.20%	**55.58%**	75.00%	32.99%	64.43%	**98.64%**	**99.09%**	52.31%	74.07%	50.23%	79.81%
DFRV$_{deep}$ [9]	78.90%	**95.87%**	43.69%	71.36%	33.51%	64.95%	87.73%	96.36%	42.13%	78.70%	56.81%	84.51%
Hybrid [60]	77.06%	94.04%	48.06%	**79.37%**	42.53%	71.39%	95.00%	97.73%	47.69%	79.63%	50.23%	80.75%
Cos	77.52%	92.66%	45.87%	76.94%	**43.30%**	71.65%	94.09%	96.36%	50.46%	81.48%	57.75%	83.57%
QCos	77.52%	92.66%	47.57%	76.94%	**43.30%**	71.13%	95.91%	**99.09%**	**53.70%**	80.09%	**58.22%**	83.57%
Cos+Sub-PM	77.98%	94.95%	47.57%	79.13%	41.24%	72.68%	91.82%	97.27%	49.07%	**83.33%**	54.93%	85.45%
QCos+Sub-PM	77.98%	94.95%	48.30%	78.64%	41.75%	**73.71%**	95.91%	98.64%	52.78%	82.87%	55.40%	**85.92%**
QCos+QSub-PM	77.52%	94.95%	48.54%	78.64%	41.75%	73.20%	95.91%	**99.09%**	52.31%	81.02%	55.87%	**85.92%**
QCos+QSub-VPM	77.06%	94.95%	48.06%	78.16%	41.24%	72.68%	95.91%	**99.09%**	**53.70%**	81.94%	56.34%	**85.92%**

9.4.4 Cross-Spectral Video Face Verification

In this section, we present some results on the IARPA JANUS Benchmark Multi-domain Face (IJB-MDF) [22] dataset. The domains in the IJB-MDF dataset are labeled as below:

- (0) visible enrollment
- (1) visible surveillance

- (2) visible gopro
- (3) visible 500m
- (4) visible 400m
- (5) visible 300m
- (6) visible 500m 400m walking
- (11) swir enrollment nofilter
- (12) swir enrollment 1150
- (13) swir enrollment 1350
- (14) swir enrollment 1550
- (15) swir 15m
- (16) swir 30m

There are a total of 251 subjects. Domains 1, 2, 3, 4, 5, 6, 15, and 16 consist of videos only, while the enrollment domains (0, 11, 12, 13, and 14) consist of still images taken in a constrained setting. Instead of performing an end-to-end evaluation, we are more interested in observing how well a feature extractor (trained on visible images) adapts to these new domains. As such, to simplify the task, we use the ground truth provided with the dataset to obtain the start and end time stamps for non-empty frames in the videos and extract all the relevant frames. The videos are captured at a frame rate of 20fps. Table 9.6 shows the distribution of the frames with respect to various domains.

We select domains 3, 4, 5, and 6 for the task of cross-spectral face recognition of remote faces. The quality of faces is sub-par with a lot of blur and lack of detail in the face. We employ the SCRFD [18] algorithm to detect faces from the video frames. The recall at a score-threshold of 0.5 is about 95%.

Table 9.6 IJB-MDF data distribution

Domain	Num videos	Num frames	Approx size of frame	Name of domain
1	358	191,971	19MB	Visible Surveillance
2	24	39,263	7 MB	Visible GoPro
3	31	56,024	6 MB	Visible 500m
4	34	61,446	4 MB	Visible 400m
5	34	61,442	1 MB	Visible 300m
6	26	24,194	7 MB	Visible 500m 400m walking
15	42	56,406	250 KB	SWIR 15m
16	42	50,368	350 KB	SWIR 30m

Table 9.7 Verification performance with SCRFD, AdaptiveWingLoss and ArcFace loss

Domain	Rank 1	Rank 2	Rank 5	Rank 10
(3) Visible 500m	20.8%	25.5%	33.2%	41.8%
(4) Visible 400m	95.0%	97.1%	98.6%	99.1%
(5) Visible 300m	98.5%	99.3%	99.7%	99.9%
(6) Visible 500m 400m walking	57.8%	64.4%	71.6%	77.5%
(3, 4, 5, 6) together	76.9%	79.5%	82.5%	85.1%

We use the AdaptiveWingLoss [53] algorithm on the cropped faces to detect the face key points. Then we perform face alignment and use the resulting images for feature extraction. For these experiments, we use a model trained on visible data (using ArcFace loss [11]) to extract features from the remote frames and evaluate face verification performance between the remote frames (probe set) and the visible enrollment images (gallery set).

Using only the frames that match the ground truth frames (removing false positives), the verification performance is shown in Table 9.7.

We can see from the results that the model adapts well to videos at 300m and 400m, but there is a definite drop in performance as we go from 400m to 500m.

9.4.5 Discussions

For the IJB-B dataset, we can see that the proposed system performs consistently better than all the results in [5] and the baseline **Cos** on identification accuracy. For open-set metric TPIR/FPIR, the proposed quality-aware cosine similarity achieves better results, but the proposed subspace similarity metric still performs better than [5] with a large margin. For the IJB-S dataset, we have similar observations: the proposed system with subspace-to-subspace similarity metric performs better than **Cos** on surveillance-to-single and surveillance-to-booking protocols, by a relatively large margin. It also achieves better accuracy than **Cos** on the surveillance-to-surveillance protocol. We notice that the fusion of Network D and E does not work well on surveillance-to-surveillance protocol, especially at higher rank accuracy. Such observations are consistent under both tracklets filtering configurations and their corresponding metrics: **with Filtering** with Top-K average accuracy and **without Filtering** with the EERR metric. The proposed system also outperforms ArcFace with a larger margin in surveillance-to-single and surveillance-to-booking protocols of IJB-S. For MBGC and FOCS datasets, from the tables and plots we can see that in general, the proposed approach performs better than **Cos** baseline, **DFRV**$_{deep}$, **DFRV**$_{px}$ and **Hybrid**.

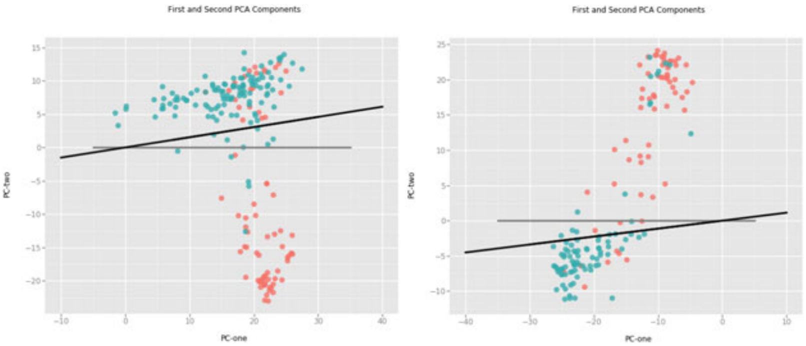


Fig. 9.9 Visualization of example templates in IJB-S. Each sample is a dot in the plot with its first two principal components as the coordinates. Samples with $d_i \geq 0.7$ are in **blue** dots and the rest samples are in **red** dots. **Gray** line and **black** line are the projection of the first subspace basis learned by **Sub** and **QSub** respectively

Figure 9.9 shows the visualization of two templates in IJB-S dataset in PCA-subspace, which illustrates the advantage of the proposed subspace learning method. In the plot, each dot corresponds to a sample in the template, where x- and y-axes correspond to the first two principal components of the samples, learned from each template respectively. Relatively high-quality detections with detection scores greater than or equal to 0.7 are represented by blue dots. Relatively low-quality detections with detection scores less than 0.7 are represented by red dots. The projections of the first subspace bases learned by **Sub** and the proposed **QSub** onto the PCA-subspace are gray and black straight lines in the plot, respectively. From the plot, we can see that, with quality-aware subspace learning, the subspaces learned by the proposed method put more weight on the high-quality sample. It fits the high-quality samples better than the low-quality ones. But the plain PCA takes each sample into account equally, which is harmful to the representation of the template.

We also compare our system with other baseline methods as part of an ablation study, from baseline cosine similarity **Cos** to the proposed quality-aware subspace-to-subspace similarity **QCos+QSub-VPM**. As we gradually modify the method by including quality-aware cosine similarity **QCos**, quality-aware subspace learning **QSub**, and variance-aware projection metric **VPM**, we can see the performance also gradually improves, especially for IJB-B and IJB-S datasets.

From the results above, we observe the following:

- The proposed system performs the best in general, which shows the effectiveness of (1) learning subspace as template representation, (2) matching video pairs using the subspace-to-subspace similarity metric and (3) utilizing quality and variance information to compute exemplars, learn subspaces and measure similarity.

- **QCos** generally performs better than **Cos**, which shows that quality-aware exemplars weigh the samples according to their quality and better represent the image sets than plain average exemplars.
- In most of the cases, **Cos+Sub-PM** achieve higher performance than **Cos**. It implies that a subspace can utilize the correlation information between samples and is a good complementary representation of exemplars as global information.
- **QCos+QSub-PM** performs better than **QCos+Sub-PM** in general. It shows that similar to **QCos**, we can learn more representative subspaces based on the quality of samples.
- **QCos+QSub-VPM** works better than **QCos+QSub-PM** in most of the experiments. It implies that by considering the variances of bases in the subspaces, **VPM** similarity is more robust to variations in the image sets.
- The improvement of the proposed system over the compared algorithms is consistent under both **with filtering** and **without filering** configurations on the IJB-S dataset. It shows that our method is effective for both high-quality and low-quality tracklets in surveillance videos.
- For IJB-S, the performance on surveillance-to-surveillance protocol is in general lower than the performance on other protocols. This is because the gallery templates of this protocol are constructed from low-quality surveillance videos, while the remaining two protocols have galleries from high-resolution still images.
- The fusion of Network D and E does not perform as well as single Network D on surveillance-to-surveillance protocol, especially at higher rank accuracy. It is probably because of the low-quality galleries in this protocol which Network E cannot represent well.
- On IJB-S, the proposed method performs better than state-of-the-art network ArcFace [11] in general, especially on surveillance-to-single and surveillance-to-booking protocols, which shows the discriminative power of the features from the proposed networks. ArcFace still performs better on surveillance-to-surveillance protocol. But the results also show that using the quality-aware subspace-to-subspace similarity improves the performance for ArcFace features as well.
- On MBGC and FOCS, ArcFace performs better in the walking-vs-walking protocol but Network D outperforms ArcFace on more challenging protocols like activity-vs-activity. Also, by applying the proposed subspace-to-subspace similarity on both features, the performance consistently improves, which shows its effectiveness on different datasets and using different features.
- For the FOCS dataset, the performance of our system surpasses the human performance, which again demonstrates the effectiveness of the proposed system.

9.5 Concluding Remarks

In this chapter, we proposed an automatic face recognition system for unconstrained video-based face recognition tasks. The proposed system learns subspaces to represent video faces and matches video pairs by subspace-to-subspace similarity metrics. We evaluated our system on four video datasets and the experimental results demonstrate the superior performance of the proposed system.

References

1. Arandjelovic, O., Shakhnarovich, G., Fisher, J., Cipolla, R., Darrell, T.: Face recognition with image sets using manifold density divergence. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 581–588. IEEE (2005)
2. Bansal, A., Nanduri, A., Castillo, C.D., Ranjan, R., Chellappa, R.: Umdfaces: An annotated face dataset for training deep networks. In: IEEE International Joint Conference on Biometrics (IJCB) (2017)
3. Basri, R., Jacobs, D.W.: Lambertian reflectance and linear subspaces. IEEE Trans. Pattern Anal. Mach. Intell. **25**(2), 218–233 (2003)
4. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: ICIP, pp. 3464–3468 (2016)
5. Chen, C.H., Chen, J.C., Castillo, C.D., Chellappa, R.: Video-based face association and identification. In: 12th FG, pp. 149–156 (2017)
6. Chen, J.C., Lin, W.A., Zheng, J., Chellappa, R.: A real-time multi-task single shot face detector. In: ICIP (2018)
7. Chen, J.C., Patel, V.M., Chellappa, R.: Unconstrained face verification using deep CNN features. In: WACV (2016)
8. Chen, J.C., Ranjan, R., Sankaranarayanan, S., Kumar, A., Chen, C.H., Patel, V.M., Castillo, C.D., Chellappa, R.: Unconstrained still/video-based face verification with deep convolutional neural networks. IJCV **126**(2), 272–291 (2018)
9. Chen, Y.C., Patel, V.M., Phillips, P.J., Chellappa, R.: Dictionary-based face recognition from video. In: ECCV (2012)
10. Chen, Y.C., Patel, V.M., Phillips, P.J., Chellappa, R.: Dictionary-based face and person recognition from unconstrained video. IEEE Access **3**, 1783–1798 (2015)
11. Deng, J., Guo, J., Niannan, X., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019)
12. Ding, C., Tao, D.: Trunk-branch ensemble convolutional neural networks for video-based face recognition. CoRR http://arxiv.org/abs/1607.05427 (2016)
13. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(2), 303–353 (1998)
14. Gong, S., Shi, Y., Jain, A.: Low quality video face recognition: Multi-mode aggregation recurrent network (marn). In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 0–0 (2019)
15. Gong, S., Shi, Y., Jain, A.K., Kalka, N.D.: Recurrent embedding aggregation network for video face recognition. CoRR http://arxiv.org/abs/1904.12019 (2019)

16. Gong, S., Shi, Y., Kalka, N.D., Jain, A.K.: Video face recognition: Component-wise feature aggregation network (c-fan). In: 2019 International Conference on Biometrics (ICB), pp. 1–8. IEEE (2019)
17. Gong, S., Shi, Y., Kalka, N.D., Jain, A.K.: Video face recognition: Component-wise feature aggregation network (C-FAN). CoRR http://arxiv.org/abs/1902.07327 (2019)
18. Guo, J., Deng, J., Lattas, A., Zafeiriou, S.: Sample and computation redistribution for efficient face detection. ArXiv preprint arXiv:2105.04714 (2021)
19. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In: ECCV (2016)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. ArXiv preprint arXiv:1506.01497 (2015)
21. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Tech. rep (2007)
22. Kalka, N.D., Duncan, J.A., Dawson, J., Otto, C.: Iarpa janus benchmark multi-domain face. In: 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp. 1–9. IEEE (2019)
23. Kalka, N.D., Maze, B., Duncan, J.A., O'Connor, K.J., Elliott, S., Hebert, K., Bryan, J., Jain, A.K.: IJB-S : IARPA Janus Surveillance Video Benchmark (2018)
24. Klare, B.F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Jain, A.K.: Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In: CVPR (2015)
25. Li, Y., Gong, S., Liddell, H.: Video-based online face recognition using identity surfaces. In: Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp. 40–46. IEEE (2001)
26. Liu, X., Chen, T., Thornton, S.M.: Eigenspace updating for non-stationary process and its application to face recognition. Pattern Recogn. **36**(9), 1945–1959 (2003)
27. Liu, X., Kumar, B., Yang, C., Tang, Q., You, J.: Dependency-aware attention control for unconstrained face recognition with image sets. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 548–565 (2018)
28. Liu, Y., Junjie, Y., Ouyang, W.: Quality aware network for set to set recognition. In: CVPR (2017)
29. Liu, Z., Hu, H., Bai, J., Li, S., Lian, S.: Feature aggregation network for video face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
30. Information Technology Laboratory, NIST.: Multiple Biomertic Grand Challenge http://www.nist.gov/itl/iad/ig/mbgc.cfm
31. Mei, T., Yang, B., Yang, S.Q., Hua, X.S.: Video collage: presenting a video sequence using a single image. Vis. Comput. **25**(1), 39–51 (2009)
32. O'Toole, A.J., Harms, J., Snow, S.L., Hurst, D.R., Pappas, M.R., Ayyad, J.H., Abdi, H.: A video database of moving faces and people. TPAMI **27**(5) (2005)
33. O'Toole, A.J., Phillips, P.J., Weimer, S., Roark, D.A., Ayyad, J., Barwick, R., Dunlop, J.: Recognizing people from dynamic and static faces and bodies: Dissecting identity with a fusion approach. Vis. Res. **51**(1) (2011)
34. Parchami, M., Bashbaghi, S., Granger, E., Sayed, S.: Using deep autoencoders to learn robust domain-invariant representations for still-to-video face recognition. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. IEEE (2017)
35. Park, U., Jain, A.K., Ross, A.: Face recognition in video: Adaptive fusion of multiple matchers. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
36. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC (2015)

37. Ranjan, R., Bansal, A., Xu, H., Sankaranarayanan, S., Chen, J., Castillo, C.D., Chellappa, R.: Crystal loss and quality pooling for unconstrained face verification and recognition. CoRR http://arxiv.org/abs/1804.01159 (2018)
38. Ranjan, R., Bansal, A., Zheng, J., Xu, H., Gleason, J., Lu, B., Nanduri, A., Chen, J.C., Castillo, C., Chellappa, R.: A fast and accurate system for face detection, identification, and verification. CoRR http://arxiv.org/abs/1809.07586 (2018)
39. Ranjan, R., Sankaranarayanan, S., Bansal, A., Bodla, N., Chen, J.C., Patel, V.M., Castillo, C.D., Chellappa, R.: Deep learning for understanding faces: Machines may be just as good, or better, than humans. IEEE Sig. Process. Mag. **35**, 66–83 (2018)
40. Ranjan, R., Sankaranarayanan, S., Castillo, C.D., Chellappa, R.: An all-in-one convolutional neural network for face analysis. In: 12th IEEE FG, vol. 00, pp. 17–24 (2017)
41. Rao, Y., Lin, J., Lu, J., Zhou, J.: Learning discriminative aggregation network for video-based face recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3781–3790 (2017)
42. Rao, Y., Lu, J., Zhou, J.: Attention-aware deep reinforcement learning for video face recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3931–3940 (2017)
43. Sankaranarayanan, S., Alavi, A., Castillo, C.D., Chellappa, R.: Triplet probabilistic embedding for face verification and clustering. CoRR http://arxiv.org/abs/1604.05417 (2016)
44. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR (2015)
45. Shakhnarovich, G., Fisher, J.W., Darrell, T.: Face recognition from long-term observations. In: European Conference on Computer Vision, pp. 851–865. Springer (2002)
46. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: NIPS (2014)
47. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: CVPR (2015)
48. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI (2017)
49. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: CVPR (2014)
50. Wang, R., Chen, X.: Manifold discriminant analysis. In: CVPRW, pp. 429–436 (2009)
51. Wang, R., Guo, H., Davis, L.S., Dai, Q.: Covariance discriminative learning: A natural and efficient approach to image set classification. In: CVPR, pp. 2496–2503 (2012)
52. Wang, R., Shan, S., Chen, X., Dai, Q., Gao, W.: Manifold-manifold distance and its application to face recognition with image sets. IEEE TIP **21**(10) (2012)
53. Wang, X., Bo, L., Fuxin, L.: Adaptive wing loss for robust face alignment via heatmap regression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6971–6981 (2019)
54. Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A.K., Duncan, J.A., Allen, K.: IARPA Janus Benchmark-B face dataset. In: CVPRW (2017)
55. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: CVPR, pp. 529–534 (2011)
56. Xu, Y., Roy-Chowdhury, A., Patel, K.: Pose and illumination invariant face recognition in video. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7. IEEE (2007)
57. Yang, J., Ren, P., Zhang, D., Chen, D., Wen, F., Li, H., Hua, G.: Neural aggregation network for video face recognition. In: CVPR, pp. 4362–4371 (2017)
58. Yang, J., Ren, P., Zhang, D., Chen, D., Wen, F., Li, H., Hua, G.: Neural aggregation network for video face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4362–4371 (2017)

59. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. CoRR http://arxiv.org/abs/1411.7923 (2014)
60. Zheng, J., Chen, J.C., Patel, V.M., Castillo, C.D., Chellappa, R.: Hybrid dictionary learning and matching for video-based face verification. In: BTAS (2019)
61. Zheng, J., Ranjan, R., Chen, C.H., Chen, J.C., Castillo, C.D., Chellappa, R.: An automatic system for unconstrained video-based face recognition. CoRR http://arxiv.org/abs/1812.04058 (2018)

Face Recognition with Synthetic Data

10

Haibo Qiu, Baosheng Yu, Dihong Gong, Zhifeng Li, Wei Liu, and Dacheng Tao

10.1 Introduction

In the last few years, face recognition has achieved extraordinary progress in a wide range of challenging problems including pose-robust face recognition [5, 24, 63], matching faces across ages [15, 17, 56, 60], across modalities [13, 14, 16, 30, 31], and occlusions [40, 49, 71]. Among these progresses, not only the very deep neural networks [22, 25, 29, 48] and sophisticated design of loss functions [10, 23, 32, 57, 61], but also large-scale training datasets [20, 26, 27] play important roles. However, it has turned out to be very difficult to further boost the performance of face recognition with the increasing number of training images collected from the Internet, especially due to the severe label noise and privacy issues [20, 55, 59]. For example, several large-scale face recognition datasets are struggling with the consent of all involved person/identities, or even have to close the access of face

H. Qiu · B. Yu (✉) · D. Tao
The University of Sydney, Sydney, Australia
e-mail: baosheng.yu@sydney.edu.au

H. Qiu
e-mail: hqiu2518@sydney.edu.au

D. Tao
e-mail: dacheng.tao@sydney.edu.au

D. Gong · Z. Li · W. Liu
Tencent Data Platform, Shenzhen, China
e-mail: gongdihong@gmail.com

Z. Li
e-mail: michaelzfli@tencent.com

W. Liu
e-mail: wl2223@columbia.edu

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_10

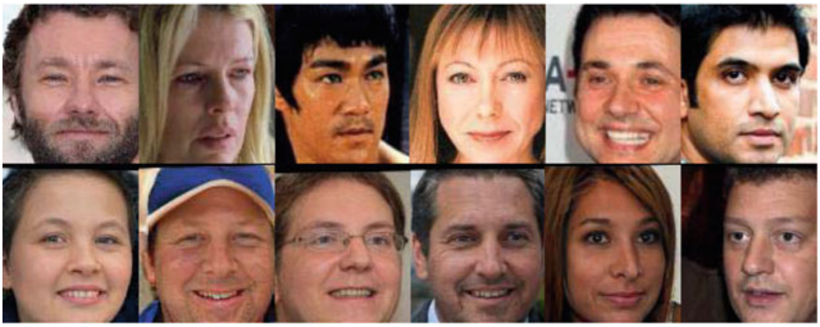


Fig. 10.1 Examples of real/synthetic face images. The first row indicates real face images from CASIA-WebFace, and the second row shows synthetic face images generated by DiscoFaceGAN [11] with the proposed identity mixup module

data from the website [20]. Meanwhile, many face training datasets also suffer from the long-tailed problem, i.e., head classes with a large number of samples and tail classes with a few number of samples [34, 37, 72]. To utilize these datasets for face recognition, people need to carefully design the network architectures and/or loss functions to alleviate the degradation on model generalizability brought by the long-tailed problem. Furthermore, the above-mentioned issues also make it difficult for people to explore the influences of different attributes (e.g., expression, pose, and illumination).

Recently, face synthesis using GANs [18] and 3DMM [3] have received increasing attention from the computer vision community, and existing methods usually focus on generating high-quality identity-preserving face images [2, 47, 65]. Some synthetic and real face images are demonstrated in Fig. 10.1. However, the problem of face recognition using synthetic face images has not been well-investigated [28, 53]. Specifically, Trigueros et al. [53] investigated the feasibility of data augmentation with photo-realistic synthetic images. Kortylewski et al. [28] further explored the pose-varying synthetic images to reduce the negative effects of dataset bias. Lately, disentangled face generation has become popular [11], which can provide the precise control of targeted face properties such as identity, pose, expression, and illumination, thus making it possible for us to systematically explore the impacts of facial properties on face recognition. Specifically, with a controllable face synthesis model, we are then capable of (1) collecting large-scale face images of non-existing identities without the risk of privacy issues; (2) exploring the impacts of different face dataset properties, such as the depth (the number of samples per identity) and the width (the number of identities); (3) analyzing the influences of different facial attributes (e.g., expression, pose, and illumination).

Despite the success of face synthesis, there is usually a significant performance gap between the models trained on synthetic and real face datasets. Through the empirical analysis, we find that (1) the poor intra-class variations in synthetic face images and (2) the domain gap between synthetic and real face datasets are the main reasons for the performance degradation. To address the above issues, we introduce identity mixup (IM) into the disentangled face generator to enlarge the intra-class variations of generated face images. Specifically, we use a convex combination of the coefficients from two different identities to form a new intermediate identity coefficient for synthetic face generation. Experimental results in Sect. 10.4 show that the identity mixup significantly improves the performance of the model trained on synthetic face images. Furthermore, we observe a significant domain gap via cross-domain evaluation: (1) training on synthetic face images and testing on real face images; (2) training on real face images and testing on synthetic face images (see more details in Sect. 10.3.2). Therefore, we further introduce the domain mixup (DM) to alleviate the domain gap, i.e., by using a convex combination of images from a large-scale synthetic dataset and a relatively small number of real face images during training. With the proposed identity mixup and domain mixup, we achieve a significant improvement over the vanilla SynFace, further pushing the boundary of face recognition performance using synthetic data.

The remainder of this chapter is structured as follows. Section 10.2 reviews existing visual tasks using synthetic data and summarizes the recent advancements on face synthesis and face recognition. Section 10.3 introduces a typical pipeline for deep face recognition with synthetic face images. Specifically, vanilla deep face recognition is introduced in Sect. 10.3.1 and the performance gap between the models trained on real and synthetic face images is described in Sect. 10.3.2. We show that the above-mentioned performance gap can be narrowed by enlarging the intra-class variations via identity mixup in Sect. 10.3.3; and leveraging a few real face images for domain adaption via domain mixup in Sect. 10.3.4. Lastly, in Sect. 10.4, (1) we discuss the impacts of synthetic datasets with different properties for face recognition, e.g., depth (the number of samples per identity) and width (the number of identities), and reveal that the width plays a more important role; (2) we systematically analyze the influences of different facial attributes on face recognition (e.g., facial pose, expression, and illumination).

10.2 Related Work

In this section, we first briefly introduce visual tasks using synthetic data. Then recent face synthesis and recognition methods are reviewed. Lastly, we discuss the mixup and its variants to indicate their relationships and differences between the proposed identity mixup and domain mixup.

Synthetic Data. Synthetic data for computer vision tasks has been widely explored, e.g., crowd counting [58], vehicle re-identification [52], semantic segmentation [6, 44, 45], 3D face reconstruction [43] and face recognition [28, 53]. According to the motivation, existing methods can be categorized into three groups: (1) It is time-consuming and expensive to collect and annotate large-scale training data [6, 43–45]; (2) It can be used to further improve the model trained on a real dataset [28, 53]; (3) It can be used to systematically analyze the impacts of different dataset attributes [28]. Among these works, [28] is the most related one to our work, while it only discusses the impacts of different head poses. Apart from facial attributes (e.g., pose, expression, and illumination), we also explore the impacts of the width and the depth of training dataset. Furthermore, we introduce identity mixup (IM) and domain mixup (DM) to increase the intra-class variations and narrow down the domain gap, leading to a significant improvement.

Face Synthesis. With the great success of GANs [1, 7, 18, 35, 36, 39, 42], face synthesis has received increasing attention and several methods have been proposed to generate identity-preserving face images [2, 47, 65]. Specifically, FF-GAN [65] utilizes 3D priors (e.g., 3DMM [3]) for high-quality face frontalization. Bao et al.[2] first disentangled identity/attributes from the face image, and then recombined different identities/attributes for identity-preserving face synthesis. FaceID-GAN [47] aims to generate identity-preserving faces by using a classifier (C) as the third player, competing with the generator (G) and cooperating with the discriminator (D). However, unlike exploring the identity-preserving property, generating face images from multiple disentangled latent spaces (i.e., different facial attributes) has not been well-investigated. Recently, DiscoFaceGAN [11] introduces a novel disentangled learning scheme for face image generation via an imitative-contrastive paradigm using 3D priors. Thus, it further enables a precise control of targeted face properties such as unknown identities, pose, expression, and illumination, yielding the flexible and high-quality face image generation.

Deep Face Recognition. Recent face recognition methods mainly focus on delivering novel loss functions for robust face recognition in the wild. The main idea is to maximize the inter-class variations and minimize the intra-class variations. For example, (1) contrastive loss [8, 21] and triplet loss [23, 66] are usually utilized to increase the Euclidean margin for better feature embedding; (2) center loss [61] aims to learn a center for each identity and then minimizes the center-aware intra-class variations; (3) Large-margin softmax loss [32, 33] and its variants such as CosFace [57] and ArcFace [10] improve the feature discrimination by adding marginal constraints to each identity.

Mixup. Mixup [68] uses the convex combinations of two data samples as a new sample for training, regularizing deep neural networks to favor a simple linear behavior in-between training samples. Vanilla mixup is usually employed on image pixels, while the generated data samples are not consistent with the real images, e.g., a mixup of two face images in the pixel level does not always form a proper new face image. Inspired by this, we introduce identity mixup to face generator via the identity coefficients, where a convex combination of two identities forms a new identity in the disentangled latent space. With the proposed identity mixup, we are also able to generate high-fidelity face images correspondingly. Recently, several mixup variants have been proposed to perform feature-level interpolation [19, 50, 51, 54], while [62] further leverages domain mixup to perform adversarial domain adaptation. Inspired by this, we perform domain adaption via domain mixup between real and synthetic face images, while the main difference is that [62] uses the mixup ratio to guide the model training, but we utilize the identity labels of both synthetic and real face images as the supervision for face recognition.

10.3 Method

In this section, we introduce face recognition with synthetic data, i.e., SynFace, and the overall pipeline is illustrated in Fig. 10.2. We first introduce deep face recognition using margin-based softmax loss functions. We then explore the performance gap between the models trained on synthetic and real datasets (SynFace and RealFace). Lastly, we introduce (1) identity mixup to enlarge the intra-class variations and (2) domain mixup to mitigate the domain gap between synthetic and real faces images.

10.3.1 Deep Face Recognition

With the great success of deep neural networks, deep learning-based embedding learning has become the mainstream technology for face recognition to maximize the inter-class variations and minimize the intra-class variations [8, 21, 23, 33]. Recently, margin-based softmax loss functions have been very popular in face recognition due to their simplicity and excellent performance, which explicitly explore the margin penalty between inter- and intra-class variations via a reformulation of softmax-based loss function [10, 32, 57, 67]. Similar to [10], we use a unified formulation for margin-based softmax loss functions as follows:

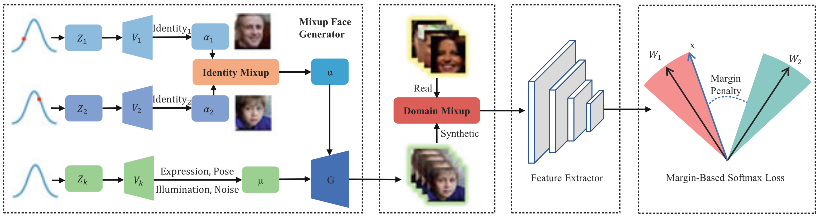


Fig. 10.2 An overview of the proposed SynFace. Firstly, the identity mixup is introduced into DiscoFaceGAN [11] to form the Mixup Face Generator, which can generate face images with different identities and their intermediate states. Next, the synthetic face images are cooperating with a few real face images via domain mixup to alleviate the domain gap. Then, the feature extractor takes the mixed face images as input and extracts the corresponding features. The extracted features are either utilized to calculate the margin-based softmax loss (where W_1, W_2 are the center weight vectors for two different classes and x is the feature vector) for model training, or employed as the face representations to perform face identification and verification tasks

$$\mathcal{L}_{margin} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cdot\delta}}{e^{s\cdot\delta}+\sum_{j\neq y_i}^{n} e^{s\cos\theta_j}}, \tag{10.1}$$

where $\delta = \cos(m_1\theta_{y_i} + m_2) - m_3, m_{1,2,3}$ are margins, N is the number of training samples, θ_j indicates the angle between the weight W_j and the feature x_i, y_i represents the ground-truth class, and s is the scale factor. Specifically, for SphereFace [32], ArcFace [10] and CosFace [57], we have the coefficients $(m_1, 0, 0)$, $(0, m_2, 0)$, and $(0, 0, m_3)$, respectively, and we use ArcFace [10] as our baseline.

10.3.2 SynFace Versus RealFace

To explore the performance gap between SynFace and RealFace, as well as the underlying causes, we perform experiments on real-world face datasets and synthetic face datasets generated by DiscoFaceGAN [11]. Specifically, for real-world face datasets, we use CASIA-WebFace [64] for training and LFW [26] for testing. For the fair comparison, we generate the synthetic version of the LFW dataset, Syn-LFW, using the same parameters (the number of samples, the number of identities, distributions of expression, pose, and illumination). For synthetic training data, we generate 10 K different identities with 50 samples per identity to form a comparable training dataset to CASIA-WebFace (containing 494,414 images from 10,575 subjects) and we refer to it as Syn_10K_50. More details of synthetic dataset construction can be found in Sect. 10.4.1. With both synthetic and real face images, we then perform the cross-domain evaluation as follows. We train two face recognition models on CASIA-WebFace and Syn_10K_50, and test them on LFW and Syn-LFW, respectively. As shown in Table 10.1, there is a clear performance gap (88.98% versus 99.18%) when testing on LFW. Meanwhile, SynFace outperforms RealFace on Syn-LFW (99.98% versus 98.85%). These observations suggest that the domain gap between synthetic and real face images contributes to the performance gap between SynFace and RealFace.

We compare the face images between Syn_10K_50 and CASIA-WebFace, and find that the synthetic face images usually lack the intra-class variations, which may be one of the reasons for the performance degradation (please refer to the supplementary materials for more illuminations). Furthermore, we also visualize the distributions of feature embeddings by using multidimensional scaling (MDS [4]) to convert the 512-dimensional feature vector

Table 10.1 The cross-domain evaluation of SynFace and RealFace using the metric of face verification accuracy (%)

Method	Training dataset	LFW	Syn-LFW
RealFace	CASIA-WebFace	99.18	98.85
SynFace	Syn_10K_50	88.98	99.98

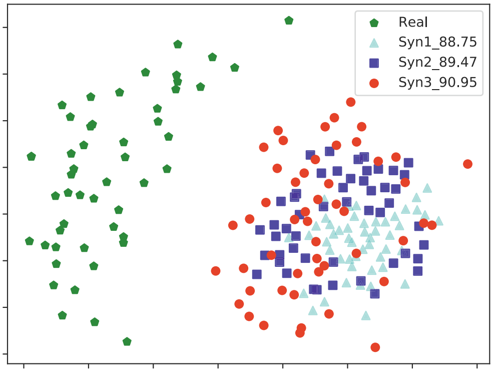


Fig. 10.3 Visualization of the feature distributions (using MDS [4]) for the samples from three different synthetic datasets (Syn1, Syn2, and Syn3) and CASIA-WebFace, which are illustrated by the cyan triangles, blue square, red circle, and green pentagon, respectively. Note that the intra-class variations of Syn1, Syn2, and Syn3 are increasing, which lead to the consistent improvements on accuracy (88.75% → 89.47% → 90.95%). Best viewed in color

into 2D space. As shown in Fig. 10.3, we randomly select 50 samples from two different classes of Syn_10K_50 and CASIA-WebFace, respectively. In particular, we observe that the cyan triangles have a much more compact distribution than the green pentagons, suggesting the poor intra-class variations in Syn_10K_50.

10.3.3 SynFace with Identity Mixup

To increase the intra-class variations of synthetic face images, we incorporate the identity mixup into DiscoFaceGAN [11] to form a new face generator for face recognition, i.e., the Mixup Face Generator, which is capable of generating different identities and their intermediate states. In this subsection, we first briefly discuss the mechanism of DiscoFaceGAN, and we then introduce how to incorporate the proposed identity mixup into the face generator.

Face Generation. DiscoFaceGAN [11] can provide the disentangled, precisely-controllable latent representations for the identity of non-existing people, expression, pose, and illumination to generated face images. Specifically, it generates realistic face images x from random noise z, which consists of five independent variables $z_i \in \mathbb{R}^{N_i}$, and each of them follows a standard normal distribution. The above five independent variables indicate independent factors for face generation: identity, expression, illumination, pose, and random noise accounting for other properties such as the background. Let $\lambda \doteq [\alpha, \beta, \gamma, \theta]$ denote the latent factors, where α, β, γ and θ indicate the identity, expression, illumination, and pose coefficient, respectively. Four simple VAEs [9] of α, β, γ, and θ are then trained for z-space to λ-space mapping, which enables training the generator to imitate the rendered faces from 3DMM [3]. The pipeline of generating a face image is to (1) first randomly sample latent variables from the standard normal distribution, (2) then feed them into the trained VAEs to obtain α, β, γ and θ coefficients, and (3) synthesize the corresponding face image by the generator using these coefficients.

Identity Mixup (IM). Inspired by the reenactment of face images [69], we propose to enlarge the intra-class variations by interpolating two different identities as a new intermediate one with changing the label correspondingly. Recalling that the coefficient α controls the identity characteristic, we thus interpolate two different identity coefficients to generate a new intermediate identity coefficient. Mathematically, it can be formulated as follows:

$$\begin{aligned} \alpha &= \varphi \cdot \alpha_1 + (1 - \varphi) \cdot \alpha_2, \\ \eta &= \varphi \cdot \eta_1 + (1 - \varphi) \cdot \eta_2, \end{aligned} \tag{10.2}$$

where α_1, α_2 are two random identity coefficients from λ-space, and η_1, η_2 are the corresponding class labels. Note that the weighted ratio φ is randomly sampled from the linear space which varies from 0.0 to 1.0 with interval being 0.05 (i.e., $np.linspace(0.0, 1.0, 21)$). Comparing to the vanilla mixup [68] which is employed at the pixel level, the proposed mixup is operating on the identity coefficient latent space, denoted as identity mixup (IM), which enlarges the intra-class variations by linearly interpolating different identities, forming the Mixup Face Generator. However, both of them can regularize the model to favor the simple linear behavior in-between training samples.

As illustrated in Fig. 10.2, the pipeline of Mixup Face Generator is first randomly sampling two different identity latent variables from the standard normal distribution, and then feeding them to the trained VAEs to obtain α_1, α_2 coefficients. The mixed identity coefficient α is obtained by identity mixup with α_1, α_2 according to Eq. (10.2), the corresponding face image is finally synthesized by the generator with α, μ coefficients (where $\mu \doteq [\beta, \gamma, \theta]$). We also visualize two groups of identity interpolation with identity mixup in Fig. 10.4. As we can see, one identity gradually and smoothly transforms to another identity as the weighted ratio φ varies from 0 to 1. Besides, it is obvious that the face images generated with intermediate identity coefficients are also high-quality.

To evaluate the identity mixup for enlarging the intra-class variations, as illustrated in Fig. 10.3, we visualize the feature embedding distributions of the same class in three synthetic datasets (containing 5 K different identities with 50 samples per identity) with different levels of identity mixup (IM) by using multidimensional scaling (MDS [4]). Note that Syn1, Syn2, and Syn3 represent the weighted ratio φ is 1.0 (i.e., no IM), 0.8 and randomly sampled from the linear space which varies from 0.6 to 1.0 with the interval being 0.05 (i.e., $np.linspace(0.6, 1.0, 11)$). It is clear that the cyan triangles (Syn1) have the smallest variations , while the red circles (Syn3) have the largest one, and the blue squares (Syn2) are in the middle position. Accordingly, the accuracy is in an increasing trend (i.e., 88.75% → 89.47% → 90.95%). Besides, 88.98% (as in Table 10.1) is boosted to 91.97% (as in Table 10.2) after utilizing identity mixup. In particular, when the baseline is weaker,

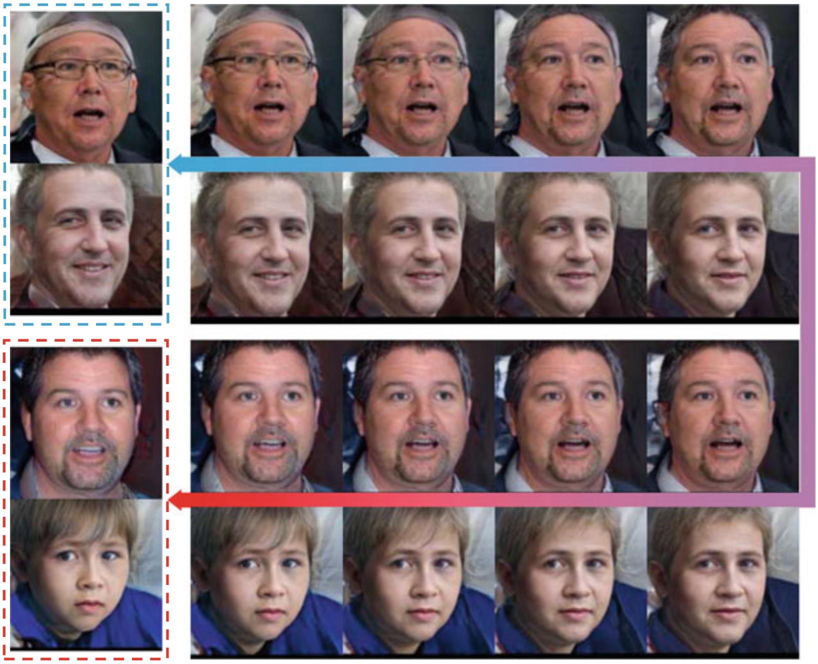


Fig. 10.4 Examples of an identity gradually and smoothly varying to another identity as the weighted ratio φ varies from 0 to 1

the improvement brought by identity mixup is larger, which is shown in Table 10.3 and Fig. 10.7.

In addition to identity mixup in the training process, we also make an attempt of employing identity mixup on the synthetic testing dataset to evaluate the model's robustness on the identity coefficient noises. Specifically, both RealFace (trained on CASIA-WebFace) and SynFace_IM (trained on Syn_10K_50 with identity mixup) are evaluated on five different synthetic testing datasets, as illustrated in Fig. 10.5. Note that Syn-LFW is the synthetic version of the LFW dataset, while Syn-LFW-R (with R $\in [0.6, 0.7, 0.8, 0.9]$) indicates employing the identity mixup with the weighted ratio R during the generation of Syn-LFW. Specifically, we mix the primary class with a random secondary class using the ratio R according to Eq. (10.2), but we keep the original label unchanged. Apparently, when R is smaller (i.e., the weight of the primary class is smaller), the corresponding testing dataset is more difficult to recognize because the secondary class impacts the identity information more heavily.

From the results of Fig. 10.5, we can find that our SynFace_IM achieves nearly perfect accuracy when R is larger than 0.6 and also obtains an impressive 97.30% result which remarkably outperforms the 87.83% accuracy by RealFace when R is 0.6. On the other hand, the accuracy of RealFace drops significantly on Syn-LFW-R when R becomes small,

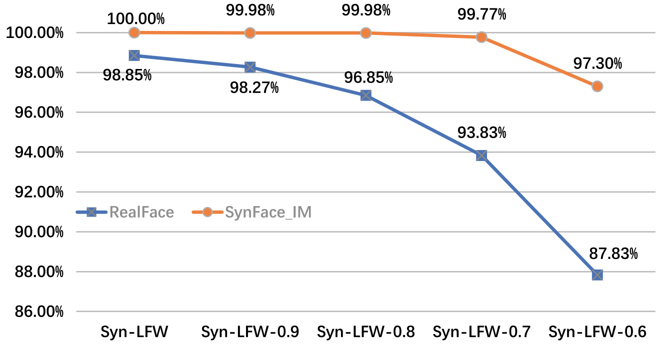


Fig. 10.5 Face verification accuracy comparison between RealFace and SynFace_IM (i.e., SynFace with Identity Mixup) on five different synthetic testing datasets. Syn-LFW is the synthetic version of the LFW dataset, while Syn-LFW-R (with R ∈ [0.6, 0.7, 0.8, 0.9]) indicates introducing identity mixup with ratio R into Syn-LFW

which suggests that the domain gap between real and synthetic face data is still large even after employing the identity mixup. Another interesting conclusion is that the current state-of-the-art face recognition model (i.e., RealFace) cannot handle the identity mixup attack. In other words, if a face image is mixup with another identity, the model cannot recognize it well. However, the proposed SynFace with identity mixup can nearly keep the accuracy under the identity mixup attack. We prefer to explore how to make the RealFace handle such an attack in future work.

10.3.4 SynFace with Domain Mixup

The lack of intra-class variation is an observable cause of the domain gap between synthetic and real faces, and SynFace can be significantly improved by the proposed identity mixup. To further narrow the performance gap between SynFace and RealFace, we introduce the domain mixup as a general domain adaptation method to alleviate the domain gap for face recognition. Specifically, we utilize large-scale synthetic face images with a small number of real-world face images with labels as the training data. When training, we perform mixup within a mini-batch of synthetic images and a mini-batch of real images, where the labels are changed accordingly as the supervision. Mathematically, the domain mixup can be formulated as follows:

$$\begin{aligned} X &= \psi \cdot X_S + (1-\psi) \cdot X_R, \\ Y &= \psi \cdot Y_S + (1-\psi) \cdot Y_R, \end{aligned} \tag{10.3}$$

where X_S, X_R indicate the synthetic and real face images, respectively, and Y_S, Y_R indicate their corresponding labels. Note that ψ is the mixup ratio which is randomly sampled from the linear space distribution from 0.0 to 1.0 with the interval being 0.05

Table 10.2 Face verification accuracies (%) of models trained on synthetic, real and mixed datasets on LFW. R_ID means the number of real identities

Method	R_ID	Samples per R_ID	Accuracy
Syn_10K_50	0	0	91.97
Real_1K_10	1K	10	87.50
Mix_1K_10	1K	10	**92.28**
Real_1K_20	1K	20	92.53
Mix_1K_20	1K	20	**95.05**
Real_2K_10	2K	10	91.22
Mix_2K_10	2K	10	**95.78**

(i.e., $np.linspace(0.0, 1.0, 21)$). For the large-scale synthetic data, we synthesize the Syn_10K_50 dataset that has 10 K different identities with 50 samples per identity. For a small set of real-world data, we utilize the first 2 K identities of CASIA-WebFace. The experimental results are shown in Table 10.2. Specifically, the first row, Syn_10K_50, indicating the baseline method without using any real face images, achieves the accuracy 91.97% using identity mixup. "Real_N_S" means the use of only real images, N identities with S samples per identity during training, while "Mix_N_S" indicates a mixture of N real identities with S samples per identity with Syn_10K_50 during training. Both identity mixup and domain mixup are employed on all the 'Mix_N_S" datasets. As demonstrated in Table 10.2, domain mixup brings a significant and consistent improvement over the baseline methods under different settings. For example, Mix_2K_10 obtains 95.78% accuracy, which significantly surpasses 91.97% achieved by Syn_10K_50 and 91.22% achieved by Real_2K_10. We conjecture that mixup with the real images can bring the real-world appearance attributes (e.g., blur and illumination) to synthetic images, which alleviate the domain gap. If we continue to increase the number of real images for training, e.g., Mix_2K_20, the performance can be further boosted from 95.78% to 97.65%.

10.4 Experiments

With the introduced Mixup Face Generator, we are able to generate large-scale face images with controllable facial attributes, including the identity, pose, expression, illumination, and other dataset characteristics such as the depth and the width. In this section, we perform an empirical analysis using synthetic face images. Specifically, we first introduce the datasets (Sect. 10.4.1) and the implementation details (Sect. 10.4.2). Then the long-tailed problem is mitigated by employing the balanced synthetic face dataset and identity mixup (Sect. 10.4.3). Lastly, we analyze the impacts of depth, width (Sect. 10.4.4), and different facial attributes (Sect. 10.4.5).

10.4.1 Datasets

- **Real Datasets.** We employ the CASIA-WebFace [64] and LFW [26] for training and testing, respectively. The CASIA-WebFace dataset contains around 500,000 web images, i.e., 494,414 images from 10,575 subjects. The LFW dataset is a widely used benchmark for face verification, which contains 13,233 face images from 5,749 identities. Following the protocol in [10], we report the verification accuracy on 6,000 testing image pairs.

- **Synthetic Datasets.** We first generate a synthetic version of LFW, in which all synthetic face images share the same properties with LFW images, e.g., expression, illumination, and pose. Specifically, for each image in LFW, we first use the 3D face reconstruction network in [12] to obtain the attribute coefficients $\mu \doteq [\beta, \gamma, \theta]$, which indicate the expression, illumination and pose coefficient, respectively. We then adopt the DiscoFaceGAN [11] to generate the face images according to these attribute coefficients with a random identity coefficient. Finally, we obtain a new dataset and refer to it as Syn-LFW, which has the same statistics as LFW with unknown identities (non-existing people). For the synthetic training dataset (e.g., Syn_10K_50), we construct it by randomly sampling latent variables from the standard normal distribution for identity, expression, pose, and illumination coefficients, respectively, leading to the same person with different expressions, poses, and illuminations in the same class. Note that the identities of Syn-LFW do not have the overlap with any synthetic training datasets.

10.4.2 Implementation Details

We use the MTCNN [70] to detect face bounding boxes and five facial landmarks (two eyes, nose and two mouth corners). All face images are then cropped, aligned (similarity transformation), and resized to 112×96 pixel as illustrated in Fig. 10.1. Similar to [10, 57], we normalize the pixel values (in [0, 255]) in RGB images to $[-1.0, 1.0]$ for training and testing. To balance the trade-off between the performance and computational complexity, we adopt the variant of ResNet [22], LResNet50E-IR, as our backbone framework, which is devised in ArcFace [10]. All models are implemented with PyTorch [38] and trained from scratch using Eight NVIDIA Tesla V100 GPUs. We use the additive angular margin loss defined in Eq. (10.1), i.e., with $(m_1, m_2, m_3) = (0, 0.5, 0)$ and $s = 30$. If not mentioned, we always set the batch size to 512. We use SGD with a momentum of 0.9 and a weight decay of 0.0005. The learning rate starts from 0.1 and is divided by 10 at the 24, 30, and 36 epochs, with 40 epochs in total.

10.4.3 Long-Tailed Face Recognition

Experimental Setup. To explore the long-tailed problem, we construct multiple synthetic datasets with the purpose that each dataset has the same number of identities (2 K) and total images (100 K) but different degrees of unbalance. Face images are generated using the equation:

$$\begin{aligned} N &= [N_1, N_2, N_3, N_4, N_5], \\ ID &= [400, 400, 400, 400, 400], \end{aligned} \tag{10.4}$$

where ID indicates the number of identities in each of the five groups, and N means the number of samples of the five groups. For example, if $N = [30, 40, 50, 60, 70]$, the corresponding synthetic dataset has 400 identities with 30 samples per identity, and the rest 1600 identities with 40, 50, 60, 70 samples per identity, respectively. We construct three different synthetic datasets by assigning N to be [2, 2, 6, 40, 200], [4, 16, 30, 80, 120] and [50, 50, 50, 50, 50], which are denoted as "2K_UB1", "2K_UB2" and "2K_50", respectively. The detailed construction process can be found in Sect. 10.4.1. Note that all the three datasets have averaged 50 samples per identity, while the first two have unbalanced distributions with the standard deviations 76.35 and 43.52, and the last one is the perfectly balanced dataset.

Empirical Analysis. We train face recognition models on the above three different synthetic datasets and the experimental results are illustrated in Fig. 10.6. We see that the model trained on the "2K_UB1" achieves the worst performance (83.80%), suggesting that the long-tailed problem or the unbalanced distribution leads to the degradation of the model performance. Comparing with the models trained on "2K_UB1" and "2K_UB2", we discover that decreasing the degree of unbalance leads to the improvement on the performance. Finally, when the model is trained on "2K_50", i.e., the perfectly balanced dataset, the accuracy is significantly improved to 86.18%. Therefore, with balanced synthetic data, the long-tailed problem can be intrinsically avoided. Besides, introducing the identity mixup for training can consistently and significantly improve the performance over all the settings.

10.4.4 Effectiveness of "Depth" and "Width"

Experimental Setup. We synthesize multiple-face datasets with different widths (the number of identities) and depths (the number of samples per identity). Let "N_S" denote the synthetic dataset containing N identities with S samples per identity, e.g., $1K\_50$ indicates the dataset having 1 K different identities and 50 samples per identity. Obviously, N and S represent the dataset's width and depth, respectively. The details of dataset construction can be found in Sect. 10.4.1.

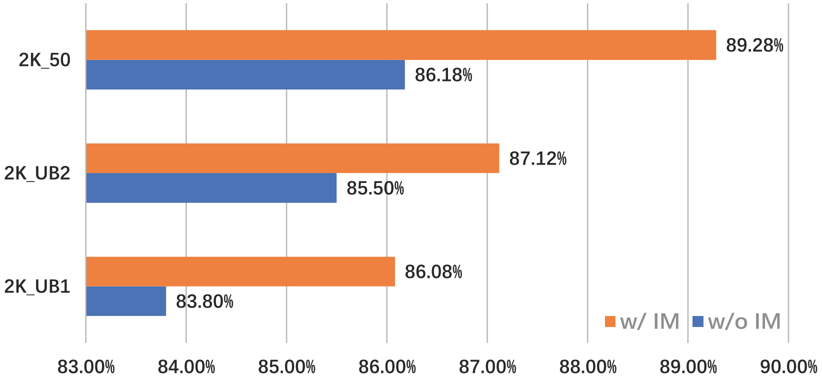


Fig. 10.6 Face verification accuracies (%) on LFW using the training datasets with decreasing imbalance, i.e., "2K_UB1", "2K_UB2", and "2K_50", where we assign N defined in Eq. (4) as [2, 2, 6, 40, 200], [4, 16, 30, 80, 120], and [50, 50, 50, 50, 50], respectively. w/ IM and w/o IM indicate whether identity mixup (IM) is used during training

Empirical Analysis. We train the same face recognition model on these synthetic datasets, and the experimental results (both w/wo identity mixup) are shown in Table 10.3. Firstly, we analyze the influence of the width of the dataset by comparing the results of (a), (b), (c), (i). From (a) to (c), we see that the accuracy dramatically increases from 83.85% to 88.75%. However, the improvement is marginal from (c) to (i), which implies that the synthetic data may suffer from the lack of inter-class variations. Observing the results of (d), (e), (f), (g), (h), (i), we conclude that the accuracy significantly improves with the increasing of dataset depth, but it is quickly saturated when the depth is larger than 20, which is in line with the observation on real data made by Schroff et al. [46]. Lastly, we see that (a) and (e) have the same number of total images (50K), while (a) outperforms (e) with a large margin, i.e., 4.37%, which reveals that the dataset width plays as the more important role than the dataset depth in term of the final face recognition accuracy. Similar observation can be found by comparing (b) and (f). Importantly, employing the identity mixup (IM) for training consistently improves the performance over all the datasets, which confirms the effectiveness of IM. The best accuracy 91.97% brought by IM significantly outperforms the original 88.98%.

10.4.5 Impacts of Different Facial Attributes

Experimental Setup. We explore the impacts of different facial attributes for face recognition (i.e., expression, pose, and illumination) by controlling face generation process. We construct four synthetic datasets that have 5 K identities and 50 samples per identity. The difference between the four datasets is the distribution of different facial attributes. Specif-

Table 10.3 Face verification accuracies (%) on LFW [64]. "N_S" implies that the corresponding dataset has N identities with S samples per identity, i.e., N and S indicate the width and depth. LFW (w/ IM) means employing the identity mixup (IM) for training

Method	ID	Samples	LFW	LFW(w/ IM)
(*a*) 1K_50	1 K	50	83.85	**87.53**
(*b*) 2K_50	2 K	50	86.18	**89.28**
(*c*) 5K_50	5 K	50	88.75	**90.95**
(*d*) 10K_2	10 K	2	78.85	**80.30**
(*e*) 10K_5	10 K	5	88.22	**88.32**
(*f*) 10K_10	10 K	10	89.48	**90.28**
(*g*) 10K_20	10 K	20	89.90	**90.87**
(*h*) 10K_30	10 K	30	89.73	**91.17**
(*i*) 10K_50	10 K	50	88.98	**91.97**

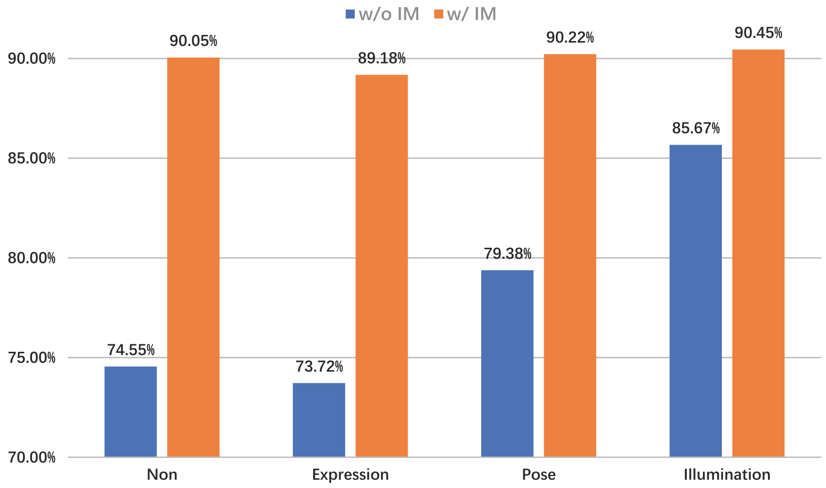


Fig. 10.7 Face verification accuracies (%) on LFW using the training datasets with variations in different facial attributes. Specifically, "Expression", "Pose", and "Illumination" indicate that we separately introduce variations in expression, pose, and illumination while keeping the other attributes unchanged. w/ IM and w/o IM indicate whether identity mixup (IM) is used during training

ically, the first dataset is referred to as "Non", since it fixes all the facial attributes. The rest three datasets are referred to as "Expression", "Pose", and "Illumination", respectively, which indicates the only changed attribute while keeping other attributes unchanged.

Empirical Analysis. As shown in Fig. 10.7, "Non" and "Expression" achieve the worst two performances 74.55% and 73.72%, respectively. Specifically, we find that "Expression" is limited to poor diversity, i.e., the generated face images mainly have the expression of "smiling" (see more demo images in the supplementary materials). Hence, there is basically only one valid sample per identity for "Non" and "Expression", causing the poor performances. Experimental results on "Pose" and "Illumination" demonstrate significant improvements over "Non", possibly due to their more diverse distributions and the testing dataset (i.e., LFW) also has similar pose and illumination. Lastly, we find that all of four settings are significantly improved with the proposed identity mixup, especially for "Non". A possible reason is that identity mixup can be regarded as a strong data augmentation method for face recognition, reducing the influences of different facial attributes on the final recognition accuracy.

10.5 Conclusion

In this chapter, we explored the potentials of synthetic data for face recognition, i.e., SynFace. We performed a systematically empirical analysis and provided novel insights on how to efficiently utilize synthetic face images for face recognition: (1) enlarging the intra-class variations of synthetic data consistently improves the performance, which can be achieved by the proposed identity mixup; (2) both the depth and width of the training synthetic dataset have significant influences on the performance, while the saturation first appears on the depth dimension, i.e., increasing the number of identities (width) is more important; (3) the impacts of different attributes vary from pose, illumination, and expression, i.e., changing pose and illumination brings significant improvements, while the generated face images suffer from a poor diversity on expression; (4) a small subset of real-world face images can greatly boost the performance of SynFace via the proposed domain mixup.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. ArXiv preprint arXiv:1701.07875 (2017)
2. Bao, J., Chen, D., Wen, F., Li, H., Hua, G.: Towards open-set identity preserving face synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6713–6722 (2018)
3. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194 (1999)
4. Borg, I., Groenen, P.J.: Modern Multidimensional Scaling: Theory and Applications. Springer Science & Business Media (2005)
5. Cao, K., Rong, Y., Li, C., Tang, X., Loy, C.C.: Pose-robust face recognition via deep residual equivariant mapping. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5187–5196 (2018)
6. Chen, Y., Li, W., Chen, X., Gool, L.V.: Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1841–1850 (2019)

7. Chen, Z., Wang, C., Yuan, B., Tao, D.: Puppeteergan: Arbitrary portrait animation with semantic-aware appearance transformation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
8. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 539–546. IEEE (2005)
9. Dai, B., Wipf, D.: Diagnosing and enhancing vae models. ArXiv preprint arXiv:1903.05789 (2019)
10. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4690–4699 (2019)
11. Deng, Y., Yang, J., Chen, D., Wen, F., Tong, X.: Disentangled and controllable face image generation via 3d imitative-contrastive learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5154–5163 (2020)
12. Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., Tong, X.: Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019)
13. Deng, Z., Peng, X., Li, Z., Qiao, Y.: Mutual component convolutional neural networks for heterogeneous face recognition. IEEE Trans. Image Process. (TIP) **28**(6), 3102–3114 (2019)
14. Gong, D., Li, Z., Huang, W., Li, X., Tao, D.: Heterogeneous face recognition: A common encoding feature discriminant approach. IEEE Trans. Image Process. (TIP) **26**(5), 2079–2089 (2017)
15. Gong, D., Li, Z., Lin, D., Liu, J., Tang, X.: Hidden factor analysis for age invariant face recognition. In: IEEE International Conference on Computer Vision (ICCV), pp. 2872–2879 (2013)
16. Gong, D., Li, Z., Liu, J., Qiao, Y.: Multi-feature canonical correlation analysis for face photo-sketch image retrieval. In: Proceedings of the 21th ACM International Conference on Multimedia, pp. 617–620 (2013)
17. Gong, D., Li, Z., Tao, D., Liu, J., Li, X.: A maximum entropy feature descriptor for age invariant face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5289–5297 (2015)
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 2672–2680 (2014)
19. Guo, H., Mao, Y., Zhang, R.: Mixup as locally linear out-of-manifold regularization. In: AAAI Conference on Artificial Intelligence (AAAI), vol. 33, pp. 3714–3722 (2019)
20. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: European Conference on Computer Vision (ECCV), pp. 87–102. Springer (2016)
21. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 1735–1742. IEEE (2006)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
23. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition, pp. 84–92. Springer (2015)
24. Huang, F.J., Zhou, Z., Zhang, H.J., Chen, T.: Pose invariant face recognition. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, pp. 245–250 (2000)

25. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708 (2017)
26. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild: A Database Forstudying Face Recognition in Unconstrained Environments (2008)
27. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4873–4882 (2016)
28. Kortylewski, A., Egger, B., Schneider, A., Gerig, T., Morel-Forster, A., Vetter, T.: Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019)
29. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
30. Li, Z., Gong, D., Li, Q., Tao, D., Li, X.: Mutual component analysis for heterogeneous face recognition. ACM Trans. Intell. Syst. Technol. (TIST) **7**(3), 1–23 (2016)
31. Li, Z., Gong, D., Qiao, Y., Tao, D.: Common feature discriminant analysis for matching infrared face images to optical face images. IEEE Trans. Image Process (TIP) **23**(6), 2436–2445 (2014)
32. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 212–220 (2017)
33. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: International Conference on Machine Learning (ICML), vol. 2, p. 7 (2016)
34. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2537–2546 (2019)
35. Mirza, M., Osindero, S.: Conditional generative adversarial nets. ArXiv preprint arXiv:1411.1784 (2014)
36. Mroueh, Y., Sercu, T., Goel, V.: Mcgan: Mean and covariance feature matching gan. ArXiv preprint arXiv:1702.08398 (2017)
37. Ouyang, W., Wang, X., Zhang, C., Yang, X.: Factors in finetuning deep model for object detection with long-tail distribution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 864–873 (2016)
38. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems (NeurIPS), pp. 8024–8035. Curran Associates, Inc. (2019). http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
39. Qi, G.J.: Loss-sensitive generative adversarial networks on lipschitz densities. Int. J. Comput. Vis. (IJCV) **128**(5), 1118–1140 (2020)
40. Qiu, H., Gong, D., Li, Z., Liu, W., Tao, D.: End2end occluded face recognition by masking corrupted features. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), pp. 1–1 (2021). https://doi.org/10.1109/TPAMI.2021.3098962
41. Qiu, H., Yu, B., Gong, D., Li, Z., Liu, W., Tao, D.: Synface: Face recognition with synthetic data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10880–10890 (2021)
42. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ArXiv preprint arXiv:1511.06434 (2015)

43. Richardson, E., Sela, M., Kimmel, R.: 3d face reconstruction by learning from synthetic data. In: 2016 Fourth International Conference on 3D vision (3DV), pp. 460–469 (2016)
44. Sakaridis, C., Dai, D., Hecker, S., Van Gool, L.: Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In: European Conference on Computer Vision (ECCV), pp. 687–704 (2018)
45. Sankaranarayanan, S., Balaji, Y., Jain, A., Nam Lim, S., Chellappa, R.: Learning from synthetic data: Addressing domain shift for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3752–3761 (2018)
46. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (2015)
47. Shen, Y., Luo, P., Yan, J., Wang, X., Tang, X.: Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 821–830 (2018)
48. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ArXiv preprint arXiv:1409.1556 (2014)
49. Song, L., Gong, D., Li, Z., Liu, C., Liu, W.: Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In: IEEE International Conference on Computer Vision (ICCV), pp. 773–782 (2019)
50. Summers, C., Dinneen, M.J.: Improved mixed-example data augmentation. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1262–1270. IEEE (2019)
51. Takahashi, R., Matsubara, T., Uehara, K.: Ricap: Random image cropping and patching data augmentation for deep cnns. In: Asian Conference on Machine Learning (ACML), pp. 786–798. PMLR (2018)
52. Tang, Z., Naphade, M., Birchfield, S., Tremblay, J., Hodge, W., Kumar, R., Wang, S., Yang, X.: Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In: IEEE International Conference on Computer Vision (ICCV), pp. 211–220 (2019)
53. Trigueros, D.S., Meng, L., Hartnett, M.: Generating photo-realistic training data to improve face recognition accuracy. ArXiv preprint arXiv:1811.00112 (2018)
54. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: International Conference on Machine Learning (ICML), pp. 6438–6447. PMLR (2019)
55. Wang, F., Chen, L., Li, C., Huang, S., Chen, Y., Qian, C., Change Loy, C.: The devil of face recognition is in the noise. In: European Conference on Computer Vision (ECCV), pp. 765–780 (2018)
56. Wang, H., Gong, D., Li, Z., Liu, W.: Decorrelated adversarial learning for age-invariant face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3527–3536 (2019)
57. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5265–5274 (2018)
58. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8198–8207 (2019)
59. Wang, X., Wang, S., Wang, J., Shi, H., Mei, T.: Co-mining: Deep face recognition with noisy labels. In: IEEE International Conference on Computer Vision (ICCV), pp. 9358–9367 (2019)

60. Wang, Y., Gong, D., Zhou, Z., Ji, X., Wang, H., Li, Z., Liu, W., Zhang, T.: Orthogonal deep features decomposition for age-invariant face recognition. In: European Conference on Computer Vision (ECCV), pp. 738–753 (2018)
61. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision (ECCV), pp. 499–515. Springer (2016)
62. Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., Zhang, W.: Adversarial domain adaptation with domain mixup. In: AAAI Conference on Artificial Intelligence (AAAI), vol. 34, pp. 6502–6509 (2020)
63. Yang, X., Jia, X., Gong, D., Yan, D.M., Li, Z., Liu, W.: Larnet: Lie algebra residual network for face recognition. In: International Conference on Machine Learning (ICML), pp. 11738–11750. PMLR (2021)
64. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. ArXiv preprint arXiv:1411.7923 (2014)
65. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Towards large-pose face frontalization in the wild. In: IEEE International Conference on Computer Vision (ICCV), pp. 3990–3999 (2017)
66. Yu, B., Liu, T., Gong, M., Ding, C., Tao, D.: Correcting the triplet selection bias for triplet loss. In: European Conference on Computer Vision (ECCV), Munich, Germany, pp. 71–87 (September 08-14, 2018)
67. Yu, B., Tao, D.: Deep metric learning with tuplet margin loss. In: IEEE International Conference on Computer Vision (ICCV), pp. 6490–6499. Seoul, Korea (October 27-November 02, 2019)
68. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: Beyond empirical risk minimization. ArXiv preprint arXiv:1710.09412 (2017)
69. Zhang, J., Zeng, X., Wang, M., Pan, Y., Liu, L., Liu, Y., Ding, Y., Fan, C.: Freenet: Multi-identity face reenactment. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5326–5335 (2020)
70. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Sig. Process. Lett. **23**(10), 1499–1503 (2016)
71. Zhang, W., Shan, S., Chen, X., Gao, W.: Local Gabor binary patterns based on Kullback-Leibler divergence for partially occluded face recognition. IEEE Sig. Process. Lett. **14**(11), 875–878 (2007)
72. Zhu, X., Anguelov, D., Ramanan, D.: Capturing long-tail distributions of object subcategories. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 915–922 (2014)

Uncertainty-Aware Face Recognition 11

Yichun Shi and Anil K. Jain

11.1 Introduction

Deep learning-based face recognition systems are mainly based on embedding methods, where facial features are compared in a latent semantic space. However, in a fully unconstrained face setting, the facial features could be ambiguous or may not even be present in the input face, leading to noisy representations that harm both the training and testing phases. In this chapter, we introduce a new type of face representation, namely probabilistic embeddings, that explicitly models the feature uncertainties in the face representations. In contrast to deterministic embeddings, which embeds each input face image/template as a point in the feature space, probabilistic embeddings represent each input as a probabilistic distribution and hence additionally model the data uncertainty. This type of representation has been shown to be (1) preferable for matching low-quality face images during testing, (2) robust against noisy data during training, and (3) good at estimating input quality for template feature fusion and input filtering.

11.2 Background: Uncertainty-Aware Deep Learning

To improve the robustness and interpretability of discriminant Deep Neural Networks (DNNs), deep uncertainty learning is getting more attention [4, 12, 13]. There are two main

Y. Shi (✉)
ByteDance, San Jose, USA
e-mail: sc2h6o@gmail.com

A. K. Jain
East Lansing, MI, USA
e-mail: jain@cse.msu.edu

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_11

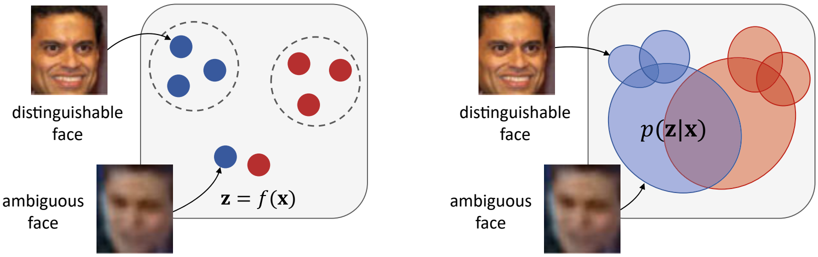
types of uncertainty: *model uncertainty* and *data uncertainty*. Model uncertainty refers to the uncertainty of model parameters given the training data and can be reduced by collecting additional training data [4, 12, 21, 23]. Data uncertainty accounts for the uncertainty in output whose primary source is the inherent noise in input data and hence cannot be eliminated with more training data [13]. The uncertainty studied in this chapter can be categorized as data uncertainty. Although techniques have been developed for estimating data uncertainty in different tasks, including classification and regression [13], they are not suitable for face embeddings since the target space is not well-defined by given labels. That is, although we are given the identity labels, they cannot directly serve as target vectors in the latent feature space. Variational Autoencoders [15] can also be regarded as a method for estimating uncertainty, but it mainly serves a generation purpose. Specific to face recognition, some studies [5, 14, 37] have leveraged the model uncertainty for analysis and learning of face representations. In contrast, in this chapter, we mainly focus on data uncertainty in face recognition and how to represent such uncertainty in face embeddings.

11.3 Probabilistic Face Embeddings (PFE)

When humans are asked to describe a face image, they not only give the description of the facial attributes, but also the confidence associated with them. For example, if the eyes are blurred in the image, a person will keep the eye size as uncertain information and focus on other features. Furthermore, if the image is completely corrupted and no attributes can be discerned, the subject may respond that he/her cannot identify this face. This kind of uncertainty (or confidence) estimation is common and important in human decision-making.

On the other hand, the representations used in state-of-the-art face recognition systems are generally confidence-agnostic. These methods depend on an embedding model (e.g., Deep Neural Networks) to give a deterministic point representation for each face image in the latent feature space [3, 19, 25, 30, 31]. A point in the latent space represents the model's estimation of the facial features in the given image. If the error in the estimation is somehow bounded, the distance between two points can effectively measure the semantic similarity between the corresponding face images. But given a low-quality input, where the expected facial features are ambiguous or absent in the image, a large shift in the embedded points is inevitable, leading to false recognition (Fig. 11.1a).

Given that face recognition systems have already achieved high recognition accuracies on relatively constrained face recognition benchmarks, e.g., LFW [9] and YTF [33], where most facial attributes can be clearly observed, recent face recognition challenges have moved on to more unconstrained scenarios, including surveillance videos [10, 17, 22]. In these tasks, any type and degree of variation could exist in the face image, where most of the desired facial features learned by the representation model could be absent. Given this lack of information, it is unlikely to find a feature set that could always match these faces accurately. Hence, the



(a) deterministic embedding

(b) probabilistic embedding

Fig. 11.1 Difference between deterministic face embeddings and probabilistic face embeddings (PFEs). Deterministic embeddings represent every face as a point in the latent space without regard to its feature ambiguity. Probabilistic face embedding (PFE) gives a distributional estimation of features in the latent space instead. **Best viewed in color**

state-of-the-art face recognition systems, which obtained over 99% accuracy on LFW, have suffered from a large performance drop on IARPA Janus benchmarks [10, 17, 22].

To address the above problems, *Probabilistic Face Embeddings (PFEs)* [27] give a distributional estimation instead of a point estimation in the latent space for each input face image (Fig. 11.1b). The mean of the distribution can be interpreted as the most likely latent feature values, while the span of the distribution represents the uncertainty of these estimations. PFE can address the unconstrained face recognition problem in a two-fold way: (1) During matching (face comparison), PFE penalizes uncertain features (dimensions) and pays more attention to more confident features. (2) For low-quality inputs, the confidence estimated by PFE can be used to reject the input or actively ask for human assistance to avoid false recognition. Besides, a natural solution can be derived to aggregate the PFE representations of a set of face images into a new distribution with lower uncertainty to increase recognition performance.

11.3.1 Limitations of Deterministic Embeddings

In this section, we explain the problems of deterministic face embeddings from both theoretical and empirical views. Let $\mathcal{X}$ denote the image space and $\mathcal{Z}$ denote the latent feature space of D dimensions. An ideal latent space $\mathcal{Z}$ should only encode *identity-salient* features and be *disentangled* from identity-irrelevant features. As such, each identity should have a unique intrinsic code $\mathbf{z} \in \mathcal{Z}$ that best represents this person and each face image $\mathbf{x} \in \mathcal{X}$ is an observation sampled from $p(\mathbf{x}|\mathbf{z})$. The process of training face embeddings can be viewed as a joint process of searching for such a latent space $\mathcal{Z}$ and learning the inverse mapping $p(\mathbf{z}|\mathbf{x})$. For deterministic embeddings, the inverse mapping is a Dirac delta function $p(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - f(\mathbf{x}))$, where f is the embedding function. Clearly, for any space $\mathcal{Z}$, given the possibility of noises in $\mathbf{x}$, it is unrealistic to recover the exact $\mathbf{z}$ and the embedded point of a low-quality input would inevitably shift away from its intrinsic $\mathbf{z}$ (no matter how much training data we have).

The question is whether this shift could be bounded such that we still have smaller intra-class distances compared to inter-class distances. However, this is unrealistic for fully unconstrained face recognition and we conduct an experiment to illustrate this. Let us start with a simple example: given a pair of identical images, a deterministic embedding will always map them to the same point, and therefore, the distance between them will always be 0, even if these images do not contain a face. This implies that "a pair of images being similar or even the same does not necessarily mean the probability of their belonging to the same person is high".

To demonstrate this, we conduct an experiment by manually degrading the high-quality images and visualizing their similarity scores. We randomly select a high-quality image of each subject from the LFW dataset [9] and manually insert Gaussian blur, occlusion, and random Gaussian noise into the faces. In particular, we linearly increase the size of Gaussian kernel, occlusion ratio, and the standard deviation of the noise to control the degradation degree. At each degradation level, we extract the feature vectors with a 64-layer CNN,[1] which is comparable to state-of-the-art face recognition systems. The features are normalized to a hyperspherical embedding space. Then, two types of cosine similarities are reported: (1) similarity between pairs of original images and their respective degraded image, and (2) similarity between degraded images of different identities. As shown in Fig. 11.2, for all three types of degradation, the genuine similarity scores decrease to 0, while the impostor similarity scores converge to 1.0! These indicate two types of errors that can be expected in a fully unconstrained scenario even when the model is very confident (very high/low similarity scores):

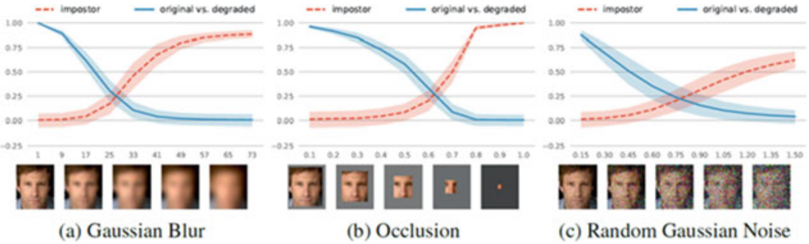
1. False accept of impostor low-quality pairs and
2. False reject of genuine cross-quality pairs.

To confirm this, we test the model on the IJB-A dataset by finding impostor/genuine image pairs with the highest/lowest scores, respectively. The situation is exactly as we hypothesized (See Fig. 11.3). We call this *Feature Ambiguity Dilemma* which is observed when the deterministic embeddings are forced to estimate the features of ambiguous faces. The experiment also implies that there exists a *dark space* where the ambiguous inputs are mapped to and the distance metric is distorted.

11.3.2 Contidional Gaussian Distributions as Probabilistic Embeddings

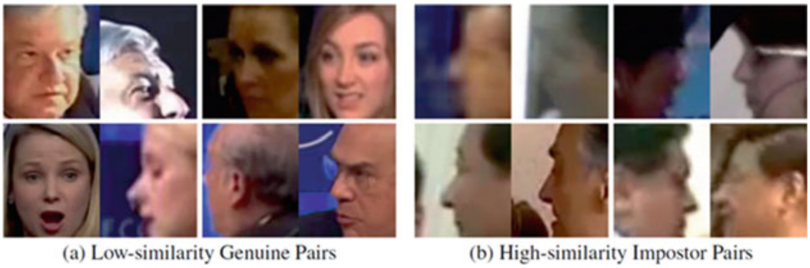
To address the aforementioned problem caused by data uncertainty, Probabilistic Face Embeddings (PFEs) encode the uncertainty into the face representation and take it into

[1] Trained on Ms-Celeb-1M [6] with AM-Softmax [29].



(a) Gaussian Blur (b) Occlusion (c) Random Gaussian Noise

Fig. 11.2 Illustration of *feature ambiguity dilemma*. The plots show the cosine similarity on LFW dataset with different degrees of degradation. Blue lines show the similarity between original images and their respective degraded versions. Red lines show the similarity between impostor pairs of degraded images. The shading indicates the standard deviation. With larger degrees of degradation, the model becomes more confident (very high/low scores) in a wrong way



(a) Low-similarity Genuine Pairs (b) High-similarity Impostor Pairs

Fig. 11.3 Example genuine pairs from IJB-A dataset estimated with the lowest similarity scores and impostor pairs with the highest similarity scores (among all possible pairs) by a 64-layer CNN model. The genuine pairs mostly consist of one high-quality and one low-quality image, while the impostor pairs are all low-quality images. Note that these pairs are not templates in the verification protocol

account during matching. Specifically, instead of building a model that gives a point estimation in the latent space, we estimate a distribution $p(\mathbf{z}|\mathbf{x})$ in the latent space to represent the potential appearance of a person's face.[2] In particular, the original PFE uses a multivariate Gaussian distribution:

$$p(\mathbf{z}|\mathbf{x}_i) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\mathbf{I}) \tag{11.1}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are both a D-dimensional vector predicted by the network from the ith input image $\mathbf{x}_i$. Here we only consider a diagonal covariance matrix to reduce the complexity of the face representation. This representation should have the following properties:

1. The center $\boldsymbol{\mu}$ should encode the most likely facial features of the input image.
2. The uncertainty $\boldsymbol{\sigma}$ should encode the model's confidence along each feature dimension.

[2] Following the notations in Sect. 11.3.1.

11.3.3 Face Matching with PFEs

Given the PFE representations of a pair of images $(\mathbf{x}_i, \mathbf{x}_j)$, we can directly measure the "likelihood" of them belonging to the same person (sharing the same latent code): $p(\mathbf{z}_i = \mathbf{z}_j)$, where $\mathbf{z}_i \sim p(\mathbf{z}|\mathbf{x}_i)$ and $\mathbf{z}_j \sim p(\mathbf{z}|\mathbf{x}_j)$. Specifically,

$$p(\mathbf{z}_i = \mathbf{z}_j) = \int p(\mathbf{z}_i|\mathbf{x}_i)p(\mathbf{z}_j|\mathbf{x}_j)\delta(\mathbf{z}_i - \mathbf{z}_j)d\mathbf{z}_i d\mathbf{z}_j. \tag{11.2}$$

In practice, we would like to use the log-likelihood instead, whose solution is given by:

$$\begin{aligned} s(\mathbf{x}_i, \mathbf{x}_j) =& \log p(\mathbf{z}_i = \mathbf{z}_j) \\ =& -\frac{1}{2}\sum_{l=1}^{D}(\frac{(\mu_i^{(l)} - \mu_j^{(l)})^2}{\sigma_i^{2(l)} + \sigma_j^{2(l)}} + \log(\sigma_i^{2(l)} + \sigma_j^{2(l)})) \\ & - const, \end{aligned} \tag{11.3}$$

where $const = \frac{D}{2}\log 2\pi$, $\mu_i^{(l)}$ refers to the lth dimension of $\boldsymbol{\mu}_i$ and similarly for $\sigma_i^{(l)}$.

This symmetric measure can be viewed as the expectation of the likelihood of one input's latent code conditioned on the other, that is

$$\begin{aligned} s(\mathbf{x}_i, \mathbf{x}_j) =& \log \int p(\mathbf{z}|\mathbf{x}_i)p(\mathbf{z}|\mathbf{x}_j)d\mathbf{z} \\ =& \log \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\mathbf{x}_i)}[p(\mathbf{z}|\mathbf{x}_j)] \\ =& \log \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\mathbf{x}_j)}[p(\mathbf{z}|\mathbf{x}_i)]. \end{aligned} \tag{11.4}$$

As such, we call it *mutual likelihood score (MLS)*. Different from KL-divergence, this score is unbounded and cannot be seen as a distance metric. It can be shown that the squared Euclidean distance is equivalent to a special case of MLS when all the uncertainties are assumed to be the same:

Property 11.1 If $\sigma_i^{(l)}$ is a fixed number for all data $\mathbf{x}_i$ and dimensions l, MLS is equivalent to a scaled and shifted negative squared Euclidean distance.

Further, when the uncertainties are allowed to be different, we note that MLS has some interesting properties that make it different from a distance metric:

1. *Attention* mechanism: the first term in the bracket in Eq. (11.3) can be seen as a weighted distance that assigns larger weights to less uncertain dimensions.

2. *Penalty* mechanism: the second term in the bracket in Eq. (11.3) can be seen as a penalty term that penalizes dimensions that have high uncertainties.
3. If either input $\mathbf{x}_i$ or $\mathbf{x}_j$ has large uncertainties, MLS will be low (because of penalty) irrespective of the distance between their mean.
4. Only if both inputs have small uncertainties and their means are close to each other, MLS could be very high.

The last two properties imply that PFE could solve the feature ambiguity dilemma if the network can effectively estimate $\boldsymbol{\sigma}_i$.

11.3.4 Template Feature Fusion with PFEs

In many cases, we have a template (set) of face images, for which we need to build a compact representation for matching. With PFEs, a conjugate formula can be derived for representation fusion (Fig. 11.4). Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a series of observations (face images) from the same identity and $p(\mathbf{z}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be the posterior distribution after the nth observation. Then, assuming all the observations are conditionally independent (given the latent code $\mathbf{z}$). It can be shown that:

$$p(\mathbf{z}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \alpha \frac{p(\mathbf{z}|\mathbf{x}_n)}{p(\mathbf{z})} p(\mathbf{z}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1}), \tag{11.5}$$

where α is a normalization factor. To simplify the notations, let us only consider the one-dimensional case below; the solution can be easily extended to the multivariate case.

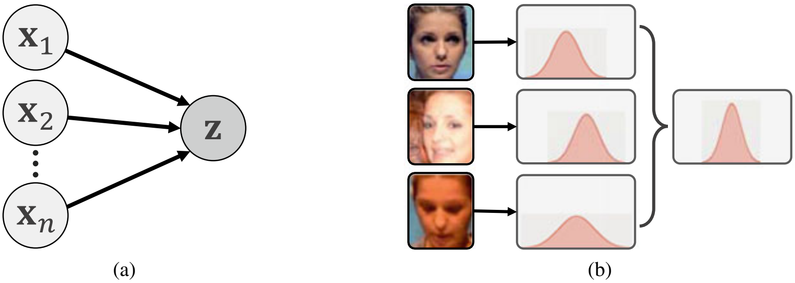


Fig. 11.4 Fusion with PFEs. **a** Illustration of the fusion process as a directed graphical model. **b** Given the Gaussian representations of faces (from the same identity), the fusion process outputs a new Gaussian distribution in the latent space with a more precise mean and lower uncertainty

If $p(\mathbf{z})$ is assumed to be a non-informative prior, i.e., $p(\mathbf{z})$ is a Gaussian distribution whose variance approaches ∞, the posterior distribution in Eq. (11.5) is a new Gaussian distribution with lower uncertainty (See supplementary material). Further, given a set of face images $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, the parameters of the fused representation can be directly given by:

$$\hat{\mu}_n = \sum_{i=1}^{n} \frac{\hat{\sigma}_n^2}{\sigma_i^2} \mu_i, \tag{11.6}$$

$$\frac{1}{\hat{\sigma}_n^2} = \sum_{i=1}^{n} \frac{1}{\sigma_i^2}. \tag{11.7}$$

In practice, because the conditional independence assumption is usually not true, e.g., video frames include a large amount of redundancy, Eq. (11.7) will be biased by the number of images in the set. Therefore, we take the dimension-wise minimum to obtain the new uncertainty.

Relationship to Quality-aware Pooling

If we consider a case where all the dimensions share the same uncertainty σ_i for ith input and let the quality value $q_i = \frac{1}{\sigma_i^2}$ be the output of the network. Then Eq. (11.6) can be written as

$$\hat{\boldsymbol{\mu}}_n = \frac{\sum_{i=1}^{n} q_i \boldsymbol{\mu}_i}{\sum_j^n q_j}. \tag{11.8}$$

If we do not use the uncertainty after fusion, the algorithm will be the same as recent quality-aware aggregation methods for set-to-set face recognition [20, 34, 35].

11.3.5 Learning Uncertainty

Note that any deterministic embedding f, if properly optimized, can indeed satisfy the properties of PFEs: (1) the embedding space is a disentangled identity-salient latent space and (2) $f(\mathbf{x})$ represents the most likely features of the given input in the latent space. As such, in this work, we consider a stage-wise training strategy: given a pre-trained embedding model f, we fix its parameters, take $\boldsymbol{\mu}(\mathbf{x}) = f(\mathbf{x})$, and optimize an additional uncertainty module to estimate $\boldsymbol{\sigma}(\mathbf{x})$. When the uncertainty module is trained on the same dataset of the embedding model, this stage-wise training strategy allows us to have a more fair comparison between PFE and the original embedding $f(\mathbf{x})$ than an end-to-end learning strategy.

The uncertainty module is a network with two fully-connected layers which share the same input as the bottleneck layer.[3] The optimization criteria are to maximize the mutual likelihood score of all genuine pairs $(\mathbf{x}_i, \mathbf{x}_j)$. Formally, the loss function to minimize is

$$\mathcal{L} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} -s(\mathbf{x}_i, \mathbf{x}_j) \tag{11.9}$$

where $\mathcal{P}$ is the set of all genuine pairs and s is defined in Eq. (11.3). In practice, the loss function is optimized within each mini-batch. Intuitively, this loss function can be understood as an alternative to maximizing $p(\mathbf{z}|\mathbf{x})$: if the latent distributions of all possible genuine pairs have a large overlap, the latent target $\mathbf{z}$ should have a large likelihood $p(\mathbf{z}|\mathbf{x})$ for any corresponding $\mathbf{x}$. Notice that because $\boldsymbol{\mu}(\mathbf{x})$ is fixed, the optimization wouldn't lead to the collapse of all the $\boldsymbol{\mu}(\mathbf{x})$ to a single point.

11.3.6 Experimentals

In this section, we test the PFE method on standard face recognition protocols to compare with deterministic embeddings. Then we conduct qualitative analysis to gain more insight into how PFE behaves.

To comprehensively evaluate the efficacy of PFEs, we conduct the experiments on 7 benchmarks, including the well-known **LFW** [9], **YTF** [33], **MegaFace** [11], and four other more unconstrained benchmarks:

CFP [26] contains 7,000 frontal/profile face photos of 500 subjects. We only test on the frontal-profile (FP) protocol, which includes 7,000 pairs of frontal-profile faces. **IJB-A** [17] is a template-based benchmark, containing 25,813 faces images of 500 subjects. Each template includes a set of still photos or video frames. Compared with previous benchmarks, the faces in IJB-A have larger variations and present a more unconstrained scenario. **IJB-C** [22] is an extension of IJB-A with 140,740 faces images of 3,531 subjects. The verification protocol of IJB-C includes more impostor pairs so we can compute True Accept Rates (TAR) at lower False Accept Rates (FAR). **IJB-S** [10] is a surveillance video benchmark containing 350 surveillance videos spanning 30 hours in total, 5,656 enrollment images, and 202 enrollment videos of 202 subjects. Many faces in this dataset are of extreme pose or low quality, making it one of the most challenging face recognition benchmarks.

[3] Bottleneck layer refers to the layer which outputs the original face embedding.

Table 11.1 Results of models trained on CASIA-WebFace. “Original” refers to the deterministic embeddings. The better performance among each base model is shown in bold numbers. “PFE” uses a mutual likelihood score for matching. IJB-A results are verification rates at FAR=0.1%

Base model	Representation	LFW	YTF	CFP-FP	IJB-A
Softmax + Center Loss [31]	Original	98.93	94.74	93.84	78.16
	PFE	**99.27**	**95.42**	**94.51**	**80.83**
Triplet [25]	Original	97.65	93.36	89.76	60.82
	PFE	**98.45**	**93.96**	**90.04**	**61.00**
A-Softmax [19]	Original	99.15	94.80	92.41	78.54
	PFE	**99.32**	**94.94**	**93.37**	**82.58**
AM-Softmax [29]	Original	99.28	95.64	94.77	84.69
	PFE	**99.55**	**95.92**	**95.06**	**87.58**

We use the CASIA-WebFace [36] and a cleaned version[4] of MS-Celeb-1M [6] as training data, from which we remove the subjects that are also included in the test datasets. In particular, 84 and 4,182 subjects were removed from CASIA-WebFace and MS-Celeb-1M, respectively.

11.3.6.1 Experiments on Different Base Embeddings

Since PFE works by converting existing deterministic embeddings, we want to evaluate how it works with different base embeddings, i.e., face representations trained with different loss functions. In particular, we implement the following state-of-the-art loss functions: Softmax+Center Loss [31], Triplet Loss [25], A-Softmax [19] and AM-Softmax [29]. To be aligned with previous work [19, 30], we train a 64-layer residual network [19] with each of these loss functions on the CASIA-WebFace dataset as base models. All the features are $\ell 2$-normalized to a hyper-spherical embedding space. Then we train the uncertainty module for each base model on the CASIA-WebFace again for 3,000 steps. We evaluate the performance on four benchmarks: LFW [9], YTF [33], CFP-FP [26] and IJB-A [17], which present different challenges in face recognition. The results are shown in Table 11.1. The PFE improves over the original representation in all cases, indicating that it is robust with different embeddings and testing scenarios.

[4] https://github.com/inlmouse/MS-Celeb-1M_WashList.

11.3.6.2 Using Uncertainty for Feature Fusion and Matching

To evaluate the effect of PFE on state-of-the-art face recognition networks, we use a different base model, which is a 64-layer network trained with AM-Softmax on the MS-Celeb-1M dataset. Then we fix the parameters and train the uncertainty module on the same dataset for 12,000 steps. In the following experiments, we compare 3 methods:

- **Baseline** only uses the original features of the 64-layer deterministic embedding along with cosine similarity for matching. Average pooling is used in the case of template/video benchmarks.
- **PFE$_{\mathbf{fuse}}$** uses the uncertainty estimation $\boldsymbol{\sigma}$ in PFE and Eq. (11.6) to aggregate the features of templates but uses cosine similarity for matching. If the uncertainty module could estimate the feature uncertainty effectively, fusion with $\boldsymbol{\sigma}$ should be able to outperform average pooling by assigning larger weights to confident features.
- **PFE$_{\mathbf{fuse+match}}$** uses $\boldsymbol{\sigma}$ both for fusion and matching (with mutual likelihood scores). Templates/videos are fused based on Eqs. (11.6) and (11.7).

In Table 11.2, we show the results on four relatively easier benchmarks: LFW, YTF, CFP, and MegaFace. Although the accuracy on LFW and YTF is nearly saturated, PFE still improves the performance of the original representation. CFP involves more large pose faces compared to LFW and YTF, and therefore, we observe a larger improvement on this benchmark. Note that MegaFace is a biased dataset: because all the probes are high-quality images from FaceScrub, the positive pairs in MegaFace are both high-quality images, while the negative pairs only contain at most one low-quality image.[5] Therefore, neither of the two types of the error caused by the feature ambiguity dilemma (Sect. 11.3.1) will show up in MegaFace and it naturally favors deterministic embeddings. However, the PFE still maintains the performance in this case. We also note that such a bias, namely the target gallery images being of higher quality than the rest of the gallery, would not exist in real-world applications.

Table 11.2 Results of the baseline and PFE trained on MS-Celeb-1M and state-of-the-art methods on LFW, YTF and MegaFace. The MegaFace verification rates are computed at FAR=0.0001%. "-" indicates that the author did report the performance on the corresponding protocol

Method	Training data	LFW	YTF	CFP-FP	MF1 Rank1	MF1 Veri.
Baseline	4.4M	99.70	97.18	92.78	79.43	92.93
PFE$_{\text{fuse}}$	4.4M	–	97.32	–	–	–
PFE$_{\text{fuse+match}}$	4.4M	99.82	97.36	93.34	78.95	92.51

[5] The negative pairs of MegaFace in the verification protocol only include those between probes and distractors.

Table 11.3 Results of the baseline and PFE model trained on MS-Celeb-1M and state-of-the-art methods on IJB-A and IJB-C

Method	Training data	IJB-A (TAR@FAR)		IJB-C (TAR@FAR)			
		0.1%	1%	0.001%	0.01%	0.1%	1%
Baseline	4.4M	93.30 ± 1.29	96.15 ± 0.71	70.10	85.37	93.61	96.91
PFE_{fuse}	4.4M	94.59 ± 0.72	95.92 ± 0.73	83.14	92.38	95.47	97.36
$\text{PFE}_{\text{fuse+match}}$	4.4M	**95.25 ± 0.89**	**97.50 ± 0.43**	**89.64**	**93.25**	**95.49**	97.17

Table 11.4 Performance comparison on three protocols of IJB-S. The performance is reported in terms of rank retrieval (closed-set) and TPIR@FPIR (open-set) instead of the media-normalized version [10]. The numbers "1%" in the second row refers to the FPIR

Method	Training data	Surv-to-Single			Surv-to-Booking			Surv-to-Surv		
		Rank-1	Rank-5	1%	Rank-1	Rank-5	1%	Rank-1	Rank-5	1%
Baseline	4.4M	50.00	59.07	7.22	47.54	56.14	14.75	9.40	17.52	0.06
PFE_{fuse}	4.4M	**53.44**	**61.40**	10.53	**55.45**	**63.17**	16.70	8.18	14.52	0.09
$\text{PFE}_{\text{fuse+match}}$	4.4M	50.16	58.33	**31.88**	53.60	61.75	**35.99**	9.20	**20.82**	**0.84**

In Table 11.3, we show the results on two more challenging datasets: IJB-A and IJB-C. The images in these unconstrained datasets present larger variations in pose, occlusion, etc., and facial features could be more ambiguous. From the results, we can see that PFE achieves a more significant improvement on these benchmarks. In particular, on IJB-C at FAR= 0.001%, PFE reduces the error rate by 64%. In addition, both of these two benchmarks are evaluated in terms of face templates, which requires a good feature fusion strategy. The results of "PFE_{fuse}" indicates that fusing the original features with the learned uncertainty also helps the performance compared to the common average pooling strategy.

In Table 11.4, we report the results on three protocols of the latest benchmark, IJB-S. Again, PFE is able to improve performance in most cases. Notice that the gallery templates in the "Surveillance-to-still" and "Surveillance-to-booking" all include high-quality frontal mugshots, which present little feature ambiguity. Therefore, we only see a slight performance gap in these two protocols. But in the most challenging "surveillance-to-surveillance" protocol, larger improvement can be achieved by using uncertainty for matching. Besides, $\text{PFE}_{\text{fuse+match}}$ improves the performance significantly on all the open-set protocols, which indicates that MLS has more impact on the absolute pairwise score than the relative ranking.

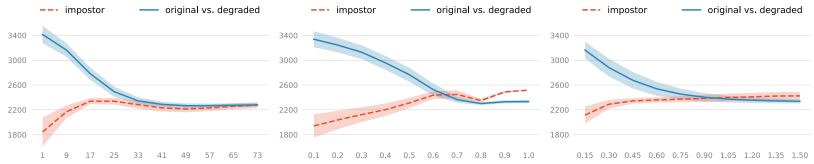
11.3.7 Qualitative Analysis

Why and when does PFE improve performance?

We first repeat the same experiments in Sect. 11.3.1 using the PFE representation and MLS. The same network is used as the base model here. As one can see in Fig. 11.5, although the scores of low-quality impostor pairs are still increasing, they converge to a point that is lower than the majority of genuine scores. Similarly, the scores of cross-quality genuine pairs converge to a point that is higher than the majority of impostor scores. This means the two types of errors discussed in Sect. 11.3.1 could be solved by PFE. This is further confirmed by the IJB-A results in Fig. 11.6. Figure 11.7 shows the distribution of estimated uncertainty on LFW, IJB-A, and IJB-S. As one can see, the "variance" of uncertainty increases in the following order: LFW $<$ IJB-A $<$ IJB-S. Compared with the performance in Sect. 11.3.6.2, we can see that PFE tends to achieve larger performance improvement on datasets with more diverse image quality.

What does DNN see and not see?

To answer this question, we train a decoder network on the original embedding, then apply it to PFE by sampling $\mathbf{z}$ from the estimated distribution $p(\mathbf{z}|\mathbf{x})$ of given $\mathbf{x}$. For a high-quality image (Fig. 11.8 Row 1), the reconstructed images tend to be very consistent without much variation, implying the model is very certain about the facial features in these images. In contrast, for a lower-quality input (Fig. 11.8 Row 2), a larger variation can be observed from the reconstructed images. In particular, attributes that can be clearly discerned from the image (e.g., thick eye-brow) are still consistent while attributes that cannot (e.g., eye shape) be discerned have larger variation. As for a mis-detected image (Fig. 11.8 Row 3), significant variation can be observed in the reconstructed images: the model does not see any salient feature in the given image.



(a) Gaussian Blur (b) Occlusion (c) Random Noise

Fig. 11.5 Repeated experiments on feature ambiguity dilemma with PFE. The same model in Fig. 11.2 is used as the base model and is converted to a PFE by training an uncertainty module. No additional training data or data augmentation is used for training

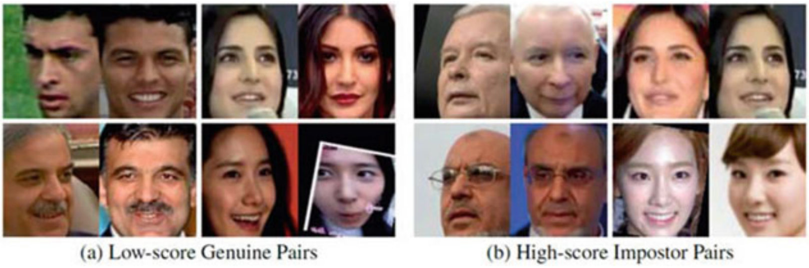


Fig. 11.6 Example genuine pairs from IJB-A dataset estimated with the lowest mutual likelihood scores and impostor pairs with the highest scores by the PFE version of the same 64-layer CNN model in Sect. 11.3.1. In comparison to Fig. 11.3, most images here are high-quality ones with clear features, which can mislead the model to be confident in a wrong way. Note that these pairs are not templates in the verification protocol

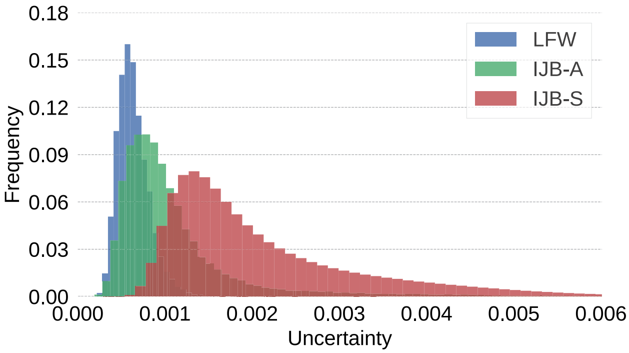


Fig. 11.7 Distribution of estimated uncertainty on different datasets. Here, "Uncertainty" refers to the harmonic mean of σ across all feature dimensions. Note that the estimated uncertainty is proportional to the complexity of the datasets. **Best viewed in color**

11.3.8 Risk-Controlled Face Recognition

In many scenarios, we may expect a higher performance than our system is able to achieve or we may want to make sure the system's performance can be controlled when facing complex application scenarios. Therefore, we would expect the model to reject input images if it is not confident. A common solution for this is to filter the images with a quality assessment tool. We show that PFE provides a natural solution for this task. We take all the images from LFW and IJB-A datasets for image-level face verification (We do not follow the original protocols here). The system is allowed to "filter out" a proportion of all images to maintain better performance. We then report the TAR@FAR$=0.001\%$ against the "Filter Out Rate". We consider two criteria for filtering: (1) the detection score of MTCNN [32] and (2) a confidence value predicted by PFE uncertainty module. Here the confidence for ith sample

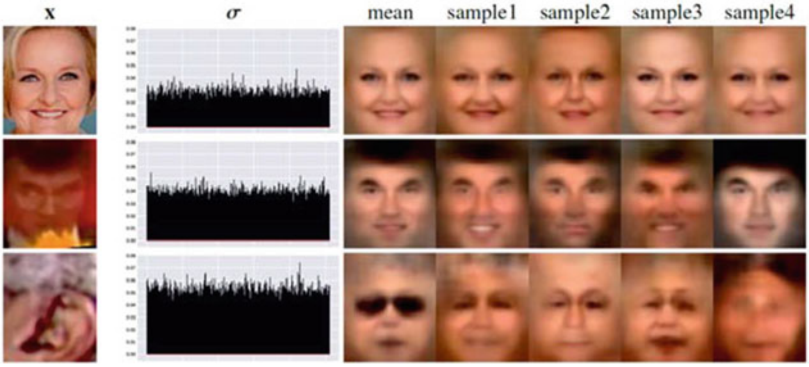


Fig. 11.8 Visualization results on a high-quality, a low-quality and a mis-detected image from IJB-A. For each input, 5 images are reconstructed by a pre-trained decoder using the mean and 4 randomly sampled $\mathbf{z}$ vectors from the estimated distribution $p(\mathbf{z}|\mathbf{x})$

is defined as the inverse of harmonic mean of $\boldsymbol{\sigma}_i$ across all dimensions. For fairness, both methods use the original deterministic embedding representations and cosine similarity for matching. To avoid saturated results, we use the model trained on CASIA-WebFace with AM-Softmax. The results are shown in Fig. 11.10. As one can see, the predicted confidence value is a better indicator of the potential recognition accuracy of the input image. This is an expected result since PFE is trained under supervision for the particular model while an external quality estimator is unaware of the kind of features used for matching by the model. Example images with high/low confidence/quality scores are shown in Fig. 11.9.

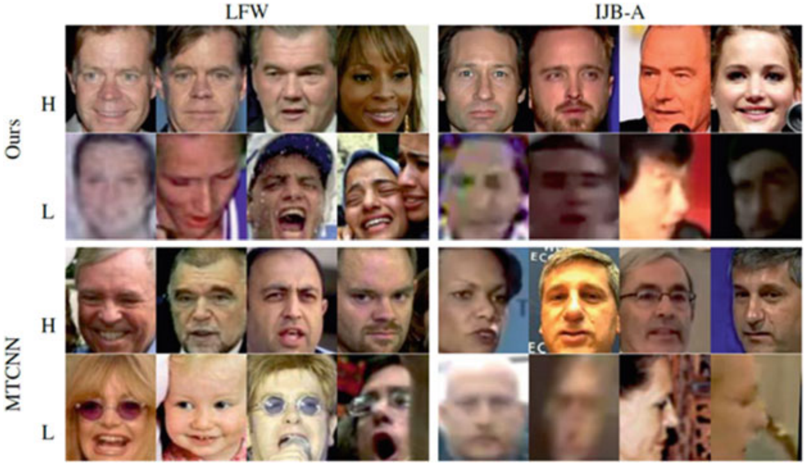
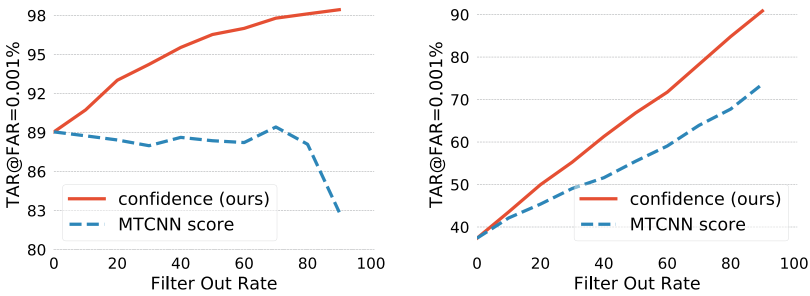


Fig. 11.9 Example images from LFW and IJB-A that are estimated with the highest (H) confidence/quality scores and the lowest (L) scores by PFE and MTCNN face detector



(a) LFW (b) IJB-A

Fig. 11.10 Comparison of verification performance on LFW and IJB-A (not the original protocol) by filtering a proportion of images using different quality criteria

11.4 Learning Representations with Data Uncertainty

The probabilistic face embeddings (PFEs) extend point-wise estimation of deep face features to distributional estimations by representing each face image as a Gaussian distribution in the feature space. In particular, the mean of the distribution is fixed as the original pre-trained face features while the variance is learned in a second stage. Therefore, later works have extended probabilistic embeddings to simultaneously learn the feature (mean) and the uncertainty (variance) of the Gaussian distribution, known as Data Uncertainty Learning (DUL) of face representations [2]. Two kinds of uncertainty learning methods were introduced by Chang et al.: one classification-based method and one regression-based. It is shown that compared with conventional representation learning methods, uncertainty-based learning is more robust against noisy training samples and improves the recognition performance on unconstrained face datasets.

11.4.1 Classification-Based DUL

Recall that in probabilistic face embeddings, each face image is represented as a Gaussian distribution in the feature space $p(\mathbf{z}_i|\mathbf{x}_i) = \mathcal{N}(\mathbf{z}_i; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\mathbf{I})$, where the mean $\boldsymbol{\mu}_i$ represents the identity features while the $\boldsymbol{\sigma}$ represents the uncertainties. The idea of classification-based DUL is to stochastically sample feature points from this distribution in a classification-based representation learning framework. This sampling process is differentiable with the re-parameterization trick as introduced in [16]:

$$\mathbf{s}_i = \boldsymbol{\mu}_i + \epsilon\boldsymbol{\sigma}_i, \quad \epsilon \sim \mathcal{N}(\mathbf{0}; \mathbf{I}). \tag{11.10}$$

Here ϵ is random noise sampled from a standard Gaussian distribution. This new features $\mathbf{s}_i$ can then be injected into other loss functions for learning face representations. For example, in the case of vanilla softmax cross-entropy loss, it would be:

$$\mathcal{L}_{softmax} = -\frac{1}{N} \Sigma_i \log \frac{e^{\mathbf{w}_{y_i}^T \mathbf{s}_i}}{\Sigma_c e^{\mathbf{w}_c^T \mathbf{s}_i}}. \tag{11.11}$$

To avoid the uncertainty $\boldsymbol{\sigma}_i$ from collapsing into Infinitesimal numbers, a regularization term is introduced during optimization. Specifically, as in the variational information bottleneck [1], a Kullback–Leibler divergence (KLD) is used as the regularization term:

$$\begin{aligned} \mathcal{L}_{KL} &= KL[\mathcal{N}(\mathbf{z}_i|\boldsymbol{\mu}_i; \boldsymbol{\sigma}_i^2\mathbf{I})|\mathcal{N}(\epsilon|\mathbf{0}, \mathbf{I})] \\ &= -\frac{1}{2}(1 + \log \boldsymbol{\sigma}_i^2 - \boldsymbol{\mu}_i^2 - \boldsymbol{\sigma}_i^2). \end{aligned} \tag{11.12}$$

Overall, the loss function for classification-based uncertainty learning would be $\mathcal{L}_{cls} = \mathcal{L}_{softmax} + \mathcal{L}_{KL}$.

11.4.2 Regression-Based DUL

Unlike the classification-based DUL, which converts the feature extraction process into a sampling process without altering the representation of target labels, regression-based DUL aims to represent each target label as a continuous vector in the feature space. And then the uncertainty learning problem can be formulated as a regression problem.

Let $\mathcal{W} \in \mathbb{R}^{D\times C}$ be the weight matrix of a pre-trained classification model. The weight vector $\mathbf{w}_i \in \mathcal{W}$ can be treated as a proxy representation vector of the ith class. The regression-based uncertainty learning aims to maximize the following log-likelihood:

$$\ln p(\mathbf{w}_c|\mathbf{x}_{i\in c}, \theta) = -\frac{(\mathbf{w}_c - \boldsymbol{\mu}_i)^2}{2\boldsymbol{\sigma}_i^2} - \frac{1}{2}\ln \boldsymbol{\sigma}_i^2 - \frac{1}{2}\ln 2\pi. \tag{11.13}$$

Since the log form of the overall likelihood is the sum of log-likelihood of each individual sample, the regression loss is given by:

$$\mathcal{L}_{rgs} = \frac{1}{2N}\Sigma_i \Sigma_{l\in D}[-\frac{(\mathbf{w}_{y_i}^{(l)} - \mu_i^{(l)})^2}{2\sigma_i^{(l)2}} - \frac{1}{2}\ln \sigma_i^{(l)2} - \frac{1}{2}\ln 2\pi], \tag{11.14}$$

where D and l refers to the dimensionality of the embeddings and lth dimension, respectively. Here, similar to the classification-based uncertainty learning, the regression loss function EL_{cls}, EL_{rgs} has a balancing effect that prevents the uncertainty estimations from being either too big or too small.

11.4.3 Experiments

In this section, we show the experimental results of DUL as well as its comparisons with PFE. The baseline models are trained on ResNet [7] with SE-blocks [8]. MS-Celeb-1M datasets [6] with 3,648,176 images of 79,891 subjects are used as the training set. All the results are originally reported in [2]. The PFE module is also re-implemented by Chang et al..

11.4.3.1 Comparisons with Baseline and PFE

To test the performance of DUL, Chang et al.experimented with three different classification loss functions, namely AM-Softmax [29], ArcFace [3], and L2-Softmax[24]. Then, corresponding models are trained for all these base models using PFE and DUL. The results are shown in Table 11.5. Note that DUL uses cosine similarity and average pooling for all the protocols. By incorporating uncertainties into the representation learning stage, DUL is able to achieve comparable or better performance than PFE on all protocols.

11.4.3.2 Training on Noisy Samples

As pointed out by Chang et al. [2], introducing data uncertainty into the learning stage is also able to improve the robustness of the model against the noisy training examples. To conduct the experiment, they pollute the original MS-Celeb-1M dataset with Gaussian

Table 11.5 Results of models trained on MS-Celeb-1M. "Original" refers to the deterministic embeddings. The better performance among each base model is shown in bold numbers. "PFE" uses mutual likelihood score for matching while DUL uses cosine similarity and average pooling "MF" refers to the rank 1 retrieval rate on MegaFace and IJB-C refers to the TPR@FPR metric

Base model	Representation	LFW	CFP-FP	MF(R1)	YTF	IJB-C	
						0.001%	0.01%
AM-Softmax [29]	Original	99.63	96.85	97.11	96.09	75.43	88.65
	PFE	99.68	94.57	97.18	96.12	86.24	92.11
	DUL$_{cls}$	**99.71**	97.28	**97.30**	**96.46**	**88.25**	**92.78**
	DUL$_{rgs}$	99.66	**97.61**	96.85	96.28	87.02	91.84
ArcFace [3]	Original	99.64	96.77	97.08	96.06	73.80	88.78
	PFE	99.68	95.34	96.55	96.32	86.69	92.28
	DUL$_{cls}$	**99.76**	97.01	**97.22**	96.20	**87.22**	**92.43**
	DUL$_{rgs}$	99.66	**97.11**	96.83	**96.38**	86.21	91.03
L2-Softmax [24]	Original	99.60	95.87	90.34	95.89	77.60	86.19
	PFE	**99.66**	86.45	90.64	95.98	79.33	87.28
	DUL$_{cls}$	99.63	**97.24**	**93.19**	**96.56**	**79.90**	**87.80**
	DUL$_{rgs}$	**99.66**	96.35	89.66	96.08	74.46	83.23

Table 11.6 Comparison of baseline model and DUL$_{cls/rgs}$ trained on noisy MS-Celeb-1M

Percent	Model	MegaFace	LFW	YTF	IJB-C		
					0.001%	0.01%	0.1%
0%	baseline	97.11	99.63	96.09	75.32	88.65	94.73
10%	baseline	96.64	99.63	96.16	64.96	86.00	94.82
	PFE	97.02	99.63	96.10	83.39	91.33	95.54
	DUL$_{cls}$	96.88	99.75	96.44	88.04	93.21	95.96
	DUL$_{rgs}$	96.05	99.71	96.46	84.74	91.56	95.30
20%	baseline	96.20	99.61	96.00	43.52	80.48	94.22
	PFE	96.90	99.61	95.86	82.03	90.89	95.38
	DUL$_{cls}$	96.37	99.71	96.68	89.01	93.24	95.97
	DUL$_{rgs}$	95.51	99.66	96.64	81.10	90.91	95.27
30%	baseline	95.72	99.60	95.45	31.51	76.09	93.11
	PFE	96.82	99.61	96.12	80.92	90.31	95.29
	DUL$_{cls}$	95.86	99.73	96.38	86.05	91.80	95.02
	DUL$_{rgs}$	94.96	99.66	96.66	81.54	91.20	95.32
40%	baseline	95.14	99.56	95.51	39.69	77.12	93.73
	PFE	96.59	99.59	95.94	77.72	89.46	94.82
	DUL$_{cls}$	95.33	99.66	96.54	84.15	92.60	95.85
	DUL$_{rgs}$	94.28	99.58	96.68	78.13	87.64	94.67

blur as noise and observe the performance trend along with the percentage of data being polluted. As shown in Table 11.6, DUL turns out to be the most robust against noisy training samples. We note that a similar observation is reported by Shi et al. [28]. In their work, Shi et al.reformulates the scaling factor in the softmax loss as an uncertainty term and shows that considering data uncertainty enables the model to learn from stronger and more diverse data augmentations.

11.5 Non-Gaussian Probabilistic Embeddings

In probabilistic face embeddings and aforementioned uncertainty-aware face representation learning, each input face image/template is represented as a Gaussian distribution in the latent feature space. Although practically this approach has achieved success in unconstrained face recognition, there remains a question of whether a Gaussian distribution is the best choice for the probabilistic representation. In fact, in most modern deep face recognition systems, the face representations are distributed on a unit hypersphere. Although we can regularize the mean of the projected Gaussian distribution onto this hypersphere, the samples from the distribution will still fall off the spherical manifold. Thus, to solve the issue, Li et al.have proposed Spherical Confidence Learning for Face Recognition [18], where non-Gaussian probabilistic embeddings are used to represent input data.

11.5.1 r-radius *von-Mises Fisher* Distribution

Typically, data points on a hypersphere can be modeled by a *von-Mises Fisher* (vMF) Distribution. Specifically for probabilistic embeddings, given an input face image data $\mathbf{x}$, a conditional *von-Mises Fisher* Distribution on d dimensional unit sphere $\mathbb{S}^{d-1} \subset \mathbb{R}^d$ is given by:

$$p(\mathbf{z}'|\mathbf{x}) = C_d(\kappa_{\mathbf{x}}) \exp(\kappa_{\mathbf{x}} \boldsymbol{\mu}_{\mathbf{x}}^T \mathbf{z}'), \tag{11.15}$$

$$C_d(\kappa_{\mathbf{x}}) = \frac{\kappa_{\mathbf{x}}^{d/2-1}}{(2\pi)^{d/2} \mathcal{I}_{d/2-1}(\kappa_{bx})}, \tag{11.16}$$

where $\mathbf{z}', \boldsymbol{\mu}_{\mathbf{x}} \in \mathbb{S}^{d-1}, \kappa_{\mathbf{x}} \geq 0$ (subscripts indicate statistical dependencies on $\mathbf{x}$) and $\mathcal{I}_\alpha$ denotes the modified Bessel function of the first kind at order α:

$$\mathcal{I}_\alpha = \Sigma_{m=0}^{\inf} \frac{1}{m!\Gamma(m+\alpha+1)} (\frac{x}{2})^{2m+\alpha}. \tag{11.17}$$

The parameters $\boldsymbol{\mu}_{\mathbf{x}}$ and $\kappa_{\mathbf{x}}$ are the mean direction and concentration parameters, respectively. The greater the value $\kappa_{\mathbf{x}}$, the higher the concentration around the mean $\boldsymbol{\mu}_{\mathbf{x}}$. The model degenerates to a uniform sphere for $\kappa_{\mathbf{x}} = 0$. Li et al. [18] further extended the distribution into a r-radius vMF that is defined over the support of an r-radius sphere $r\mathbb{S}^{d-1}$. Formally, for any $\mathbf{z} \in r\mathbb{S}^{d-1}$, there exists a one-to-one correspondence between z' and z such that $\mathbf{z} = r\mathbf{z}'$. Then, the r-radius vMF density (denoted as r-vMF($\boldsymbol{\mu}_{\mathbf{x}}, \kappa_{\mathbf{x}}$)) can be obtained by:

$$p(\mathbf{z}|\mathbf{x}) = \frac{C_d(\kappa_{\mathbf{x}})}{r^{d-1}} \exp(\kappa_{\mathbf{x}} \boldsymbol{\mu}_{\mathbf{x}}^T \mathbf{z}') \tag{11.18}$$

11.5.2 Spherical Confidence Face (SCF)

Modern deep face recognition networks are usually trained with a classifier loss, where the weight vectors in the last classification represent one the of target classes. Here, let $\mathbf{w}_{\mathbf{x}}$ denote the weight vector of the target class c of a sample image $\mathbf{x} \in c$. In SCF, $q(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - \mathbf{w}_{\mathbf{x}})$ is used to represent class c, where δ is the Dirac delta function. The loss function is thus to minimize the KL-divergence between $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$, which is equivalent to the cross entropy:

$$\begin{aligned}\mathcal{L}_{SCF} &= -\int_{r\mathbb{S}^{d-1}} (\mathbf{z}|\mathbf{x}) \log p(\mathbf{z}|\mathbf{x}) d\mathbf{z} \\ &= -\frac{\kappa(\mathbf{x})}{r}\boldsymbol{\mu}(\mathbf{x})^T \mathbf{w}_{\mathbf{x}\in c} - (\frac{d}{2}-1)\log\kappa(\mathbf{x}) \\ &\quad + \log(\mathcal{I}_{d/2-1}(\kappa(\mathbf{x}))) + \frac{d}{2}\log 2\pi r^2.\end{aligned} \tag{11.19}$$

Similar to PFE, this loss function is applied to learn confidence scores with fixed deterministic embeddings $\boldsymbol{\mu}(\mathbf{x})$ as backbones. However, by using the class representations, the loss function can be trained without sampling data pairs. As shown by Li et al. [18], data points that are closer to their target representation $\mathbf{w}_{\mathbf{x}}$ tend to have higher confidence $\kappa_{\mathbf{x}}$.

11.5.3 Feature Comparison

SCF also uses the Mutual Likelihood Score (MLS) as the comparison function between face images/templates. Unlike the original PFE, this MLS is defined over the aforementioned r-radius vMFs:

$$\begin{aligned}s(\mathbf{x}_i, \mathbf{x}_j) &= \log \int\int_{r\mathbb{S}^{d-1}\times r\mathbb{S}^{d-1}} p(\mathbf{z}_i|\mathbf{x}_i)p(\mathbf{z}_j|\mathbf{x}_j)\delta(\mathbf{z}_i - \mathbf{z}_j)d\mathbf{z}_i d\mathbf{z}_j \\ &= \log C_d(\kappa_i) + \log C_d(\kappa_j) - \log C_d(\tilde{\kappa}) - d\log r,\end{aligned} \tag{11.20}$$

where $\tilde{\kappa} = \|\mathbf{p}\|_2$, $\mathbf{p} = (\kappa_i\boldsymbol{\mu}_i + \kappa_j\boldsymbol{\mu}_j)$. We refer the readers to [18] for detailed derivations.

11.5.4 Feature Pooling

Similar to Gaussian PFE, the spherical confidence of r-radius vMF can be used for feature pooling. Given two sets of images $x_i \in A$ and $x_j \in B$, the fused representation is:

$$\mathbf{z}_A = \frac{\Sigma_{x_i\in A}\kappa_i\boldsymbol{\mu}_i}{\Sigma_{x_i\in A}\kappa_i}, \quad \mathbf{z}_B = \frac{\Sigma_{x_j\in B}\kappa_j\boldsymbol{\mu}_j}{\Sigma_{x_j\in B}\kappa_j}. \tag{11.21}$$

The similarity between the two sets is then measured by the cosine similarity $\cos < \mathbf{z}_A, \mathbf{z}_B >$.

11.5.5 Experiments

In this section, we show the experimental results of SCF as well as its comparisons with PFE. ResNet100 and ResNet34 [7] are employed as deterministic embedding backbones. For both SCF and PFE, the mean direction module $\boldsymbol{\mu}$ is initialized by deterministic embeddings and fixed throughout the training. MS1MV2 is used as the training set. The PFE module is also

Table 11.7 Comparison of baseline models, PFE and SCF on MS1MV2 dataset using ResNet100 as the backbone network. For IJB-C, "0.001%" and "0.01%" refer to the False Accept Rate in the verification protocol. For MegaFace, ID refers to "Identification", while "Ver" refers to the verification (TAR@FAR=0.0001%)

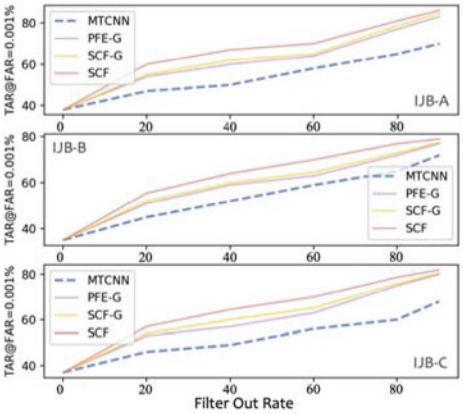
Method	LFW	CFP-FP	AgeDB	MegaFace		IJB-C	
				ID	Ver	0.001%	0.01%
CosFace	99.78	98.45	98.03	80.56	96.56	93.86	95.95
+ PFE	**99.80**	98.56	98.15	80.44	96.49	94.09	96.04
+ SCF-G	99.79	98.54	98.14	80.57	96.61	94.15	96.02
+ SCF	**99.80**	**98.59**	**98.26**	**80.93**	**96.90**	**94.78**	**96.22**
ArcFace	99.77	98.27	98.28	81.03	96.98	93.15	95.60
+ PFE	99.78	98.33	98.21	80.53	96.43	92.95	95.32
+ SCF-G	99.79	98.31	98.23	81.23	97.11	93.85	95.33
+ SCF	**99.82**	**98.40**	**98.30**	**81.40**	**97.15**	**94.04**	**96.09**

re-implemented by Li et al.. Besides the spherical version of SCF, they also implemented a Gaussian version, denoted as SCF-G for comparison.

An overview of the comparison results from Li et al. [18] is shown in Table 11.7. Note that they also report the performance on many other benchmarks while we only select the representative ones here. A ResNet100 is used as the backbone for all these datasets. Two backbone networks are trained with CosFace [30] and ArcFace [3], respectively, as the deterministic embeddings for initialization. In the first row of each part, the performance of the original backbones are reported. The following rows (PFE, SCF-G, and SCF) report the performance of different variants of probabilistic embeddings. According to the results, the SCF loss function also works when it is applied to the Gaussian probabilistic embeddings. Furthermore, SCF constantly outperforms PFE and SCF-G because it better aligns with the feature distribution of the backbone embeddings. In the original paper, the authors also observed that SCF leads to less improvement on deeper networks than on shallower networks (ResNet34), and they hypothesize that this is because deeper embeddings already exhibit high separability and less ambiguous samples lie on the classification boundaries.

Li et al.also conducted an experiment of risk-controlled face recognition experiment, where a certain number of face images can be filtered out during face verification on IJB-A, IJB-B, and IJB-C. The results are shown in Fig. 11.11. It can be seen that both PFE and SCF-G outperform face detector (MTCNN) as quality indicator for input face images. SCF further outperforms its Gaussian counterpart on the task by taking the non-Gaussianity of the feature distribution into consideration.

Fig. 11.11 Experiments of risk-controlled face recognition on IJB-A, IJB-B, and IJB-C



11.6 Summary

In this chapter, we introduced the motivation of data uncertainty estimation in deep face recognition systems as well as its applications. From a probabilistic perspective, traditional deep face embeddings can be viewed as deterministic face embeddings, which do not take intrinsic data uncertainty of image samples into account. And therefore they will inevitably fail on ambiguous samples that are hardly recognizable. To solve the issue, Probabilistic Face Embedding (PFE) is introduced to represent each face image/template as a Gaussian distribution in the feature space. The variances of these distributions are then used as uncertainty estimation for feature comparison, feature pooling, and quality assessment. Data Uncertainty Learning (DUL) further extends the uncertainty estimation into the learning stage of backbone neural networks and improves their robustness against noisy training samples. Spherical Gaussian Face (SCF) extends SCF to von-Mises Fisher distributions to model the uncertainty on a spherical feature space, which better aligns with the feature distribution of most current deep face recognition systems.

References

1. Alemi, A.A., Fischer, I., Dillon, J.V., Murphy, K.: Deep variational information bottleneck. In: ICLR (2017)
2. Chang, J., Lan, Z., Cheng, C., Wei, Y.: Data uncertainty learning in face recognition. In: CVPR (2020)
3. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019)
4. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML (2016)
5. Gong, S., Boddeti, V.N., Jain, A.K.: On the capacity of face representation. arXiv:1709.10433 (2017)

6. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: ECCV (2016)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
8. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)
9. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Tech. rep (2007)
10. Kalka, N.D., Maze, B., Duncan, J.A., O'Connor, K.J., Elliott, S., Hebert, K., Bryan, J., Jain, A.K.: IJB-S : IARPA Janus Surveillance Video Benchmark . In: BTAS (2018)
11. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: CVPR (2016)
12. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In: BMVC (2015)
13. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: NIPS (2017)
14. Khan, S., Hayat, M., Zamir, W., Shen, J., Shao, L.: Striking the right balance with uncertainty. arXiv:1901.07590 (2019)
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2013)
16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
17. Klare, B.F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Jain, A.K.: Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In: CVPR (2015)
18. Li, S., Xu, J., Xu, X., Shen, P., Li, S., Hooi, B.: Spherical confidence learning for face recognition. In: CVPR (2021)
19. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: CVPR (2017)
20. Liu, Y., Yan, J., Ouyang, W.: Quality aware network for set to set recognition. In: CVPR (2017)
21. MacKay, D.J.: A practical bayesian framework for backpropagation networks. Neural Comput. (1992)
22. Maze, B., Adams, J., Duncan, J.A., Kalka, N., Miller, T., Charles, O., Jain, A.K., Tyler, N., Anderson, J., Cheney, J., Grother, P.: Iarpa janus benchmark-c: Face dataset and protocol. In: ICB (2018)
23. Neal, R.M.: Bayesian Learning for Neural Networks. Ph.D. Thesis, University of Toronto (1995)
24. Ranjan, R., Castillo, C.D., Chellappa, R.: L2-constrained softmax loss for discriminative face verification. arXiv:1703.09507 (2017)
25. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR (2015)
26. Sengupta, S., Chen, J.C., Castillo, C., Patel, V.M., Chellappa, R., Jacobs, D.W.: Frontal to profile face verification in the wild. In: WACV (2016)
27. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: ICCV (2019)
28. Shi, Y., Yu, X., Sohn, K., Chandraker, M., Jain, A.K.: Towards universal representation learning for deep face recognition. In: CVPR (2020)
29. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Sig. Process, Lett (2018)
30. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: CVPR (2018)
31. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV (2016)

32. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV (2016)
33. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: CVPR (2011)
34. Xie, W., Zisserman, A.: Multicolumn networks for face recognition. In: BMVC (2018)
35. Yang, J., Ren, P., Zhang, D., Chen, D., Wen, F., Li, H., Hua, G.: Neural aggregation network for video face recognition. In: CVPR (2017)
36. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv:1411.7923 (2014)
37. Zafar, U., Ghafoor, M., Zia, T., Ahmed, G., Latif, A., Malik, K.R., Sharif, A.M.: Face recognition with bayesian convolutional networks for robust surveillance systems. EURASIP J, Image Video Process (2019)

Reducing Bias in Face Recognition 12

Sixue Gong, Xiaoming Liu, and Anil K. Jain

12.1 The Bias in Face Recognition

In this chapter, we estimate the demographic bias in FR (Face Recognition) algorithms and introduce two methods to mitigate the demographic impact on FR performance. The goal of this research is to learn a fair face representation, where the faces of every group could be equally well-represented. Specifically, we explore de-biasing approaches by designing two different network architectures using deep learning. Meanwhile, we evaluate the model's demographic bias on various datasets to show how much bias is mitigated in our attempt at improving the fairness of face representations extracted from CNNs.

In the first method, we present a de-biasing adversarial network (DebFace) that learns to extract disentangled feature representations for both unbiased face recognition and demographics estimation. The proposed network consists of one identity classifier and three demographic classifiers (for gender, age, and race) that are trained to distinguish identity and demographic attributes, respectively. Adversarial learning is adopted to minimize correlation among feature factors so as to abate bias influence from other factors. We also design a scheme to combine demographics with identity features to strengthen the robustness of face representation in different demographic groups.

The second method, group adaptive classifier (GAC), learns to mitigate bias by using adaptive convolution kernels and attention mechanisms on faces based on their demographic

S. Gong (✉) · X. Liu · A. K. Jain
Department of Computer Science & Engineering, Michigan State University, Michigan, USA
e-mail: gongsixu@msu.edu

X. Liu
e-mail: liuxm@cse.msu.edu

A. K. Jain
e-mail: jain@egr.msu.edu

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_12

attributes. The adaptive module comprises kernel masks and channel-wise attention maps for each demographic group so as to activate different facial regions for identification, leading to more discriminative features pertinent to their demographics. We also introduce an automated adaptation strategy which determines whether to apply adaptation to a certain layer by iteratively computing the dissimilarity among demographic-adaptive parameters, thereby increasing the efficiency of the adaptation learning.

The experimental results on benchmark face datasets (e.g., RFW [92], LFW [39], IJB-A [47] and IJB-C [63]) show that our approach is able to reduce bias in face recognition on various demographic groups as well as maintain the competitive performance.

12.2 Fairness Learning and De-biasing Algorithms

We start by reviewing recent advances in fairness learning and de-biasing algorithms. Previous efforts on fairness techniques are proposed to prevent machine learning models from utilizing statistical bias in training data, including adversarial training [3, 34, 62, 95], subgroup constraint optimization [43, 96, 114], data pre-processing (e.g., weighted sampling [28], and data transformation [6]), and algorithm post-processing [45, 69]. For example, Alexander et al. [4] develop an algorithm to mitigate the hidden biases within training data to uncover deep learning bias. Another example of promising approaches in fair representations is to preserve all discerning information about the data attributes or task-related attributes but eliminate the prejudicial effects from sensitive factors by adversarial training [16, 32, 65, 79, 107]. Locatello et al. [58] show the feature disentanglement is consistently correlated with increasing fairness of general purpose representations by analyzing $12,600$ SOTA models.

In the FR community, the biased performance of FR algorithms is not only an issue of current DNN-based models but also in prior-DNN era. For example, Klare et al. [46] propose to train classifiers separately on each demographic group to reduce bias from hand-crafted face representations. In the DNN era, however, the data-driven FR models inherit most of their bias from large-scale face datasets with highly-imbalanced distribution [103, 109]. Several efforts have been made to address data bias in FR. The work of [8] presents multiple data-driven factors (e.g., image quality and population statistics) for assessing bias in FR algorithms. The study in [84] also shows the correlation of image quality and FR bias. Dooley et al. [23] construct an improved face dataset with better quality and use it to study bias in FR via comparing various algorithms by human reviewers.

To mitigate data bias in FR, center-based feature transfer learning [103] and large margin feature augmentation [93] are proposed to augment features of under-represented identities and equalize identity distribution. These studies mainly focus on bias from insufficient identity samples, but ignore the influence of demographic imbalance on the dataset. In contrast, the studies in [70, 92] address the demographic bias in FR by leveraging unlabeled faces to improve the performance in groups with fewer samples. Wang et al. [91] collect a race-balanced face dataset and propose skewness-aware reinforcement learning to mitigate racial bias in FR. What is noteworthy is that a balanced dataset does not necessarily benefit the bias mitigation in FR [30]. Another approach to data bias is to utilize synthetic face data, such as the fully annotated face images synthesized by [49], and the face images via transformation on racial characteristics constructed by Yucer et al. [104]. They later introduce an alternative methodology on race bias using face phenotype attributes [105].

Apart from data pre-processing, an alternative direction is feature post-processing. For instance, gender bias is mitigated by transforming a pre-trained deep face representation by minimizing the intra-class variance on each gender group via von Mises-Fisher loss [14]. Another adds-on framework is proposed to improve the accuracy of a given face representation while reducing its bias in performance via triplet loss whose triplets are carefully sampled based on sensitive factors [75]. Unlike prior work, we design a GAC framework to customize the classifier for each demographic group, which, if successful, would lead to mitigated bias. This framework is presented in the following Sect. 12.5.

Inspired by adversarial training in machine learning, a gender-neutral face representation is proposed to reduce the gender information present in face embeddings extracted from any well-trained FR network [20]. Our second de-biasing framework, *DebFace*, leverages a similar idea, which disentangles face representations to de-bias both FR and demographic attribute estimation. Section 12.4 discusses DebFace in more details.

12.3 Problem Definition

We now give a specific definition of the problem addressed in this chapter. The ultimate goal of unbiased face recognition is that, given a face recognition system, there is no statistically significant difference among the performance in different categories of face images. Despite the research on pose-invariant face recognition that aims for equal performance on all poses [86, 102], we believe that it is inappropriate to define variations like pose, illumination, or resolution, as the categories. These are instantaneous *image-related* variations with intrinsic bias. For instance, large-pose or low-resolution faces are inherently harder to be recognized than frontal-view high-resolution faces.

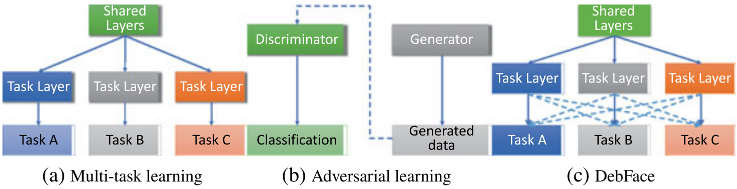
Instead, we would like to define *subject-related* properties such as demographic attributes as the categories. *A face recognition system is* ***biased*** *if it performs worse on certain demographic cohorts.* For practical applications, it is important to consider what demographic biases may exist, and whether these are intrinsic biases across demographic cohorts or algorithmic biases derived from the algorithm itself. This motivates us to analyze the demographic influence on face recognition performance and strive to reduce algorithmic bias for face recognition systems. One may achieve this by training on a dataset containing uniform samples over the cohort space. However, the demographic distribution of a dataset is often imbalanced and underrepresents demographic minorities while overrepresenting majorities. Naively re-sampling a balanced training dataset may still induce bias since the diversity of latent variables is different across cohorts and the instances cannot be treated fairly during training. To mitigate demographic bias, we propose two face de-biasing frameworks that reduce demographic bias over face identity features while maintaining the overall verification performance in the meantime.

12.4 Jointly De-biasing Face Recognition and Demographic Attribute Estimation

In this section, we introduce another framework to address the influence of demographic bias on face recognition. With the technique of adversarial learning, we attack this issue from a different perspective. Specifically, we assume that if the face representation does not carry discriminative information of demographic attributes, it would be unbiased in terms of demographics. Given this assumption, one common way to remove demographic information from face representations is to perform feature disentanglement via adversarial learning (Fig. 12.1b). That is, the classifier of demographic attributes can be used to encourage the identity representation to *not* carry demographic information. However, one issue of this common approach is that the demographic classifier itself could be biased (e.g., the race classifier could be biased on gender), and hence it will act differently while disentangling faces of different cohorts. This is clearly undesirable as it leads to demographic biased identity representation.

To resolve the chicken-and-egg problem, we propose to *jointly* learn unbiased representations for both the identity and demographic attributes. Specifically, starting from a multi-task learning framework that learns disentangled feature representations of gender, age, race, and identity, respectively, we request the classifier of each task to act as adversarial supervision for the other tasks (e.g., the dash arrows in Fig. 12.1c). These four classifiers help each other to achieve better feature disentanglement, resulting in unbiased feature representations for both the identity and demographic attributes. As shown in Fig. 12.1, our framework is in sharp contrast to both multi-task learning and adversarial learning.

Moreover, since the features are disentangled into the demographic and identity, our face representations also contribute to privacy-preserving applications. It is worth noticing that such identity representations contain little demographic information, which could undermine



(a) Multi-task learning (b) Adversarial learning (c) DebFace

Fig. 12.1 Methods to learn different tasks simultaneously. Solid lines are typical feature flow in CNN, while dash lines are adversarial losses

the recognition competence since demographic features are *part* of identity-related facial appearance. To retain the recognition accuracy on demographic biased face datasets, we propose another network that combines the demographic features with the demographic-free identity features to generate a new identity representation for face recognition.

The key contributions and findings of this work are:

◇ A thorough analysis of deep learning-based face recognition performance on three different demographics: (i) gender, (ii) age, and (iii) race.

◇ A de-biasing face recognition framework, called DebFace, that generates disentangled representations for both identity and demographics recognition while jointly removing discriminative information from other counterparts.

◇ The identity representation from DebFace (DebFace-ID) shows lower bias on different demographic cohorts and also achieves SOTA face verification results on demographic-unbiased face recognition.

◇ The demographic attribute estimations via DebFace are less biased across other demographic cohorts.

◇ Combining ID with demographics results in more discriminative features for face recognition on biased datasets.

12.4.1 Adversarial Learning and Disentangled Representation

We first review previous work related to adversarial learning and representation disentanglement. Adversarial learning [73] has been well explored in many computer vision applications. For example, Generative Adversarial Networks (GANs) [26] employ adversarial learning to train a generator by competing with a discriminator that distinguishes real images from synthetic ones. Adversarial learning has also been applied to domain adaptation [60, 83, 87, 88]. An issue of current interest is to learn interpretable representations with semantic meaning [100]. Many studies have been learning factors of variations in the data by supervised learning [55–57, 85, 86], or semi-supervised/unsupervised learning [44, 59, 66, 113], referred to as disentangled representation. For supervised disentangled feature learning, adversarial networks are utilized to extract features that only contain discriminative information of a target task. For face recognition, Liu et al. [57] propose a disentangled representation by training an adversarial autoencoder to extract features that can capture identity discrimination and its complementary knowledge. In contrast, our proposed DebFace differs from prior works in that each branch of the multi-task network acts as both a generator to its branch and discriminators to other branches (Fig. 12.1c).

12.4.2 Methodology

12.4.2.1 Algorithm Design

The proposed network takes advantage of the relationship between demographics and face identities. On the one hand, demographic characteristics are highly correlated with face features. On the other hand, demographic attributes are heterogeneous in terms of data type and semantics [31]. Being male, for example, does not necessarily indicate a specific age or race of an individual. Accordingly, we present a framework that jointly generates demographic features and identity features from a single face image by considering both the aforementioned attribute correlation and attribute heterogeneity in a DNN.

While our main goal is to mitigate demographic bias from face representation, we observe that demographic estimations are biased as well (see Fig. 12.5). How can we remove the bias of face recognition when demographic estimations themselves are biased? Cook et al. [15] investigated this effect and found the performance of face recognition is affected by multiple demographic covariates. We propose a de-biasing network, DebFace, that disentangles the representation into gender (DebFace-G), age (DebFace-A), race (DebFace-R), and identity (DebFace-ID), to decrease bias of both face recognition and demographic estimations. Using adversarial learning, the proposed method is capable of jointly learning multiple discriminative representations while ensuring that each classifier cannot distinguish among classes through non-corresponding representations.

Though less biased, DebFace-ID loses demographic cues that are useful for identification. In particular, race and gender are two critical components that constitute face patterns. Hence, we desire to incorporate race and gender with DebFace-ID to obtain a more integrated face representation. We employ a light-weight fully-connected network to aggregate the representations into a face representation (DemoID) with the same dimensionality as DebFace-ID.

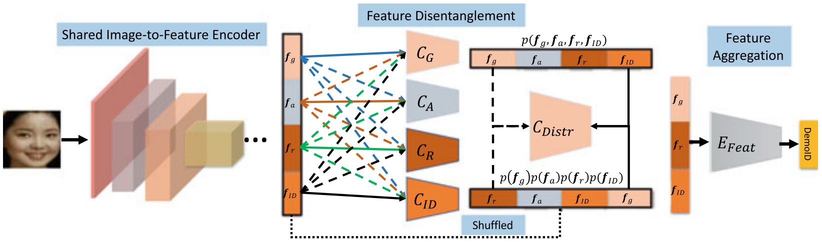


Fig. 12.2 Overview of the proposed De-biasing face (DebFace) network. DebFace is composed of three major blocks, i.e., a shared feature encoding block, a feature disentangling block, and a feature aggregation block. The solid arrows represent the forward inference, and the dashed arrows stand for adversarial training. During inference, either DebFace-ID (i.e., $\mathbf{f}_{ID}$) or DemoID can be used for face matching given the desired trade-off between biasness and accuracy

12.4.2.2 Network Architecture

Figure 12.2 gives an overview of the proposed DebFace network. It consists of four components: the shared image-to-feature encoder E_{Img}, the four attribute classifiers (including gender C_G, age C_A, race C_R, and identity C_{ID}), the distribution classifier C_{Distr}, and the feature aggregation network E_{Feat}. We assume access to N labeled training samples $\{(\mathbf{x}^{(i)}, y_g^{(i)}, y_a^{(i)}, y_r^{(i)}, y_{id}^{(i)})\}_{i=1}^N$. Our approach takes an image $\mathbf{x}^{(i)}$ as the input of E_{Img}. The encoder projects $\mathbf{x}^{(i)}$ to its feature representation $E_{Img}(\mathbf{x}^{(i)})$. The feature representation is then decoupled into four D-dimensional feature vectors, gender $\mathbf{f}_g^{(i)}$, age $\mathbf{f}_a^{(i)}$, race $\mathbf{f}_r^{(i)}$, and identity $\mathbf{f}_{ID}^{(i)}$, respectively. Next, each attribute classifier operates on the corresponding feature vector to correctly classify the target attribute by optimizing parameters of both E_{Img} and the respective classifier C_*. For a demographic attribute with K categories, the learning objective $\mathcal{L}_{C_{Demo}}(\mathbf{x}, y_{Demo}; E_{Img}, C_{Demo})$ is the standard cross-entropy loss function. For the $n-$identity classification, we adopt AM-Softmax [89] as the objective function $\mathcal{L}_{C_{ID}}(\mathbf{x}, y_{id}; E_{Img}, C_{ID})$. To de-bias all of the feature representations, adversarial loss $\mathcal{L}_{Adv}(\mathbf{x}, y_{Demo}, y_{id}; E_{Img}, C_{Demo}, C_{ID})$ is applied to the above four classifiers such that each of them will not be able to predict correct labels when operating irrelevant feature vectors. Specifically, given a classifier, the remaining three attribute feature vectors are imposed on it and attempt to mislead the classifier by only optimizing the parameters of E_{Img}. To further improve the disentanglement, we also reduce the mutual information among the attribute features by introducing a distribution classifier C_{Distr}. C_{Distr} is trained to identify whether an input representation is sampled from the joint distribution $p(\mathbf{f}_g, \mathbf{f}_a, \mathbf{f}_r, \mathbf{f}_{ID})$ or the multiplication of margin distributions $p(\mathbf{f}_g)p(\mathbf{f}_a)p(\mathbf{f}_r)p(\mathbf{f}_{ID})$ via a binary cross-entropy loss $\mathcal{L}_{C_{Distr}}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr})$, where y_{Distr} is the distribution label. Similar to adversarial loss, a factorization objective function $\mathcal{L}_{Fact}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr})$ is utilized to restrain the C_{Distr} from distinguishing the real distribution and thus minimizes the mutual information of the four attribute representations. Both adversarial loss and factorization loss are detailed in Sect. 12.4.2.3. Altogether, DebFace endeavors to minimize the joint loss:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, y_{Demo}, y_{id}, y_{Distr}; E_{Img}, C_{Demo}, C_{ID}, C_{Distr}) = \\ &\mathcal{L}_{C_{Demo}}(\mathbf{x}, y_{Demo}; E_{Img}, C_{Demo}) \\ &+ \mathcal{L}_{C_{ID}}(\mathbf{x}, y_{id}; E_{Img}, C_{ID}) \\ &+ \mathcal{L}_{C_{Distr}}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr}) \\ &+ \lambda\mathcal{L}_{Adv}(\mathbf{x}, y_{Demo}, y_{id}; E_{Img}, C_{Demo}, C_{ID}) \\ &+ \nu\mathcal{L}_{Fact}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr}),\end{aligned} \tag{12.1}$$

where λ and ν are hyper-parameters determining how much the representation is decomposed and decorrelated in each training iteration.

The discriminative demographic features in DebFace-ID are weakened by removing demographic information. Fortunately, our de-biasing network preserves all pertinent demographic features in a disentangled way. Basically, we train another multi-layer perceptron

(MLP) E_{Feat} to aggregate DebFace-ID and the demographic embeddings into a unified face representation DemoID. Since age generally does not pertain to a person's identity, we only consider gender and race as the identity-informative attributes. The aggregated embedding, $\mathbf{f}_{DemoID} = E_{feat}(\mathbf{f}_{ID}, \mathbf{f}_g, \mathbf{f}_r)$, is supervised by an identity-based triplet loss:

$$\mathcal{L}_{E_{Feat}} = \frac{1}{M}\sum_{i=1}^{M}[\|\mathbf{f}^{(i)}_{DemoID^a} - \mathbf{f}^{(i)}_{DemoID^p}\|_2^2 - \|\mathbf{f}^{(i)}_{DemoID^a} - \mathbf{f}^{(i)}_{DemoID^n}\|_2^2 + \alpha]_+, \quad (12.2)$$

where $\{\mathbf{f}^{(i)}_{DemoID^a}, \mathbf{f}^{(i)}_{DemoID^p}, \mathbf{f}^{(i)}_{DemoID^n}\}$ is the ith triplet consisting of an anchor, a positive, and a negative DemoID representation, M is the number of hard triplets in a mini-batch. $[x]_+ = \max(0, x)$, and α is the margin.

12.4.2.3 Adversarial Training and Disentanglement

As discussed in Sect. 12.4.2.2, the adversarial loss aims to minimize the task-independent information semantically, while the factorization loss strives to dwindle the interfering information statistically. We employ both losses to disentangle the representation extracted by E_{Img}. We introduce the adversarial loss as a means to learn a representation that is invariant in terms of certain attributes, where a classifier trained on it cannot correctly classify those attributes using that representation. We take one of the attributes, e.g., gender, as an example to illustrate the adversarial objective. First of all, for a demographic representation $\mathbf{f}_{Demo}$, we learn a gender classifier on $\mathbf{f}_{Demo}$ by optimizing the classification loss $\mathcal{L}_{C_G}(\mathbf{x}, y_{Demo}; E_{Img}, C_G)$. Secondly, for the same gender classifier, we intend to maximize the chaos of the predicted distribution [41]. It is well known that a uniform distribution has the highest entropy and presents the most randomness. Hence, we train the classifier to predict the probability distribution as close as possible to a uniform distribution over the category space by minimizing the cross-entropy:

$$\mathcal{L}^G_{Adv}(\mathbf{x}, y_{Demo}, y_{id}; E_{Img}, C_G) = -\sum_{k=1}^{K_G}\frac{1}{K_G}\cdot\left(\log\frac{e^{C_G(\mathbf{f}_{Demo})_k}}{\sum_{j=1}^{K_G} e^{C_G(\mathbf{f}_{Demo})_j}} + \log\frac{e^{C_G(\mathbf{f}_{ID})_k}}{\sum_{j=1}^{K_G} e^{C_G(\mathbf{f}_{ID})_j}}\right), \quad (12.3)$$

where K_G is the number of categories in gender,[1] and the ground-truth label is no longer an one-hot vector, but a K_G-dimensional vector with all elements being $\frac{1}{K_G}$. The above loss function corresponds to the dash lines in Fig. 12.2. It strives for gender-invariance by finding a representation that makes the gender classifier C_G perform poorly. We minimize the adversarial loss by only updating parameters in E_{Img}.

We further decorrelate the representations by reducing the mutual information across attributes. By definition, the mutual information is the relative entropy (KL divergence)

[1] In our case, $K_G = 2$, i.e., male and female.

between the joint distribution and the product distribution. To increase uncorrelation, we add a distribution classifier C_{Distr} that is trained to simply perform a binary classification using $\mathcal{L}_{C_{Distr}}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr})$ on samples $\mathbf{f}_{Distr}$ from both the joint distribution and dot product distribution. Similar to adversarial learning, we factorize the representations by tricking the classifier via the same samples so that the predictions are close to random guesses,

$$\mathcal{L}_{Fact}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr}) = -\sum_{i=1}^{2} \frac{1}{2} \log \frac{e^{C_{Distr}(\mathbf{f}_{Distr})_i}}{\sum_{j=1}^{2} e^{C_{Distr}(\mathbf{f}_{Distr})_j}}. \tag{12.4}$$

In each mini-batch, we consider $E_{Img}(\mathbf{x})$ as samples of the joint distribution $p(\mathbf{f}_g, \mathbf{f}_a, \mathbf{f}_r, \mathbf{f}_{ID})$. We randomly shuffle feature vectors of each attribute, and re-concatenate them into $4D$-dimension, which are approximated as samples of the product distribution $p(\mathbf{f}_g)p(\mathbf{f}_a)p(\mathbf{f}_r)p(\mathbf{f}_{ID})$. During factorization, we only update E_{Img} to minimize mutual information between decomposed features.

12.4.3 Experiments

12.4.3.1 Datasets and Pre-processing

We utilize 15 total face datasets in this work, for learning the demographic estimation models, the baseline face recognition model, DebFace model as well as their evaluation. To be specific, CACD [10], IMDB [71], UTKFace [112], AgeDB [64], AFAD [67], AAF [13], FG-NET [1], RFW [92], IMFDB-CVIT [76], Asian-DeepGlint [2], and PCSO [17] are the datasets for training and testing demographic estimation models; and MS-Celeb-1M [29], LFW [39], IJB-A [47], and IJB-C [63] are for learning and evaluating face verification models. Table 12.1 reports the statistics of training and testing datasets involved in all the experiments of both GAC and DebFace, including the total number of face images, the total number of subjects (identities), and whether the dataset contains the annotation of gender, age, race, or identity (ID). All faces are detected by MTCNN [108]. Each face image is cropped and resized to 112 × 112 pixels using a similarity transformation based on the detected landmarks.[2]

12.4.3.2 Implementation Details

DebFace is trained on a cleaned version of MS-Celeb-1M [18], using the ArcFace architecture [18] with 50 layers for the encoder E_{Img}. Since there are no demographic labels in MS-Celeb-1M, we first train three demographic attribute estimation models for gender, age, and race, respectively. For age estimation, the model is trained on the combination of CACD, IMDB, UTKFace, AgeDB, AFAD, and AAF datasets. The gender estimation model

[2] https://yanweifu.github.io/FG_NET_data.

Table 12.1 Statistics of training and testing datasets used in the paper

Dataset	# Of images	# Of subjects	Contains the label of			
			Gender	Age	Race	ID
CACD [10]	163, 446	2, 000	No	Yes	No	Yes
IMDB [71]	460, 723	20, 284	Yes	Yes	No	Yes
UTKFace [112]	24, 106	–	Yes	Yes	Yes	No
AgeDB [64]	16, 488	567	Yes	Yes	No	Yes
AFAD [67]	165, 515	–	Yes	Yes	Yes^a	No
AAF [13]	13, 322	13, 322	Yes	Yes	No	Yes
FG-NET	1, 002	82	No	Yes	No	Yes
RFW [92]	665, 807	–	No	No	Yes	Partial
BUPT-Balancedface [91]	1, 251, 430	28, 000	No	No	Yes	Yes
IMFDB-CVIT [76]	34, 512	100	Yes	Age Groups	Yes*	Yes
Asian-DeepGlint [2]	2, 830, 146	93, 979	No	No	Yes^a	Yes
MS-Celeb-1M [29]	5, 822, 653	85, 742	No	No	No	Yes
PCSO [17]	1, 447, 607	5, 749	Yes	Yes	Yes	Yes
LFW [39]	13, 233	5, 749	No	No	No	Yes
IJB-A [47]	25, 813	500	Yes	Yes	Skin Tone	Yes
IJB-C [63]	31, 334	3, 531	Yes	Yes	Skin Tone	Yes

a East Asian
* Indian

is trained on the same datasets except CACD which contains no gender labels. We combine AFAD, RFW, IMFDB-CVIT, and PCSO for race estimation training. All three models use ResNet [33] with 34 layers for age, 18 layers for gender and race. We discuss the evaluation results of the demographic attribute estimation models in Sect. 12.6.

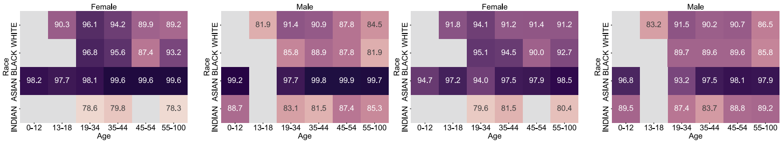
We predict the demographic labels of MS-Celeb-1M with the well-trained demographic models. Our DebFace is then trained on the re-labeled MS-Celeb-1M using SGD with a momentum of 0.9, a weight decay of 0.01, and a batch size of 256. The learning rate starts from 0.1 and drops to 0.0001 following the schedule at 8, 13, and 15 epochs. The dimensionality of the embedding layer of E_{Img} is 4×512, i.e., each attribute representation (gender, age, race, ID) is a 512-*dim* vector. We keep the hyper-parameter setting of AM-Softmax as [18]: $s = 64$ and $m = 0.5$. The feature aggregation network E_{Feat} comprises of two linear residual units with P-ReLU and BatchNorm in between. E_{Feat} is trained on MS-Celeb-1M by SGD with a learning rate of 0.01. The triplet loss margin α is 1.0. The disentangled features of gender, race, and identity are concatenated as a 3×512-*dim* vector, which is inputted into E_{Feat}. The network is then trained to output a 512-*dim* representation for face recognition on biased datasets.

12.4.3.3 De-biasing Face Verification

Baseline: We compare DebFace-ID with a regular face representation model which has the same architecture as the shared feature encoder of DebFace. Referred to as BaseFace, this baseline model is also trained on MS-Celeb-1M, with the representation dimension of 512.

To show the efficacy of DebFace-ID on bias mitigation, we evaluate the verification performance of DebFace-ID and BaseFace on faces from each demographic cohort. There are 48 total cohorts given the combination of demographic attributes including 2 gender (male, female), 4 race[3] (Black, White, East Asian, Indian), and 6 age group ($0-12$, $13-18$, $19-34$, $35-44$, $45-54$, $55-100$). We combine CACD, AgeDB, CVIT, and a subset of Asian-DeepGlint as the testing set. Overlapped identities among these datasets are removed. IMDB is excluded from the testing set due to its massive number of wrong ID labels. For datasets without certain demographic labels, we simply use the corresponding models to predict the labels. We report the Area Under the Curve (AUC) of the Receiver Operating Characteristics (ROC). We define the degree of bias, termed *biasness*, as the standard deviation of performance across cohorts.

Figure 12.3 shows the face verification results of BaseFace and DebFace-ID on each cohort. That is, for a particular face representation (e.g., DebFace-ID), we report its AUC on each cohort by putting the number in the corresponding cell. From these heatmaps, we observe that both DebFace-ID and BaseFace present bias in face verification, where the performance on some cohorts is significantly worse, especially the cohorts of Indian female and elderly people. Compared to BaseFace, DebFace-ID suggests less bias and the difference of AUC is smaller, where the heatmap exhibits smoother edges. Figure 12.4 shows the performance of face verification on 12 demographic cohorts. Both DebFace-ID and BaseFace present similar relative accuracies across cohorts. For example, both algorithms

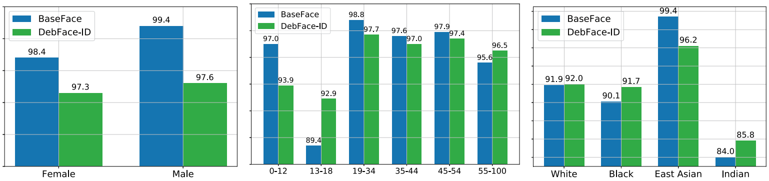


(a) BaseFace

(b) DebFace-ID

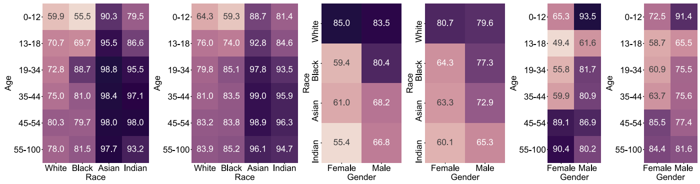
Fig. 12.3 Face Verification AUC (%) on each demographic cohort. The cohorts are chosen based on the three attributes, i.e., gender, age, and race. To fit the results into a 2D plot, we show the performance of male and female separately. Due to the limited number of face images in some cohorts, their results are gray cells

[3] To clarify, we consider two race groups, Black and White; and two ethnicity groups, East Asian and Indian. The word race denotes both race and ethnicity here.



(a) Gender (b) Age (c) Race

Fig. 12.4 The overall performance of face verification AUC (%) on gender, age, and race



(a) BaseGender (b) DebFace-G (c) BaseAge (d) DebFace-A (e) BaseRace (f) DebFace-R

Fig. 12.5 Classification accuracy (%) of demographic attribute estimations on faces of different cohorts, by DebFace and the baselines. For simplicity, we use DebFace-G, DebFace-A, and DebFace-R to represent the gender, age, and race classifier of DebFace

perform worse on the younger age cohorts than on adults; and the performance on the Indian is significantly lower than on the other races. DebFace-ID decreases the bias by gaining discriminative face features for cohorts with less images in spite of the reduction in the performance on cohorts with more samples.

12.4.3.4 De-biasing Demographic Attribute Estimation

Baseline: We further explore the bias of demographic attribute estimation and compare demographic attribute classifiers of DebFace with baseline estimation models. We train three demographic estimation models, namely, gender estimation (BaseGender), age estimation (BaseAge), and race estimation (BaseRace), on the same training set as DebFace. For fairness, all three models have the same architecture as the shared layers of DebFace.

We combine the four datasets mentioned in Sect. 12.4.3.3 with IMDB as the global testing set. As all demographic estimations are treated as classification problems, the classification accuracy is used as the performance metric. As shown in Fig. 12.5, all demographic attribute estimations present significant bias. For gender estimation, both algorithms perform worse on the White and Black cohorts than on East Asian and Indian. In addition, the performance on young children is significantly worse than on adults. In general, the race estimation models perform better on the male cohort than on female. Compared to gender, race estimation shows a higher bias in terms of age. Both baseline methods and DebFace perform worse on cohorts in age between 13 and 44 than in other age groups.

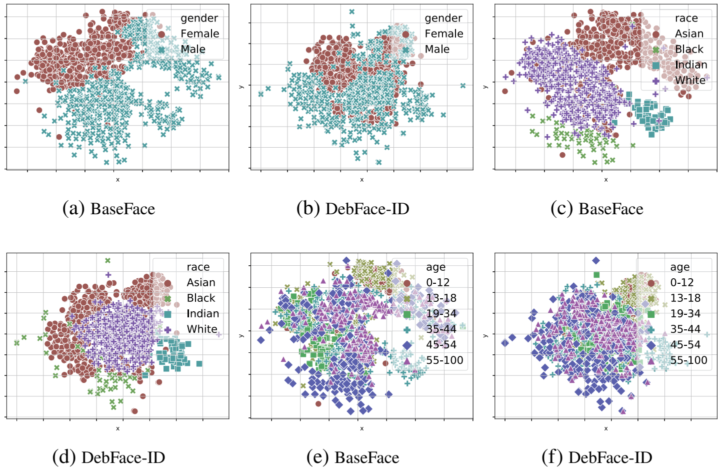
Table 12.2 Biasness of face recognition and demographic attribute estimation

Method	Face verification				Demographic estimation		
	All	Gender	Age	Race	Gender	Age	Race
Baseline	6.83	0.50	3.13	5.49	12.38	10.83	14.58
DebFace	**5.07**	**0.15**	**1.83**	**3.70**	**10.22**	**7.61**	**10.00**

Similar to race, age estimation still achieves better performance on male than on female. Moreover, the white cohort shows dominant advantages over other races in age estimation. In spite of the existing bias in demographic attribute estimations, the proposed DebFace is still able to mitigate bias derived from algorithms. Compared to Fig. 12.5a, c, e, cells in Fig. 12.5b, d, f present more uniform colors. We summarize the biasness of DebFace and baseline models for both face recognition and demographic attribute estimations in Table 12.2. In general, we observe DebFace substantially reduces biasness for both tasks. For the task with larger biasness, the reduction of biasness is larger.

12.4.3.5 Analysis of Disentanglement

We notice that DebFace still suffers from unequal performance in different demographic groups. It is because there are other latent variables besides the demographics, such as image quality or capture conditions that could lead to biased performance. Such variables



(a) BaseFace (b) DebFace-ID (c) BaseFace

(d) DebFace-ID (e) BaseFace (f) DebFace-ID

Fig. 12.6 The distribution of face identity representations of BaseFace and DebFace. Both collections of feature vectors are extracted from images of the same dataset. Different colors and shapes represent different demographic attributes. Zoom in for details

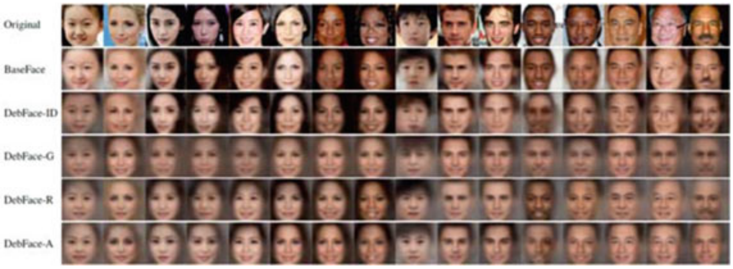


Fig. 12.7 Reconstructed Images using Face and Demographic Representations. The first row is the original face images. From the second row to the bottom, the face images are reconstructed from (2) BaseFace; (3) DebFace-ID; (4) DebFace-G; (5) DebFace-R; (6) DebFace-A. Zoom in for details

are difficult to control in pre-collected large face datasets. In the framework of DebFace, it is also related to the degree of feature disentanglement. fully disentangling is supposed to completely remove the factors of bias from demographic information. To illustrate the feature disentanglement of DebFace, we show the demographic discriminative ability of face representations by using these features to estimate gender, age, and race. Specifically, we first extract identity features of images from the testing set in Sect. 12.4.3.1 and split them into training and testing sets. Given demographic labels, the face features are fed into a two-layer fully-connected network, learning to classify one of the demographic attributes. Table 12.3 reports the demographic classification accuracy on the testing set. For all three demographic estimations, DebFace-ID presents much lower accuracies than BaseFace, indicating the decline of demographic information in DebFace-ID. We also plot the distribution of identity representations in the feature space of BaseFace and DebFace-ID. From the testing set in Sect. 12.4.3.3, we randomly select 50 subjects in each demographic group and one image of each subject. BaseFace and DebFace-ID are extracted from the selected image set and are then projected from 512-*dim* to 2-*dim* by t-SNE. Figure 12.6 shows their t-SNE feature distributions. We observe that BaseFace presents clear demographic clusters, while the demographic clusters of DebFace-ID, as a result of disentanglement, mostly overlap with each other.

To visualize the disentangled feature representations of DebFace, we train a decoder that reconstructs face images from the representations. Four face decoders are trained separately for each disentangled component, i.e., gender, age, race, and ID. In addition, we train another decoder to reconstruct faces from BaseFace for comparison. As shown in Fig. 12.7, both BaseFace and DebFace-ID maintain the identified features of the original faces, while DebFace-ID presents less demographic characteristics. No race or age, but gender features can be observed on faces reconstructed from DebFace-G. Meanwhile, we can still recognize race and age attributes on faces generated from DebFace-R and DebFace-A.

Table 12.3 Demographic classification accuracy (%) by face features

Method	Gender	Race	Age
BaseFace	95.27	89.82	78.14
DebFace-ID	73.36	61.79	49.91

Table 12.4 Face verification accuracy (%) on RFW dataset

Method	Gender	Race	Age
BaseFace	95.27	89.82	78.14
DebFace-ID	73.36	61.79	49.91

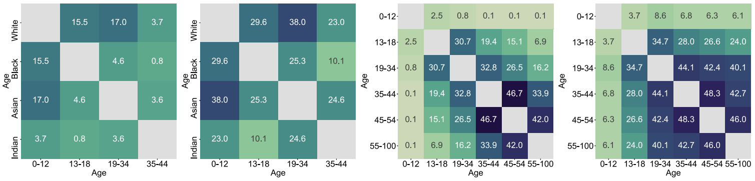
12.4.3.6 Face Verification on Public Testing Datasets

We report the performance of three different settings, using (1) BaseFace, the same baseline in Sect. 12.4.3.3, (2) DebFace-ID, and (3) the fused representation DemoID. Table 12.5 reports face verification results on three public benchmarks: LFW, IJB-A, and IJB-C. On LFW, DemoID outperforms BaseFace while maintaining similar accuracy compared to SOTA algorithms. On IJB-A/C, DemoID outperforms all prior works except PFE [77]. Although DebFace-ID shows lower discrimination, TAR at lower FAR on IJB-C is higher than that of BaseFace. To evaluate DebFace on a racially balanced testing dataset RFW [92] and compare with the work [91], we train a DebFace model on BUPT-Balancedface [91] dataset. The new model is trained to reduce racial bias by disentangling ID and race. Table 12.4 reports the verification results on RFW. While DebFace-ID gives a slightly lower face verification accuracy, it improves the biasness over [91].

We observe that DebFace-ID is less discriminative than BaseFace, or DemoID, since demographics are essential components of face features. To understand the deterioration of DebFace, we analyze the effect of demographic heterogeneity on face verification by showing the tendency for one demographic group to experience a false accept error rela-

Table 12.5 Verification Performance on LFW, IJB-A, and IJB-C

Method	LFW (%)	Method	IJB-A (%)	IJB-C @ FAR (%)		
			0.1% FAR	0.001%	0.01%	0.1%
DeepFace+ [82]	97.35	Yin et al. [101]	73.9 ± 4.2	–	–	69.3
CosFace [90]	99.73	Cao et al. [7]	90.4 ± 1.4	74.7	84.0	91.0
ArcFace [18]	99.83	Multicolumn [98]	92.0 ± 1.3	77.1	86.2	92.7
PFE [77]	99.82	PFE [77]	95.3 ± 0.9	89.6	93.3	95.5
BaseFace	99.38	*BaseFace*	90.2 ± 1.1	80.2	88.0	92.9
DebFace-ID	98.97	*DebFace-ID*	87.6 ± 0.9	82.0	88.1	89.5
DemoID	99.50	*DemoID*	92.2 ± 0.8	83.2	89.4	92.9



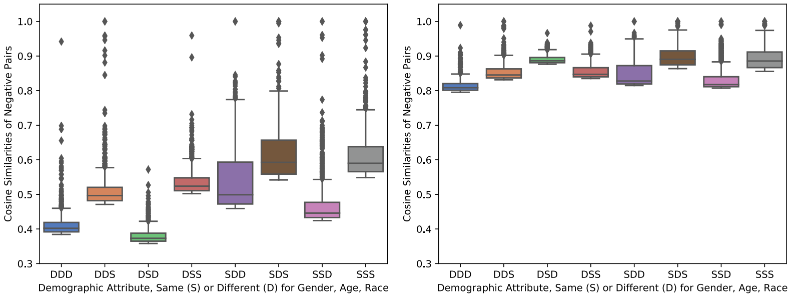
(a) BaseFace: Race (b) DebFace-ID: Race (c) BaseFace: Age (d) DebFace-ID: Age

Fig. 12.8 The percentage of false accepted cross race or age pairs at 1% FAR

tive to another group. For any two demographic cohorts, we check the number of falsely accepted pairs that are from different groups at 1% FAR. Figure 12.8 shows the percentage of such falsely accepted demographic-heterogeneous pairs. Compared to BaseFace, DebFace exhibits more cross-demographic pairs that are falsely accepted, resulting in the performance decline on demographically biased datasets. Due to the demographic information reduction, DebFace-ID is more susceptible to errors between demographic groups. In the sense of de-biasing, it is preferable to decouple demographic information from identity features. However, if we prefer to maintain the overall performance across all demographics, we can still aggregate all the relevant information. It is an application-dependent trade-off between accuracy and de-biasing. DebFace balances the accuracy vs. bias trade-off by generating both debiased identity and debiased demographic representations, which may be aggregated into DemoID if bias is less of a concern.

12.4.3.7 Distributions of Scores

We follow the work of [36] that investigates the effect of demographic homogeneity and heterogeneity on face recognition. We first randomly select images from CACD, AgeDB, CVIT, and Asian-DeepGlint datasets, and extract the corresponding feature vectors by using the models of BaseFace and DebFace, respectively. Given their demographic attributes, we put those images into separate groups depending on whether their gender, age, and race are the same or not. For each group, a fixed false alarm rate (the percentage of the face pairs from the same subjects being falsely verified as from different subjects) is set to 1%. Among the falsely verified pairs, we plot the top 10th percentile scores of the negative face pairs (a pair of face images that are from different subjects) given their demographic attributes. As shown in Fig. 12.9a, b, we observe that the similarities of DebFace are higher than those of BaseFace. One of the possible reasons is that the demographic information is disentangled from the identity features of DebFace, increasing the overall pair-wise similarities between faces of different identities. In terms of de-biasing, DebFace also reflects smaller differences in the score distribution with respect to the homogeneity and heterogeneity of demographics.



(a) BaseFace

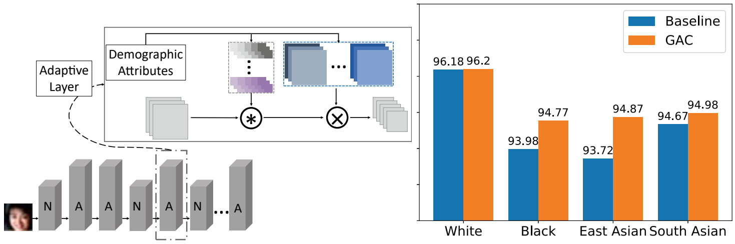
(b) DebFace

Fig. 12.9 BaseFace and DebFace distributions of the similarity scores of the imposter pairs across homogeneous versus heterogeneous gender, age, and race categories

12.5 Mitigating Face Recognition Bias via Group Adaptive Classifier

In spite of the effectiveness of DebFace in mitigating demographic bias, it degenerates the overall recognition performance as well. This motivates us to find anther solution to this problem such that the biasness can be reduced without impairing the average recognition performance. In this section, we introduce our second approach to mitigate face recognition bias via group adaptive classifier (GAC). The main idea of GAC is to optimize the face representation learning on every demographic group in a single network, despite demographically imbalanced training data. Conceptually, we may categorize face features into two types of patterns: *general pattern* is shared by all faces; *differential pattern* is relevant to demographic attributes. When the differential pattern of one specific demographic group dominates training data, the network learns to predict identities mainly based on that pattern as it is more convenient to minimize the loss than using other patterns, thus bringing bias toward faces of that specific group. One solution is to give the network more capacity to broaden its scope for multiple face patterns from different demographic groups. An unbiased FR model shall rely on not only unique patterns for recognition of different groups, but also general patterns of all faces for improved generalizability. Accordingly, in Fig. 12.10, we propose GAC to explicitly learn these different feature patterns. GAC includes two modules: the adaptive layer and the automation module. The adaptive layer in GAC comprises adaptive convolution kernels and channel-wise attention maps where each kernel and attention map tackle faces in *one* demographic group. We also introduce a new objective function to GAC, which diminishes the variation of average intra-class distance between demographic groups.

Prior works on dynamic CNNs introduce adaptive convolutions to either every layer [42, 94, 99] or manually specified layers [35, 61, 81]. In contrast, this work proposes an automation module to choose which layers to apply adaptations. As we observe, not all convolutional layers require adaptive kernels for bias mitigation (see Fig. 12.16a). At any layer of GAC,



(a) GAC (b) Bias (1.11 → 0.60)

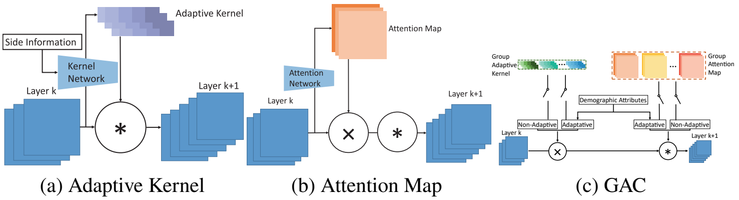
Fig. 12.10 **a** Our proposed group adaptive classifier (GAC) automatically chooses between non-adaptive ("N") and adaptive ("A") layer in a multi-layer network, where the latter uses demographic-group-specific kernel and attention. **b** Compared to the baseline with the 50-layer ArcFace backbone, GAC improves face verification accuracy in most groups of RFW dataset [92], especially under-represented groups, leading to mitigated FR bias. GAC reduces biasness from 1.11 to 0.60

only kernels expressing high dissimilarity are considered as demographic-adaptive kernels. For those with low dissimilarity, their average kernel is shared by all input images in that layer. Thus, the proposed network progressively learns to select the optimal structure for the demographic-adaptive learning. This enables that both non-adaptive layers with shared kernels and adaptive layers are jointly learned in a unified network.

Contributions of this work are summarized as: (1) A new face recognition algorithm that reduces demographic bias and increases the robustness of representations for faces in every demographic group by adopting adaptive convolutions and attention techniques; (2) A new adaptation mechanism that automatically determines the layers to employ dynamic kernels and attention maps; (3) The proposed method achieves SOTA performance on a demographic-balanced dataset and three benchmarks.

12.5.1 Adaptive Neural Networks

Since the main technique applied in GAC is adaptive neural network, we first review recent work related to adaptive learning. Three types of CNN-based adaptive learning techniques are related to our work: adaptive architectures, adaptive kernels, and attention mechanisms. Adaptive architectures design new performance-based neural functions or structures, e.g., neuron selection hidden layers [38], and automatic CNN expansion for FR [111]. As CNN advances many AI fields, prior works propose dynamic kernels to realize content-adaptive convolutions. Li et al. [50] propose a shape-driven kernel for facial trait recognition where each landmark-centered patch has a unique kernel. A convolution fusion strategy for graph neural networks is introduced by [24] where a set of varying-size filters are used per layer. The works of [22] and [51] use a kernel selection scheme to automatically adjust the receptive field size based on inputs. To better suit input data, [21] splits training data into clusters and



(a) Adaptive Kernel (b) Attention Map (c) GAC

Fig. 12.11 A comparison of approaches in adaptive CNNs

learns an exclusive kernel per cluster. Li et al. [52] introduce an adaptive CNN for object detection that transfers pre-trained CNNs to a target domain by selecting useful kernels per layer. Alternatively, one may feed input images or features into a kernel function to dynamically generate convolution kernels [40, 48, 80, 106]. Despite its effectiveness, such individual adaptation may not be suitable given the diversity of faces in demographic groups. Our work is most related to the side information adaptive convolution [42], where in each layer a sub-network inputs auxiliary information to generate filter weights. We mainly differ in that GAC automatically learns where to use adaptive kernels in a multi-layer CNN (see Fig. 12.11a, c), thus more efficient and capable of applying to a deeper CNN.

As the human perception process naturally selects the most pertinent piece of information, attention mechanisms are designed for a variety of tasks, e.g., detection [110], recognition [12], image captioning [11], tracking [9], pose estimation [81], and segmentation [61]. Typically, attention weights are estimated by feeding images or feature maps into a shared network, composed of convolutional and pooling layers [5, 12, 53, 78] or multi-layer perceptron (MLP) [37, 54, 72, 97]. Apart from feature-based attention, Hou et al. [35] propose a correlation-guided cross-attention map for few-shot classification where the correlation between the class feature and query feature generates the attention weights. The work of [99] introduces a cross-channel communication block to encourage information exchange across channels at the convolutional layer. To accelerate the channel interaction, Wang et al. [94] propose a 1D convolution across channels for attention prediction. Different from prior work, our attention maps are constructed by demographic information (see Fig. 12.11b, c), which improves the robustness of face representations in every demographic group.

12.5.2 Methodology

12.5.2.1 Overview

Our goal is to train a FR network that is impartial to individuals in different demographic groups. Unlike image-related variations, e.g., large-poses, or low-resolution faces which are harder to be recognized, demographic attributes are subject-related properties with no apparent impact over recognizability of identity, at least from a layman's perspective. Thus, an unbiased FR system should be able to obtain equally salient features for faces across

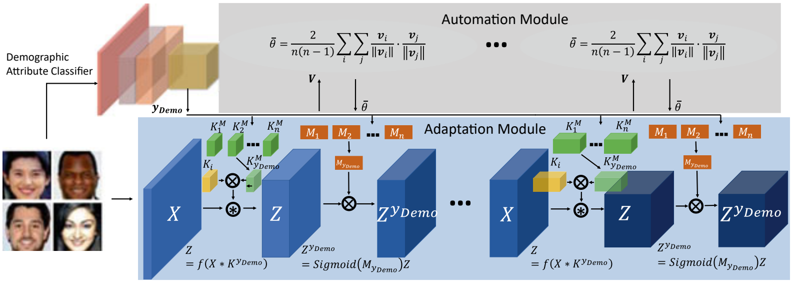

Automation Module
$\bar{\theta} = \frac{2}{n(n-1)}\sum_i\sum_j \frac{v_i}{\|v_i\|}\cdot\frac{v_j}{\|v_j\|}$
Demographic Attribute Classifier
y_{Demo}
V
$\bar{\theta}$
Adaptation Module
K_1^M K_2^M K_n^M
M_1 M_2 M_n
K_i
$K_{y_{Demo}}^M$
$M_{y_{Demo}}$
X
Z
$Z^{y_{Demo}}$
$Z = f(X * K^{y_{Demo}})$
$Z^{y_{Demo}} = Sigmoid(M_{y_{Demo}})Z$

Fig. 12.12 Overview of the proposed GAC for mitigating FR bias. GAC contains two major modules: the adaptive layer and the automation module. The adaptive layer consists of adaptive kernels and attention maps. The automation module is employed to decide whether a layer should be adaptive or not

demographic groups. However, due to imbalanced demographic distributions and inherent face differences between groups, it was shown that certain groups achieve higher performance even with hand-crafted features [46]. Thus, it is impractical to extract features from different demographic groups that exhibit equal discriminability. Despite such disparity, a FR algorithm can still be designed to *mitigate* the difference in performance.

To this end, we propose a CNN-based group adaptive classifier that utilizes dynamic kernels and attention maps to boost FR performance in all demographic groups considered here. Specifically, GAC has two main modules, an adaptive layer and an automation module. In an adaptive layer, face images or feature maps are convolved with a unique kernel for each demographic group, and multiplied with adaptive attention maps to obtain demographic-differential features for faces in a certain group. The automation module determines the layers of the network that adaptive kernels and attention maps should be applied to. As shown in Fig. 12.12, given an aligned face, and its identity label y_{ID}, a pre-trained demographic classifier first estimates its demographic attribute y_{Demo}. With y_{Demo}, the image is then fed into a recognition network with multiple demographic-adaptive layers to estimate its identity. In the following, we present these two modules.

12.5.2.2 Adaptive Layer

Adaptive Convolution. For a standard convolution in CNN, an image or feature map from the previous layer $X \in \mathbb{R}^{c\times h^X \times w^X}$ is convolved with a single kernel matrix $K \in \mathbb{R}^{k\times c\times h^K \times w^K}$, where c is the number of input channels, k the number of filters, h^X and w^X the input size, and h^K and w^K the filter size. Such an operation shares the kernel with every input going through the layer, and is thus agnostic to demographic content, resulting in limited capacity to represent minority groups. To mitigate the bias in convolution, we introduce a trainable matrix of kernel masks $K^M \in \mathbb{R}^{n\times c\times h^K \times w^K}$, where n is the number of demographic groups. In the forward pass, the demographic label y_{Demo} and kernel matrix K^M are fed into the adaptive convolutional layer to generate demographic-adaptive filters. Let $K_i \in \mathbb{R}^{c \times h^K \times w^K}$ denote the ith channel of the shared filter. The ith channel of adaptive filter for group y_{Demo} is:

$$K_i^{y_{Demo}} = K_i \bigotimes K_{y_{Demo}}^M, \tag{12.5}$$

where $K_{y_{Demo}}^M \in \mathbb{R}^{c \times h^K \times w^K}$ is the ${y_{Demo}}^{th}$ kernel mask for group y_{Demo}, and $\bigotimes$ denotes element-wise multiplication. Then the ith channel of the output feature map is given by $Z_i = f(X * K_i^{y_{Demo}})$, where $*$ denotes convolution, and $f(\cdot)$ is activation. Unlike conventional convolution, samples in every demographic group have a unique kernel $K^{y_{Demo}}$.

Adaptive Attention. Each channel filter in a CNN plays an important role in every dimension of the final representation, which can be viewed as a semantic pattern detector [11]. In the adaptive convolution, however, the values of a kernel mask are broadcast along the channel dimension, indicating that the weight selection is spatially varied but channel-wise joint. Hence, we introduce a channel-wise attention mechanism to enhance the face features that are demographic-adaptive. First, a trainable matrix of channel attention maps $M \in \mathbb{R}^{n \times k}$ is initialized in every adaptive attention layer. Given y_{Demo} and the current feature map $Z \in \mathbb{R}^{k \times h^Z \times w^Z}$, where h^Z and w^Z are the height and width of Z, the ith channel of the new feature map is calculated by

$$Z_i^{y_{Demo}} = \text{Sigmoid}(M_{y_{Demo}i}) \cdot Z_i, \tag{12.6}$$

where $M_{y_{Demo}i}$ is the entry in the ${y_{Demo}}^{th}$ row of M for the demographic group y_{Demo} at ith column. In contrast to the adaptive convolution, elements of each demographic attention map $M_{y_{Demo}}$ diverge in a channel-wise manner, while the single attention weight $M_{y_{Demo}i}$ is spatially shared by the entire matrix $Z_i \in \mathbb{R}^{h^Z \times w^Z}$. The two adaptive matrices, K^M and M, are jointly tuned with all the other parameters supervised by the classification loss.

Unlike dynamic CNNs [42] where additional networks are engaged to produce input-variant kernel or attention map, our adaptiveness is yielded by a simple thresholding function directly pointing to the demographic group with no auxiliary networks. Although the kernel network in [42] can generate continuous kernels without enlarging the parameter space, further encoding is required if the side inputs for kernel network are discrete variables. Our approach, in contrast, divides kernels into clusters so that the branch parameter learning can stick to a specific group without interference from individual uncertainties, making it suitable for discrete domain adaptation. Furthermore, the adaptive kernel masks in GAC are more efficient in terms of the number of additional parameters. Compared to a non-adaptive layer, the number of additional parameters of GAC is $n \times c \times h^K \times w^K$, while that of [42] is $s \times k \times c \times h^K \times w^K$ if the kernel network is a one-layer MLP, where s is the dimension of input side information. Thus, for one adaptive layer, [42] has $\frac{s \times k}{n}$ times more parameters than ours, which can be substantial given the typical large value of k, the number of filters.

12.5.2.3 Automation Module

Though faces in different demographic groups are adaptively processed by various kernels and attention maps, it is inefficient to use such adaptations in *every* layer of a deep CNN. To relieve the burden of unnecessary parameters and avoid empirical trimming, we adopt a similarity fusion process to automatically determine the adaptive layers. Since the same fusion scheme can be applied to both types of adaptation, we take the adaptive convolution as an example to illustrate this automatic scheme.

First, a matrix composed of n kernel masks is initialized in every convolutional layer. As training continues, each kernel mask is updated independently to reduce classification loss for each demographic group. Second, we reshape the kernel masks into 1D vectors $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$, where $\mathbf{v}_i \in \mathbb{R}^l$, $l = c \times w^K \times h^K$ is the kernel mask of the ith demographic group. Next, we compute the Cosine similarity between two kernel vectors, $\theta_{ij} = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \cdot \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|}$, where $1 \leq i, j \leq n$. The average similarity of all pair-wise similarities is obtained by $\overline{\theta} = \frac{2}{n(n-1)} \sum_i \sum_j \theta_{ij}, i \neq j$. If the dissimilarity $-\overline{\theta}$ is lower than a pre-defined threshold τ, the kernel parameters in this layer reveal the demographic-agnostic property. Hence, we merge the n kernels into a single kernel by averaging along the group dimension. By definition, a lower τ implies more adaptive layers. Given an array of $\{-\theta_i\}^t$ (t is the total number of convolutional layers), we first sort the elements from the smallest to the highest, and this way, layers whose $-\theta_i$ values are larger than τ will be adaptive. Thus, when τ decreases, more layers will be adaptive. In the subsequent training, this single kernel can still be updated separately for each demographic group, as the kernel may become demographic-adaptive in later epochs. We monitor the similarity trend of the adaptive kernels in each layer until $\overline{\theta}$ is stable.

12.5.2.4 De-biasing Objective Function

Apart from the objective function for face identity classification, we also adopt a regress loss function to narrow the gap of the intra-class distance between demographic groups. Let $g(\cdot)$ denote the inference function of GAC, and I_{ijg} is the ith image of subject j in group g. Then, the feature representation of image I_{ijg} is given by $\mathbf{r}_{ijg} = g(I_{ijg}, \mathbf{w})$, where $\mathbf{w}$ denotes the GAC parameters. Assuming the feature distribution of each subject is a Gaussian distribution with an identity covariance matrix (hyper-sphere), we utilize the average Euclidean distance to every subject center as the intra-class distance of each subject. In particular, we first compute the center point of each identity-sphere:

$$\boldsymbol{\mu}_{jg} = \frac{1}{N} \sum_{i=1}^{N} g(I_{ijg}, \mathbf{w}), \tag{12.7}$$

where N is the total number of face images of subject j. The average intra-class distance of subject j is as follows:

$$Dist_{jg} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{r}_{ijg} - \boldsymbol{\mu}_{jg})^T(\mathbf{r}_{ijg} - \boldsymbol{\mu}_{jg}). \tag{12.8}$$

We then compute the intra-class distance for all subjects in group g as $Dist_g = \frac{1}{Q}\sum_{j=1}^{Q} Dist_{jg}$, where Q is the number of total subjects in group g. This allows us to lower the difference of intra-class distance by

$$\mathcal{L}_{bias} = \frac{\lambda}{Q \times n}\sum_{g=1}^{n}\sum_{j=1}^{Q}\left| Dist_{jg} - \frac{1}{n}\sum_{g=1}^{n} Dist_g \right|, \tag{12.9}$$

where λ is the coefficient for the de-biasing objective.

12.5.3 Experiments

Datasets Our bias study uses RFW dataset [92] for testing and BUPT-Balancedface dataset [91] for training. RFW consists of faces in four race/ethnic groups: White, Black, East Asian, and South Asian.[4] Each group contains ∼10K images of 3K individuals for face verification. BUPT-Balancedface contains 1.3M images of 28K celebrities and is approximately race-balanced with 7K identities per race. Other than race, we also study gender bias. We combine IMDB [71], UTKFace [112], AgeDB [64], AAF [13], AFAD [67] to train a gender classifier, which estimates gender of faces in RFW and BUPT-Balancedface. The statistics of the datasets are reported in Table 12.1. All face images are cropped and resized to 112 × 112 pixels via landmarks detected by RetinaFace [19].

Implementation Details We train a baseline network and GAC on BUPT-Balancedface, using the 50-layer ArcFace architecture [18]. The classification loss is an additive Cosine margin in Cosface [90], with the scale and margin respectively as $s = 64$ and $m = 0.5$. Training is optimized by SGD with a batch size 256. The learning rate starts from 0.1 and drops to 0.0001 following the schedule at 8, 13, 15 epochs for the baseline, and 5, 17, 19 epochs for GAC. We set $\lambda = 0.1$ for the intra-distance de-biasing. $\tau = -0.2$ is chosen for automatic adaptation in GAC. Our FR models are trained to extract a 512-dim representation. Our demographic classifier uses a 18-layer ResNet [33]. Comparing GAC and the baseline, the average feature extraction speed per image on Nvidia 1080Ti GPU is 1.4ms and 1.1ms, and the number of model parameters is 44.0M and 43.6M, respectively.

Performance Metrics The common group fairness criteria like demographic parity distance [58] are improper to evaluate the fairness of learnt representations, since they are designed to measure the independence properties of random variables. However, in FR the sensitive demographic characteristics are tied to identities, making these two variables cor-

[4] RFW [92] uses Caucasian, African, Asian, and Indian to name demographic groups. We adopt these groups and accordingly rename to White, Black, East Asian, and South Asian for clearer race/ethnicity definition.

related. The NIST report uses false negative and false positive for each demographic group to measure the fairness [27]. Instead of plotting false negative vs. false positives, we adopt a compact quantitative metric, i.e., the standard deviation (STD) of the performance in different demographic groups, previously introduced in [25, 91] and called "biasness." As bias is considered as systematic error of the estimated values compared to the actual values, here, we assume the average performance to be the actual value. For each demographic group, its biasness is the error between the average and the performance on the demographic group. The overall biasness is the expectation of all group errors, which is the STD of performance across groups. We also report average accuracy (Avg) to show the overall FR performance.

12.5.3.1 Results on RFW Protocol

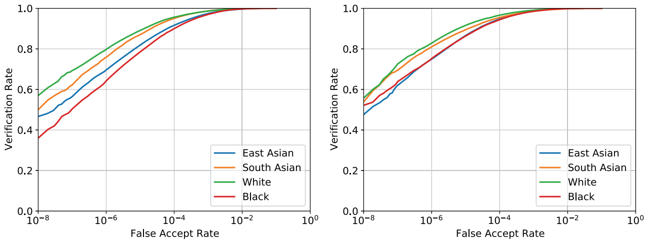
We follow RFW face verification protocol with 6K pairs per race/ethnicity. The models are trained on BUPT-Balancedface with ground-truth race and identity labels.

Compare with SOTA. We compare the GAC with four SOTA algorithms on RFW protocol, namely, ACNN [42], RL-RBN [91], PFE [77], and DebFace [25]. Since the approach in ACNN [42] is related to GAC, we re-implement it and apply to the bias mitigation problem. First, we train a race classifier with the cross-entropy loss on BUPT-Balancedface. Then the softmax output of our race classifier is fed to a filter manifold network (FMN) to generate adaptive filter weights. Here, FMN is a two-layer MLP with a ReLU in between. Similar to GAC, race probabilities are considered as auxiliary information for face representation learning. We also compare with the SOTA approach PFE [77] by training it on BUPT-Balancedface. As shown in Table 12.6, GAC is superior to SOTA w.r.t. average performance and feature fairness. Compared to kernel masks in GAC, the FMN in ACNN [42] contains more trainable parameters. Applying it to each convolutional layer is prone to overfitting. In fact, the layers that are adaptive in GAC ($\tau = -0.2$) are set to be the FMN-based convolution in ACNN. As the race data is a four-element input in our case, using extra kernel networks adds complexity to the FR network, which degrades the verification performance. Even though PFE performs the best on standard benchmarks (Table 12.15), it still exhibits high biasness. Our GAC outperforms PFE on RFW in both biasness and average performance. Compared to DebFace [25], in which demographic attributes are disentangled from the identity representations, GAC achieves higher verification performance by optimizing the classification for each demographic group, with a lower biasness as well.

To further present the superiority of GAC over the baseline model in terms of bias, we plot Receiver Operating Characteristic (ROC) curves to show the values of True Acceptance Rate (TAR) at various values of False Acceptance Rate (FAR). Figure 12.13 shows the ROC performance of GAC and the baseline model on RFW. We see that the curves of demographic groups generated by GAC present smaller gaps in TAR at every FAR, which demonstrates the de-biasing capability of GAC. Figure 12.14 shows pairs of false positives (two faces falsely verified as the same identity) and false negatives in RFW dataset.

Table 12.6 Performance comparison with SOTA on the RFW protocol [92]. The results marked by (*) are directly copied from [91]

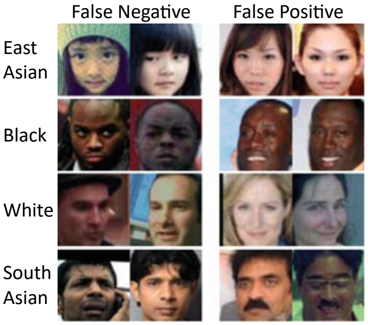
Method	White	Black	East Asian	South Asian	Avg (↑)	STD (↓)
RL-RBN [91]	96.27	95.00	94.82	94.68	95.19	0.63
ACNN [42]	96.12	94.00	93.67	94.55	94.58	0.94
PFE [77]	**96.38**	**95.17**	94.27	94.60	95.11	0.93
ArcFace [18]	96.18*	94.67*	93.72*	93.98*	94.64	0.96
CosFace [90]	95.12*	93.93*	92.98*	92.93*	93.74	0.89
DebFace [25]	95.95	93.67	94.33	94.78	94.68	0.83
GAC	96.20	94.77	**94.87**	**94.98**	**95.21**	**0.58**



(a) Baseline (b) GAC

Fig. 12.13 ROC of **a** baseline and **b** GAC evaluated on all pairs of RFW

Fig. 12.14 8 false positive and false negative pairs on RFW given by the baseline but successfully verified by GAC



Ablation on Adaptive Strategies. To investigate the efficacy of our network design, we conduct three ablation studies: adaptive mechanisms, number of convolutional layers, and demographic information. For adaptive mechanisms, since deep feature maps contain both spatial and channel-wise information, we study the relationship among adaptive kernels, spatial and channel-wise attentions, and their impact on bias mitigation. We also study the impact of τ in our automation module. Apart from the baseline and GAC, we ablate

Table 12.7 Ablation of adaptive strategies on the RFW protocol [92]

Method	White	Black	East Asian	South Asian	Avg (↑)	STD (↓)
Baseline	96.18	93.98	93.72	94.67	94.64	1.11
GAC-Channel	95.95	93.67	94.33	94.78	94.68	0.83
GAC-Kernel	96.23	94.40	94.27	94.80	94.93	0.78
GAC-Spatial	95.97	93.20	93.67	93.93	94.19	1.06
GAC-CS	96.22	93.95	94.32	**95.12**	94.65	0.87
GAC-CSK	96.18	93.58	94.28	94.83	94.72	0.95
GAC-($\tau = 0$)	96.18	93.97	93.88	94.77	94.70	0.92
GAC-($\tau = -0.1$)	**96.25**	94.25	94.83	94.72	95.01	0.75
GAC-($\tau = -0.2$)	96.20	**94.77**	**94.87**	94.98	**95.21**	**0.58**

eight variants: (1) GAC-Channel: channel-wise attention for race-differential feature; (2) GAC-Kernel: adaptive convolution with race-specific kernels; (3) GAC-Spatial: only spatial attention is added to baseline; (4) GAC-CS: both channel-wise and spatial attention; (5) GAC-CSK: combine adaptive convolution with spatial and channel-wise attention; (6,7,8) GAC-($\tau = *$): set τ to $*$.

From Table 12.7, we make several observations: (1) the baseline model is the most biased across race groups. (2) spatial attention mitigates the race bias at the cost of verification accuracy and is less effective on learning fair features than other adaptive techniques. This is probably because spatial contents, especially local layout information, only reside at earlier CNN layers, where the spatial dimensions are gradually decreased by the latter convolutions and poolings. Thus, semantic details like demographic attributes are hardly encoded spatially. (3) Compared to GAC, combining adaptive kernels with both spatial and channel-wise attention increases the number of parameters, lowering the performance. (4) As τ determines the number of adaptive layers in GAC, it has a great impact on the performance. A small τ may increase redundant adaptive layers, while the adaptation layers may lack in capacity if too large.

Ablation on Depths and Demographic Labels. Both the adaptive layers and de-biasing loss in GAC can be applied to CNN in any depth. In this ablation, we train both the baseline and GAC ($\lambda = 0.1$, $\tau = -0.2$) in ArcFace architecture with three different numbers of layers: 34, 50, and 100. As the training of GAC relies on demographic information, the error and bias in demographic labels might impact the bias reduction of GAC. Thus, we also ablate with different demographic information, (1) ground-truth: the race/ethnicity labels provided by RFW; (2) estimated: the labels predicted by a pre-trained race estimation model; (3) random: the demographic label randomly assigned to each face.

Table 12.8 Ablation of CNN depths and demographics on RFW protocol [92]

Method	White	Black	East Asian	South Asian	Avg (↑)	STD (↓)
Number of layers						
ArcFace-34	96.13	93.15	92.85	93.03	93.78	1.36
GAC-ArcFace-34	96.02	94.12	94.10	94.22	94.62	0.81
ArcFace-50	96.18	93.98	93.72	94.67	94.64	1.11
GAC-ArcFace-50	96.20	94.77	94.87	94.98	95.21	0.58
ArcFace-100	96.23	93.83	94.27	94.80	94.78	0.91
GAC-ArcFace-100	96.43	94.53	94.90	95.03	95.22	0.72
Race/Ethnicity labels						
Ground-truth	96.20	94.77	94.87	94.98	95.21	0.58
Estimated	96.27	94.40	94.32	94.77	94.94	0.79
Random	95.95	93.10	94.18	94.82	94.50	1.03

As shown in Table 12.8, compared to the baselines, GAC successfully reduces the STD at different number of layers. We see that the model with least number of layers presents the most bias, and the bias reduction by GAC is the most as well. The noise and bias in demographic labels do, however, impair the performance of GAC. With estimated demographics, the biasness is higher than that of the model with ground-truth supervision. Meanwhile, the model trained with random demographics has the highest biasness. Even so, using estimated attributes during testing still improves fairness in face recognition compared to baseline. This indicates the efficacy of GAC even in the absence of ground-truth labels.

Ablation on λ. We use λ to control the weight of de-biasing loss. Table 12.9 reports the results of GAC trained with different values of λ. When $\lambda = 0$, de-biasing loss is removed in training. The results indicate a larger λ leads to lower biasness at the cost of overall accuracy.

Ablation on Automation Module

Here, we also ablate GAC with two variants to show the efficiency of its automation module: (i) *Ada-All*, i.e., all the convolutional layers are adaptive and (ii) *Ada-8*, i.e., the same 8 layers as GAC are set to be adaptive starting from the beginning of the training process, with no automation module (our best GAC model has 8 adaptive layers). As in Table 12.11, with automation module, GAC achieves higher average accuracy and lower biasness than the other two models.

12.5.3.2 Results on Gender and Race Groups

We now extend demographic attributes to both gender and race. First, we train two classifiers that predict gender and race/ethnicity of a face image. The classification accuracy of gender

Table 12.9 Ablations on λ on RFW protocol (%)

λ	White	Black	East Asian	South Asian	Avg (↑)	STD (↓)
0	96.23	94.65	94.93	95.12	95.23	0.60
0.1	96.20	94.77	94.87	94.98	95.21	0.58
0.5	94.89	94.00	93.67	94.55	94.28	0.47

Table 12.10 Verification Accuracy (%) of 5-fold cross-validation on 8 groups of RFW [92]

Method	Gender	White	Black	East Asian	South Asian	Avg (↑)	STD (↓)
Baseline	Male	97.49 ± 0.08	96.94 ± 0.26	97.29 ± 0.09	97.03 ± 0.13	96.96 ± 0.03	0.69 ± 0.04
	Female	97.19 ± 0.10	97.93 ± 0.11	95.71 ± 0.11	96.01 ± 0.08		
AL+Manual	Male	98.57 ± 0.10	98.05 ± 0.17	98.50 ± 0.12	**98.36 ± 0.02**	98.09 ± 0.05	0.66 ± 0.07
	Female	98.12 ± 0.18	**98.97 ± 0.13**	96.83 ± 0.19	97.33 ± 0.13		
GAC	Male	**98.75 ± 0.04**	**98.18 ± 0.20**	**98.55 ± 0.07**	98.31 ± 0.12	**98.19 ± 0.06**	**0.56 ± 0.05**
	Female	**98.26 ± 0.16**	98.80 ± 0.15	**97.09 ± 0.12**	**97.56 ± 0.10**		

Table 12.11 Ablations on the automation module on RFW protocol (%)

Method	White	Black	East Asian	South Asian	Avg (↑)	STD (↓)
Ada-All	93.22	90.95	91.32	92.12	91.90	0.87
Ada-8	96.25	94.40	94.35	95.12	95.03	0.77
GAC	96.20	94.77	94.87	94.98	95.21	0.58

and race/ethnicity is 85% and 81%,[5] respectively. Then, these fixed classifiers are affiliated with GAC to provide demographic information for learning adaptive kernels and attention maps. We merge BUPT-Balancedface and RFW, and split the subjects into 5 sets for each of 8 demographic groups. In 5-fold cross-validation, each time a model is trained on 4 sets and tested on the remaining set. Table 12.12 reports the statistics of each data fold for the cross-validation experiment on BUPT-Balancedface and RFW datasets.

Here we demonstrate the efficacy of the automation module for GAC. We compare it to the scheme of manually designing (AL+Manual) that adds adaptive kernels and attention maps to a subset of layers. Specifically, the first block in every residual unit is chosen to be the adaptive convolution layer, and channel-wise attentions are applied to the feature map outputted by the last block in each residual unit. As we use 4 residual units and each block has 2 convolutional layers, the manual scheme involves 8 adaptive convolutional layers and

[5] This seemingly low accuracy is mainly due to the large dataset we assembled for training and testing gender/race classifiers. Our demographic classifier has been shown to perform comparably as SOTA on common benchmarks. While demographic estimation errors impact the training, testing, and evaluation of bias mitigation algorithms, the evaluation is of the most concern as demographic label errors may greatly impact the biasness calculation. Thus, future development may include either manually cleaning the labels, or designing a biasness metric robust to label errors.

Table 12.12 Statistics of dataset folds in the cross-validation experiment

Fold	White (#)		Black (#)		East Asian (#)		South Asian (#)	
	Subjects	Images	Subjects	Images	Subjects	Images	Subjects	Images
1	1, 991	68, 159	1, 999	67, 880	1, 898	67, 104	1, 996	57, 628
2	1, 991	67, 499	1, 999	65, 736	1, 898	66, 258	1, 996	57, 159
3	1, 991	66, 091	1, 999	65, 670	1, 898	67, 696	1, 996	56, 247
4	1, 991	66, 333	1, 999	67, 757	1, 898	65, 341	1, 996	57, 665
5	1, 994	68, 597	1, 999	67, 747	1, 898	68, 763	2, 000	56, 703

Table 12.13 Verification (%) on gender groups of IJB-C (TAR @ 0.1% FAR)

Model	Male	Female	Avg (↑)	STD (↓)
Baseline	89.72	79.57	84.64	5.08
GAC	88.25	83.74	86.00	2.26

4 groups of channel-wise attention maps. As in Table 12.10, automatic adaptation is more effective in enhancing the discriminability and fairness of face representations. Figure 12.16a shows the dissimilarity of kernel masks in the convolutional layers changes during training epochs under three thresholds τ. A lower τ results in more adaptive layers. We see the layers that are determined to be adaptive do vary across both layers (vertically) and training time (horizontally), which shows the importance of our automatic mechanism.

Since IJB-C also provides gender labels, we evaluate our GAC-gender model (see Sect. 4.2 of the main paper) on IJB-C as well. Specifically, we compute the verification TAR at 0.1% FAR on the pairs of female faces and male faces, respectively. Table 12.13 reports the TAR @ 0.1% FAR on gender groups of IJB-C. The biasness of GAC is still lower than the baseline for different gender groups of IJB-C.

12.5.3.3 Analysis on Intrinsic Bias and Data Bias

For all the algorithms listed in Table 12.1 of the main paper, the performance is higher in White group than those in the other three groups, even though all the models are trained on a demographic-balanced dataset, BUPT-Balancedface [91]. In this section, we further investigate the intrinsic bias of face recognition between demographic groups and the impact of the data bias in the training set. *Are non-White faces inherently difficult to be recognized for existing algorithms? Or, are face images in BUPT-Balancedface (the training set) and RFW* [92] *(testing set) biased toward the White group?*

To this end, we train our GAC network using training sets with different race/ethnicity distributions and evaluate them on RFW. In total, we conduct four experiments, in which we gradually reduce the total number of subjects in the White group from the BUPT-

Table 12.14 Verification accuracy (%) on the RFW protocol [92] with varying race/ethnicity distribution in the training set

Training Ratio	White	Black	East Asian	South Asian	Avg (↑)	STD (↓)
7 : 7 : 7 : 7	96.20	94.77	94.87	94.98	95.21	0.58
5 : 7 : 7 : 7	96.53	94.67	94.55	95.40	95.29	0.79
3.5 : 7 : 7 : 7	96.48	94.52	94.45	95.32	95.19	0.82
1 : 7 : 7 : 7	95.45	94.28	94.47	95.13	94.83	0.48
0 : 7 : 7 : 7	92.63	92.27	92.32	93.37	92.65	0.44

Balancedface dataset. To construct a new training set, subjects from the non-White groups in BUPT-Balancedface remain the same, while a subset of subjects is randomly picked from the White group. As a result, the ratios between non-White groups are consistently the same, and the ratios of White, Black, East Asian, South Asian are $\{5:7:7:7\}$, $\{3.5:7:7:7\}$, $\{1:7:7:7\}$, $\{0:7:7:7\}$ in the four experiments, respectively. In the last setting, we completely remove White from the training set.

Table 12.14 reports the face verification accuracy of models trained with different race/ethnicity distributions on RFW. For comparison, we also put our results on the balanced dataset here (with ratio $\{7:7:7:7\}$), where all images in BUPT-Balancedface are used for training. From the given results, we see several observations: (1) It shows that the White group still outperforms the non-White groups for the first three experiments. Even without any White subjects in the training set, the accuracy on the White testing set is still higher than those on the testing images in Black and East Asian groups. This suggests that White faces are either intrinsically easier to be verified or face images in the White group of RFW are less challenging. (2) With the decline in the total number of White subjects, the average performance declines as well. In fact, for all these groups, the performance suffers from the decrease in the number of White faces. This indicates that face images in the White groups are helpful to boost the face recognition performance for both White and non-White faces. In other words, faces from the White group benefit the representation learning of global patterns for face recognition in general. (3) Opposite to our intuition, the biasness is lower with less number of White faces, while the data bias is actually increased by adding the imbalance to the training set.

12.5.3.4 Results on Standard Benchmark Datasets

While our GAC mitigates bias, we also hope it can perform well on standard benchmarks. Therefore, we evaluate GAC on standard benchmarks without considering demographic impacts, including LFW [39], IJB-A [47], and IJB-C [63]. These datasets exhibit an imbalanced distribution in demographics. For a fair comparison with SOTA, instead of using ground-truth demographics, we train GAC on Ms-Celeb-1M [29] with the demographic

Table 12.15 Verification performance on LFW, IJB-A, and IJB-C. [Key: **Best**, *Second*, Third Best]

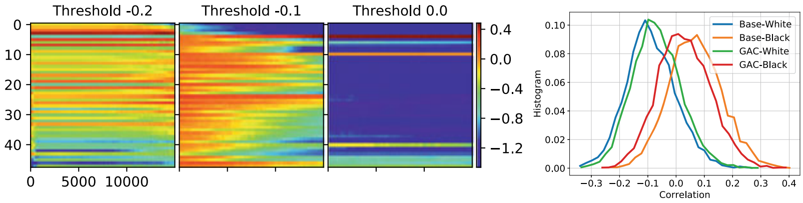
Method	LFW (%)	Method	IJB-A (%)	IJB-C @ FAR (%)		
			0.1% FAR	0.001%	0.01%	0.1%
DeepFace+ [82]	97.35	Yin et al. [101]	73.9 ± 4.2	–	–	69.3
CosFace [90]	99.73	Cao et al. [7]	90.4 ± 1.4	74.7	84.0	91.0
ArcFace [18]	**99.83**	Multicolumn [98]	*92.0 ± 1.3*	77.1	86.2	92.7
PFE [77]	*99.82*	PFE [77]	**95.3 ± 0.9**	**89.6**	**93.3**	**95.5**
Baseline	99.75	Baseline	90.2 ± 1.1	80.2	88.0	92.9
GAC	99.78	GAC	91.3 ± 1.2	*83.5*	*89.2*	*93.7*

attributes estimated by the classifier pre-trained in Sect. 12.5.3.2. As in Table 12.15, GAC outperforms the baseline and performs comparable to SOTA.

12.5.3.5 Visualization and Analysis on Bias of FR

Visualization To understand the adaptive kernels in GAC, we visualize the feature maps at an adaptive layer for faces of various demographics, via a Pytorch visualization tool [68]. We visualize important face regions pertaining to the FR decision by using a gradient-weighted class activation mapping (Grad-CAM) [74]. Grad-CAM uses the gradients back from the final layer corresponding to an input identity, and guides the target feature map to highlight import regions for identity predicting. Figure 12.15 shows that, compared to the baseline, the salient regions of GAC demonstrate more diversity on faces from different groups. This illustrates the variability of network parameters in GAC across different groups.

Bias via Local Geometry In addition to STD, we explain the bias phenomenon via the local geometry of a given face representation in each demographic group. We assume that the statistics of neighbors of a given point (representation) reflects certain properties of its manifold (local geometry). Thus, we illustrate the pair-wise correlation of face representa-



(a) Average similarities under different τ

(b) Pair-wise correlation

Fig. 12.15 The first row shows the average faces of different groups in RFW. The next two rows show gradient-weighted class activation heatmaps [74] at the 43th convolutional layer of the GAC and baseline. The higher diversity of heatmaps in GAC shows the variability of parameters in GAC across groups

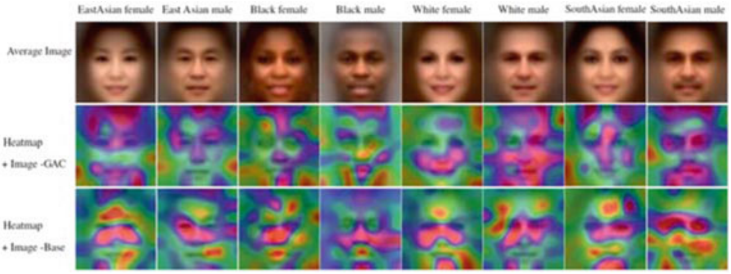


Fig. 12.16 **a** For each of the three τ in automatic adaptation, we show the average similarities of pair-wise demographic kernel masks, i.e., $\overline{\theta}$, at 1-48 layers (y-axis), and 1-15K training steps (x-axis). The number of adaptive layers in three cases, i.e., $\sum_1^{48}(\overline{\theta} > \tau)$ at 15K^{th} step, are 12, 8, and 2, respectively. **b** With two race groups (White, Black in PCSO [46]) and two models (baseline, GAC), for each of the four combinations, we compute pair-wise correlation of face representations using any two of 1K subjects in the same race, and plot the histogram of correlations. GAC reduces the difference/bias of two distributions

Table 12.16 Distribution of ratios between minimum inter-class distance and maximum intra-class distance of face features in 4 race groups of RFW. GAC exhibits higher ratios, and more similar distributions to the reference

Race	Mean		StaD		Relative entropy	
	Baseline	GAC	Baseline	GAC	Baseline	GAC
White	1.15	1.17	0.30	0.31	0.0	0.0
Black	1.07	1.10	0.27	0.28	0.61	0.43
East Asian	1.08	1.10	0.31	0.32	0.65	0.58
South Asian	1.15	1.18	0.31	0.32	0.19	0.13

tions. To minimize variations caused by other variables, we use constrained frontal faces of a mug shot dataset, PCSO [46]. We randomly select 1K White and 1K Black subjects from PCSO, and compute their pair-wise correlation within each race. In Fig. 12.16b, Base-White representations show lower inter-class correlation than Base-Black, i.e., faces in the White group are over-represented by the baseline than the Black group. In contrast, GAC-White and GAC-Black show more similarity in their correlation histograms.

As PCSO has few Asian subjects, we use RFW for another examination of the local geometry in 4 groups. That is, after normalizing the representations, we compute the pair-wise Euclidean distance and measure the ratio between the minimum distance of inter-subjects pairs and the maximum distance of intra-subject pairs. We compute the mean and standard deviation (StaD) of ratio distributions in 4 groups, by two models. Also, we gauge the relative entropy to measure the deviation of distributions from each other. For simplicity, we choose White group as the reference distribution. As shown in Table 12.16, while GAC has minor improvement over baseline in the mean, it gives smaller relative entropy in the

Table 12.17 Network complexity and inference time

Model	Input resolution	# Parameters (M)	MACs (G)	Inference (ms)
Baseline	112 × 112	43.58	5.96	1.1
GAC	112 × 112	44.00	9.82	1.4

Table 12.18 Gender distribution of the datasets for gender estimation

Dataset	# Of images	
	Male	Female
Training	321,590	229,000
Testing	15,715	10,835

other 3 groups, indicating that the ratio distributions of other races in GAC are more similar, i.e., less biased, to the reference distribution. These results demonstrate the capability of GAC to increase the fairness of face representations.

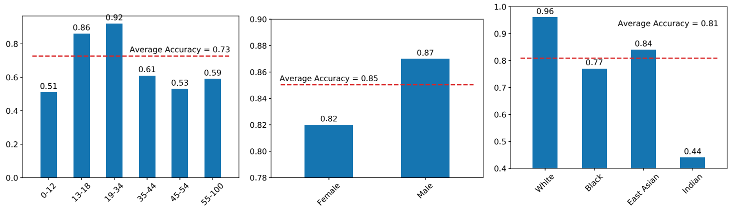
12.5.3.6 Network Complexity and FLOPs

Table 12.17 summarizes the network complexity of GAC and the baseline in terms of the number of parameters, multiplier–accumulator, and inference times. While we agree the number of parameters will increase with the number of demographic categories, it will not necessarily increase the inference time, which is more important for real-time applications.

12.6 Demographic Estimation

We train three demographic estimation models to annotate age, gender, and race information of the face images in BUPT-Balancedface and MS-Celeb-1M for training GAC and DebFace. For all three models, we randomly sample equal numbers of images from each class and set the batch size to 300. The training process ends after $35K$th iteration. All hyper-parameters are chosen by evaluations on a separate validation set. Below, we give the details of model learning and estimation performance of each demographic.

Gender: We combine IMDB, UTKFace, AgeDB, AFAD, and AAF datasets for learning the gender estimation model. Similar to age, 90% of the images in the combined datasets are used for training, and the remaining 10% are used for validation. Table 12.18 reports the total number of female and male face images in the training and testing set. More images belong to male faces in both training and testing set. Figure 12.17b shows the gender estimation performance on the validation set. The performance on male images is slightly better than that on female images.



(a) Age (b) Gender (c) Race

Fig. 12.17 Demographic Attribute Classification Accuracy on each group. The red dashed line refers to the average accuracy on all images in the testing set

Table 12.19 Race distribution of the datasets for race estimation

Dataset	# Of images			
	White	Black	East Asian	Indian
Training	468,139	150,585	162,075	78,260
Testing	9,469	4,115	3,336	3,748

Table 12.20 Age distribution of the datasets for age estimation

Dataset	# Of images in the age group					
	0–12	13–18	19–34	35–44	45–54	55–100
Training	9,539	29,135	353,901	171,328	93,506	59,599
Testing	1,085	2,681	13,848	8,414	5,479	4,690

Race: We combine AFAD, RFW, IMFDB-CVIT, and PCSO datasets for training the race estimation model. UTKFace is used as validation set. Table 12.19 reports the total number of images in each race category of the training and testing set. Similar to age and gender, the performance of race estimation is highly correlated to the race distribution in the training set. Most of the images are within the White group, while the Indian group has the least number of images. Therefore, the performance on White faces is much higher than that on Indian faces.

Age: We combine CACD, IMDB, UTKFace, AgeDB, AFAD, and AAF datasets for learning the age estimation model. 90% of the images in the combined datasets are used for training, and the remaining 10% are used for validation. Table 12.20 reports the total number of images in each age group of the training and testing set, respectively. Figure 12.17a shows the age estimation performance on the validation set. The majority of the images come from the age 19 to 34 group. Therefore, the age estimation performs the best on this group. The performance on the young children and middle to old age group is significantly worse than the majority group.

It is clear that all the demographic models present biased performance with respect to different cohorts. These demographic models are used to label the BUPT-Balancedface and MS-Celeb-1M for training GAC and DebFace. Thus, in addition to the bias from the dataset itself, we also add label bias to it. Since DebFace employs supervised feature disentanglement, we only strive to reduce the data bias instead of the label bias.

12.7 Conclusion

This chapter tackles the issue of demographic bias in FR by learning fair face representations. We present two de-biasing FR networks, GAC and DebFace, to mitigate demographic bias in FR. In particular, GAC is proposed to improve the robustness of representations for every demographic group considered here. Both adaptive convolution kernels and channel-wise attention maps are introduced to GAC. We further add an automatic adaptation module to determine whether to use adaptations in a given layer. Our findings suggest that faces can be better represented by using layers adaptive to different demographic groups, leading to more balanced performance gains for all groups. Unlike GAC, DebFace mitigates mutual bias across identities and demographic attributes recognition by adversarially learning the disentangled representation for gender, race, and age estimation, and face recognition simultaneously. We empirically demonstrate that DebFace can reduce bias not only in face recognition but in demographic attribute estimation as well.

References

1. https://yanweifu.github.io/FG_NET_data
2. http://trillionpairs.deepglint.com/overview
3. Alvi, M., Zisserman, A., Nellåker, C.: Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In: ECCV (2018)
4. Amini, A., Soleimany, A., Schwarting, W., Bhatia, S., Rus, D.: Uncovering and mitigating algorithmic bias through learned latent structure. In: AAAI/ACM Conference on AI, Ethics, and Society (2019)
5. Bastidas, A.A., Tang, H.: Channel attention networks. In: CVPR Workshops (2019)
6. Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K.N., Varshney, K.R.: Optimized pre-processing for discrimination prevention. In: NeurIPS (2017)
7. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGface2: A dataset for recognising faces across pose and age. In: FRGC. IEEE (2018)
8. Cavazos, J.G., Phillips, P.J., Castillo, C.D., O'Toole, A.J.: Accuracy comparison across face recognition algorithms: Where are we on measuring race bias? IEEE Trans. Biomet. Behav. Identity Sci. **3**(1), 101–111 (2021). https://doi.org/10.1109/TBIOM.2020.3027269
9. Chen, B., Li, P., Sun, C., Wang, D., Yang, G., Lu, H.: Multi attention module for visual tracking. Pattern Recognit. (2019)
10. Chen, B.C., Chen, C.S., Hsu, W.H.: Cross-age reference coding for age-invariant face recognition and retrieval. In: ECCV (2014)

11. Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., Chua, T.S.: SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In: CVPR (2017)
12. Chen, Z., Liu, L., Sa, I., Ge, Z., Chli, M.: Learning context flexible attention model for long-term visual place recognition. IEEE Robot. Autom. Lett. (2018)
13. Cheng, J., Li, Y., Wang, J., Yu, L., Wang, S.: Exploiting effective facial patches for robust gender recognition. Tsinghua Sci. Technol. **24**(3), 333–345 (2019)
14. Conti, J.R., Noiry, N., Clemencon, S., Despiegel, V., Gentric, S.: Mitigating gender bias in face recognition using the von mises-fisher mixture model. In: ICML. PMLR (2022)
15. Cook, C.M., Howard, J.J., Sirotin, Y.B., Tipton, J.L., Vemury, A.R.: Demographic effects in facial recognition and their dependence on image acquisition: An evaluation of eleven commercial systems. Behavior, and Identity Science. In: IEEE Transactions on Biometrics (2019)
16. Creager, E., Madras, D., Jacobsen, J.H., Weis, M., Swersky, K., Pitassi, T., Zemel, R.: Flexibly fair representation learning by disentanglement. In: ICML (2019)
17. Deb, D., Best-Rowden, L., Jain, A.K.: Face recognition performance under aging. In: CVPRW (2017)
18. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019)
19. Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., Zafeiriou, S.: Retinaface: Single-stage dense face localisation in the wild. ArXiv preprint arXiv:1905.00641 (2019)
20. Dhar, P., Gleason, J., Souri, H., Castillo, C.D., Chellappa, R.: Towards gender-neutral face descriptors for mitigating bias in face recognition. ArXiv preprint arXiv:2006.07845 (2020)
21. Ding, C., Li, Y., Xia, Y., Wei, W., Zhang, L., Zhang, Y.: Convolutional neural networks based hyperspectral image classification method with adaptive kernels. In: Remote Sensing (2017)
22. Ding, C., Li, Y., Xia, Y., Zhang, L., Zhang, Y.: Automatic kernel size determination for deep neural networks based hyperspectral image classification. In: Remote Sensing (2018)
23. Dooley, S., Downing, R., Wei, G., Shankar, N., Thymes, B., Thorkelsdottir, G., Kurtz-Miott, T., Mattson, R., Obiwumi, O., Cherepanova, V., et al.: Comparing human and machine bias in face recognition. ArXiv preprint arXiv:2110.08396 (2021)
24. Du, J., Zhang, S., Wu, G., Moura, J.M., Kar, S.: Topology adaptive graph convolutional networks. ArXiv preprint arXiv:1710.10370 (2017)
25. Gong, S., Liu, X., Jain, A.K.: Jointly de-biasing face recognition and demographic attribute estimation. In: ECCV (2020)
26. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
27. Grother, P., Ngan, M., Hanaoka, K.: Face recognition vendor test (FRVT) part 3: Demographic effects. In: Technical Report, National Institute of Standards and Technology (2019). https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8280.pdf
28. Grover, A., Song, J., Kapoor, A., Tran, K., Agarwal, A., Horvitz, E.J., Ermon, S.: Bias correction of learned generative models using likelihood-free importance weighting. In: NeurIPS (2019)
29. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: ECCV. Springer (2016)
30. Gwilliam, M., Hegde, S., Tinubu, L., Hanson, A.: Rethinking common assumptions to mitigate racial bias in face recognition datasets. In: ICCVW (2021)
31. Han, H., Anil, K.J., Shan, S., Chen, X.: Heterogeneous face attribute estimation: A deep multi-task learning approach. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1–1 (2017)
32. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: NeurIPS (2016)
33. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

34. Hendricks, L.A., Burns, K., Saenko, K., Darrell, T., Rohrbach, A.: Women also snowboard: Overcoming bias in captioning models. In: ECCV (2018)
35. Hou, R., Chang, H., Bingpeng, M., Shan, S., Chen, X.: Cross attention network for few-shot classification. In: NeurIPS (2019)
36. Howard, J., Sirotin, Y., Vemury, A.: The effect of broad and specific demographic homogeneity on the imposter distributions and false match rates in face recognition algorithm performance. In: IEEE BTAS (2019)
37. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)
38. Hu, T.K., Lin, Y.Y., Hsiu, P.C.: Learning adaptive hidden layers for mobile gesture recognition. In: AAAI (2018)
39. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst (2007)
40. Jia, X., De Brabandere, B., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. In: NeurIPS (2016)
41. Jourabloo, A., Yin, X., Liu, X.: Attribute preserved face de-identification. In: ICB (2015)
42. Kang, D., Dhar, D., Chan, A.: Incorporating side information by adaptive convolution. In: NeurIPS (2017)
43. Kearns, M., Neel, S., Roth, A., Wu, Z.S.: An empirical study of rich subgroup fairness for machine learning. In: Proceedings of the Conference on Fairness, Accountability, and Transparency (2019)
44. Kim, H., Mnih, A.: Disentangling by factorising. ArXiv preprint arXiv:1802.05983 (2018)
45. Kim, M.P., Ghorbani, A., Zou, J.: Multiaccuracy: Black-box post-processing for fairness in classification. In: AAAI/ACM (2019)
46. Klare, B.F., Burge, M.J., Klontz, J.C., Bruegge Richard, W.V., Jain, A.K.: Face recognition performance: Role of demographic information. IEEE Trans. Inf. Forensics Secur. **7**(6), 1789–1801 (2012)
47. Klare, B.F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Jain, A.K.: Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In: CVPR (2015)
48. Klein, B., Wolf, L., Afek, Y.: A dynamic convolutional layer for short range weather prediction. In: CVPR (2015)
49. Kortylewski, A., Egger, B., Schneider, A., Gerig, T., Morel-Forster, A., Vetter, T.: Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In: CVPRW (2019)
50. Li, S., Xing, J., Niu, Z., Shan, S., Yan, S.: Shape driven kernel adaptation in convolutional neural network for robust facial traits recognition. In: CVPR (2015)
51. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: CVPR (2019)
52. Li, X., Ye, M., Liu, Y., Zhu, C.: Adaptive deep convolutional neural networks for scene-specific object detection. IEEE Transactions on Circuits and Systems for Video Technology (2017)
53. Ling, H., Wu, J., Huang, J., Chen, J., Li, P.: Attention-based convolutional neural network for deep face recognition. Multimedia Tools and Applications (2020)
54. Linsley, D., Schiebler, D., Eberhardt, S., Serre, T.: Learning what and where to attend. In: ICLR (2019)
55. Liu, F., Zeng, D., Zhao, Q., Liu, X.: Disentangling features in 3D face shapes for joint face reconstruction and recognition. In: CVPR (2018)
56. Liu, Y., Wang, Z., Jin, H., Wassell, I.: Multi-task adversarial network for disentangled feature learning. In: CVPR (2018)

57. Liu, Y., Wei, F., Shao, J., Sheng, L., Yan, J., Wang, X.: Exploring disentangled feature representation beyond face identification. In: CVPR (2018)
58. Locatello, F., Abbati, G., Rainforth, T., Bauer, S., Schölkopf, B., Bachem, O.: On the fairness of disentangled representations. In: NeurIPS (2019)
59. Locatello, F., Bauer, S., Lucic, M., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. arXiv preprint arXiv:1811.12359 (2018)
60. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: NIPS (2018)
61. Lu, X., Wang, W., Ma, C., Shen, J., Shao, L., Porikli, F.: See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In: CVPR (2019)
62. Madras, D., Creager, E., Pitassi, T., Zemel, R.: Learning adversarially fair and transferable representations. In: ICML (2018)
63. Maze, B., Adams, J., Duncan, J.A., Kalka, N., Miller, T., Otto, C., Jain, A.K., Niggel, W.T., Anderson, J., Cheney, J., et al.: Iarpa janus benchmark-c: Face dataset and protocol. In: ICB (2018)
64. Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., Zafeiriou, S.: Agedb: the first manually collected, in-the-wild age database. In: CVPRW (2017)
65. Moyer, D., Gao, S., Brekelmans, R., Galstyan, A., Ver Steeg, G.: Invariant representations without adversarial training. In: NeurIPS (2018)
66. Narayanaswamy, S., Paige, T.B., Van de Meent, J.W., Desmaison, A., Goodman, N., Kohli, P., Wood, F., Torr, P.: Learning disentangled representations with semi-supervised deep generative models. In: NIPS (2017)
67. Niu, Z., Zhou, M., Wang, L., Gao, X., Hua, G.: Ordinal regression with multiple output cnn for age estimation. In: CVPR (2016)
68. Ozbulak, U.: Pytorch cnn visualizations. https://github.com/utkuozbulak/pytorch-cnn-visualizations (2019)
69. Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., Weinberger, K.Q.: On fairness and calibration. In: NeurIPS (2017)
70. Qin, H.: Asymmetric rejection loss for fairer face recognition. arXiv preprint arXiv:2002.03276 (2020)
71. Rothe, R., Timofte, R., Van Gool, L.: Deep expectation of real and apparent age from a single image without facial landmarks. IJCV (2018)
72. Sadiq, M., Shi, D., Guo, M., Cheng, X.: Facial landmark detection via attention-adaptive deep network. IEEE Access (2019)
73. Schmidhuber, J.: Learning factorial codes by predictability minimization. Neural Comput. **4**(6), 863–879 (1992)
74. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
75. Serna, I., Morales, A., Fierrez, J., Obradovich, N.: Sensitive loss: Improving accuracy and fairness of face representations with discrimination-aware deep learning. Artif. Intell. **305**, 103682 (2022)
76. Setty, S., Husain, M., Beham, P., Gudavalli, J., Kandasamy, M., Vaddi, R., Hemadri, V., Karure, J.C., Raju, R., Rajan, Kumar, V., Jawahar, C.V.: Indian Movie Face Database: A Benchmark for Face Recognition Under Wide Variations. In: NCVPRIPG (2013)
77. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: ICCV (2019)
78. Sindagi, V.A., Patel, V.M.: Ha-ccn: Hierarchical attention-based crowd counting network. IEEE Transactions on Image Processing (2019)

79. Song, J., Kalluri, P., Grover, A., Zhao, S., Ermon, S.: Learning controllable fair representations. In: ICAIS (2019)
80. Su, H., Jampani, V., Sun, D., Gallo, O., Learned-Miller, E., Kautz, J.: Pixel-adaptive convolutional neural networks. In: CVPR (2019)
81. Su, K., Yu, D., Xu, Z., Geng, X., Wang, C.: Multi-person pose estimation with enhanced channel-wise and spatial information. In: CVPR (2019)
82. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: CVPR (2014)
83. Tao, C., Lv, F., Duan, L., Wu, M.: Minimax entropy network: Learning category-invariant features for domain adaptation. arXiv preprint arXiv:1904.09601 (2019)
84. Terhörst, P., Kolf, J.N., Damer, N., Kirchbuchner, F., Kuijper, A.: Face quality estimation and its correlation to demographic and non-demographic bias in face recognition. In: IJCB (2020). 10.1109/IJCB48548.2020.9304865
85. Tran, L., Yin, X., Liu, X.: Disentangled representation learning GAN for pose-invariant face recognition. In: CVPR (2017)
86. Tran, L., Yin, X., Liu, X.: Representation learning by rotating your faces. IEEE Trans. on Pattern Analysis and Machine Intelligence **41**(12), 3007–3021 (2019)
87. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: CVPR (2015)
88. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)
89. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Signal Process. Lett. **25**(7), 926–930 (2018)
90. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: CVPR (2018)
91. Wang, M., Deng, W.: Mitigating bias in face recognition using skewness-aware reinforcement learning. In: CVPR (2020)
92. Wang, M., Deng, W., Hu, J., Tao, X., Huang, Y.: Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In: ICCV (2019)
93. Wang, P., Su, F., Zhao, Z., Guo, Y., Zhao, Y., Zhuang, B.: Deep class-skewed learning for face recognition. Neurocomputing (2019)
94. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. arXiv preprint arXiv:1910.03151 (2019)
95. Wang, T., Zhao, J., Yatskar, M., Chang, K.W., Ordonez, V.: Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In: CVPR (2019)
96. Wang, Z., Qinami, K., Karakozis, Y., Genova, K., Nair, P., Hata, K., Russakovsky, O.: Towards fairness in visual recognition: Effective strategies for bias mitigation. arXiv preprint arXiv:1911.11834 (2019)
97. Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: ECCV (2018)
98. Xie, W., Zisserman, A.: Multicolumn networks for face recognition. arXiv preprint arXiv:1807.09192 (2018)
99. Yang, J., Ren, Z., Gan, C., Zhu, H., Parikh, D.: Cross-channel communication networks. In: NeurIPS (2019)
100. Yin, B., Tran, L., Li, H., Shen, X., Liu, X.: Towards interpretable face recognition. In: ICCV (2019)
101. Yin, X., Liu, X.: Multi-task convolutional neural network for pose-invariant face recognition. IEEE Trans. Image Processing **27**(2), 964–975 (2017)

102. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Towards large-pose face frontalization in the wild. In: ICCV (2017)
103. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Feature transfer learning for face recognition with under-represented data. In: CVPR (2019)
104. Yucer, S., Akcay, S., Al-Moubayed, N., Breckon, T.P.: Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation. In: CVPRW (2020)
105. Yucer, S., Tektas, F., Al Moubayed, N., Breckon, T.P.: Measuring hidden bias within face recognition via racial phenotypes. In: WACV (2022)
106. Zamora Esquivel, J., Cruz Vargas, A., Lopez Meyer, P., Tickoo, O.: Adaptive convolutional kernels. In: ICCV Workshops (2019)
107. Zemel, R., Wu, Y., Swersky, K., Pitassi, T., Dwork, C.: Learning fair representations. In: ICML (2013)
108. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters (2016)
109. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: CVPR (2017)
110. Zhang, X., Wang, T., Qi, J., Lu, H., Wang, G.: Progressive attention guided recurrent network for salient object detection. In: CVPR (2018)
111. Zhang, Y., Zhao, D., Sun, J., Zou, G., Li, W.: Adaptive convolutional neural network and its application in face recognition. Neural Processing Letters (2016)
112. Zhang, Z., Song, Y., Qi, H.: Age progression/regression by conditional adversarial autoencoder. In: CVPR (2017)
113. Zhang, Z., Tran, L., Yin, X., Atoum, Y., Wan, J., Wang, N., Liu, X.: Gait recognition via disentangled representation learning. In: CVPR (2019)
114. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., Chang, K.W.: Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In: EMNLP (2017)

13 Adversarial Attacks on Face Recognition

Xiao Yang and Jun Zhu

13.1 Introduction

Face recognition is becoming a prevailing authentication solution in numerous biometric applications thanks to the rapid development of deep neural networks (DNNs) [18, 37, 39]. Empowered by the excellent performance of DNNs, face recognition models are widely deployed in various safety-critical scenarios ranging from finance/payment to automated surveillance systems. Despite its booming development, recent research in adversarial machine learning has revealed that face recognition models based on DNNs are highly vulnerable to adversarial examples [14, 40], which are maliciously generated to mislead a target model. Therefore, it will lead to serious consequences or security problems in real-world applications, such as deceiving the payment system in vending machines [13] and unlocking a mobile phone or car [41].

Extensive efforts have been devoted to crafting adversarial examples (i.e., adversarial attacks) on face recognition models, which can be conducive to evaluating model robustness [42, 51]. Adversarial attacks in the digital world [11, 35, 49, 51] add minimal perturbations to face images in the *digital* space, aiming to evade being recognized or to impersonate another identity. Some commercial face recognition APIs can also be attacked by adversarial examples in the black-box manner [11]. Besides, adversarial attacks in the physical space are characterized by adding adversarial patches that can be carefully attached to faces. These patches are subsequently captured by a camera and fed into a face recognition model

X. Yang (✉) · J. Zhu
Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
e-mail: yangxiao19@mails.tsinghua.edu.cn

J. Zhu
e-mail: dcszj@mail.tsinghua.edu.cn

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_13

to mislead the prediction. Some studies have shown the success of physical attacks against state-of-the-art face recognition models with regard to different attack types, such as eyeglass frames [34, 35, 44], hats [23], and stickers [16, 36].

In this chapter, we first briefly introduce the threat model on face recognition. Next, we describe some typical adversarial attacks on face recognition in both digital space and physical space, respectively. Besides, we present the methods related to adversarial defenses on face recognition, including input transformation and adversarial training. Furthermore, some positive applications of adversarial attacks are considered, such as making adversarial perturbations overlaid on facial images so that the original identities can be concealed without sacrificing the visual quality. Lastly, we elaborately discuss the growing future and unresolved problems of adversarial attacks on face recognition.

13.2 Threat Model

Face recognition usually focuses on solving the two sub-tasks: (1) face verification that distinguishes whether a pair of facial images belong to the same identity [14]; and (2) face identification that predicts the identity of a test facial image. We mainly consider face verification in this chapter, since the setting can be naturally extended to face identification.

In face verification, the feature distance between a pair of images $\{\boldsymbol{x}_1, \boldsymbol{x}_2\} \subset \mathcal{X}$ can be calculated as

$$\mathcal{D}_f(\boldsymbol{x}_1, \boldsymbol{x}_2) = \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\|_2^2, \tag{13.1}$$

where f extracts a normalized feature representation in $\mathbb{R}^d$ for an input face image. Note that this definition is consistent with the commonly used cosine similarity metric. Then the prediction of face verification can be formulated as

$$C(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mathbb{I}(\mathcal{D}_f(\boldsymbol{x}_1, \boldsymbol{x}_2) < \delta), \tag{13.2}$$

where $\mathbb{I}$ is the indicator function, and δ is a threshold. When $C(\boldsymbol{x}_1, \boldsymbol{x}_2) = 1$, the two images are recognized as the same identity, otherwise different identities (Fig. 13.1).

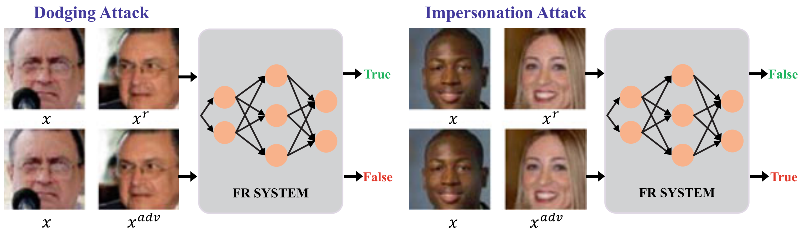


Fig. 13.1 Demonstration of the adversarial attack on face recognition. Given a pair of face images, the attacker can craft adversarial examples to achieve the dodging or impersonation attack with small perturbations for misleading the prediction of face recognition (FR) system

13.2.1 Adversary's Goals

There are generally two types of attacks with different goals for face verification, namely, dodging attack and impersonation attack, as detailed below.

Dodging attacks. In a dodging attack, an adversary seeks to make one face misidentified for misleading a face recognition system. In general, dodging attacks are of great interest in bypassing a face recognition system in surveillance. Formally, given a pair of face images $\boldsymbol{x}$ and $\boldsymbol{x}^r$ with the same identity, the adversary aims to modify $\boldsymbol{x}$ to craft an adversarial image $\boldsymbol{x}^{adv}$ that cannot be recognized by the model, i.e., to make $C(\boldsymbol{x}^{adv}, \boldsymbol{x}^r) = 0$.

Impersonation attacks. Impersonation attacks focus on generating an adversarial example that can be identified as another target identity, which is generally more difficult than dodging attacks. One attacker can try to camouflage his/her face to be identified as an authorized user for fooling the face authentication systems. Formally, given a pair of face images $\boldsymbol{x}$ and $\boldsymbol{x}^r$ with two different identities, the adversary will generate an adversarial image $\boldsymbol{x}^{adv}$ that is recognized as the target identity of $\boldsymbol{x}^r$, i.e., to make $C(\boldsymbol{x}^{adv}, \boldsymbol{x}^r) = 1$.

13.2.2 Adversary's Capabilities

The adversary's capability can be very different due to budget constraints, such as the perturbation budget in the digital space and the area budget for printing adversarial patches [34, 35] in the physical world.

Digital space. Adversarial examples in the digital space are usually assumed to be indistinguishable from the original ones from visual observations [14, 40], and the adversary can only introduce small modifications to the inputs. Recent research has widely adopted the ℓ_p additive perturbation setting, where the adversary has the ability to add a small perturbation measured by the ℓ_p norms (e.g., $p = \infty$ or $p = 2$) to the original input. To achieve the adversary's goal, an adversary can optimize the feature distance between the adversarial image $\boldsymbol{x}^{adv}$ and the counterpart face image $\boldsymbol{x}^r$, meanwhile keeping a small distance between $\boldsymbol{x}^{adv}$ and $\boldsymbol{x}$ in the input space.

For dodging attacks, an adversarial image can be crafted by maximizing the distance between $\boldsymbol{x}'$ and $\boldsymbol{x}^r$ in the feature space as

$$\boldsymbol{x}^{adv} = \underset{\boldsymbol{x}':\|\boldsymbol{x}'-\boldsymbol{x}\|_p \leq \epsilon}{\arg\max} \mathcal{D}_f(\boldsymbol{x}', \boldsymbol{x}^r), \tag{13.3}$$

where ϵ is a small constant that characterizes the level of perturbation. By solving problem (13.3), the face recognition model will be likely to mistakenly identify $\boldsymbol{x}^{adv}$ and $\boldsymbol{x}^r$ as different identities (as their feature distance can be larger than a predefined threshold δ).

For impersonation attacks, the adversary can similarly formulate the problem by minimizing the distance between $\boldsymbol{x}'$ and $\boldsymbol{x}^r$ as

$$\boldsymbol{x}^{adv} = \mathop{\arg\min}_{\boldsymbol{x}':\|\boldsymbol{x}'-\boldsymbol{x}\|_p \le \epsilon} \mathcal{D}_f(\boldsymbol{x}', \boldsymbol{x}^r). \tag{13.4}$$

Thus, the feature representation of $\boldsymbol{x}^{adv}$ will resemble that of $\boldsymbol{x}^r$, such that they are recognized as the same identity by the face recognition model.

Physical space. Adversarial examples in the physical space are usually required physically wearable for real human faces. They are thus realized by first generating adversarial perturbations confined to a specific region in the digital space and then printing adversarial patches (e.g., eyeglass frames, hats, etc).

For dodging attacks, an adversarial patch can be generated by maximizing the distance between $\boldsymbol{x}'$ and $\boldsymbol{x}^r$ in the confined region as

$$\begin{gathered}\boldsymbol{x}^{adv} = \arg\max \mathcal{D}_f(\boldsymbol{x}', \boldsymbol{x}^r),\\ \text{s.t. } \|\mathrm{M} \odot \boldsymbol{x}' - \mathrm{M} \odot \boldsymbol{x}\|_p \le \epsilon, \quad (\mathbf{1} - \mathrm{M}) \odot \boldsymbol{x}' = (\mathbf{1} - \mathrm{M}) \odot \boldsymbol{x},\end{gathered} \tag{13.5}$$

where $\mathrm{M} \in \{0, 1\}^d$ is a binary mask, $\odot$ is the element-wise dot product, and d is the dimension of the face image. By solving the constrained problem (13.5), the face recognition model can misidentify them as different identities by only modifying a confined region. Similarly, the definition can be extended to the impersonation attack by changing the optimization direction from problem (13.5).

13.2.3 Adversary's Knowledge

An adversary can have different levels of knowledge of the target face recognition models to craft adversarial examples, including white-box and black-box attacks. In the white-box case, the adversary can obtain accurate knowledge of detailed information from the target face recognition model, including architectures, parameters, and gradients of the loss regarding the input. In the black-box scenario, there are two general types including query-based attacks [7, 11] and transfer-based attacks [9]. The former attack leverages plenty of query feedback from the target model to generate adversarial examples. The latter directly relies on the transferability of adversarial examples, which assumes the availability of a substitute model based on which the adversarial examples can be generated.

13.3 Digital Adversarial Attacks

In this section, we introduce some attack methods in the digital space under both white-box and black-box (transfer-based and query-based) settings.

13.3.1 White-Box Attacks

We first summarize some typical white-box adversarial attack methods that are adopted for evaluating the robustness of face recognition models. And we only introduce these methods for dodging attacks as a default, since the extension to impersonation attacks is straightforward.

Fast Gradient Sign Method (FGSM) [14] crafts an adversarial example given a pair of images $\boldsymbol{x}$ and $\boldsymbol{x}^r$ with the same identity under the ℓ_∞ norm as

$$\boldsymbol{x}^{adv} = \boldsymbol{x} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{D}_f(\boldsymbol{x}, \boldsymbol{x}^r)), \tag{13.6}$$

where $\nabla_{\boldsymbol{x}} \mathcal{D}_f$ is the gradient of the feature distance w.r.t. $\boldsymbol{x}$, and sign$(\cdot)$ is the sign function to make the perturbation meet the ℓ_∞ norm bound. It can be extended to an ℓ_2 attack as

$$\boldsymbol{x}^{adv} = \boldsymbol{x} + \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} \mathcal{D}_f(\boldsymbol{x}, \boldsymbol{x}^r)}{\|\nabla_{\boldsymbol{x}} \mathcal{D}_f(\boldsymbol{x}, \boldsymbol{x}^r)\|_2}. \tag{13.7}$$

Basic Iterative Method (BIM) [24] iteratively takes multiple gradient updates based on FGSM as

$$\boldsymbol{x}_{t+1}^{adv} = \text{clip}_{\boldsymbol{x},\epsilon}\big(\boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{D}_f(\boldsymbol{x}_t^{adv}, \boldsymbol{x}^r))\big), \tag{13.8}$$

where clip$_{\boldsymbol{x},\epsilon}$ projects the adversarial example to satisfy the ℓ_∞ constraint and α is a small step size. It can also be extended to an ℓ_2 attack similar to FGSM. The projected gradient descent (PGD) [26] can be also viewed as a variant of BIM by adopting random starts.

Carlini & Wagner's Method (C&W) [6] is a powerful optimization-based attack method. It takes a Lagrangian form of the constrained optimization problem and adopts Adam [22] for optimization, which is quite effective for ℓ_2 attacks. However, the direct extension of the C&W method to face recognition is problematic since C&W used the loss function defined on the logits of the classification models. A suitable attack objective function was presented in [51] for face recognition systems. Specifically, for dodging attacks, the optimization problem can be formulated as

$$\boldsymbol{x}^{adv} = \arg\min_{\boldsymbol{x}'} \big\{\|\boldsymbol{x}' - \boldsymbol{x}\|_2^2 + c \cdot \max(\delta - \mathcal{D}_f(\boldsymbol{x}', \boldsymbol{x}^r), 0)\big\}, \tag{13.9}$$

where c is a parameter to balance the two loss terms, whose optimal value is discovered by binary search. Besides, δ is the threshold of the face verification model in Eq. (13.2).

13.3.2 Transfer-Based Black-Box Attacks

Under this black-box setting, the attackers have no access to the parameters of gradients of the model being attacked. Instead, adversarial examples are first generated by using the attack methods against a substitute face recognition model and then transferred to attack the black-box models. Therefore, the transferability of the attack examples is a key factor in the success of such attack methods. Some representative attack methods are presented below.

Momentum Iterative Method (MIM) [9] proposes to improve the transferability of adversarial examples by integrating a momentum term into BIM as

$$\begin{aligned} \boldsymbol{g}_{t+1} &= \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} \mathcal{D}_f(\boldsymbol{x}_t^{adv}, \boldsymbol{x}^r)}{\|\nabla_{\boldsymbol{x}} \mathcal{D}_f(\boldsymbol{x}_t^{adv}, \boldsymbol{x}^r)\|_1}; \\ \boldsymbol{x}_{t+1}^{adv} &= \text{clip}_{\boldsymbol{x},\epsilon}(\boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\boldsymbol{g}_{t+1})). \end{aligned} \tag{13.10}$$

By integrating the momentum term into the iterative process of the white-box attacks, this method can stabilize the update directions and avoid the local optima by input diversity.

Diverse Inputs Method (DIM) [46] relies on a stochastic transformation function to generate transferable adversarial examples at each iteration, which can be denoted as

$$\boldsymbol{x}_{t+1}^{adv} = \text{clip}_{\boldsymbol{x},\epsilon}\big(\boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{D}_f(T(\boldsymbol{x}_t^{adv}; p), \boldsymbol{x}^r))\big), \tag{13.11}$$

where $T(\boldsymbol{x}_t^{adv}; p)$ refers to some transformation to diversify the input with a probability p. Thus, they are incorporated into the attack process to create hard and diverse input patterns, which obtain higher success rates for black-box models and maintain similar success rates for white-box models.

Translation-Invariant Method (TIM) [10] proposes a translation-invariant attack method to generate more transferable adversarial examples against the defense models. Thus, the translation-invariant method can be integrated into BIM by convolving the gradient with the predefined kernel $\boldsymbol{W}$ as

$$\boldsymbol{x}_{t+1}^{adv} = \text{clip}_{\boldsymbol{x},\epsilon}\big(\boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\boldsymbol{W} * \nabla_{\boldsymbol{x}} \mathcal{D}_f(\boldsymbol{x}_t^{adv}, \boldsymbol{x}^r))\big). \tag{13.12}$$

By optimizing the objective in Eq. (13.12), TIM can mitigate the effect of different discriminative regions between models. The crafted adversarial examples are less sensitive to the white-box model, meanwhile enhancing the transferability of adversarial examples by gradient diversity. Therefore, TIM can better achieve evading the defense models by transferable adversarial examples.

Landmark-Guided Cutout (LGC) [51] proposes to leverage the special characteristics of a face recognition model to improve the black-box transferability of adversarial examples. LGC builds on an observation that the existing face recognition models have different attention maps for predictions, as illustrated in Fig. 13.2. Therefore, the crafted adversarial examples will rely on the discriminative local area of the substitute model, making it hard to transfer to the other models with different discriminative areas. The iterative procedure of LGC can be denoted as

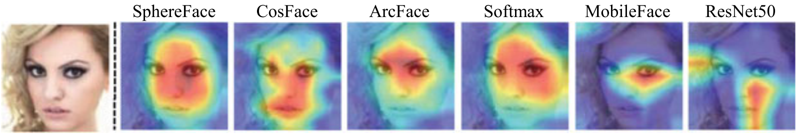


Fig. 13.2 The illustration of the attention maps highlighting the discriminative regions of the different models from [51]

$$\begin{aligned} \boldsymbol{g}_{t+1} &= \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} \mathcal{D}_f(M_t \odot \boldsymbol{x}_t^{adv}, \boldsymbol{x}^r)}{\|\nabla_{\boldsymbol{x}} \mathcal{D}_f(M_t \odot \boldsymbol{x}_t^{adv}, \boldsymbol{x}^r)\|_1}; \\ \boldsymbol{x}_{t+1}^{adv} &= \text{clip}_{\boldsymbol{x},\epsilon}(\boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\boldsymbol{g}_{t+1})), \end{aligned} \tag{13.13}$$

where $M_t \in \{0, 1\}^d$ is a binary mask, $\odot$ is the element-wise dot product, and d is the dimension of the face image. In the t-th iteration, after initializing the values of M_t as 1, some randomly sampled fixed-size small square regions are set to 0 to form M_t. By optimizing the object in Eq. (13.13), an adversary can occlude units from prominent input regions of the images, making the network focus on less prominent regions and obtain more transferable adversarial examples.

13.3.3 Query-Based Black-Box Attacks

Although the white-box access to the model is unavailable in the black-box setting, query-based methods [11, 21] generally require a large number of queries to generate an adversarial example with a minimum perturbation or converge to a large perturbation with few queries. Dong et al. [11] proposed the Evolutionary Attack method, which adopted a query-based black-box setting for attacking a real-world face recognition system. This method models the local geometry of the search directions and reduces the dimension of the search space. Specifically, the objective of an impersonation attack can be achieved by solving the black-box optimization problem as

$$\min_{\boldsymbol{x}'} \mathcal{D}_f\left(\boldsymbol{x}', \boldsymbol{x}\right), \quad \text{s.t. } \hat{C}\left(f\left(\boldsymbol{x}'\right)\right) = 1, \tag{13.14}$$

where $\hat{C}(\cdot)$ is an adversarial criterion that takes 1 if the attack requirement is satisfied and 0 otherwise. To achieve this, they adopted a valuable and straightforward variant of the covariance matrix adaptation evaluation strategy (CMA-ES) [17] for black-box optimization. To accelerate this algorithm, they proposed to model the local geometry of the search directions for appropriately sampling the random noise. Besides, the characteristics of the digital faces were incorporated to reduce the dimensions of search space. Experimental results also demonstrated that the proposed method can achieve faster convergence and smaller distortions against the state-of-the-art face recognition models, compared with other methods in both dodging and impersonation settings.

13.3.4 Universal Adversarial Attacks

The aforementioned methods (e.g., FGSM and MIM) compute image-specific adversarial perturbations by performing gradient updates iteratively. Different from image-specific attacks, image-agnostic attacks belong to image-independent (universal) methods. The first pipeline is to learn a universal perturbation by iterative optimization. For instance, UAP [29] proposes to mislead a model by adding a learned universal noise vector. Another pipeline of attacks introduces a learned universal function (generative model) [38, 48] on the data distribution independent of specific instances. Generally, the training objectives of the generative model $\mathcal{G}_\theta$ seek to minimize the training error on the perturbed image of the generator for achieving an impersonation attack as

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}[\mathcal{D}_f\big(\boldsymbol{x}+\mathcal{G}_\theta(\boldsymbol{x}), \boldsymbol{x}^r\big)], \quad \text{s.t. } \|\mathcal{G}_\theta(\boldsymbol{x})\|_\infty \le \epsilon, \tag{13.15}$$

which adopts an end-to-end training paradigm with the goal of generating adversarial images to mislead the face recognition model. By solving problem (13.15), this method can obtain a generative model by minimizing the distance of $\boldsymbol{x}$ and $\boldsymbol{x}^r$ in the training dataset. Once the parameter θ of the generator $\mathcal{G}_\theta$ is trained completely, the adversarial example $\boldsymbol{x}^{adv}$ can be crafted by $\boldsymbol{x}^{adv} = \boldsymbol{x} + \mathcal{G}_\theta(\boldsymbol{x})$ for any given face image $\boldsymbol{x}$, which only requires an inference for this face image $\boldsymbol{x}$. Note that universal adversarial attacks can promote more general and transferable adversarial examples [30, 48] since the universal perturbation or function can alleviate the data-specific overfitting problem by training on an unlabeled dataset.

13.4 Physical Adversarial Attacks

We now introduce physical adversarial attacks that aim to deceive face recognition models in the physical world [3, 14, 40]. Therefore, a specific characteristic of physical adversarial examples is making them physically wearable for real human faces. Next, we describe two common attack types for effectively achieving physical adversarial attacks, i.e., patch-based and light-based ones.

13.4.1 Patch-Based Physical Attacks

Patch-based physical adversarial examples are usually realized by first generating adversarial perturbations confined to a specific region in the digital space and then printing adversarial patches (e.g., eyeglass frames, hats, etc). Some volunteers will be subsequently asked to attach them and test the attack performance against face recognition models under a specific environment.

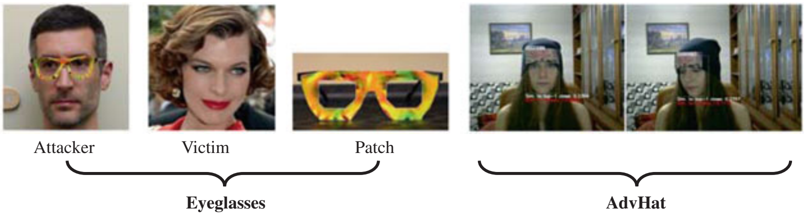


Fig. 13.3 Some physically realizable adversarial patches of the eyeglass frame [35] and hat [23] with the aim of misleading face recognition models

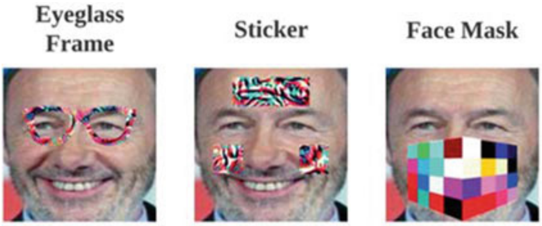


Fig. 13.4 Some other physically realizable adversarial patches of the stickers and masks from [42] with the aim of misleading face recognition models

Some research has explored the case that the adversarial perturbations are confined to a specifically designed confined region, as illustrated in Figs. 13.3 and 13.4. This can be achieved by setting a binary mask $\mathrm{M} \in \{0, 1\}^d$, and d is the dimension of the face images. The procedure can make the adversarial perturbations confined in the pixels where the value of the mask is 1, which can be denoted as

$$\boldsymbol{x}_{t+1}^{adv} = \mathrm{proj}_D\big(\boldsymbol{x}_t^{adv} + \alpha \cdot \mathrm{M} \odot \mathrm{sign}(\nabla_{\boldsymbol{x}} \mathcal{D}_f(\boldsymbol{x}_t^{adv}, \boldsymbol{x}^r))\big), \tag{13.16}$$

where $D = \{\boldsymbol{x} : \|\mathrm{M} \odot \boldsymbol{x} - \mathrm{M} \odot \boldsymbol{x}^r\|_\infty \leq \epsilon\}$. By iteratively performing this updating rule, the attackers can obtain the final adversarial patches. Once crafted in the whole attack generation pipeline, the adversarial patches will be posted on the attack to mislead the black-box face recognition system. The expectation over transformation method [2] has been proposed to make the physical adversarial patches robust under diverse physical variations. To further boost the black-box transferability of adversarial patches, Xiao et al. [44] proposed to regularize the adversarial patch by optimizing it on a low-dimensional manifold. By optimizing the adversarial perturbations on the latent space of a pre-trained generative model, the adversarial perturbations exhibit strong semantic patterns inherent to the face image, meanwhile obtaining an excellent performance on the black-box transferability.

To generate a physically realizable adversarial patch for fooling face recognition models, Sharif et al. [35] proposed the mask type of Eyeglasses. Specifically, they first introduced a readily available digital replica of eyeglass frames and utilized a printer to print the front plane of the eyeglass frames on paper. The color was iteratively updated through the gradient descent method to generate adversarial perturbations. Consequently, the adversarial frames can evade the recognition model by only occupying about 6.5% of the whole face image pixels. Figure 13.3 also illustrates an impersonation attack by wearing 2D-printing or 3D-printing glass frames. Besides, AdvHat [23] adopted the mask type of Hat to achieve an impersonation attack. They proposed to simulate the off-plane bending as a parabolic transformation in the 3D space which maps each point of the sticker to the new point. Furthermore, the 3D affine transformation was applied to the sticker to simulate the corresponding pitch rotation based on the obtained coordinates. To preserve the smoothness of adversarial perturbations, the optimization was iteratively updated by minimizing total variation (TV) [28] as

$$\mathrm{TV}(\boldsymbol{n}) = \sum_{i,j} \left(\left(\boldsymbol{n}_{i,j} - \boldsymbol{n}_{i+1,j} \right)^2 + \left(\boldsymbol{n}_{i,j} - \boldsymbol{n}_{i,j+1} \right)^2 \right)^{1/2}, \tag{13.17}$$

where $\boldsymbol{n}_{i,j}$ denotes a pixel in $\boldsymbol{n}$ at coordinate $\{i, j\}$. The objective will be lower if the values of adjacent pixels are closer to each other, meaning that the perturbation is smoother. Therefore, the smoothness of the adversarial examples will be promoted and the physical realizability is also improved.

Although AdvHat considered simple geometric transformations of the patch, it would inevitably result in unsatisfying performance when fitting the patch to the real 3D face due to the deformation. Yang et al. [50] proposed a 3D-aware attack method—Face3DAdv to craft robust adversarial patches, which can naturally stitch a patch onto the face to make the adversarial patch more versatile and realistic by fully leveraging the recent advances in 3D face modeling, as illustrated in Fig. 13.5. Moreover, Face3DAdv also exploited profitable 3D face transformations and realistic physical variations based on the 3D simulator. Experiments also showed obvious improvements over the previous 2D attack methods against different face recognition models in diverse physical conditions of 3D transformations, lighting variations, etc.

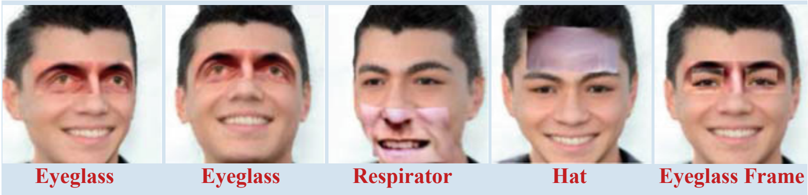


Fig. 13.5 The 3D adversarial examples are derived from different physically realizable attacks from Yang et al. [50]

13.4.2 Light-Based Physical Attacks

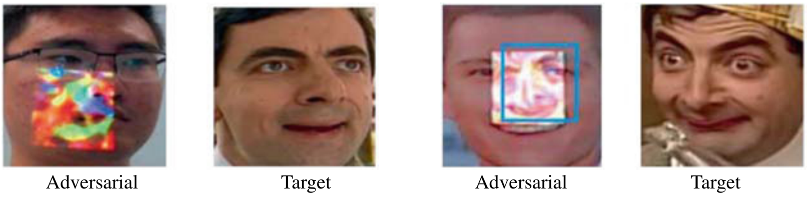
An adversary can first generate a digital adversarial pattern using one or more face images of the target identity. Rather than printing 2D or 3D adversarial patches, Nguyen et al. proposed the adversarial light projection attack [31] on face recognition systems. There are two steps in the generation phase: (1) calibrate the camera-projector setup based on the specific environment and calculate the adversarial pattern in the digital domain for dodging or impersonation attacks; (2) project the computed digital adversarial pattern onto the adversary's face using the projector to attack the deployed face recognition system.

Specifically, the adversarial pattern in the digital domain for dodging attacks can be represented as

$$\boldsymbol{x}_{t+1}^{adv} = \mathrm{clip}_{\boldsymbol{x},\epsilon}\big(\boldsymbol{x}_t^{adv} + \alpha \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}} \mathcal{D}_f(T(\boldsymbol{x}_t^{adv}), \boldsymbol{x}^r))\big), \tag{13.18}$$

where $T(\cdot)$ refers to some transformation operations to diversify the input. After the generation, this method will also consider two calibration steps integral to the effectiveness of the physical attack. First, the position calibration aims to ensure that the adversarial pattern crafted in the digital domain can be projected onto the appropriate region of the attacker while conducting the attack. Second, the color calibration focuses on the reproduction of digital adversarial examples with high fidelity by the projector. Some light-based physical adversarial examples are presented in Fig. 13.6, which can achieve white-box and black-box impersonation attacks against different recognition models.

Besides, Zhou et al. proposed a novel infrared-based technique capable of stealthily morphing one's facial features to impersonate a different individual. Meanwhile, they also developed a new algorithm to search for adversarial examples under the constraints of the limitations of commercial-off-the-shelf LEDs. Experimentally, a large-scale study on the LFW dataset was conducted, which showed the attack success rates of over 70%.



(a) White-box Impersonation Attack (b) Black-box Impersonation Attack

Fig. 13.6 Some examples of white-box and black-box impersonation attacks from [31]. The face image with adversarial light projected in the physical domain will make the face recognition model predict the targeted identity

13.5 Adversarial Defense for Face Recognition

Adversarial attacks raise a large amount of security issues to face recognition systems in diverse settings. Extensive research has also concentrated on making face recognition models robust to various adversarial attacks. The defense strategies can be generally divided into two categories: (1) input transformation: modifying the altered input throughout testing; and (2) adversarial training: injecting adversarial examples into training data.

13.5.1 Input Transformation

Some defenses can transform the inputs before feeding them into deep neural networks, including JPEG compression [12] and bit-depth reduction [47], and total variance minimization [15]. Besides, some works chose to add randomness into the input [45] for mitigating adversarial effects. Therefore, these input transformation methods can be naturally incorporated into the face recognition models for adversarial defenses in the testing phase. A pipeline of the randomization-based defense mechanism is also illustrated in Fig. 13.7. However, these methods generally rely on vanishing gradients or random gradients to prohibit adversarial attacks. Some works [1, 19] demonstrated that these input transformation methods can be defeated by adopting adaptive attacks.

13.5.2 Adversarial Training

One of the most effective methods of defending adversarial attacks is adversarial training [5, 25, 52], where the authors proposed to generate adversarial examples online and augment

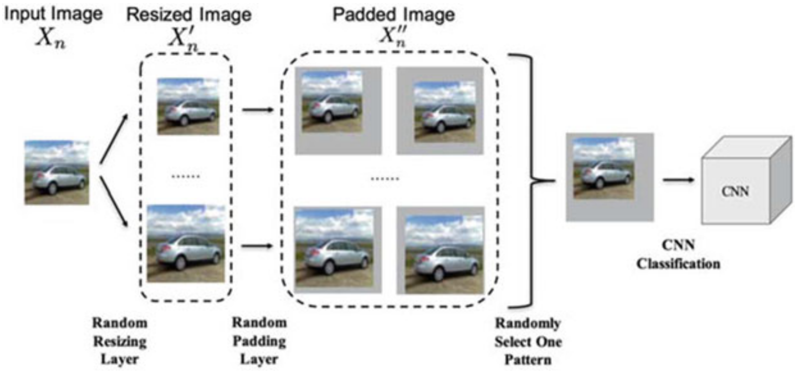


Fig. 13.7 A pipeline of the randomization-based defense mechanism from [45], including randomly resizing, padding, and selecting images

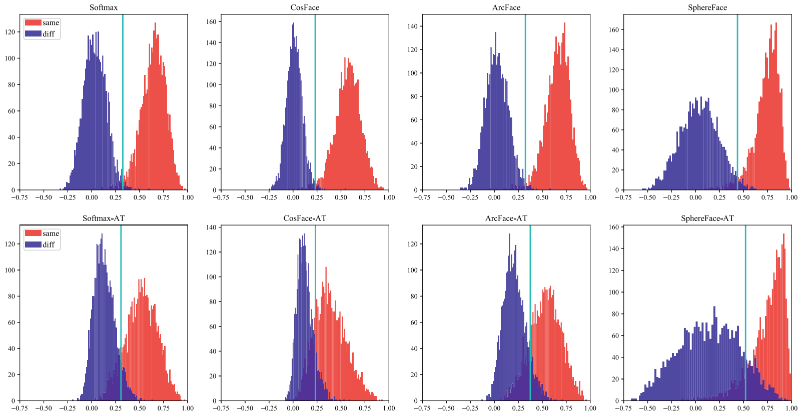


Fig. 13.8 Distance distributions of both same and different pairs on the LFW dataset [20] under normal training and adversarial training from [51]. The cyan lines refer to the thresholds

them into the training data in a mixed manner, i.e., each mini-batch of training data consists of a mixture of clean and adversarial samples. PGD-AT [27], as the most popular one, formulates the adversarial training procedure as a min-max problem. Therefore, adversarial training methods in face recognition can be formulated as a two-stage framework:

$$\min_{\boldsymbol{\omega},\mathbf{W}} \frac{1}{n} \sum_{i=1}^{n} \max_{\boldsymbol{\eta}_i \in \mathcal{S}} \mathcal{L}(f(\boldsymbol{x}_i + \boldsymbol{\eta}_i), y_i, \mathbf{W}), \tag{13.19}$$

where $f(\cdot)$ is the feature extractor with parameters $\boldsymbol{\omega}$, the matrix $\mathbf{W} = (W_1, ..., W_C)$ is the weight matrix for the task with C labels, $\mathcal{L}$ is a cross-entropy loss, and $\mathcal{S} = \{\boldsymbol{\eta} : \|\boldsymbol{\eta}\|_\infty \leq \epsilon\}$ is a set of allowed points around $\boldsymbol{x}$ with the perturbation ϵ. Adversarial examples are crafted in the inner maximization, and model parameters are optimized by solving the outer minimization. Thus they are iteratively executed in training until model parameters $\boldsymbol{\omega}$ and $\mathbf{W}$ converge. The project gradient descent method (PGD) [27] has been generally applied in the inner optimization, which is denoted by taking multiple steps as

$$\boldsymbol{\eta}_i^{t+1} = \prod_{\mathcal{S}} \left(\boldsymbol{\eta}_i^t + \alpha \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\eta}_i^t), y_i, \mathbf{W}))\right), \tag{13.20}$$

where $\boldsymbol{\eta}_i$ is the adversarial perturbation at the t-th step, α is the step size and $\prod(\cdot)$ is a projection function in $\mathcal{S}$. Since the problems of inner maximization and outer minimization are mutually coupled, they are iteratively completed in the training phase until the model parameters converge. Among the defense methods studied in [51], adversarial training is still the most robust method, which presents consistent performance on different face loss constraints. Besides, adversarial training still results in a reduction of natural accuracy, which also accords with the performance of the general image classification (Fig. 13.8).

To defend physically realizable attacks on face recognition systems, Wu et al. [43] proposed to adopt adversarial training with the rectangular occlusion attacks. Specifically, rectangular occlusion attacks are introduced by locating a small adversarially crafted rectangle among a collection of possible regions in a face image. Plenty of experiments also demonstrated that the proposed defense method can effectively improve the robustness against the eyeglass frame attack for VGG-based face recognition system [32].

13.6 Positive Applications of Adversarial Attacks

With the growing ubiquity of deep neural networks, face recognition systems are increasingly applied by private companies, government agencies, and commercial surveillance services. These systems can typically deal with personal data by scraping social profiles from user images. As a byproduct, they also increase the potential risks for privacy leakage of personal information. Therefore, it is imperative to provide users with an effective method to protect private information from being unconsciously identified. Recent research has found that adversarial examples can mislead a face recognition system [14, 35, 40, 51] by overlaying adversarial perturbations on the original images, thus becoming an appealing mechanism to apply an adversarial perturbation to conceal one's identity.

Fawkes [33] was developed to prevent social media images from being used by unauthorized facial recognition systems based on adversarial attacks, which fooled unauthorized facial recognition models by introducing adversarial examples into training data. Based on this, users can add imperceptible pixel-level changes to their own photos before releasing them. When used to train face recognition models, these images produce the models that effectively make natural images of the user misidentified. Experimentally, a 95+% protection success rate was provided by Fawkes against various face recognition models.

LowKey [8] designed a black-box adversarial attack on facial recognition models, which moved the feature space representations of gallery faces. Experimentally, LowKey conducted the evaluations on a large collection of images and identities. As a comparison, Fawkes assumed that face recognition practitioners trained their models on each individual's data and performed evaluations on small datasets. Experimental results also demonstrated the effective performance of LowKey against commercial black-box APIs, including Amazon Recognition and Microsoft Azure Face.

Recent TIP-IM [49] involved some novel considerations for preventing identity leakage against unauthorized recognition systems from the user's perspective, such as targeted protection, natural outputs, black-box face systems, and unknown gallery sets. Besides, they proposed to generate an adversarial identity mask, where multi-target sets are introduced with a well-matched optimization mechanism to guarantee black-box effectiveness. Furthermore, *maximum mean discrepancy* (MMD) [4] was introduced as an effective non-parametric and differentiable metric for naturalness, which can be capable of comparing two data distributions and evaluating the imperceptibility of the generated images. Figure 13.9 also presented

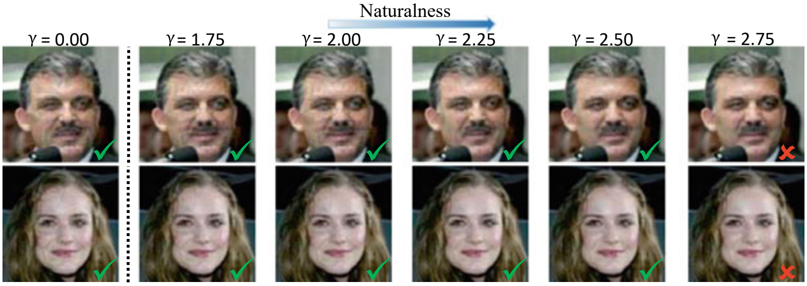


Fig. 13.9 Some examples of face identity protection based on adversarial attacks with different coefficients from [49]. Green hook refers to success, which also implies a trade-off between effectiveness and naturalness

a trade-off between effectiveness and naturalness. As the coefficient increases, the visual quality of the generated images gets better based on different metrics. Therefore, it can be seen that, to an extent, TIP-IM can control the degree of the generated protected images when conditioning on different hyper-parameters.

13.7 Discussion

With the evolution of new technologies regarding adversarial attacks, many works have achieved impressive performance on attack or defense against face recognition models. However, some problems still remain largely open and are calling for further investigation to develop robust face recognition models.

First, one problem is how to craft effective physical adversarial patches for achieving impersonation attacks against commercial face recognition services, which usually incorporate strong defensive mechanisms with face anti-spoofing. Such attack strategies are valuable to test the robustness of commercial systems. Since a 3D texture-based patch does not change the depth of a face image, it may be more conducive to passing commercial defense services steadily. Therefore, a 3D texture-based attack may be a feasible solution regarding effectiveness and imperceptibility against commercial face recognition services.

Besides, although the adversarial perturbations generated by the existing methods have a small intensity change, they may still sacrifice the visual quality for human perception due to the artifacts. Thus, future technologies should also consider visually natural characteristics from the corresponding original ones, otherwise, it may introduce undesirable appearances as a result.

Furthermore, the existing adversarial attacks are remarkably demonstrated to be image-specific, and image-agnostic (universal) attacks against face recognition models are still worth considering. Universal adversarial attacks have the ability to craft universal adversarial perturbations given one face target, meanwhile generating strong transferable patterns. Therefore, universal attacks should be an essential concern in future research.

Lastly, adversarial training is becoming the most robust method among the defensive strategies, which can adaptively integrate with different loss functions from face recognition for seeking robust performance. However, adversarial training still results in a reduction of natural accuracy and high training cost in face recognition. These matters inhibit the researchers from designing practical defensive techniques in automatic face recognition systems. Therefore, future works will be encouraged to propose efficient and practical defenses against various confined attacks on face recognition.

References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: International Conference on Machine Learning (ICML) (2018)
2. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. In: International Conference on Machine Learning (ICML) (2018)
3. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. In: Joint European conference on machine learning and knowledge discovery in databases, pp. 387–402. Springer (2013)
4. Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., Smola, A.J.: Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics **22**(14), e49–e57 (2006)
5. Brendel, W., Rauber, J., Kurakin, A., Papernot, N., Veliqi, B., Mohanty, S.P., Laurent, F., Salathé, M., Bethge, M., Yu, Y., et al.: Adversarial vision challenge. In: The NeurIPS'18 Competition, pp. 129–153. Springer (2020)
6. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy (2017)
7. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J.: Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15–26. ACM (2017)
8. Cherepanova, V., Goldblum, M., Foley, H., Duan, S., Dickerson, J., Taylor, G., Goldstein, T.: Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition. ArXiv preprint arXiv:2101.07922 (2021)
9. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
10. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
11. Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., Zhu, J.: Efficient decision-based black-box adversarial attacks on face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
12. Dziugaite, G.K., Ghahramani, Z., Roy, D.M.: A study of the effect of jpg compression on adversarial images. ArXiv preprint arXiv:1608.00853 (2016)

13. GeekPwn, A.F.M.C.: (2020). http://2020.geekpwn.org/zh/index.html Accessed: 2020-10-24
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (ICLR) (2015)
15. Guo, C., Rana, M., Cisse, M., van der Maaten, L.: Countering adversarial images using input transformations. In: International Conference on Learning Representations (ICLR) (2018)
16. Guo, Y., Wei, X., Wang, G., Zhang, B.: Meaningful adversarial stickers for face recognition in physical world. ArXiv preprint arXiv:2104.06728 (2021)
17. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evol. Comput. **9**(2), 159–195 (2001)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
19. He, W., Li, B., Song, D.: Decision boundary analysis of adversarial examples. In: International Conference on Learning Representations (ICLR) (2018)
20. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In: Technical report (2007)
21. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: International Conference on Machine Learning (ICML) (2018)
22. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
23. Komkov, S., Petiushko, A.: Advhat: Real-world adversarial attack on arcface face id system. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 819–826. IEEE (2021)
24. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. In: International Conference on Learning Representations (ICLR) Workshops (2017)
25. Kurakin, A., Goodfellow, I., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., Xie, C., et al.: Adversarial attacks and defences competition. ArXiv preprint arXiv:1804.00097 (2018)
26. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (ICLR) (2018)
27. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (ICLR) (2018)
28. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5188–5196 (2015)
29. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1765–1773 (2017)
30. Naseer, M.M., Khan, S.H., Khan, M.H., Khan, F.S., Porikli, F.: Cross-domain transferability of adversarial perturbations. In: Advances in Neural Information Processing Systems, pp. 12905–12915 (2019)
31. Nguyen, D.L., Arora, S.S., Wu, Y., Yang, H.: Adversarial light projection attacks on face recognition systems: A feasibility study. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 814–815 (2020)
32. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Association (2015)
33. Shan, S., Wenger, E., Zhang, J., Li, H., Zheng, H., Zhao, B.Y.: Fawkes: Protecting privacy against unauthorized deep learning models. In: 29th {USENIX} Security Symposium ({USENIX} Security 20), pp. 1589–1604 (2020)

34. Sharif, M., Bhagavatula, S., Bauer: Adversarial generative nets: Neural network attacks on state-of-the-art face recognition. ArXiv preprint arXiv:1801.00349 (2017)
35. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1528–1540. ACM (2016)
36. Shen, M., Yu, H., Zhu, L., Xu, K., Li, Q., Hu, J.: Effective and robust physical-world attacks on deep learning face recognition systems. IEEE Trans. Inf. Forensics Secur. **16**, 4063–4077 (2021)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
38. Song, Y., Shu, R., Kushman, N., Ermon, S.: Constructing unrestricted adversarial examples with generative models. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
39. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
40. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (ICLR) (2014)
41. Technologies, V.: http://visagetechnologies.com/face-recognition-in-cars/ Accessed: 2020-10-9 (2020)
42. Tong, L., Chen, Z., Ni, J., Cheng, W., Song, D., Chen, H., Vorobeychik, Y.: Facesec: A fine-grained robustness evaluation framework for face recognition systems. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13254–13263 (2021)
43. Wu, T., Tong, L., Vorobeychik, Y.: Defending against physically realizable attacks on image classification. ArXiv preprint arXiv:1909.09552 (2019)
44. Xiao, Z., Gao, X., Fu, C., Dong, Y., Gao, W., Zhang, X., Zhou, J., Zhu, J.: Improving transferability of adversarial patches on face recognition with generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11845–11854 (2021)
45. Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.: Mitigating adversarial effects through randomization. In: International Conference on Learning Representations (ICLR) (2018)
46. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
47. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. In: Proceedings of the Network and Distributed System Security Symposium (NDSS) (2018)
48. Yang, X., Dong, Y., Pang, T., Su, H., Zhu, J.: Boosting transferability of targeted adversarial examples via hierarchical generative networks. ArXiv preprint arXiv:2107.01809 (2021)
49. Yang, X., Dong, Y., Pang, T., Su, H., Zhu, J., Chen, Y., Xue, H.: Towards face encryption by generating adversarial identity masks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3897–3907 (2021)
50. Yang, X., Dong, Y., Pang, T., Xiao, Z., Su, H., Zhu, J.: Controllable evaluation and generation of physical adversarial patch on face recognition. ArXiv e-prints pp. arXiv–2203 (2022)
51. Yang, X., Yang, D., Dong, Y., Yu, W., Su, H., Zhu, J.: Robfr: Benchmarking adversarial robustness on face recognition. ArXiv preprint arXiv:2007.04118 (2020)
52. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning (ICML) (2019)

14 Heterogeneous Face Recognition

Decheng Liu, Nannan Wang, and Xinbo Gao

14.1 Introduction

Face recognition is one of the most important applications in computer vision. This is because face recognition could achieve efficient and convenient identity verification in uncontrolled scenarios. With the development of deep learning models in image processing, superior recognition accuracy has been achieved recently [6, 59]. However, real face images are captured through different sources, such as sketch artists and infrared imaging devices, called **heterogeneous faces**. Furthermore, matching face images in different modalities, which is referred to as **heterogeneous face recognition (HFR)**, is now attracting growing attention in both biometrics research and industry.

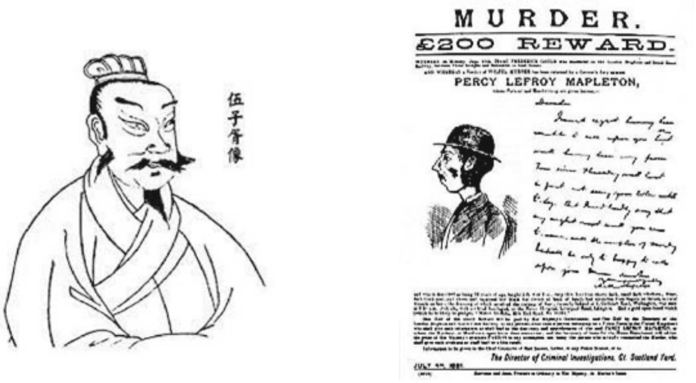
For example, face sketches are desired in criminal investigations when the frontal photos or videos of suspects are not available. The police need to generate face sketches by hand or specific software according to the descriptions of eyewitnesses. Actually, the earliest face sketch is created in Spring and Autumn and Warring States Periods of China. The king of Chu forces the army to catch Wu Zixu according to the hand-drawn portrait sketch (as shown

D. Liu
State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, Shaanxi 710071, China
e-mail: dchliu@xidian.edu.cn

N. Wang (✉)
State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi 710071, China
e-mail: nnwang@xidian.edu.cn

X. Gao
Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
e-mail: gaoxb@cqupt.edu.cn

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_14



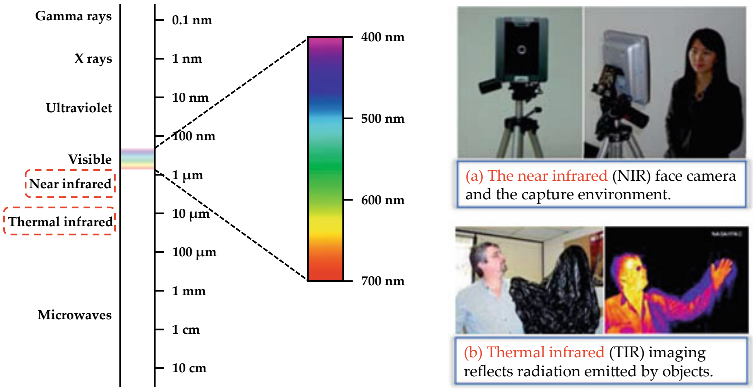
(a) The biography of Wu Zixu (in Spring and Autumn and Warring States Period of China)

(b) The biography of Mapleton (in 1880s)

Fig. 14.1 The examples of face sketches application in the history

in Fig. 14.1a). Additionally, the first police composite face sketch to appear on a "Wanted" poster in a newspaper, which shows the hand-drawn sketch of murder called Mapleton in 1881 (as shown in Fig. 14.1b). Nowadays, face sketches are also usually generated by specific software [38]. It is because the police artists always need much time to draw sketches by hand, but these composite sketches are directly combined by choosing suitable face components, which indeed takes less time.

The cross-spectrum face images are also important for heterogeneous face recognition. As shown in Fig. 14.2, the spectrogram shows the range of visible light is 400 nm to 700 nm, and the range of near-infrared and thermal infrared is higher than 700 nm. As we all know, near-infrared images (NIR) are usually acquired in poor illumination environment,



(a) The near infrared (NIR) face camera and the capture environment.

(b) Thermal infrared (TIR) imaging reflects radiation emitted by objects.

Fig. 14.2 The illustration of spectrogram and other spectrum images (like NIR and TIR)

which is inherently less sensitive to light influence. Matching the probe NIR to the gallery, visual face images has raised more concerns recently. Similarly, the thermal infrared (TIR) faces are captured by specific TIR camera, which could effectively reflect radiation emitted by objects.

Overall, traditional homogeneous face recognition methods perform poorly in most heterogeneous face scenarios due to the large discrepancy between face images in different modalities. Thus, heterogeneous face analysis is still an important and challenging problem in law enforcement. In addition, there still exists a strong request to learn more meaningful and interpretable representations of heterogeneous faces.

14.1.1 Literature Review

In this section, we give a comprehensive literature review about HFR algorithms. Due to the great discrepancies, conventional homogeneous face recognition methods perform poorly by directly identifying the probe image from gallery images in most HFR scenarios. Existing approaches can be generally grouped into three categories: synthesis-based methods, common space projection-based methods, and feature descriptor-based methods. Here we first show the basic procedures of these three kinds of methods. (1) Feature descriptor-based methods [8, 10, 22, 23, 71] first represent face images with local feature descriptors. These encoded descriptors can then be utilized for recognition. *However, most existing methods of this category represent an image ignoring the special spatial structure of faces, which is crucial for face recognition in reality.* (2) Synthesis-based methods [11, 28, 34, 55, 61, 64] first transform the heterogeneous face images into the same modality. Once the synthesized photos are generated from non-photograph images or vice versa, conventional face recognition algorithms can be applied directly. *However, the synthesis process is actually more difficult than recognition and the performance of these methods heavily depends on the fidelity of the synthesized images.* (3) Common space projection-based methods [19, 25, 30, 42, 43, 51] attempt to project face images in different modalities into a common subspace where the discrepancy is minimized. Then heterogeneous face images can be matched directly in this common subspace. *However, the projection procedure generally causes the information loss which decreases the recognition performance.* With the development of deep learning, researchers pay more attention to the synthesis-based and common space-based HFR algorithms.

14.1.1.1 Synthesis-Based Methods

The aim of synthesis based HFR is to synthesize different modality images first, and then recognize identity. The illustration of different categories in synthesis based heterogeneous face methods is shown in Fig. 14.3. Existing synthesis-based HFR can be divided into exemplar-

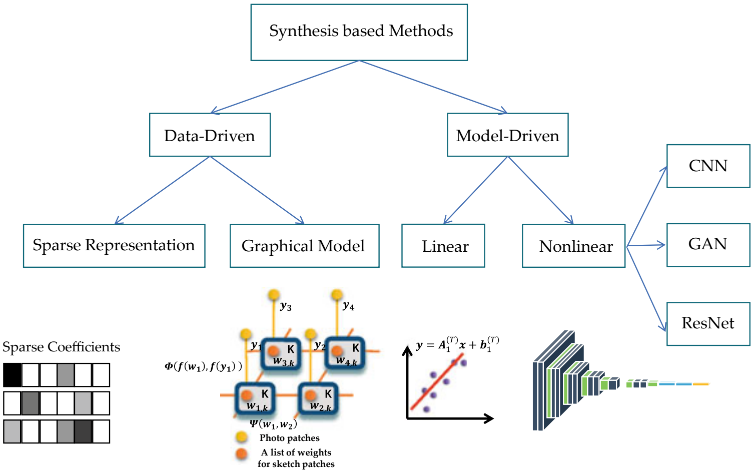


Fig. 14.3 The illustration of different categories in synthesi-based heterogeneous face methods

based methods (also called data-driven-based methods) and regression-based methods (also called model-driven-based methods).

Exemplar-based methods can be further roughly categorized into sparse representation based methods and graphic model based methods. Note that more category descriptions can be found in [47, 75]. Tang and Wang [56] pioneered the exemplar-based approach by computing a global eigen transformation for synthesizing face sketches from face photos. However, whole face photos and face sketches cannot be simply explained by a linear transformation, especially when the hair region is considered. [35] presented a patch-based approach with the idea of locally linear approximating global nonlinear. It represents each target sketch patch by a linear combination of some candidate sketch patches in the training set using locally linear embedding (LLE). The drawback is that each patch was independently synthesized and thus compatible relationship between the neighboring image patches was neglected. In order to tackle this problem, [60] employed multiscale Markov Random Field (MRF) to introduce probabilistic relationships between neighboring image patches. Their method synthesizes a face sketch by selecting the "best" candidate patches that maximize the a posteriori estimation of their MRF model. The weakness is that it can not synthesize new patches that do not exist in the training set and its optimization is NP hard. Zhou et al. [73] presented a Markov Weight Fields (MWF) model to improve the aforementioned problem by introducing the linear combination into the MRF. Wang et al. [58] proposed a Bayesian framework that provided an interpretation to existing face sketch synthesis methods from a probabilistic graphical view. Sparse representation [67] was also applied to face sketch synthesis to compute the weight combination. Wang et al. [63] proposed to adaptively determine the number of nearest neighbors by sparse representation and [12] presented a sparse-representation-based enhancement strategy to enhance the quality of the synthesized photos and sketches. Most of the **exemplar-based methods** have high computational complexity. To solve this problem, [53] proposed a real-time face sketch synthesis method by considering face sketch synthesis as an image denoising problem with the aid of GPU. Wang et al. [57] employ offline random sampling in place of online K-NN search to improve the efficiency of neighbor selection. Locality constraint is introduced to model the distinct correlations between the test patch and random sampled patches.

Regression-based methods have won more and more attention recently profiting from its real-time speed and end-to-end property. Chang et al. [4] adopted kernel ridge regression to synthesize face sketch patches from face photos. Zhang et al. [70] proposed to use support vector regression to express the high-frequency mappings between photo patches and sketch patches. Zhu and Wang [74] adopted a divide and conquer strategy. Photo-sketch patch pairs are firstly divided into many different clusters, each of which is equipped with a ridge regression model to learn the mapping between photo patches and sketch patches. CNN has greatly promoted the development of the nonlinear regression model [67]. Zhang et al. [69] adopted a fully convolutional network (FCN) to directly model the complex nonlinear mapping between face photos and sketches. Liu et al. [32] proposed the novel iterative local re-ranking algorithm to process diverse synthesis faces, which are generated by different attributes clues. More deep learning based synthesis methods description are shown in [75]. Most of this kind of methods are inspired by the deep generative model architectures.

14.1.1.2 Feature Descriptor-Based Methods

Klare et al. [23] proposed a local feature-based discriminant analysis (LFDA) framework through scale invariant feature transform (SIFT) feature [39] and multi-scale local binary pattern (MLBP) features [45]. A face descriptor based on coupled information-theoretic encoding was designed for matching face sketches with photos by Zhang et al. [71]. The coupled information-theoretic projection tree was introduced and was further extended to the randomized forest with different sampling patterns. Another face descriptor called local radon binary pattern (LRBP) was proposed in [8]. The face images were projected onto the radon space and encoded by local binary patterns (LBP). A histogram of averaged oriented gradients (HAOG) face descriptor was proposed to reduce the modality difference [10]. Lei et al. [26] proposed a discriminant image filter learning method that benefitted from LBP like face representation for matching NIR to VIS face images. Alex et al. [1] proposed a local difference of Gaussian binary pattern (LDoGBP) for face recognition across modalities. Bhatt et al. [3] proposed a discriminative approach for matching forensic sketches to mug shots employing multi-scale circular Weber's local descriptor (MCWLD) and an evolutionary memetic optimization algorithm. Klare and Jain [22] represented heterogeneous face images through their nonlinear kernel similarities to a collection of prototype face images. Liao et al. [29] firstly utilized a difference of Gaussian filter for matching heterogeneous images. Considering the fact that many law enforcement agencies employ facial composite software to create composite sketches, Han et al. [13] proposed a component-based approach for matching composite sketches to mug shot photos. Liu et al. [37] further fuse different face components discriminative information to boost recognition performance.

With the development of deep learning, deep feature extraction-based methods have drawn more and more attention. Mittal et al. [44] presented a transfer learning-based representation method. Lu et al. [40] proposed an unsupervised feature learning method that learns features from raw pixels. Wu et al. [66] utilized the nuclear norm constraint to increase the correlation between two modalities. The Wasserstein distance is utilized to learn invariant features for NIR-VIS face recognition [15]. This kind of method would be utilized with high computational complexity. Note that more algorithms descriptions of feature-based HFR could be found in [31, 47].

14.1.1.3 Common Space-Based Methods

In order to minimize the intra-modality differences, Lin and Tang [30] proposed a common discriminant feature extraction (CDFE) approach to map heterogeneous features into a common feature space. The canonical correlation analysis (CCA) was applied to learn the correlation between NIR and VIS face images by Yi *et al.* [68]. Lei and Li [25] proposed a subspace learning framework for heterogeneous face matching, which is called coupled spectral regression (CSR). They later improved the CSR by learning the projections based on all samples from all modalities [27]. Sharma and Jacobs [51] used partial least squares (PLS) to linearly map images from different modalities to a common linear subspace. A cross modal metric learning (CMML) algorithm was proposed by Mignon and Jurie [43] to learn a discriminative latent space. Both the positive and negative constraints were considered in metric learning procedure. Kan et al. [19] proposed a multi-view discriminant analysis (MvDA) method to obtain a discriminant common space for recognition. The correlations from both inter-view and intra-view were exploited.

Nowadays, the deep learning model could also be regarded as the nonlinear common space projection in the field. Sharma and Jacobs [52] proposed the partial least squares algorithm to learn the linear mapping between different face modalities. A multi-view discriminant analysis (MvDA) method [20] was proposed to exploit both inter-view and intra-view correlations of heterogeneous face images. He et al. [14] proposed an invariant deep representation approach to map different modalities of images into a common space. Liu et al. [38] directly utilized the deep learning model as the mapping function, which is also integrated with the extra semantic attribute information. Yet the projection procedure may lose some discriminative information.

14.2 Feature Descriptor-Based HFR

As mentioned in the above section, the feature descriptor-based methods aim to extract robust face feature descriptors according to the property of heterogeneous faces. Considering this kind of HFR algorithm always focuses on the capture specific cross-modality characteristics in the image level, these feature descriptor-based methods would not directly choose deep learning network as the feature extraction. Although feature descriptor-based HFR consume much computational resources, they achieve good performance in recognition task with good generalization, even in cross-datasets evaluation experiments.

To further state details of feature descriptor-based methods, we choose the representative HFR to show more details here. The graphical representation-based HFR (G-HFR) is proposed by researchers in Xidian University [47]. The framework of G-HFR is shown in Fig. 14.4. The key components are the suitable heterogeneous face representation extraction and similarity score. Here we take face sketch-photo recognition as an example to describe the proposed method, which could be easily extend to other HFR scenarios. To effectively represent the cross-modality spatial information in both sketches and photos, the representation dataset composed of face sketch-photo pairs is constructed to extract the graphical features of the gallery and probe images. For convenience of descriptions, we denote the representation dataset with M face sketch-photo pairs $\{(\mathbf{s}^1, \mathbf{p}^1), \cdots, (\mathbf{s}^M, \mathbf{p}^M)\}$, we first divide each face image into N overlapping patches. The probe sketch $\mathbf{t}$ and the gallery photos $\{\mathbf{g}^1, \cdots, \mathbf{g}^L\}$ are also divided into N overlapping patches. Here L denotes the number of photos in the gallery. For a probe sketch patch $\mathbf{y}_i (i = 1, 2, \cdots, N)$, we can find K nearest sketch patches from the sketches in the representation dataset within the search region around the location of $\mathbf{y}_i$. The probe sketch patch $\mathbf{y}_i$ can then be regarded as a linear combination of the K nearest sketch patches $\{\mathbf{y}_{i,1}, \cdots, \mathbf{y}_{i,K}\}$ weighted by a column

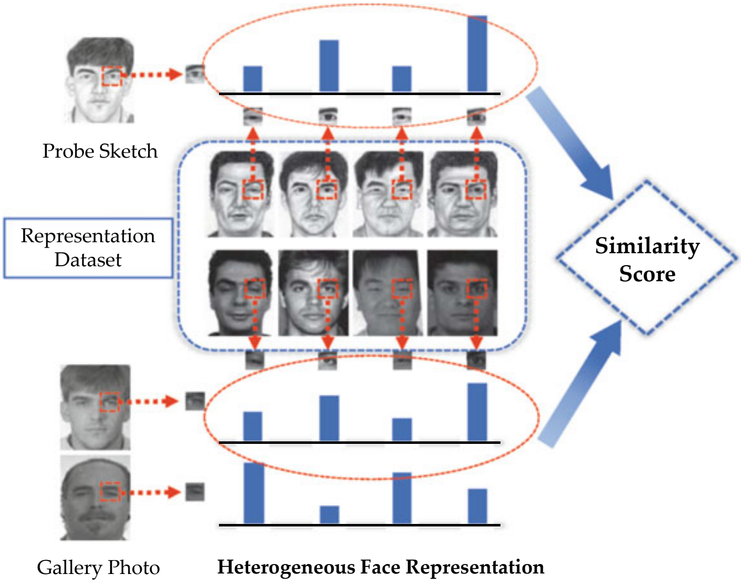


Fig. 14.4 Overview of the proposed graphical representation-based heterogeneous face recognition

vector $\mathbf{w}_{\mathbf{y}_i} = (w_{\mathbf{y}_{i,1}}, \cdots, w_{\mathbf{y}_{i,K}})^T$. As shown in Fig. 14.4, these learned weight vector $\mathbf{w}_{\mathbf{y}_i}$ is regarded as the extracted feature of the probe sketch patch $\mathbf{y}_i$. Similarly, for a gallery photo patch $\mathbf{x}_i^l$ from the l-th gallery photo $\mathbf{g}^l$, where $l = 1, 2, \cdots, L$, we can also find K nearest photo patches from the photos in the representation dataset and reconstruct the photo patch by a linear combination of these K nearest photo patches weighted by $\mathbf{w}_{\mathbf{x}_i^l}$. Thus, the weight vector $\mathbf{w}_{\mathbf{x}_i^l}$ is regarded as the extracted feature of the gallery photo patch $\mathbf{x}_i^l$. More details could be shown in [47]. Once effective graphic face features are extracted, how to calculate similarities becomes important for the performance evaluation. The following subsection shows the designed similarity measurement.

14.2.1 Graphical Representation

Here we introduce algorithm details of the graphical representation extraction. The Markov network is utilized to model all patches from a probe sketch or from gallery photos (as shown in Fig. 14.5).

The joint probability of the probe sketch patches and the weights is defined as,

$$\begin{aligned} & p(\mathbf{w}_{\mathbf{y}_1}, \cdots, \mathbf{w}_{\mathbf{y}_N}, \mathbf{y}_1, \cdots, \mathbf{y}_N) \\ = & \prod_i \Phi(\mathbf{f}(\mathbf{y}_i), \mathbf{f}(\mathbf{w}_{\mathbf{y}_i})) \prod_{(i,j)\in\Xi} \Psi(\mathbf{w}_{\mathbf{y}_i}, \mathbf{w}_{\mathbf{y}_j}), \end{aligned} \tag{14.1}$$

where $(i, j) \in \Xi$ denotes that the i-th probe sketch patch and the j-th probe sketch patch are adjacent. Ξ represents the edge set in the sketch layer of the Markov networks. $\mathbf{f}(\mathbf{y}_i)$ means the feature extracted from the probe sketch patch $\mathbf{y}_i$ and $\mathbf{f}(\mathbf{w}_{\mathbf{y}_i})$ denotes the linear combination of features extracted from neighboring sketch patches in the representation dataset, *i.e.* $\mathbf{f}(\mathbf{w}_{\mathbf{y}_i}) = \sum_{k=1}^{K} w_{\mathbf{y}_{i,k}} \mathbf{f}(\mathbf{y}_{i,k})$. $\Phi(\mathbf{f}(\mathbf{y}_i), \mathbf{f}(\mathbf{w}_{\mathbf{y}_i}))$ is the local evidence function, and $\Psi(\mathbf{w}_{\mathbf{y}_i}, \mathbf{w}_{\mathbf{y}_j})$ is the neighboring compatibility function.

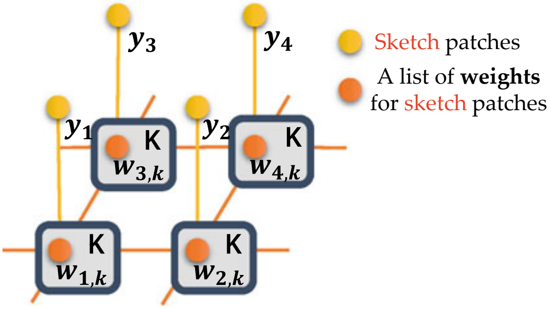


Fig. 14.5 The illustration of the graphical representation in the introduced G-HFR method [47]

The local evidence function $\Phi(\mathbf{f}(\mathbf{y}_i), \mathbf{f}(\mathbf{w}_{\mathbf{y}_i}))$ is defined as,

$$\begin{aligned} &\Phi(\mathbf{f}(\mathbf{y}_i), \mathbf{f}(\mathbf{w}_{\mathbf{y}_i})) \\ \propto\ &\exp\{-\|\mathbf{f}(\mathbf{y}_i) - \sum_{k=1}^{K} w_{\mathbf{y}_{i,k}} \mathbf{f}(\mathbf{y}_{i,k})\|^2 / 2\delta_\Phi^2\}. \end{aligned} \tag{14.2}$$

The rationale behind the local evidence function is that $\sum_{k=1}^{K} w_{\mathbf{y}_{i,k}} \mathbf{f}(\mathbf{y}_{i,k})$ should be similar to $\mathbf{f}(\mathbf{y}_i)$. Then the weight vector $\mathbf{w}_{\mathbf{y}_i}$ is regarded as a representation of the probe sketch patch $\mathbf{y}_i$.

The neighboring compatibility function $\Psi(\mathbf{w}_{\mathbf{y}_i}, \mathbf{w}_{\mathbf{y}_j})$ is defined as,

$$\begin{aligned} &\Psi(\mathbf{w}_{\mathbf{y}_i}, \mathbf{w}_{\mathbf{y}_j}) \\ \propto\ &\exp\{-\|\sum_{k=1}^{K} w_{\mathbf{y}_{i,k}} \mathbf{o}_{i,k}^{j} - \sum_{k=1}^{K} w_{\mathbf{y}_{j,k}} \mathbf{o}_{j,k}^{i}\|^2 / 2\delta_\Psi^2\}, \end{aligned} \tag{14.3}$$

where $\mathbf{o}_{i,k}^{j}$ represents the vector consisting of intensity values extracted from the overlapping area (between the i-th probe sketch patch and the j-th probe sketch patch) in the k-th nearest sketch patch of the i-th probe sketch patch. The neighboring compatibility function is utilized to guarantee that neighboring patches have compatible overlaps. The details of maximizing the joint probability function (14.1) are shown in [47].

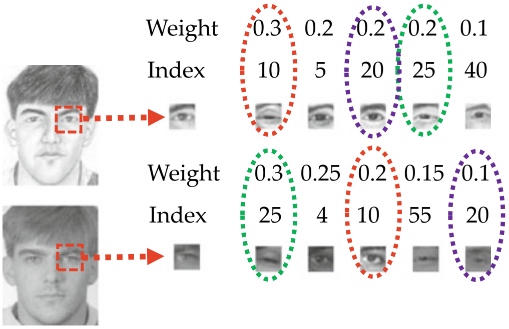
14.2.2 Similarity Metric

The suitable similarity measurement is designed for the mentioned graphic heterogeneous face features. Here we denote the extracted features of the sketch and photo as $\mathbf{W_t}$ and $\mathbf{W}_{\mathbf{g}^l}$. Then, the similarity score of each coupled patch pair is calculated. Researchers find the characteristics of the proposed graphical representation, *i.e.*, two graphical representations corresponding to the same position in coupled heterogeneous face images share similar semantic meanings. For example, $w_{\mathbf{y}_{i,z}}$ and $w_{\mathbf{x}_{i,z}^l}$ represent the weights of the sketch patch and photo patch from the z-th ($z = 1, 2, \cdots, M$) sketch-photo pair in the representation dataset. Thus, the weights, which share the same neighbors in the extracted features, are used to calculate similarity scores.

As shown in Fig. 14.6, the similarity score between the probe sketch patch $\mathbf{y}_i$ and the gallery photo patch $\mathbf{x}_i^l$ is calculated as the sum of the weights sharing the same nearest neighbors.

$$s(\mathbf{y}_i, \mathbf{x}_i^l) = 0.5 \sum_{z=1}^{M} n_z (w_{\mathbf{y}_{i,z}} + w_{\mathbf{x}_{i,z}^l}), \tag{14.4}$$

Fig. 14.6 The illustration of the similarity metric in G-HFR algorithm [47]



where

$$n_z = \begin{cases} 1, & w_{\mathbf{y}_{i,z}} > 0 \quad \text{and} \quad w_{\mathbf{x}^l_{i,z}} > 0 \\ 0, & \text{otherwise.} \end{cases}$$

The average of the similarity scores on all patch positions can be regarded as the final similarity score between the probe sketch and the gallery photo. For better understanding, we give an example in Fig. 14.6. When the weights and the indexes are given for probe sketch and gallery photo, the similarity score is calculated by $Score = 0.5 \times (0.2 + 0.1 + 0.3 + 0.2 + 0.3 + 0.2) = 0.65$.

14.3 Face Synthesis-Based HFR

Face synthesis-based methods are also an important branch of HFR. This kind of HFR often consists of two necessary steps: (1) synthesizing different modality faces into the same modality images; (2) recognizing the synthesis of homogeneous face identities. The advantage of synthesis-based HFR is that the model inference is visualized because the quality of synthesis faces could affect the following recognition performance. Thus, the goal of synthesis-based HFR is to synthesize high-quality face images.

Here we choose the representative heterogeneous face synthesis method and roughly describe the algorithm details. The probabilistic graphical model-based face synthesis method (DPGM) is proposed [75] to generate high-quality reconstructed images, even in poor light variations and cluttered backgrounds. The framework is shown in Fig. 14.7. The key components are as follows: (1) deep patch representation extraction; (2) candidate patch selection; (3) deep graphical representation learning. Firstly, the deep learning model is utilized to extract deep patch representations for test photos and all training sketches. And then, the patch selection strategy is designed to select candidate sketch patches for each test photo patch. The deep graphical representation learning is introduced to obtain the best weights for sketch patch reconstruction. Finally, each reconstructed sketch patch is obtained by weighted recombining the candidate sketch patches.

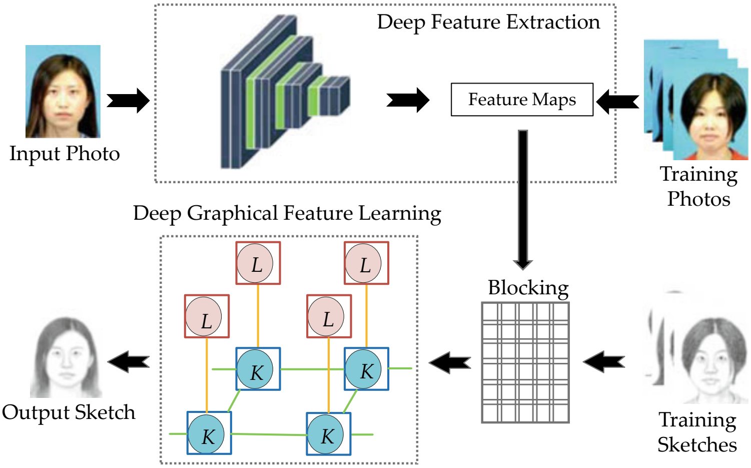


Fig. 14.7 The overall framework of the introduced probabilistic graphical model-based face synthesis method [75]

14.3.1 Deep Patch Representation Extraction

With the development of deep generative model, researchers naturally want to choose deep learning model to extract better face representations. Here the training dataset includes M face photo-sketch pairs and a test photo t. The deep collaborative networks are trained with M face photo-sketch pairs. The aim of the deep collaborative networks is to learn two opposite mappings: $G : photo \to sketch$ and $F : sketch \to photo$. Researchers hope that the learned model can help us map the test photo and training sketches into uniform deep image representations. The more details of the network architecture are shown are [75].

Assuming $\mathbf{t}_i$ is the i-th test photo patch, where $i = 1, 2, ..., N$. The designed deep patch representation of the test photo patch can be represented as the linear combination of feature maps weighted by vector $\mathbf{u}_i$, which is denoted as,

$$\mathrm{D}(\mathbf{t}_i) = \sum_{l=1}^{L} u_{i,l} \mathrm{d}_l(\mathbf{t}_i), \tag{14.5}$$

where $\mathrm{d}_l(\mathbf{t}_i)$ refers to l-th feature map of patch $\mathbf{t}_i$. $u_{i,l}$ denotes the weight of l-th feature map, $l = 1, 2, ..., L$, and $\sum_{l=1}^{L} u_{i,l} = 1$. Note that the weights for deep patch representation at different locations in photo patches are different, and each feature map has a uniform initial weight.

14.3.2 Candidate Patch Selection

The test photo patch is denoted as $\mathbf{t}_i$ and its deep patch representation is denoted as $\mathrm{D}(\mathbf{t}_i)$. The aim is to find K training sketch patches $\{\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, ..., \mathbf{y}_{i,K}\}$ that most like $\mathbf{t}_i$ according to the Euclidean distance of representation within the search region around the location of $\mathbf{t}_i$ as the candidate sketch patches for reconstruction. Once the deep representation model is trained, we could directly map the test photo patch and training sketch patches into deep patch representations. Figure 14.8 shows the difference between the matching strategy of DPGM and other former methods. More accurate candidate sketch patches can be selected directly and better weight combination for sketch patch reconstruction can be obtained.

Thus, the target sketch patch $\mathbf{y}_i$ can be synthesized by the linear combination of K candidate sketch patches weighted by the K-dimensional vector $\mathbf{w}_i$:

$$\mathbf{y}_i = \sum_{k=1}^{K} w_{i,k}\mathbf{y}_{i,k}, \tag{14.6}$$

where $w_{i,k}$ denotes the weight of the k-th candidate sketch and $\sum_{k=1}^{K} w_{i,k} = 1$.

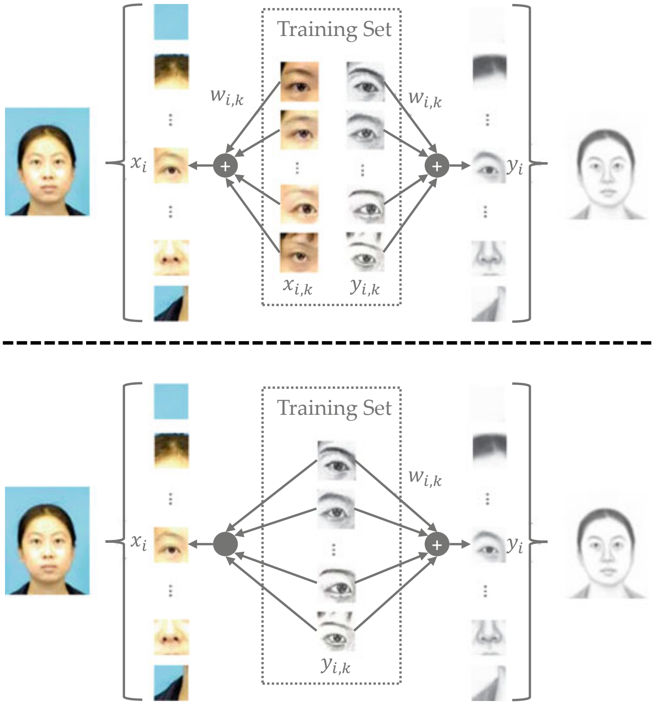


Fig. 14.8 Difference between the matching strategy of DPGM and other similar related algorithm

14.3.3 Deep Graphical Representation Learning

Similar with [47], researches also introduce the graphical model to model the distribution over the weights for deep patch representation and the distribution over the weights for sketch patch reconstruction. Given the observed variables $\mathbf{t}_i$, the model need to infer weights for deep patch representation $\mathbf{u}_i$ and weights for sketch patch reconstruction $\mathbf{w}_i$. $\mathbf{u}_i$ determines the representation ability of deep patch representation. Note that our reconstruction method is conducted at patch level and what we ultimately needed is optimal reconstruction weights for candidate sketch patches at different spatial locations. The joint probability of $\mathbf{u}_i$, $\mathbf{w}_i$, and $\mathbf{t}_i$ $\forall i \in \{1, ..., N\}$, is formulated as:

$$\begin{aligned} & p(\mathbf{t}_1, ..., \mathbf{t}_N, \mathbf{u}_1, ..., \mathbf{u}_N, \mathbf{w}_1, ..., \mathbf{w}_N) \\ & \propto \prod_{i=1}^{N} \Phi(\mathbf{t}_i, \mathbf{u}_i, \mathbf{w}_i) \prod_{(i,j)\in\Xi} \Psi(\mathbf{w}_i, \mathbf{w}_j) \prod_{i=1}^{N} \Upsilon(\mathbf{u}_i), \end{aligned} \tag{14.7}$$

where $\Phi(\mathbf{t}_i, \mathbf{u}_i, \mathbf{w}_i)$ is the local evidence function:

$$\begin{aligned} & \Phi(\mathbf{t}_i, \mathbf{u}_i, \mathbf{w}_i) \\ & = \exp\{-\sum_{l=1}^{L} u_{i,l} \| \mathrm{d}_l(\mathbf{t}_i) - \sum_{k=1}^{K} w_{i,k} \mathrm{d}_l(\mathbf{y}_{i,k}) \|^2 / 2\delta_D^2\}. \end{aligned} \tag{14.8}$$

and $\Psi(\mathbf{w}_i, \mathbf{w}_j)$ is the neighboring compatibility function:

$$\begin{aligned} & \Psi(\mathbf{w}_i, \mathbf{w}_j) \\ & = \exp\{-\| \sum_{k=1}^{K} w_{i,k} \mathbf{o}_{i,k}^{j} - \sum_{k=1}^{K} w_{j,k} \mathbf{o}_{j,k}^{i} \|^2 / 2\delta_S^2\}. \end{aligned} \tag{14.9}$$

and $\Upsilon(\mathbf{u}_i)$ is the regularization function:

$$\Upsilon(\mathbf{u}_i) = \exp\{-\lambda \|\mathbf{u}_i\|^2\}. \tag{14.10}$$

Here $\mathrm{d}_l(\mathbf{t}_i)$ means the l-th feature map of patch $\mathbf{t}_i$ and $\mathrm{d}_l(\mathbf{y}_{i,k})$ means the l-th feature map of k-th candidate patch. $(i, j) \in \Xi$ means the i-th and j-th patches are neighbors. $\mathbf{o}_{i,k}^{j}$ represents the overlapping area between the candidate sketch patch $\mathbf{y}_{i,k}$ and the j-th patch. λ balances the regularization term with the other two terms. The posterior probability can be written as:

$$\begin{aligned} & p(\mathbf{u}_1, ..., \mathbf{u}_N, \mathbf{w}_1, ..., \mathbf{w}_N | \mathbf{t}_1, ..., \mathbf{t}_N) \\ & = \frac{1}{Z} p(\mathbf{t}_1, ..., \mathbf{t}_N, \mathbf{u}_1, ..., \mathbf{u}_N, \mathbf{w}_1, ..., \mathbf{w}_N), \end{aligned} \tag{14.11}$$

where $Z = p(\mathbf{t}_1, ..., \mathbf{t}_N)$ is a normalization term. The details of maximizing the posterior probability are shown in [75].

14.4 Common Space-Based HFR

In this section, we introduce the common space-based HFR methods. As mentioned above, the common space based HFR is developed rapidly due the superior performance of deep learning. Due to the different data distributions of heterogeneous face images, traditional face recognition couldn't be directly applied to identify heterogeneous faces. Thus, the common space based algorithm aims to design the mapping function to project faces of different modalities into a common space to reduce modality gap. Nowadays, more and more researchers find the deep learning model could be utilized to learn the cross-modality mapping nonlinear function.

To expound the algorithm details more clearly, we introduce one representative HFR method here. The heterogeneous face interpretable representation method (HFIDR) is proposed [31] to learn the suitable mapping function for projecting input heterogeneous faces. As known to all, face sketches are generated according to the description of eyewitnesses when photos of the suspect are limited. Thus, these indeed exist shape exaggerations and distortions in face sketches compared with photos. However, humans can easily recognize the identity according to a distortional sketch, rather than learning from enough sketches. Inspired by the specific generation procedure, researchers aim to learn the latent identity information in heterogeneous faces. Furthermore, the interpretable disentangled face representation is designed where each dimension could contain reasonable meaning and acquire latent identity information. The framework of HFIDR is shown in Fig. 14.9. It is noted that we take face sketches recognition and synthesis as an example to describe the proposed method, which would be generalized to other heterogeneous face scenarios.

14.4.1 Network Architecture

The input face sketches and photos are separately denoted as $\{s_i\}_{i=1}^{N}$ and $\{p_j\}_{j=1}^{N}$, where N is the number of face images. Given one input sketch s_i and photo p_j, we constrain these two images belong to different identities. Note that the proposed interpretable representation should contain the modality part, the identity part and the redundant part. More explanations are described in [31]. Here the modality part $v_{\text{mod}} \in R^{N_{\text{mod}}}$ is an one-hot vector. N_{mod} refers to the number of different modalities of heterogeneous face images. The encoder model G_{enc} is utilized to encode the identity information $v_{id} \in R^{N_{id}}$, which represents the heterogeneous face identity relevant information. Therefore, the designed interpretable disentangled representations are separately denoted as follows:

$$z_{s_i} = [v^{s}_{\text{mod}}, v^{i}_{id}, v_{noise}], \tag{14.12}$$

$$z_{p_j} = [v^{p}_{\text{mod}}, v^{j}_{id}, v_{noise}]. \tag{14.13}$$

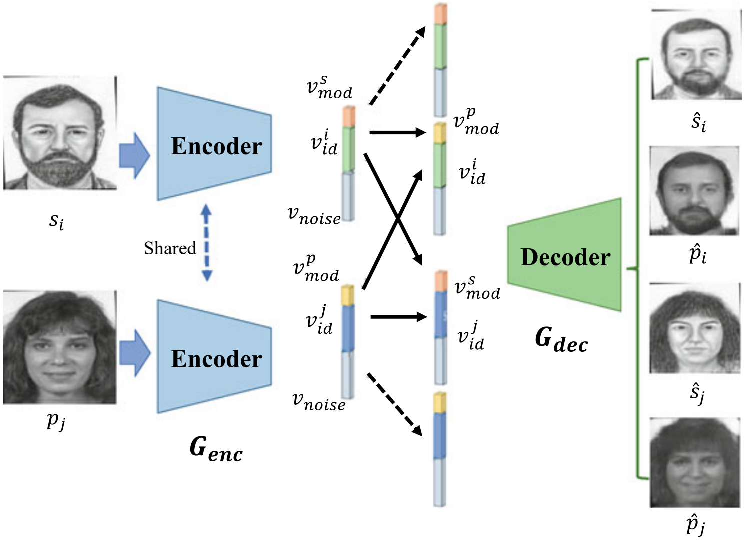


Fig. 14.9 Overview of the introduced heterogeneous face interpretable disentangled representation method [31]

Later, researchers choose the exchange strategy to force the encoder model learn the latent modality-invariant identity information. Thus, these two new recombined representations are denoted by z_{s_j} and z_{p_i},

$$z_{s_j} = [v^s_{\text{mod}}, v^j_{id}, v_{noise}], \tag{14.14}$$

$$z_{p_i} = [v^p_{\text{mod}}, v^i_{id}, v_{noise}]. \tag{14.15}$$

The decoder model G_{dec} generates two reconstructed images and the following two recombined images. Inspired by the adversarial training strategy [18], researchers additionally design two symmetry discriminators to improve synthesis quality. The first conditional discriminator D_s learns to classify between fake tuples (input photo p_j and synthesized sketch $\hat{s}_j$) and real tuples (input photo p_j and reference sketch s_j). On the contrary, the other conditional discriminator D_p learns to classify between fake tuples (input sketch s_i and synthesized photo $\hat{p}_i$) and real tuples (input sketch s_i and reference photo p_i). The generator G_{enc} and G_{dec} learn to fool these discriminators. More details of these discriminators architectures are shown in [31].

14.4.2 Loss Function

Here we introduce the designed loss function when training HFIDR model. The reconstruction loss is used to reconstruct face sketches and photos as follows:

$$\begin{aligned} L_{recon} = \mathrm{E}[\left\|s_i - G_{dec}(z_{s_i})\right\| + \left\|s_j - G_{dec}(z_{s_j})\right\| \\ + \left\|p_i - G_{dec}(z_{p_i})\right\| + \left\|p_j - G_{dec}(z_{p_j})\right\|]. \end{aligned} \tag{14.16}$$

For improve the synthesis performance, these two adversarial losses are formulated as

$$\begin{aligned} L^s_{adv}(G_{enc}, G_{dec,} D_p) = \mathrm{E}[\log(D_s(p_j, s_j))] \\ + \mathrm{E}[log(1 - D_s(p_j, G_{dec}(z_{s_j})))], \end{aligned} \tag{14.17}$$

$$\begin{aligned} L^p_{adv}(G_{enc}, G_{dec,} D_p) = \mathrm{E}[\log(D_p(s_i, p_i))] \\ + E[log(1 - D_p(s_i, G_{dec}(z_{p_i})))], \end{aligned} \tag{14.18}$$

where L^s_{adv} forces the synthesized sketch $\hat{s}_j$ to be closer to the distribution of face sketches, and L^p_{adv} forces the synthesized photo $\hat{p}_i$ to be closer to the distribution of face photos.

Additionally, the simple softmax loss is employed to recognize different identities, which is formulated as

$$\begin{aligned} L_{id} = \mathrm{E}[-log(p(y_i|G_{dec}(z_{s_i}))) - log(p(y_i|G_{dec}(z_{p_i}))) \\ - log(p(y_j|G_{dec}(z_{s_j}))) - log(p(y_j|G_{dec}(z_{p_j})))], \end{aligned} \tag{14.19}$$

where $y_{i,}y_j \in \{y_k\}^M_{k=1}$. M denotes the number of face identities. Synthesized face images $G_{dec}(z_{s_i})$ and $G_{dec}(z_{p_i})$ belong to the identity class y_i, synthesized images $G_{dec}(z_{s_j})$ and $G_{dec}(z_{p_j})$ belong to the identity class y_j. Similarly, we also utilize simple binary classifier to distinguish different modalities as follows:

$$\begin{aligned} L_{\mathrm{mod}} = \mathrm{E}[-log(p(m_s|G_{dec}(z_{s_i}))) - log(p(m_s|G_{dec}(z_{s_j}))) \\ - log(p(m_p|G_{dec}(z_{p_i}))) - log(p(m_p|G_{dec}(z_{p_j})))], \end{aligned} \tag{14.20}$$

where $m_{s,}m_p \in \{0, 1\}$. Here $p(m_s|G_{dec}(z_{s_i}))$ and $p(m_s|G_{dec}(z_{s_j}))$ is the predicted probabilities of modalities of generated images are sketches, $p(m_p|G_{dec}(z_{p_i}))$ and $p(m_p|G_{dec}(z_{p_j}))$ is the predicted probabilities of modalities of generated images are photos.

The final total objective function is weighted of four mentioned loss terms:

$$\begin{aligned} L_{total} = \lambda_{recon} L_{recon} + \lambda_{adv}(L^s_{adv} + L^p_{adv}) \\ + \lambda_{id} L_{id} + \lambda_{\mathrm{mod}} L_{\mathrm{mod}}, \end{aligned} \tag{14.21}$$

where parameters λ_{recon}, λ_{adv}, λ_{id}, and λ_{mod} balance the contribution of four loss terms.

14.5 Experiments

14.5.1 Databases

In this subsection, we would introduce some typical public heterogeneous face datasets. Example face images are shown in Fig. 14.10. It noted that recent works mainly focus on recognize face sketches and NIR faces. However, HFR methods can be easily extend in other heterogeneous face scenarios.

The CUHK Face Sketch FERET (CUFSF) Database [72] contains 1194 subjects, with photos from the FERET database and face sketches are drawn by the artist. To mimic the real-world scenarios, the photos have more illumination variations and the sketches have more shape exaggerations in the CUFSF sketch database. The 500 sketch-photo pairs are selected as training set, and the rest pairs serve as the testing set.

PRIP Viewed Software-Generated Composite Database (PRIP-VSGC) [24] contains 123 subjects, with photos from the AR database [41] and composite sketches created using FACES [7] and Identi-Kit [17]. The composite sketches are created with facial composite software kits which synthesize a sketch by selecting a collection of facial components from candidate patterns. The 123 composite sketches generated using Identi-Kit software.

For NIR-VIS face images analysis, we conduct experiments on the Oulu-CASIA NIR-VIS database [5]. Oulu-CASIA NIR-VIS database contains 80 subjects, with 50 subjects from Oulu University and 30 subjects from CASIA. Each subject comprises six expressions. All images are aligned and cropped to 128x128 by five facial landmarks, which is the same with [65]. With the same protocol [65], we randomly select 20 subjects as the training set and 20 subjects as the testing set. Note that there exist 96 images for each subject, with 48 NIR images and 48 VIS images. In the test stage, we use the VIS images of 20 subjects as the gallery set and the NIR images of these subjects as the probe set.

The CASIA NIR-VIS 2.0 dataset is the challenging NIR-VIS dataset, with large cross-modality variations. This dataset contains 725 subjects, and these images are organized into

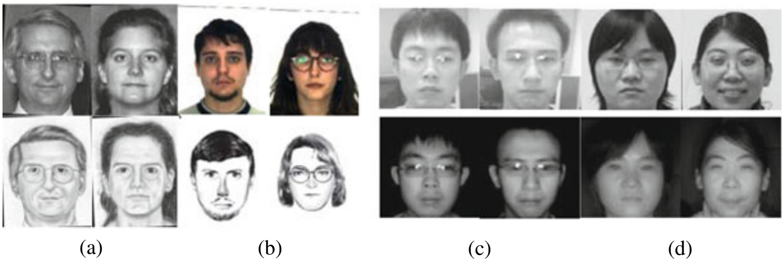


Fig. 14.10 The illustration of heterogeneous face databases. **a** the CUHK face sketch FERET (CUFSF) database. **b** PRIP-VSGC composite sketch database. **c** the Oulu CASIA NIR-VIS database. **d** the CASIA NIR-VIS 2.0 database

Table 14.1 Recognition accuracies of the State-of-the-Art methods, the introduced HFIDR method and G-HFR method on the CUFSF sketch database

Algorithm	Accuracy (%)	Synthesis task
Fisherface [2]	28.82	No
PLS [52]	51.00	No
P-RS [21]	83.95	No
LRBP [9]	91.12	No
VGG [46]	39.65	No
MvDA [20]	55.50	No
SeetaFace [36]	16.57	No
G-HFR [47]	96.04	No
SGR-DA [48]	96.97	No
HFIDR [31]	99.39	Yes

two views. View 1 is used for parameter tuning and view 2 is used for evaluation. We follow the protocols in [65] and evaluate our method with 10-fold experiments.

14.5.2 Heterogeneous Face Recognition Results

We compare the introduced HFIDR method (in Sect. 14.4) and G-HFR method (in Sect. 14.2) with representative related methods on the CUFSF sketch database as shown in Table 14.1. The two conventional face recognition methods (VGG and SeetaFace) achieve poor recognition performance because of the large modality discrepancy. The two common space-based methods (PLS and MvDA) only achieve 51.00% and 55.50% at rank-1. The rest modality invariant feature based methods can achieve better recognition performance. The HFIDR could further achieve 99.39% at rank-1, which indicates the discriminative latent identity information could be effectively captured.

As shown in Table 14.2, the DVR [65] method reduces the modality discrepancy and separately achieves 99.30% and 100% at rank-1 when the backbone is LigthCNN-9 and LightCNN-29. On the Oulu-CASIA NIR-VIS dataset, the mentioned HFDIR could achieve rank-1 accuracy of 100% with the help of the interpretable disentangled representation structure, even utilize the Lightcnn-9 as the encoder backbone. The experimental results demonstrate the effectiveness and robustness of the introduced method (details in Sect. 14.4) on extracting modality invariant identity information.

Table 14.2 Recognition accuracies of the State-of-the-Art methods, the introduced HFIDR method on the Oulu-CASIA NIR-VIS Database

Algorithm	Accuracy (%)	Synthesis task
KDSR [16]	66.90	No
P-RS [21]	62.20	No
H2(LBP3) [50]	70.80	No
TRIVET [33]	92.20	No
IDR [14]	94.30	No
ADFL [54]	95.50	No
CDL [66]	94.30	No
W-CNN [15]	98.00	No
DVR(LightCNN-9) [65]	99.30	No
DVR(LightCNN-29) [65]	100.00	No
HFIDR(LightCNN-9) [31]	100.00	Yes
HFIDR(LightCNN-29) [31]	100.00	Yes

Table 14.3 Recognition accuracies of the State-of-the-Art methods, the introduced HFIDR method on the CASIA NIR-VIS 2.0 Database

Algorithm	Accuracy (%)	Synthesis task
KDSR [16]	37.50	No
H2(LBP3) [50]	43.80	No
HFR-CNN [49]	85.90	No
TRIVET [33]	95.70	No
IDR [14]	97.30	No
ADFL [54]	98.20	No
CDL [66]	98.60	No
W-CNN [15]	98.70	No
DVR(LightCNN-9) [65]	99.10	No
DVR(LightCNN-29) [65]	99.70	No
HFIDR(LightCNN-9) [31]	87.48	Yes
HFIDR(LightCNN-29) [31]	98.64	Yes

On the CASIA NIR-VIS 2.0 dataset, the introduced HFIDR could achieve 87.48% on LighCNN-9, and 98.64% on LightCNN-29 at rank-1. It is because that HFIDR is more suitable for pairwise heterogeneous face scenarios [38], while there exist large intra-class cross-modality variations, like lighting, expression and pose, and unpaired data on the CASIA NIR-VIS 2.0 database. Additionally, experimental results demonstrated larger and more robust network can achieve better performance (Table 14.3).

14.5.3 Heterogeneous Face Synthesis Results

In this section, we mainly show the comparison experimental results of the introduced DPGM algorithm (in Sect. 14.3). The qualitative evaluation and quantitative evaluation experiments are conducted with related face synthesis methods for further analysis.

Qualitative evaluation The structural similarity index metric (SSIM) [62] is deployed to objectively evaluate the visually perceptual quality of the synthesized sketches by different methods on CUFS database and CUFSF database. The reference image is the original sketch drawn by artists while the distorted image is the synthesized sketch. The statistics of average SSIM scores on the CUFS database and the CUFSF database are shown in Fig. 14.11. The horizontal axis labels represent the SSIM score from 0 to 1. The vertical axis means the percentage of synthesized sketch, whose SSIM scores are not smaller than the score marked on the horizontal axis. Table 14.4 gives the average SSIM score on the CUFS database and the CUFSF database.

Quantitative evaluation Figure 14.12 shows some synthesized face sketches by different exemplar based methods on the CUFS database. As can be seen, blurring appeared in some dominant facial regions on the results from the LLE method, the MWF method, the SSD method, the SFS method and the SFS-SVR method. Synthesized results of the MRF method have some deformations and patch mismatch around the face region. Even the most recently proposed algorithms such as the Bayesian method and the RSLCR method are exist with aforementioned defects. The introduced DPGM method (in Sect. 14.3) performs well whether in facial details or in the hair and background area. The DPGM method can generate sketches with high visual quality and sharper edges.

Figure 14.13 shows some synthesized face sketches by different regression based methods on the CUFS database. As can be seen, the results of FCN are very blurry due to the poor representation ability of their network. Pix2pix model can generate images with sharper textures which can improve the visual perception quality. The results of the introduced DPGM method (in Sect. 14.3) tend to have high visual quality whether in normal or bad environment conditions.

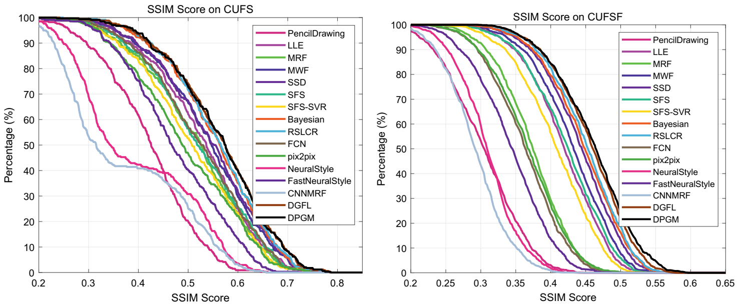


Fig. 14.11 Statistics of SSIM scores on the CUFS database and the CUFSF database

Table 14.4 Average SSIM score (%) on the CUFS database and the cufsf database

Methods	CUFS (%)	CUFSF (%)
PencilDrawing	42.07	30.33
LLE	52.58	41.76
MRF	51.32	37.24
MWF	53.93	42.99
SSD	54.20	44.09
SFS	51.90	42.11
SFS-SVR	50.59	40.62
Bayesian	55.43	44.54
RSLCR	55.72	44.96
FCN	52.14	36.22
pix2pix	49.39	36.65
NeuralStyle	39.50	30.89
FastNeuralStyle	47.41	34.27
CNNMRF	37.20	29.25
DGFL	**56.45**	**45.62**
DPGM	**56.39**	**46.00**

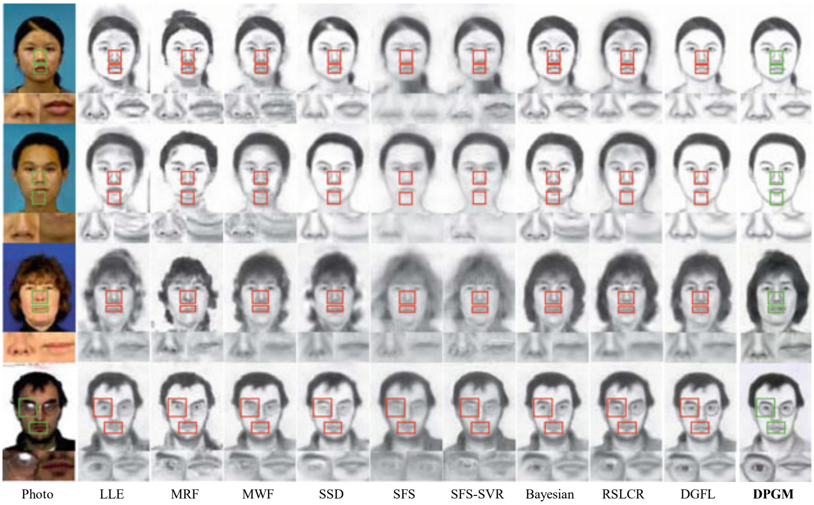


Fig. 14.12 Synthesized sketches on the CUFS database by examplar based methods (LLE, MRF, MWF, SSD, SFS, SFS-SVR, Bayesian, RSLCR, DGFL) and the introduced DPGM method (in Sect. 14.3)

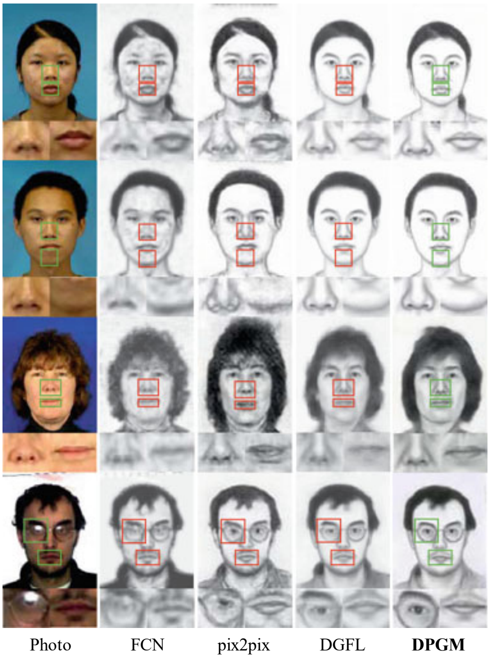


Fig. 14.13 Synthesized sketches on the CUFS database by regression based methods (FCN, pix2pix), the DGFL method and the introduced DPGM method (in Sect. 14.3)

Figure 14.14 shows some synthesized face sketches by different neural style transfer methods on the CUFS database. Because of the lack of structure information, the results generated by NeuralStyle possess extreme messy texture. By combining a Markov Random Field (MRF) and a CNN, CNNMRF is able to preserve some structure information. The results generated by FastNeuralStyle do not possess messy texture.

Figure 14.15 illustrates some face sketches synthesized by different methods on face photos with extreme lighting variance. Since deep patch representation is more robust to these noises than pixel intensity, the introduced DPGM method (in Sect. 14.3) can reconstruct sketches with virtually no distortion. This advantage is of vital importance in real world applications. More experimental results analysis can be found in [75].

14.6 Conclusion

Heterogeneous face recognition is still a challenging problem in biometric analysis and computer vision. Firstly, we defined the HFR as the difficult task in real-world scenarios and analyze existing problems. The comprehensive literature review is shown in Sect. 14.2. We further describe the advantages and disadvantages of the mentioned three kinds of HFR methods: (1) feature descriptor-based HFR; (2) synthesis-based HFR; (3) common space-based HFR. To clarify algorithm details clearly, we further introduced three representative

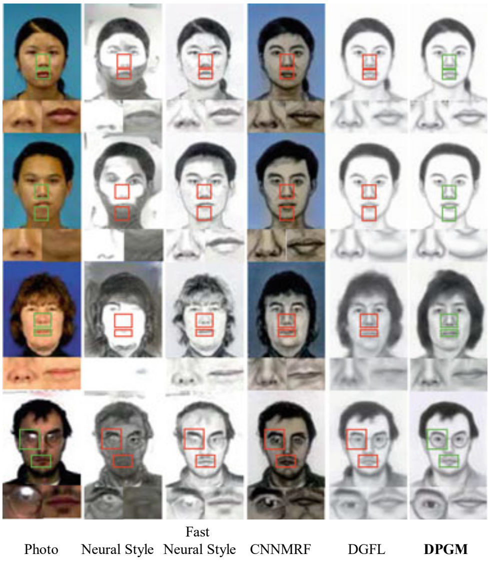


Fig. 14.14 Synthesized sketches on the CUFS database by nerual style transfer (NeuralStyle, Fast-NeuralStyle, CNNMRF), the DGFL method and the introduced DPGM method (in Sect. 14.3)

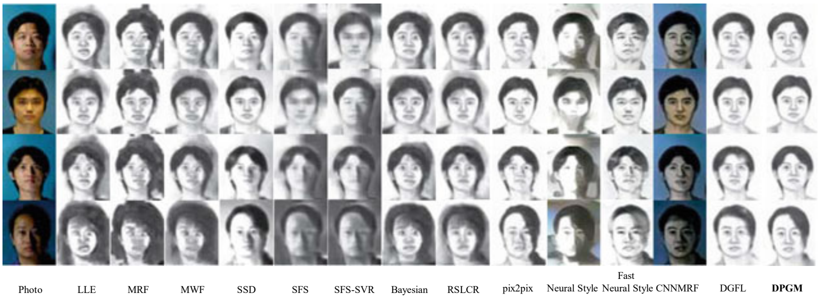


Fig. 14.15 Synthesized sketches of the photos with extreme lighting variance by different methods

HFR methods: the G-HFR method [47], the DPGM method [75] and the HFIDR method [31]. Moreover, experimental settings and results are also shown compared with other related HFR methods in Sect. 14.5. Finally, we conclude this chapter by presenting some possible future avenues of HFR task. *(1) More robust HFR models should be carefully designed to be suitable in multiple HFR scenarios: face sketch, NIR image, TIR image, low-resolution image, etc. (2) The interpretable HFR model should be explored in the future. It is because the HFR is often deployed in social security scenes, and how to generate the credible identity recognition is an interesting topic. (3) Considering the specific generation of face sketches, the visual human forgetting process should be explored and introduced in the HFR framework. We think the combination of computer vision and cognitive psychology would* *provide a better research approach. (4) For cross-modality face synthesis tasks, researchers should make generated faces become more similar in identity-level, but not only in pixel-level.* We hope this chapter will inspire more related works, and heterogeneous face recognition would draw more attention in the future.

References

1. Alex, A., Asari, V., Mathew, A.: Local difference of gaussian binary pattern: robust features for face sketch recognition. In: Proc. IEEE Int. Conf. Syst. Man & Cybern., pp. 1211–1216 (2013)
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)
3. Bhatt, H.S., Bharadwaj, S., Singh, R., Vatsa, M.: Memetically optimized MCWLD for matching sketches with digital face images. IEEE Trans. Inf. Forens. Security **5**(10), 1522–1535 (2012)
4. Chang, L., Zhou, M., Deng, X., Wu, Z., Han, Y.: Face sketch synthesis via multivariate output regression. In: International Conference on Human-Computer Interaction, pp. 555–561 (2011)
5. Chen, J., Yi, D., Yang, J., Zhao, G., Li, S.Z., Pietikainen, M.: Learning mappings for face synthesis from near infrared to visual light images. In: IEEE Conf. Comput. Vis. Pattern Recogn., pp. 156–163 (2009)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: IEEE Conf. Comput. Vis. Pattern Recogn., pp. 4690–4699 (2019)
7. FACES. [Online]. Available: http://www.iqbiometrix.com
8. Galoogahi, H., Sim, T.: Face sketch recognition by local radon binary pattern. In: Proc. IEEE Int. Conf. Image Process., pp. 1837–1840 (2012)
9. Galoogahi, H.K., Sim, T.: Face sketch recognition by local radon binary pattern: Lrbp. In: IEEE Int. Conf. Image Process., pp. 1837–1840 (2012)
10. Galoogahi, H., Sim, T.: Inter-modality face sketch recognition. In: Proc. IEEE Int. Conf. Multimedia & Expo, pp. 224–229 (2012)
11. Gao, X., Zhong, J., Li, J., Tian, C.: Face sketch synthesis using E-HMM and selective ensemble. IEEE Trans. Circuits Syst. Video Technol. **18**(4), 487–496 (2008)
12. Gao, X., Wang, N., Tao, D., Li, X.: Face sketch-photo synthesis and retrieval using sparse representation. IEEE Trans. Circuits Syst. Video Technol. **22**(8), 1213–1226 (2012)
13. Han, H., Klare, B., Bonnen, K., Jain, A.: Matching composite sketches to face photos: a component-based approach. IEEE Trans. Inf. Forens. Security **8**(1), 191–204 (2013)
14. He, R., Wu, X., Sun, Z., Tan, T.: Learning invariant deep representation for nir-vis face recognition. In: Proc. AAAI. Conf. Artificical Intell. (2017)
15. He, R., Wu, X., Sun, Z., Tan, T.: Wasserstein cnn: Learning invariant features for nir-vis face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **41**(7), 1761–1773 (2018)
16. Huang, X., Lei, Z., Fan, M., Wang, X., Li, S.Z.: Regularized discriminative spectral regression method for heterogeneous face matching. IEEE Trans. Image Process. **22**(1), 353–362 (2012)
17. Identi-Kit. [Online]. Available: http://www.identikit.net/
18. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conf. Comput. Vis. Pattern Recogn., pp. 1125–1134 (2017)
19. Kan, M., Shan, S., Zhang, H., Lao, S., Chen, X.: Multi-view discriminant analysis. In: Proc. Eur. Conf. Comput. Vis., pp. 808–821 (2012)
20. Kan, M., Shan, S., Zhang, H., Lao, S., Chen, X.: Multi-view discriminant analysis. IEEE Trans. Pattern Anal. Mach. Intell. **38**(1), 188–194 (2016)

21. Klare, B.F., Jain, A.K.: Heterogeneous face recognition using kernel prototype similarities. IEEE Trans. Pattern Anal. Mach. Intell. **35**(6), 1410–1422 (2013)
22. Klare, B.F., Jain, A.K.: Heterogeneous face recognition using kernel prototype similarities. IEEE Trans. Pattern Anal. Mach. Intell. **35**(6), 1410–1422 (2013)
23. Klare, B., Li, Z., Jain, A.: Matching forensic sketches to mug shot photos. IEEE Trans. Pattern Anal. Mach. Intell. **33**(3), 639–646 (2011)
24. Klum, S., Han, H., Klare, B., Jain, A.K.: The FaceSketchID system: Matching facial composites to mugshots. IEEE Trans. Inf. Forens. Security **9**(12), 2248–2263 (2014)
25. Lei, Z., Li, S.: Coupled spectral regression for matching heterogeneous faces. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 1123–1128 (2009)
26. Lei, Z., Yi, D., Li, S.: Discriminant image filter learning for face recognition with local binary pattern like representation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 2512–2517 (2012)
27. Lei, Z., Zhou, C., Yi, D., Jain, A., Li, S.: An improved coupled spectral regression for heterogeneous face recognition. In: Proc. Int. Conf. Biom., pp. 7–12 (2012)
28. Li, J., Hao, P., Zhang, C., Dou, M.: Hallucinating faces from thermal infrared images. In: Proc. IEEE Int. Conf. Image Process., pp. 465–468 (2008)
29. Liao, S., Yi, D., Lei, Z., Qin, R., Li, S.Z.: Heterogeneous face recognition from local structures of normalized appearance. In: Int. Conf. Biometrics, pp. 209–218. Springer (2009)
30. Lin, D., Tang, X.: Inter-modality face recognition. In: Proc. 9th Eur. Conf. Comput. Vis., pp. 13–26 (2006)
31. Liu, D., Gao, X., Peng, C., Wang, N., Li, J.: Heterogeneous face interpretable disentangled representation for joint face recognition and synthesis. In: IEEE Transactions on Neural Networks and Learning Systems (2021)
32. Liu, D., Gao, X., Wang, N., Peng, C., Li, J.: Iterative local re-ranking with attribute guided synthesis for face sketch recognition. Pattern Recognit. **109** (2021)
33. Liu, X., Song, L., Wu, X., Tan, T.: Transferring deep representation for nir-vis heterogeneous face recognition. In: Proc. Int. Conf. Biometrics, pp. 1–8 (2016)
34. Liu, Q., Tang, X., Jin, H., Lu, H., Ma, S.: A nonlinear approach for face sketch synthesis and recognition. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 1005–1010 (2005)
35. Liu, Q., Tang, X., Jin, H., Lu, H., Ma, S.: A nonlinear approach for face sketch synthesis and recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1005–1010 (2005)
36. Liu, X., Kan, M., Wu, W., Shan, S., Chen, X.: Viplfacenet: an open source deep face recognition sdk. Front. Comp. Sci. **11**(2), 208–218 (2017)
37. Liu, D., Li, J., Wang, N., Peng, C., Gao, X.: Composite components-based face sketch recognition. Neurocomputing **302**, 46–54 (2018)
38. Liu, D., Gao, X., Wang, N., Li, J., Peng, C.: Coupled attribute learning for heterogeneous face recognition. IEEE Trans. Neural Netw. Learn. Syst. **31**(11), 4699–4712 (2020)
39. Lowe, D.: Distinctive image features from scale-invariant key-points. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
40. Lu, J., Liong, V.E., Zhou, J.: Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition. IEEE Trans. Pattern Anal. Mach, Intell (2018)
41. Martinez, A., Benavente, R.: The AR Face Database. Tech. Rep. 24, CVC, Barcelona, Spain (1998)
42. Mignon, A., Jurie, F.: A new metric learning approach for cross modal matching. In: Proc. Asian Conf. Comput. Vis. 1–14 (2012)
43. Mignon, A., Jurie, F.: CMML: a new metric learning approach for cross modal matching. In: Proc. Asian Conf. Comput. Vis. 1–14 (2012)

44. Mittal, P., Vatsa, M., Singh, R.: Composite sketch recognition via deep network. In: Proc. Int. Conf. Biom, pp. 1091–1097 (2015)
45. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
46. Parkhi, O.M., Vedaldi, A., Zisserman, A., et al.: Deep face recognition. In: Proc. British Conf. Mach. Vis. 1, 6 (2015)
47. Peng, C., Gao, X., Wang, N., Li, J.: Graphical representation for heterogeneous face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(2), 301–312 (2017)
48. Peng, C., Gao, X., Wang, N., Li, J.: Sparse graphical representation based discriminant analysis for heterogeneous face recognition. Sig. Process. **156**, 46–61 (2019)
49. Saxena, S., Verbeek, J.: Heterogeneous face recognition with cnns. In: Proc. Eur. Conf. Comput. Vis., pp. 483–491. Springer (2016)
50. Shao, M., Fu, Y.: Cross-modality feature learning through generic hierarchical hyperlingual-words. IEEE Trans. Neural Netw. Learn. Syst. **28**(2), 451–463 (2017)
51. Sharma, A., Jacobs, D.: Bypass synthesis: PLS for face recognition with pose, low-resolution and sketch. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 593–600 (2011)
52. Sharma, A., Jacobs, D.W.: Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch, pp. 593–600 (2011)
53. Song, Y., Bao, L., Yang, Q., M., Y.: Real-time exemplar-based face sketch synthesis. In: Proceedings of Eureopean Conference on Computer Vision, pp. 800–813 (2014)
54. Song, L., Zhang, M., Wu, X., He, R.: Adversarial discriminative heterogeneous face recognition. In: AAAI Conf. Artif. Intell. (2018)
55. Tang, X., Wang, X.: Face sketch synthesis and recognition. In: Proc. IEEE Int. Conf. Comput. Vis., pp. 687–694 (2003)
56. Tang, X., Wang, X.: Face sketch synthesis and recognition. In: Proceedings of IEEE International Conference on Computer Vision, pp. 687–694 (2003)
57. Wang, N., Gao, X., Li, J.: Random sampling and locality constraint for fast face sketch synthesis. ArXiv preprint arXiv:1701.01911 (2017)
58. Wang, N., Gao, X., Sun, L., Li, J.: Bayesian face sketch synthesis. IEEE Trans. Image Process. **PP**, 1–11 (2017)
59. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: IEEE Conf. Comput. Vis. Pattern Recogn., pp. 5265–5274 (2018)
60. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(11), 1955–1967 (2009)
61. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(11), 1955–1967 (2009)
62. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
63. Wang, N., Li, J., Tao, D., Li, X., Gao, X.: Heterogeneous image transformation. Pattern Recogn. Lett. **34**(1), 77–84 (2013)
64. Wang, N., Tao, D., Gao, X., Li, X., Li, J.: Transductive face sketch-photo synthesis. IEEE Trans. Neural Netw. Learn. Syst. **24**(9), 1364–1376 (2013)
65. Wu, X., Huang, H., Patel, V.M., He, R., Sun, Z.: Disentangled variational representation for heterogeneous face recognition. In: AAAI Conf. Artif. Intell., pp. 9005–9012 (2019)
66. Wu, X., Song, L., He, R., Tan, T.: Coupled deep learning for heterogeneous face recognition. In: AAAI Conf. Artificial Intell. (2018)

67. Yang, M., Zhang, L., Feng, X., Zhang, D.: Sparse representation based fisher discrimination dictionary learning for image classification. Int. J. Comput. Vis. **109**(3), 209–232 (2014)
68. Yi, D., Liu, R., Chu, R., Lei, Z., Li, S.: Face matching between near infrared and visible light images. In: Proc. Int. Conf. Biom., pp. 523–530 (2007)
69. Zhang, L., Lin, L., Wu, X., Ding, S., Zhang, L.: End-to-end photo-sketch generation via fully convolutional representation learning. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 627–634 (2015)
70. Zhang, J., Wang, N., Gao, X., Tao, D., Li, X.: Face sketch-photo synthesis based on support vector regression. In: IEEE International Conference on Image Processing, pp. 1125–1128 (2011)
71. Zhang, W., Wang, X., Tang, X.: Coupled information-theoretic encoding for face photo-sketch recognition. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 513–520 (2011)
72. Zhang, W., Wang, X., Tang, X.: Coupled information-theoretic encoding for face photo-sketch recognition. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 513–520 (2011)
73. Zhou, H., Kuang, Z., Wong, K.: Markov weight fields for face sketch synthesis. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1091–1097 (2012)
74. Zhu, M., Wang, N.: A simple and fast method for face sketch synthesis. In: International Conference on Internet Multimedia Computing and Service, pp. 168–171 (2016)
75. Zhu, M., Li, J., Wang, N., Gao, X.: Learning deep patch representation for probabilistic graphical model-based face sketch synthesis. Int. J. Comput. Vis. **129**(6), 1820–1836 (2021)

3D Face Recognition 15

Di Huang and Hongyu Yang

15.1 Introduction

The past decades have witnessed the tremendous efforts made on Face Analysis (FA). In spite of great progress achieved so far within the field [1–5], faces recorded by 2D images (or videos) are still not reliable enough as a biometric trait, especially in the presence of illumination and pose changes. With the rapid development in 3D imaging systems, 3D (or 2.5D) scans have been expected as a major alternative to deal with the unsolved issues.

3D data convey exact geometry information of faces and are theoretically complementary to 2D images/videos that capture texture clues. During the last ten years, 3D FA has received increasing attention from both the academia and industry, along with the release of several milestone benchmarks, such as FRGC [6], Bosphorus [7], BU-3DFE [8], and BU-4DFE [9]. The investigations not only cover 3D shape based FA [10–13], but also include 3D+2D multi-modal FA [14–16] and 3D aided 2D FA (i.e., heterogeneous or asymmetric 3D-2D FA) [17, 18],[1] involving in various applications, e.g., Face Recognition (FR), Facial Expression Recognition (FER), Gender and Ethnicity Classification (GEC), Age Estimation (AE), *etc.*

Since the emergence of 3D FA, many hand-crafted approaches have been proposed with consistent performance gains reported in public databases, handling the reputed difficulties of expression variations as well as internal/external occlusions [19, 20]. Please refer to

[1] We mainly focus on 3D shape based face analysis. Unless stated, all related work discussed only makes use of 3D geometry clues.

D. Huang (✉) · H. Yang
Beihang University, Beijing, China
e-mail: dhuang@buaa.edu.cn

H. Yang
e-mail: hongyuyang@buaa.edu.cn

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6_15

[21–23] for a number of comprehensive surveys. More recently, deep learning approaches have dominated this community, which are developing in three major trends: (1) designing more sophisticated deep neural networks for higher accuracies; (2) exploring more effective solutions to consumer-grade depth sensors for better practicability; and (3) building more powerful generic 3D face models to facilitate related applications. We elaborate these trends and introduce some attempts by our team in the following.

Trend I: Deep learning models for 3D FA

In the era of deep learning, the studies on 3D FA are not as extensive as the ones in the 2D domain, thus limiting its pervasion. As stated in [24], the main reason lies in that top deep learning models generally demand a huge amount of data, while 3D face acquisition is not as easy as that of 2D images. Specifically, current successful deep models on 3D FA are still Convolutional Neural Networks (CNN) based [25–29], where facial depth images are generated from irregular point-clouds as input. This is either fulfilled by fine surface registration on probe faces or rich pose augmentation on training faces. For the former, even though more advanced alignment methods are available [30], frontalizing a given face scan of an arbitrary pose is rather challenging, because uncertainty is incurred in partially missing data by self-occlusions. For the latter, as it is unrealistic to synthesize training faces with continuous viewpoint changes in the 3D space, the predefined discrete ones are probably inconsistent to that of the probe, leading to errors. Both the facts indicate that the property of rotation invariance of 3D data is not sufficiently exploited in those methods.

On the other side, point-cloud deep learning is widely investigated and shape characteristics of surfaces are hierarchically encoded from disordered points. As well-known representatives, PointNet [31], PointNet++ [32], PointCNN [33], *etc.*, prove their abilities in 3D object detection, classification, and segmentation, which suggests the potential in more tasks. Nevertheless, human faces are deformable and full of fine-grained geometric details, much more complex than the general objects only with coarse-grained rigid shapes. This makes it not straightforward to adapt vanilla geometry deep learning models to 3D FA, which is also confirmed by the large margin for ameliorated accuracies of the preliminary attempts [34].

Targeting the issues above, we propose a novel deep learning approach, namely Fast and Light Manifold CNN (FLM-CNN), and demonstrate its effectiveness in 3D FER [35]. Considering that the representation ability of the point-cloud-based models is limited by the MLP-based framework, we design the model according to that of the Manifold CNNs, which applies patch-based operators to launch convolution calculation on manifold meshes. Different from the existing manifold CNN models, such as Geodesic CNNs (GCNN) [36], Anisotropic CNNs (ACNN) [37], and Mixture Model CNNs (MoNet) [38], FLM-CNN adopts a human vision inspired pooling structure and a multi-scale encoding strategy to enhance geometry representation, which highlights the differences of facial shapes between individual expressions in an efficient manner. Moreover, we present a sampling tree-based preprocessing method, and it greatly reduces memory cost without much information loss of original data, benefiting data augmentation. More importantly, thanks to the property of manifold CNN features of being rotation-invariant, the proposed method displays a high robustness to pose changes.

Trend II: Representation of depth images of consumer-grade sensors

The face models used in the state-of-the-art 3D FA systems are of high-quality as the ones in FRGC [6], Bosphorus [7], *etc.*, recorded by specialized equipments. In the early years, the devices to capture such data, e.g., Minolta VIVID 910, may take dozens of seconds for a single session, and during this period, faces are required to keep still, which makes it unsuitable for on-line FA scenarios, especially when users are not so cooperative. Along with the continuous revolution in both hardware and software, the following versions, e.g., 3dMD and Artec3D, are able to provide dynamic flows of 3D face scans of a high resolution at the rate of tens of frames per second. But they are at rather high prices, generally hundreds or even thousands of times more expensive than 2D cameras. Moreover, they are usually big in size and not convenient to operate and it thus leaves a hard problem to implement systems based on them in practical conditions.

The recent advent of low-cost and real-time 3D scanning devices, such as Microsoft Kinect and Intel Realsense, makes it possible to collect and exploit 3D data in our daily life. Low-cost 3D data (or with the texture counterpart, i.e., RGB-D data) have received increasing attention in the academia in various aspects, including action recognition [39], object detection [40], scene classification [41], *etc.* In contrast to the aforementioned tasks, FR using low-cost 3D data is more challenging, because the compromise between cost and accuracy by such sensors makes data much more noisy, leading to serious loss of important details. Some preliminary attempts have been made, and the best result is up to 100% [42], indicating its feasibility to some extent. Nevertheless, the score is not sufficiently convincing, because the subjects in the evaluation dataset are not many enough and only with limited variations.

To address these issues, we build a large-scale database consisting of low-cost Kinect 3D face videos, namely Lock3DFace, for 3D FR [43]. To the best of our knowledge, Lock3DFace is currently one of the largest low-cost 3D face databases for public academic use. The 3D samples are highly noisy and contain a diversity of variations in expression, pose, occlusion, time lapse, and their corresponding texture and near infrared channels have changes in lighting condition and radiation intensity, supporting the scenarios of 2D FR, near infrared FR, multi-modal FR, and heterogeneous FR. We then present a lightweight and efficient deep approach [44], namely, Led3D, to 3D FR using such low-quality depth images, for both higher accuracy and higher efficiency. To achieve this, Led3D works in two ways, i.e., a new lightweight CNN architecture as well as bigger and finer training data. In particular, to balance accuracy and efficiency, it focuses on an enhanced lightweight network rather than stubbornly deepening the model. The backbone network contains only 4 convolutional layers, and to make a high accuracy, we propose a Multi-Scale Feature Fusion (MSFF) module and a Spatial Attention Vectorization (SAV) module. The former combines features at different levels in an efficient way, improving the representation of low-quality face data, and the latter highlights important spatial facial clues when summarizing local features and outperforms the widely used Global Average Pooling (GAP) for FR. Furthermore, to deal with the problem of inadequate data in deep models, a preprocessing pipeline and a data augmentation scheme for low-quality 3D face data are proposed, generating a finer and bigger training set.

Trend III: Generic 3D face models for diverse applications

3D Morphable Face Models (3DMMs) are well-reputed statistical models, established by learning techniques upon prior distributions of facial shapes and textures from a set of samples with dense correspondence, aiming at rendering realistic faces of a high variety. Since a morphable representation is unique across different downstream tasks where the geometry and appearance are separately controllable, 3DMMs are pervasively exploited in many face analysis applications. In 3DMMs, the most fundamental issue lies in the way to generate latent morphable representations, and during the past two decades, along with data improvement in scale, diversity, and quality [45–48], remarkable progresses have been achieved. The methods are initially linear model based [49–51] and further extended to multilinear model based [52–54], in which different modes are individually encoded. Unfortunately, for the relatively limited representation ability of linear models, these methods are not so competent at handling the cases with complicated variations, e.g., exaggerated expressions. In the context of deep learning, a number of nonlinear model-based methods have been investigated with the input of 2D images [55, 56] or 3D meshes [57–60] by using CNNs or Graph Neural Networks (GNNs) for their strong representation power. They indeed deliver some performance gains; however, restricted by the resolution of discrete representing strategies on input data, facial priors are not sufficiently captured, incurring loss of shape details.

Recently, several studies on Implicit Neural Representations (INRs) [61–64] have shown that 3D geometries can be precisely modeled by learning continuous deep implicit functions. They describe an input observation as a low-dimensional shape embedding and estimate the Signed Distance Function (SDF) or the occupancy value of a query point so that the surface of an arbitrary resolution and topology can be defined by an isocontour. Due to the continuous parameterization and consistent representation, INRs prove superior to the discrete voxels, point-clouds and meshes, and report decent results in shape reconstruction [65–68] and surface registration [69–71]. Such an advantage suggests an alternative to 3DMM that can fulfill accurate correspondence and fine-grained modeling in a unified network. Nevertheless, unlike the objects with apparent shape differences and limited non-rigid variations such as indoor scenes and human bodies, all face surfaces look very similar but include more complex deformations, where multiple identities and rich expressions deeply interweave with each other, making current INR methods problematic in face modeling, as evidenced by the preliminary attempt [72]. Another difficulty is that implicit functions primarily require watertight input, which is not friendly to facial surfaces.

To deal with the problems, we introduce a novel versatile 3D face morphable model, namely ImFace, which substantially upgrades conventional 3DMMs by learning INRs [73]. To capture nonlinear facial geometry changes, ImFace builds separate INR sub-networks to explicitly disentangle shape morphs into two deformation fields for identity and expression respectively, and an improved auto-decoder embedding learning strategy is introduced to extend the latent space of expressions to allow more diverse details. In this way, inter-individual differences and fine-grained deformations can be accurately modeled, which simultaneously takes into account the flexibility when applied to related tasks. Furthermore, inspired by linear blend skinning [74], a Neural Blend-Field is presented to decompose the entire facial deformation or geometry into semantically meaningful regions encoded by a set of local implicit functions and adaptively blend them through a lightweight module, leading to more sophisticated representations with reduced parameters. Besides, a new preprocessing pipeline is designed, which bypasses the need of watertight face data as in existing SDF-based INR models and works well for various facial surfaces, *i.e.*, either hardware-acquired or artificially synthesized.

The remainder of this chapter detailedly introduces our solutions and experiments. Specifically, Sect. 15.2 presents the fast and light manifold CNN for 3D FER. Section 15.3 describes low-quality depth image-based 3D face analysis, including the Lock3DFace dataset and the lightweight Led3D FR model. The nonlinear 3D morphable face model is displayed in Sect. 15.4.

15.2 Fast and Light Manifold CNN-based 3D FER

3D FER has received persistently increasing attention during the last several decades. On the one hand, expressions are the consequences of shape deformations produced by facial muscle movements, which are better recorded in the 3D modality. On the other hand, 3D FER is more tolerant to the unsolved challenging factors in 2D, possessing the invariance to illumination variations and the convenience in pose correction. Here, we propose a Manifold CNN model-based approach (FLM-CNN) to 3D FER and we introduce it in the following.

15.2.1 Method

15.2.1.1 MoNet Revisit

In manifold CNNs, 3D shapes are modeled as 2D differentiable manifolds denoted by X. Let $f : X \to \mathbb{R}$ be the real functions defined on X. The major role of the patch operator is to map the value of function f at the neighborhood of point $x (x \in X)$ into a patch with a regular shape so that convolution can be conducted on it. In the previous literature, the patch operator acting on f at x is usually denoted as $D(x) f$, and we follow it in this study.

MoNet is a general framework of manifold CNNs, where GCNN and ACNN can be deemed as their particular instances. The patch operator is formulated as

$$D_j(x)f = \sum_{y \in \mathcal{N}(x)} \omega_j(\boldsymbol{u}(x, y))f(y), \qquad j = 1, 2, \ldots, J \tag{15.1}$$

where J denotes the dimensionality of the patch; $\mathcal{N}(x)$ denotes the neighborhood of x; and $\boldsymbol{u}(x, y)$ is any kind of local coordinate of y relative to x. ω is a weighting function to interpolate the value of patch j, chosen as Gaussian kernels in [38]:

$$\omega_j(\boldsymbol{u}) = exp(-\frac{1}{2}(\boldsymbol{u} - \boldsymbol{\mu}_j)^T \textstyle\sum_j^{-1}(\boldsymbol{u} - \boldsymbol{\mu}_j)), \tag{15.2}$$

where $\boldsymbol{\mu}_j$ and $\sum_j$ are the mean vector and covariance matrix of a Gaussian kernel, respectively.

Based on (15.1) and (15.2), we can see that there are two major steps in MoNet (and all manifold CNNs): one is computing the weighting function of the patch operator, ω_j; and another is generating the patch, $D_j(x)f$. In the first stage, MoNet makes use of learnable Gaussian kernels, which are more powerful than the fixed templates used in GCNN and ACNN. However, the mixture model-based scheme is more complex and the weighting function has to be repetitively computed in training, consuming much time. In the second stage, manifold CNNs, e.g., GCNN, ACNN, and Monet need huge memory when generating patches, and it is indeed a problem to large input data with thousands of points or more, e.g., 3D face models acquired by scanners. The two issues hold back the application of MoNet to 3D FER.

Therefore, we propose a novel manifold CNN model, namely FLM-CNN, which is faster and lighter. Two improvements, i.e., human vision inspired weighting (in Sect. 15.2.1.2) as well as sampling tree-based preprocessing (in Sect. 15.2.1.3), are presented to handle the two limitations, respectively. The framework is shown in Fig. 15.1.

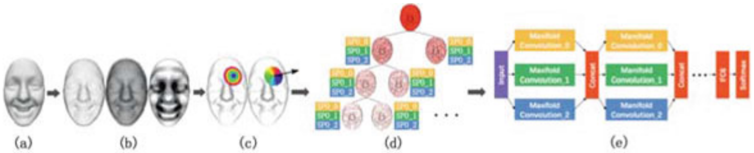


Fig. 15.1 Method Overview: **a** original face scan; **b** differential geometry quantities; **c** geodesic distance based local polar coordinates; **d** sampling tree-based patch operator; and **e** FLM-CNN structure. From Chen et al. [35], with permission

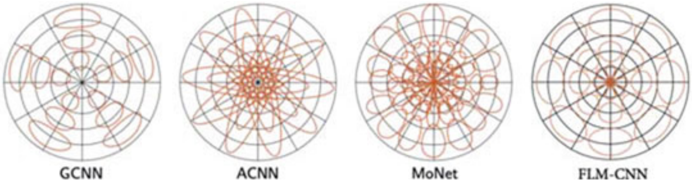


Fig. 15.2 Comparison of different weighting schemes. From Chen et al. [35], with permission

15.2.1.2 Human Vision Inspired Weighting

The weighting function in MoNet contributes to generating more discriminative features, but it consumes much more time in training, compared with the hand-crafted ones. In our FLM-CNN, to speed up the training phase while keeping the features sufficiently distinctive, we employ a human vision inspired weighting strategy, as it proves effective in a number of studies, such as DAISY [75] and HSOG [76].

Specifically, our FLM-CNN adopts a Gaussian-based weighting function, where the mean and variance are designed to simulate the human vision mechanism. In such vision systems, the receptive field is basically modeled as a set of concentric circles as in Fig. 15.2. The information captured closer to the center is assigned bigger weights than that farther away, emphasizing its importance. In this case, we set the mean and variance as

$$\mu_{\rho_i} = (\frac{i-0.5}{N_\rho - 0.5})^2 \rho_0, \qquad \sigma^2_{\rho_i} = (\frac{i\rho_0}{2N^2_\rho}) \tag{15.3}$$

$$\mu_{\theta_j} = \frac{2\pi j}{N_\theta}, \qquad \sigma^2_{\theta_j} = (\frac{\pi}{2N_\theta})^2 \tag{15.4}$$

in the local geodesic polar coordinate system, where $i = 1, 2, \ldots, N_\rho$ and $j = 1, 2, \ldots, N_\theta$. There are N_ρ and N_θ Gaussian kernels in radial and angular directions, respectively. μ and σ^2 denote the mean and variance; and ρ_0 denotes the radius of local geodesic disc (a local neighborhood, explained in Sect. 15.2.1.5). From (15.3), we can see that μ_{ρ_i} and $\sigma^2_{\rho_i}$ are not uniform along the radial direction. Kernels closer to the center of the receptive field distribute in a denser manner and have smaller variance. It guarantees that important information is highlighted.

To achieve rotation invariance, the origin of the angular coordinate is required. We select the minimum principal curvature direction as the origin as in ACNN. Due to the bi-directional property, we rotate the convolution kernel twice and hold the maximum. Formally, the manifold convolution can be formulated as

$$(f * g)(x) = \max_{\Delta\theta \in \{0,\pi\}} \int_0^{2\pi} \int_0^{\rho_{max}} g(\rho, \theta + \Delta\theta)(D(x)f)(\rho, \theta)\, d\rho\, d\theta \tag{15.5}$$

where $\Delta\theta$ is either 0 or π, and g denotes the parameters of convolution kernels to be trained.

From Fig. 15.2, we can see the difference in the patch operator weighting between FLM-CNN and some related counterparts in generalizations of convolution on the manifold.

15.2.1.3 Sampling Tree Based Preprocessing

Existing manifold CNNs mainly focus on shape correspondence which is a point-level classification task. To the best of our knowledge, they have not been explored for object-level classification as batches of large 3D point-clouds or meshes result in unaffordable memory consumption. Besides, object-level classification is more challenging to manifold CNNs since it is not straightforward to integrate unordered point-wise features into a global one. Masci et al. [36] calculate the covariance matrix of the features of the last convolutional layer as global representation, but such simple statistics causes inevitable loss of spatial and characteristic information.

Regarding 2D CNNs, pooling layers play a significant role in reducing feature dimensionality and enlarging receptive fields of trailing convolutional layers without increasing the kernel size. Max-pooling is the most popular down-sampling way but it is not proper for manifold CNNs, because patch operators need to be recalculated after down-sampling which slows down training and increases memory usage. It suggests that down-sampling should be applied in advance.

Random sampling gives an easy choice but it tends to drop important cues at the same time, leading to performance decrease. Therefore we propose a sampling tree-based preprocessing technique, to control memory cost and keep useful information. A sampling tree is a hierarchical data structure that stores down-sampling point indices of a 3D scan. Figure 15.1d shows an illustration of a sampling tree. The root node stores all the point indices of the original 3D scan. All nodes except leaf ones split into some child nodes to preserve the point indices sampled from their parent nodes. The point indices of child nodes born from the same parent are expected to be complementary. They should be as distinct as possible and their union should cover all the point indices of the parent node. The information of a parent node can thus be completely and almost non-repeatedly transferred into its child nodes. The numbers of point indices of the nodes at the same depth are suggested to be equal, in which case we can conveniently execute batch training. The node splitting process stops if the receptive field of the points of the deepest nodes is approximately equivalent to that of a full 3D scan. Once the sampling tree is built, we produce a set of paths from the root node to the leaf ones, and each path can be treated as a down-sampling strategy of the 3D scan in FLM-CNN. Thus, a 3D scan can be augmented into several depending on the number of leaf nodes.

Before training, we calculated patch operators on every node except the root of the sampling tree, and they are called Sampling Patch Operators (SPO) in this study. The SPO of node n_i acts on the point features of its parent node n_{p_i} and produces patch-based point features of n_i. In the discrete case, the SPO of n_i can be expressed as

$$(D(x)f)(\rho,\theta) = \sum_{y \in (\mathcal{N}(x) \cap n_{p_i})} \omega_{\rho,\theta}(x,y) f(y), \quad x \in n_i \tag{15.6}$$

where the weighting function of SPO is represented as an $N_\rho N_\theta N_{n_i} \times N_{n_{p_i}}$ sparse matrix. For a 3D scan with N points, the size of the weighting function matrix of the original patch operator is $N_\rho N_\theta N_{n_i} \times N_{n_{p_i}}$, which can be thousands of times larger than SPO of leaf nodes. Consequently, SPO helps to save much memory in training.

15.2.1.4 Implementation Details of FLM-CNN

Based on manifold convolution and SPOs, we construct FLM-CNN, and Fig. 15.3 shows its architecture. There are five manifold convolutional layers, and each layer has three convolutional kernels of different scales, as in Google Inception [77], to achieve multi-scale representation. We also add 1×1 convolutions (can be seen as a point-wise fully-connected layer) before the last three manifold convolutional layers, for computation reduction and rectified linear activation [77]. Through the five convolutional layers, every sampling point in the fifth layer can represent the whole 3D scan and can be used for classification. Two point-wise fully-connected layers are in the following and the output of the last layer is activated by the softmax function. Finally, FLM-CNN is trained by minimizing the cross-entropy loss.

In training, batch encapsulation of SPOs is not easy to launch. One reason lies in that discrete SPOs are sparse matrices which have different quantities of non-zero values, and another is that the SPO matrices of the first convolutional layer are of different sizes since the root nodes store the original 3D scans which usually have different numbers of points. In this case, we concatenate SPO matrices in the diagonal direction and concatenate feature matrices along the vertical axis. Figure 15.4 demonstrates this operation. By sampling the same number of points on nodes at the same depth, we guarantee that each SPO in the same convolutional layer has the same number of rows. Therefore, SPOs can produce patch features in the same shape, which can be fed into the subsequent convolutional layers in a batch style.

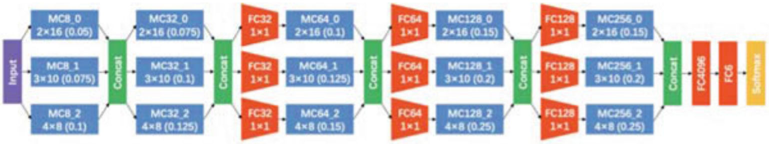


Fig. 15.3 Architecture of FLM-CNN. *MC* denotes the manifold convolutional layer, below which are kernel size and the radius of the local geodesic disc; and *FC* denotes the point-wise fully-connected layer which can also be seen as a 1×1 convolutional layer. From Chen et al. [35], with permission

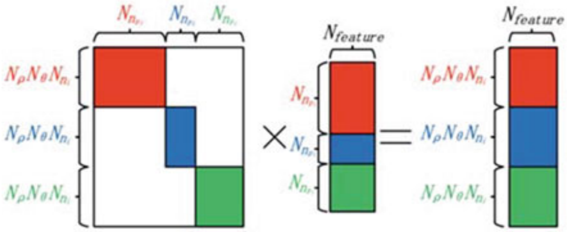

$N_{n_{p_i}}$
$N_{n_{p_i}}$
$N_{n_{p_i}}$
$N_{feature}$
$N_{feature}$
$N_\rho N_\theta N_{n_i}$
$N_\rho N_\theta N_{n_i}$
$N_\rho N_\theta N_{n_i}$
$N_{n_{p_i}}$
$N_{n_{p_i}}$
$N_{n_{p_i}}$
×
=
$N_\rho N_\theta N_{n_i}$
$N_\rho N_\theta N_{n_i}$
$N_\rho N_\theta N_{n_i}$

Fig. 15.4 Illustration of batch encapsulation of SPOs ($N_{n_{p_i}}$ of SPOs can be different from each other). From Chen et al. [35], with permission

15.2.1.5 Rotation-Invariant 3D FER

Thanks to the structure of FLM-CNN, it can be applied to 3D FER. Figure 15.1 shows an overview of the framework. Firstly, multiple order differential geometry quantities, including original coordinates, normal vectors, and the shape index values [78] are taken as the input data. Next, we locate the neighborhood of each point and compute the local polar coordinates according to geodesic distance. Further, we build the sampling tree and generate SPOs of 3D faces. Finally, we construct FLM-CNN for 3D FER.

Coordinate System independent Differential Quantities. Coordinate, normal, and shape index convey the original property of the given surface and its first and second-order derivatives, and their joint use delivers a more comprehensive description of shape characteristics of 3D faces. We follow the way to calculate them as in [29, 79, 80].

Among the three features, shape index is invariant to rotations, whose values range from 0 to 1 and uniquely represent local shapes; unfortunately, the values of original coordinates and normal vectors are dependent on the coordinate system. To make the method insensitive to pose changes, we adopt Principal Component Analysis (PCA) to generate surface specific coordinates and normal vectors. For each surface, we build an intrinsic coordinate system in which the basis vectors are first selected as the eigenvectors of the coordinate covariance matrix and the original coordinates can then be decomposed using the intrinsic basis vectors. The new coordinates are thus independent of extrinsic coordinate systems. For normal vectors, the transformation process is the same.

Neighborhood and Local Polar Coordinates. For computation simplicity, we just consider the neighborhood of each point. 3D face scans are scaled into a unit sphere, and a fixed threshold ρ_0 in terms of geodesic distance is set to locate the neighborhood. Geodesic distances are computed by a fast and exact discrete geodesic algorithm named Vertex-oriented Triangle Propagation (VTP) [81]. VTP is based on wavefront propagation and works in a continuous Dijkstra style. We stop it when the maximum geodesic distance of the visited neighborhood reaches the fixed threshold. By VTP, we find the neighborhood of point x, a geodesic disc expressed as $\mathcal{N}(x) = \{y : d_X(x, y) \leq \rho_0\}$, where ρ_0 is called its radius.

Once we have the neighborhood, the next is to generate local polar coordinates. Based on the geodesic distance of the neighborhood to x, we compute the radial coordinate, ρ. Figure 15.5 illustrates the computation procedure of angular coordinates. The geodesic path between the neighborhood and x can be recorded by VTP. All the geodesic paths go

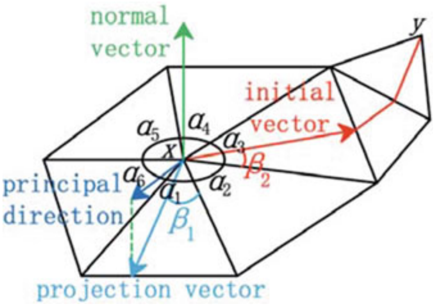


Fig. 15.5 Computation of angular coordinates. From Chen et al. [35], with permission

through the 1-ring triangles of x, and the angular coordinate of the neighborhood is computed according to the direction of the initial vector on the 1-ring triangles. The minimum principal curvature vector is then projected onto the 1-ring triangles along the normal vector of x and the direction of the projection vector is treated as the origin. Thus we can compute the angle between the initial vector and the origin. As seen in Fig. 15.5, the angular coordinate of y can be computed as $\theta_y^{'} = \beta_1 + \alpha_2 + \beta_2$. The sum of angles $\sum_{i=1}^{6} \alpha_i$ is not necessarily equal to 2π because of non-zero surface curvature at x, and the final angular coordinate of y is scaled as $\theta_y = \theta_y^{'} / \sum_{i=1}^{6} \alpha_i$. The procedure for boundary points is similar by filling up some triangles.

Sampling Tree and Patch Operator. For 3D face models, the sampling tree is generated, and the SPOs of all the nodes are computed according to (15.6). The radius of the geodesic disc, ρ_0, can be viewed as the receptive field of the convolution kernel. In general, the receptive field is small in shallow layers for local features and is large in deep layers for global features. As a result, ρ_0 is small for the nodes close to the root and large for the ones close to the leaf.

15.2.2 Experiments

15.2.2.1 Protocols

We adopt the standard identity-independent protocols as in most previous studies so that direct comparison can be made. Specifically, there are two protocols (P1 and P2), in both of which 60 out of 100 subjects in BU-3DFE are considered, and the samples of the two highest intensities with a total number of 720 (60×6×2) are used. In P1, we randomly select 60 persons and fix them during the whole experiment. In each round, we give a random split for 10-fold cross-validation, where 648 samples of 54 persons (90%) are employed for training (48 persons for model building and 6 persons for model selection) and 72 samples of 6 persons (10%) for testing. In P2, the only difference lies in that 60 persons are randomly chosen in every round. For both the two protocols, experiments are conducted 100 rounds and the average score is reported as a stable accuracy.

Table 15.1 Performance comparison with state-of-the-art methods on BU-3DFE (the scores marked by * are the results achieved by combining shape and texture cues). From Chen et al. [35], with permission

Method	Landmarks	Registration	Model	Accuracy (%)	
				P1	P2
Wang et al. (2006) [82]	64 Manu.	No	Hand-crafted	61.79	–
Soyel et al. (2007) [83]	11 Manu.	No	Hand-crafted	67.52	–
Tang et al. (2008) [84]	83 Manu.	No	Hand-crafted	74.51	–
Gong et al. (2009) [85]	Not needed	Yes	Hand-crafted	76.22	–
Berretti et al. (2010) [86]	27 Manu.	Yes	Hand-crafted	–	77.54
Lemaire et al. (2011) [87]	21 Auto.	Yes	Hand-crafted	75.76	–
Lemaire et al. (2013) [88]	Not needed	Yes	Hand-crafted	76.61	–
Li et al. (2012) [89]	Not needed	Yes	Hand-crafted	–	80.14
Zeng et al. (2013) [90]	3 Auto.	Yes	Hand-crafted	–	70.93
Yang et al. (2015) [79]	Not needed	Yes	Hand-crafted	84.80	82.73
Azazi et al. (2015) [91]	11 Auto.	Yes	Hand-crafted	–	85.81*
Zhen et al. (2015) [80]	3 Auto.	Yes	Hand-crafted	84.50	83.20
Li et al. (2015) [92]	49 Auto.	Yes	Hand-crafted	82.70	–
Li et al. (2015) [93]	Not needed	Yes	Deep	83.96	82.81
Oyedotun et al. (2017) [28]	Not needed	Yes	Deep	84.72	–
Li et al. (2017) [29]	Not needed	Yes	Deep	86.86*	–
FLM-CNN	Not needed	No	Deep	86.67	85.96

15.2.2.2 Results

Comparison. We compare our results with the state-of-the-art ones on BU-3DFE under both the protocols, P1 and P2. The comparison is shown in Table 15.1, and we can see that our method outperforms all the others that report the results in the 3D modality, including both the hand-crafted and deep solutions. The result in [29] achieved under P1 seems superior to ours, but it should be noted that this score is actually based on a combination of shape and texture clues (they do not provide the 3D result separately), while FLM-CNN only uses the shape information. Besides, our method does not require any landmarks or global registration which is necessary in the other counterparts. All the facts indicate the effectiveness of the proposed method in 3D FER. Table 15.2 gives the confusion matrix using P1.

Robustness to Pose Variations. As BU-3DFE only contains frontal faces, we rotate the samples to certain angles and remove the invisible triangles, to simulate self-occlusions.

To be specific, as in P1, we randomly select 60 persons, and the samples of the two highest expression intensities are exploited. 54 persons are used to train our FLM-CNN, and the other 6 persons are utilized for the test. Each test scan is synthesized to 16 poses (Yaw and Pitch: −80°, −60°, −40°, −20°, 20°, 40°, 60°, 80°) to generate faces. Figure 15.6 visualizes rotated views and their corresponding frontal views. We report the results of the 16 angles plus the original individually.

Table 15.2 Confusion matrix using P1 on BU-3DFE. From Chen et al. [35], with permission

%	AN	DI	FE	HA	SA	SU
AN	**85.28**	3.89	1.94	0.56	8.33	0.00
DI	1.11	**88.33**	5.56	1.11	1.94	1.95
FE	1.39	7.50	**76.94**	7.50	2.50	4.17
HA	0.00	1.39	6.11	**92.50**	0.00	0.00
SA	13.33	1.11	3.61	0.00	**80.56**	1.39
SU	0.00	1.11	2.50	0.00	0.00	**96.39**

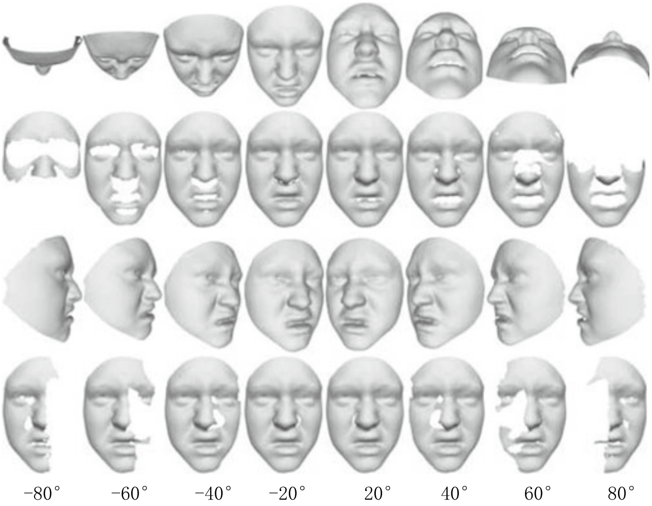


Fig. 15.6 Visualization of rotated faces: the first and third rows show faces rotated in the pitch and yaw directions; and the second and fourth rows display the corresponding faces in the frontal view. From Chen et al. [35], with permission

The accuracies of 17 poses (plus 0° are depicted in Fig. 15.7, and we can see that pose changes indeed impose a negative impact on FER, since they cause data missing in some facial parts that are probably important to expressions. This fact can be evidenced by the consistent drop in accuracy as the pose angle in the yaw or pitch direction increases.

Besides, we also notice that the degradation by the angle changes in the yaw direction is not as serious as that in pitch. The left and right face parts possess redundant information due to their symmetric structure. Regarding the pitch case, we can see that the scores of the lower face parts are generally better than those of the upper ones. As we know, the upper face region (e.g., forehead and nose) is relatively rigid and the lower area (e.g., mouth) is relatively non-rigid, and they present different clues in recognizing expressions. More importantly, thanks to the property of being rotation-invariant, the proposed approach shows a good robustness

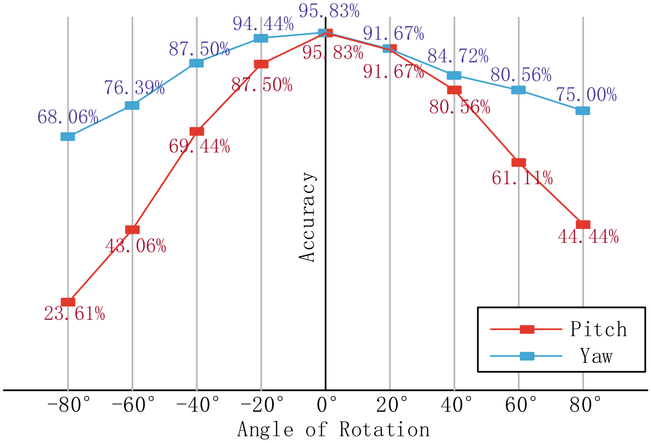


Fig. 15.7 Results of pose changes on BU-3DFE

to such pose variations, and the results on the samples of moderate pose changes basically remain stable. It clearly suggests that our approach has a promising potential to deal with 3D FER in more practical scenarios.

Computational Cost and Memory Usage. We reproduce MoNet according to [38] and compare it with our FLM-CNN. The statistical data is obtained by TensorBoard. For simplicity, both MoNet and FLM-CNN are equipped with two convolutional layers. We calculate the original patch operator for MoNet and the SPO for FLM-CNN. Then, we train both of them in 100 iterations with the batch size of 1. MoNet spends 124 seconds and occupies 3988MB memory, while FLM-CNN costs only around 1 second and 32MB.

It is worth noting that the version of FLM-CNN used in 3D FER is more complex, but its time and memory costs are still under control (13 seconds and 804MB). In contrast, the cost of the standard version of MoNet in [38], which contains three convolutional layers, tends to be too heavy for this experiment. All of the time evaluation is achieved on a single GPU (GeForce GTX TITAN X).

15.3 Low-Quality Depth Image Based 3D FR

Although 3D data provide solutions to deal with the unsolved issues, i.e., illumination and pose variations in the 2D domain, those techniques still suffer from a very high hardware cost, which impedes their practical application. Recently, with the rapid development in 3D data acquisition, the devices have appeared which are able to capture dynamic 3D data in the real time. Kinect is one of the representatives and has received increasing attentions due to its personal affordable price and operation simplicity. In this section, we introduce a large-scale database consisting of low-cost Kinect 3D face videos, namely Lock3DFace

[43], for 3D face analysis. Moreover, Led3D, a 3D FR approach using low-quality data, targeting an efficient and accurate deep learning solution, is described.

15.3.1 Lock3DFace: A Large-Scale Database of Low-Cost Kinect 3D Faces

Zhang et al. [43] present a large-scale dataset of low-cost Kinect faces, namely Lock3DFace, aiming to thoroughly investigate low-cost 3D FR and comprehensively compare the approaches. To the best of our knowledge, Lock3DFace is the largest database of low-cost 3D face models publicly available, which consists of 5,711 video samples with a diversity of variations in expression, pose, occlusion, and time lapse, belonging to 509 individuals. In each raw face record, the clues in the texture and near infrared modalities are also provided, supporting the scenarios of 2D FR, near infrared FR, multi-modal (RGB-D) FR, and heterogeneous FR. In the subsequent, we introduce its details.

15.3.1.1 Data Acquisition

Lock3DFace is acquired using Kinect V2. Kinect V2 updates the 2D camera of the original Kinect to a higher resolution one that can be used for color video recording. Moreover, it has an increased field of view, thus reducing the amount of distance needed between the user and the sensor for optimal configuration.

All the data are captured under a moderately controlled indoor environment with natural light in the daytime. The participants are asked to sit in front of the Kinect sensor fixed on the holder and are not allowed to move rapidly when the video is recording for 2-3 seconds. Three types of modalities, i.e., color, depth, and infrared are collected in individual channels at the same time. The color frames are recorded with the size of 1,920 × 1,080, and the depth and infrared frames are of the resolution of 512 × 424. There are in total 509 volunteers who participate in the collection process. Among them, 377 are male and 122 are female, and their ages distribute in the range of 16–36 years old. See Fig. 15.8 for more details. All the major challenges in FR are considered, involving the changes in expression, pose, and occlusion. The dataset contains two separate sessions with a long interval up to 7 months. All the 509 subjects join the first session, while 169 join the second session, thereby presenting time lapse variations as well. Regarding an individual subject, in each session, at least two video clips are made in the categories of neutral-frontal, expression, pose, and occlusion.

15.3.1.2 Challenges

To comprehensively evaluate FR methods, especially to simulate complex conditions in the real world, volunteers are required to present different expressions, poses, and occlusions in each session, forming five categories of frontal-neutral, expression, pose, occlusion, and time. Some examples of an individual are demonstrated in Fig. 15.9, from which we can see

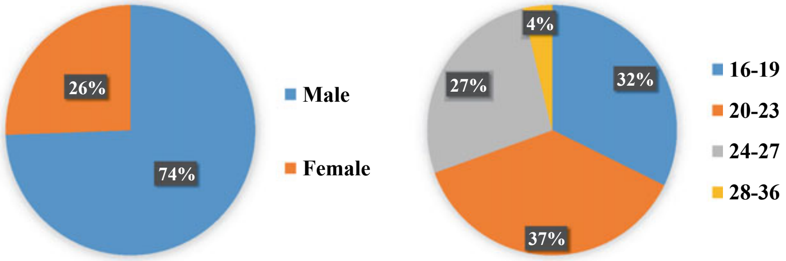


Fig. 15.8 Data distribution of Lock3DFace of **a** gender and **b** age. From [43], with permission

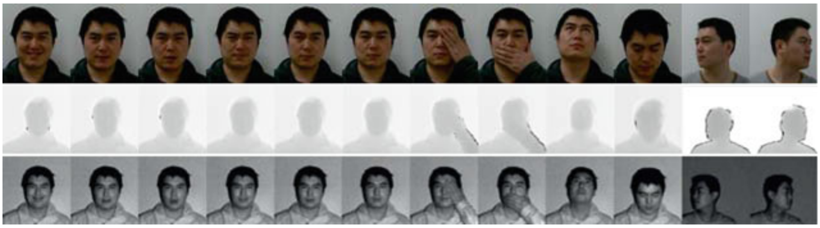


Fig. 15.9 Sample illustration of different challenges in the database. From left to right, happiness, anger, surprise, disgust, fear, neutral face, left face occluded by hand, mouth occluded by hand, looking-up, looking-down, left face profile, and right face profile are displayed in order. Upper row: RGB images; middle row: depth images; and bottom row: near infrared images which share the same coordinate with the depth maps. From [43], with permission

that a large diversity of variations are included, and it is a distinct property of Lock3DFace. Table 15.3 shows its data organization in terms of different variations.

15.3.1.3 Preprocessing

To improve the convenience of the researchers to work with Lock3DFace, along with the database, a preprocessed version of the data is provided. On the one hand, some fiducial points are manually marked on the first frame of each RGB and infrared facial video clip respectively, and the corresponding ones on the depth map are then easily obtained due to the point-to-point correspondence with the infrared map. These landmarks are a few distinct anthropometric points shared by all human beings, including the nose tip, two inner corners of eyes, and two corners of mouth. They offer the simplicity in face cropping, pose correction, feature extraction, *etc.* in face analysis. Additionally, such points can be regarded as ground-truth to evaluate the techniques of 3D facial landmarking on low-cost data.

On the other hand, the depth images captured by Kinect are very noisy, and unlike the RGB data, they cannot be directly used for feature extraction in FR. Therefore, a pipeline is provided to deal with the low-cost 3D data, including spike and outlier removing, hole filling, and smoothing. Specifically, the phase-space method in [94] is employed to exclude

Table 15.3 Data organization of the Lock3DFace database in terms of different challenges. From [43], with permission

Variations	Session-1		Session-2	
	Sub.	Sample	Sub.	Sample
Neutral-frontal	509	1014	169	338
Expression		1287		338
Pose		1014		338
Occlusion		1004		338
Total		4319		1352

the spike. The values of some pixels on the depth map are sensed as 0 when they cannot be precisely measured. To solve this problem, thresholding is applied, and a non-negative threshold is set in order to remove those unmeasurable pixels, and the missing data can then be filled using the cubic interpolation technique. To remove the noise, the bilateral filter [95] is adopted, a simple, non-iterative method that has the property of edge-preserving, and during smoothing, it retains the shape information as much as possible that is supposed to contribute in FR.

15.3.2 Led3D: A Lightweight and Efficient Deep Model to Low-cost 3D FR

Lock3DFace is the first comprehensive dataset that is suitable for evaluating methods on 3D FR using low-quality depth images, and it provides baseline results using Iterative Closet Points (ICP). Later, Cui et al. [96] present a deep model-based baseline. They both illustrate the feasibility of the identification on low-quality 3D face data. Unfortunately, very little research has investigated this issue. In [44], Mu et al. present a lightweight and efficient deep approach, namely Led3D, to bridge this gap.

15.3.2.1 An Efficient and Accurate Network

Led3D presents a CNN-based approach to improve the accuracy and efficiency. For fast inference, the network has to be shallower, with a smaller number of parameters, leading to lower memory cost. Thus, the backbone contains only 4 blocks which have 32, 64, 128, and 256 convolution filters, respectively. Each block is composed of a convolution layer with a kernel size of 3 × 3, a batch normalization layer, and a ReLU activation layer. As shown in Fig. 15.10b, the blocks are very compact. To enhance the accuracy, a Multi-Scale Feature Fusion (MSFF) module and a Spatial Attention Vectorization (SAV) module are proposed. MSFF is used to fuse multi-scale features from each block for comprehensive representation and SAV emphasizes important spatial information, both of which improve

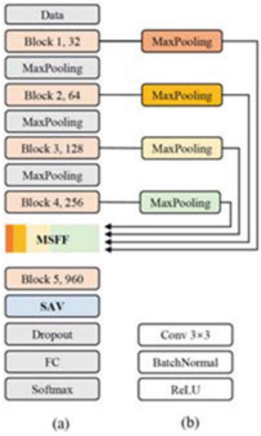


Fig. 15.10 **a** The architecture of Led3D for 3D FR with low-quality data, including a Multi-Scale Feature Fusion (MSFF) module and a Spatial Attention Vectorization (SAV) module; and **b** details of the 'Block' used in **a**. From Mu et al. [44], with permission

the discriminative capacity of the resulting feature. A dropout layer between SAV and the Fully-Connected (FC) layer is then applied, to overcome over-fitting. At the end of the network, a Softmax layer is utilized with the cross-entropy loss to guide network training. The whole architecture is shown in Fig. 15.10.

15.3.2.2 Multi-Scale Feature Fusion

CNN has a hierarchical architecture which is a stack of multiple convolutional layers. Individual layers learn different information. It is natural to combine the features at different layers for better representation. Led3D extracts the feature maps from each of the four convolutional blocks, corresponding to information captured by different Receptive Fields (RFs). All the feature maps are then down sampled to a fixed size by max-pooling for fast processing, and they are further concatenated in the channel dimension. Furthermore, the feature maps at different scales are integrated by another convolution layer consisting of 960 3×3 kernels (Block 5). In this way, a more discriminative feature can be efficiently generated to represent the 3D face of a low-quality. In addition, during model training, the convolution layers in the backbone are directed both by the successive layers as well as the neighboring ones, which can speed up the convergence of the network.

15.3.2.3 Spatial Attention Vectorization

For the aligned faces, corresponding areas contain fixed facial components. In high-level feature maps, each pixel encodes a specific area of the input image, and the area size is dependent on the receptive field, thus including fixed semantic information. But the Global Average Pooling (GAP) layer used in main-stream CNN architectures clearly ignores such correspondence. Therefore, Led3D investigates another feature generation method which is as efficient as GAP and keeps the spatial cues. In particular, a Spatial Attention Vectorization

(SAV) module is proposed to replace GAP. SAV is implemented by adding an attention weight map to each feature map. In this case, the contributions of pixels at different locations can be separately emphasized in training, and the weights are then fixed for inference. In the network, SAV is applied to the feature maps produced by MSFF, which previously integrates both the low-level and high-level features. In SAV, there are 960 convolution filters related to 960 feature maps, whose kernel size is 8×8, the same as that of feature maps. After training the model by massive faces, SAV sets corresponding weights for each feature map, taking both the strength of abstract representation and spatial information of the input face into account. Thus, the feature vector conveys more discriminative cues than GAP, benefiting FR. Compared with the ones of the counterparts, the feature learned by Led3D is more compact and separable.

15.3.2.4 Bigger Training Data

A data augmentation scheme is specifically proposed to improve the quantity. We also consider a new scenario that probably appears in the real world, namely, 3D FR across quality, where the gallery set includes high-quality data and the probe samples are of low-quality, and discuss how to handle the data for this case.

Data Augmentation Since previous public databases of low-quality data are small and CNNs are data hungry, we launch data augmentation techniques to generate more samples for training Led3D. Apart from the widely used pose augmentation (out-of-plane rotation), we propose two new schemes (shape jittering and shape scaling) to adapt to 3D FR on low-quality data. The generated samples are shown in Fig. 15.11c.

1. *Pose Generating.* Given a point-cloud 3D face, faces with richer pose variations are synthesized by adjusting the virtual camera parameters. We generate new facial point-clouds in the range of $[-60°, 60°]$ on yaw and $[-40°, 40°]$ on pitch, with the interval of 20°. For each generated face, we compute depth and normal images.
2. *Shape Jittering.* Low-quality faces (in Lock3DFace) usually have very rough surfaces. Motivated by this, we add the Gaussian noise to augmented 3D faces to simulate such changes. By properly controlling the noise level, we do not change the identity information. The Gaussian noise we use has 0 mean and 2e-5 variance, on the normalized point-clouds. We find that such parameters lead to significant performance enhancement.
3. *Shape Scaling.* When the faces are collected by 3D cameras, the distance between the face and the camera is not fixed. Actually, there exist moderate changes on that distance, and the cropped faces thus have varying sizes. To simulate this change, firstly, we binarize the depth face to compute a mask image. Then, we zoom in the depth face image with 1.1 times. Finally, we render the new depth face, which is cropped from the enlarged one via the mask.

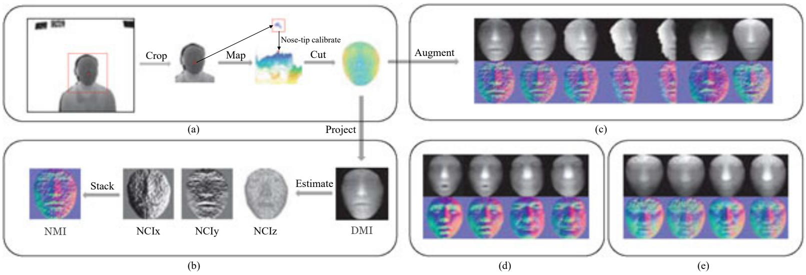


Fig. 15.11 Pipeline of data improvement: **a** facial surface refinement; **b** generation of Depth Map Images (DMI) and Normal Map Images (NMI); **c** augmentation of 3D face samples; **d** generation of low-quality data from high-quality ones; and **e** generation of low-quality and high-quality data of virtual identities. From Mu et al. [44], with permission

Cross-quality Data Generation High-precision scanners capture high-quality 3D faces with smooth surfaces, leading to better FR performance. However, such scanners are in big volume and expensive, thus difficult to pervade for on-line scenarios. In comparison, low-quality sensors are more widely used. In the real world, a popular setting is: high-quality data work as gallery and low-quality data are used as probes. To simulate this setting, we convert the high-quality data (from FRGC v2 and Bospohrus in our case) with smooth surfaces to low-quality ones with rough surfaces. Random disturbance is added on high-quality face point-clouds to generate low-quality like depth maps.

Specifically, a 3D face and the disturbance can be represented as $F_i = [x_p, y_p, z_p]$ and $D_i = [d_p]$, respectively. Here $i = 1, ..., N$, $p = 1, ..., P$ and $D_i \sim N(0, 16)$; N is the number of 3D faces and P is the number of vertices of a 3D face. The generated low-quality like face $F_i^l = [x_p, y_p, z_p^l]$ can be obtained by $z_p^l = z_p + d_p$. Then, we use a maximum filter with a kernel size of 3×3 on every generated face to amplify the effect of the disturbance. Examples of generated low-quality faces from high-quality ones are shown in Fig. 15.11d. Furthermore, we use the virtual ID generation method in [101] to generate new individuals (identities) to increase the data size for cross-quality model training. The sample is shown in Fig. 15.11e.

15.3.3 Experiments

15.3.3.1 Settings and Protocols

3D FR on Low-quality Data. All the depth face images (or normal face images) are resized to 128×128, and to adapt to other counterpart networks, the input image is scaled to suitable solutions. The models are pre-trained on the combination of FRGC v2 and Bosphorus, and then fine-tuned on Lock3DFace.

Cross-quality 3D FR. To explore this new scenario, experiments are performed on Bosphorus. The training set contains the augmented high-quality normal face data, the generated low-quality normal face data, and the synthesized virtual face data on FRGC v2, with totally 122,150 faces of 1,000 identities. For test, the first faces of the neutral expression in high-quality of all the 105 individuals are used as a gallery and the remaining ones are processed into a low quality as probes.

15.3.3.2 Results

3D FR on Low-quality Data. Table 15.4 reports the rank-one accuracies of Led3D model and four state-of-the-art CNNs on Lock3DFace, compared with the baseline method [43]. We can see that Led3D achieves the best average scores in all the settings, showing its effectiveness. However, for the training data without augmentation, the scores of all the CNN methods on the subset (PS) are lower than Baseline [43] using ICP-based registration. The reason lies in that the training data are not sufficient and do not contain faces with pose variations. Once augmentation techniques are applied to training data, the accuracies of CNN models are significantly improved on the test subsets.

Table 15.5 shows that Led3D outperforms the state-of-the-art methods, where the training and testing data are separated by subjects. The results for Inception V2 are reported by [96]. They pre-train Inception V2 on their private dataset, which contains 845K faces of 747 identities. Unlike [96], Led3D is trained from scratch and evaluated on depth faces and normal faces. The model reports an accuracy of 81.02% on depth faces, around 1.17% higher than that in [96]. In addition, it achieves 84.22% by concatenating the feature of depth and normal, suggesting that these two features have complementary information.

Cross-quality 3D FR. The results of cross-quality 3D FR are reported in Table 15.6. Led3D achieves 91.27% accuracy for HL (high-quality in gallery and low-quality in probe) and 90.7% for LL (low-quality in both gallery and probe), both of which are significantly superior to the ones reached by Inception V2, the major counterpart used in [96]. It illustrates that Led3D is also competent at recognizing 3D face across the change in data quality, where its generalization ability is highlighted.

Runtime. The run-time of the four CNNs and Led3D are evaluated on Jetson TX2, which is one of the fastest, most power-efficient embedded AI edge device. The run-time is computed on a single inference using MXNet 1.2 and python 2.7. The device is set in different power modes and computes in different processors. As shown in Table 15.7, the Led3D model runs at a speed of 136 FPS in the high-power mode, which is much faster than MobileNet V2. If using the ARM core process, it also achieves 15 FPS, faster than MobileNet V2 as well. It verifies that Led3D is efficient and can be deployed on edge devices to achieve real-time 3D FR using low-quality data.

Table 15.4 Performance comparison in terms of rank-one score on Lock3DFace using different training sets. From Mu et al. [44], with permission

Model	Training data	Evaluation type	Test subset				
			FE	OC	PS	TM	AVG
Baseline [43]	No augmentation	Video based	74.12	28.57	18.63	13.17	34.53
VGG-16 [97]	No augmentation	Video based	74.49	27.19	8.97	7.61	34.55
ResNet-34 [98]			63.06	21.81	12.92	5.82	30.2
Inception V2 [99]			78.07	35.36	**14.4**	7.46	39.13
MobileNet V2 [100]			73.72	27.49	10.75	7.01	34.73
Led3D			**79.78**	**36.95**	12.33	**19.85**	**41.65**
VGG-16 [97]	With augmentation	Video based	79.63	36.95	21.7	12.84	42.8
ResNet-34 [98]			62.83	20.32	22.56	5.07	32.23
Inception V2 [99]			80.48	32.17	33.23	12.54	44.77
MobileNet V2 [100]			85.38	32.77	28.3	10.6	44.92
Led3D			**86.94**	**48.01**	**37.67**	**26.12**	**54.28**

FE: expression. PS: pose. OC: occlusion. TM: time.

Table 15.5 Performance in terms of rank-one recognition rate (%) of 3D FR using low-quality data on Lock3DFace using the protocol in [96]

Test	Inception V2 [96]	Led3D	
Subset	Depth	Depth	Depth&Normal
NU	99.55	**99.62**	**99.62**
FE	98.03	97.62	**98.17**
PS	65.26	64.81	**70.38**
OC	**81.62**	68.93	78.10
TM	55.79	64.97	**65.28**
Total	79.85	81.02	**84.22**

15.4 Nonlinear 3D Morphable Face Model

Zheng et al. [73] present a novel 3D morphable face model, namely ImFace, to learn a nonlinear and continuous space with Implicit Neural Representations (INRs). As Fig. 15.12 shows, it builds two explicitly disentangled deformation fields to model complex shapes associated with identities and expressions, respectively, and designs an improved learning

Table 15.6 Performance of cross-quality 3D FR (HL: high-quality in gallery and low-quality in probe; LL: low-quality in both gallery and probe). From Mu et al. [44], with permission

Model	Bosphorus	
	HL	LL
Inception V2 [99]	78.56	77.23
Led3D	**91.27**	**90.70**

Table 15.7 Comparison in terms of running speed (FPS) with four CNNs on Jetson TX2. Low-Power Mode means the default setting of Max-Q, and High-Power Mode means the maximum clock frequency setting of Max-N. From Mu et al. [44], with permission

Model	Jetson TX2			
	Low-power mode		High-power mode	
	GPU	ARM	GPU	ARM
VGG-16 [97]	7.09	0.43	11.13	0.88
ResNet-34 [98]	8.44	0.58	13.08	1.14
Inception V2 [99]	24.33	2.90	39.02	5.16
MobileNet V2 [100]	35.41	3.16	60.41	5.62
Led3D	**46.26**	**9.77**	**135.93**	**15.66**

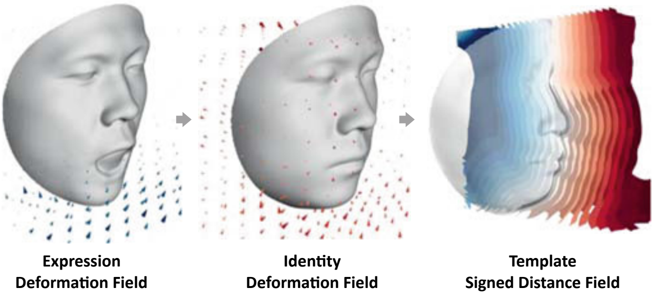


Fig. 15.12 ImFace encodes complex face variations by two explicitly disentangled deformation fields with respect to a template face, resulting in a morphable implicit representation for 3D face

strategy to extend embeddings of expressions to allow more diverse changes. A Neural Blend-Field is further introduced to learn sophisticated details by adaptively blending a series of local fields.

15.4.1 Method

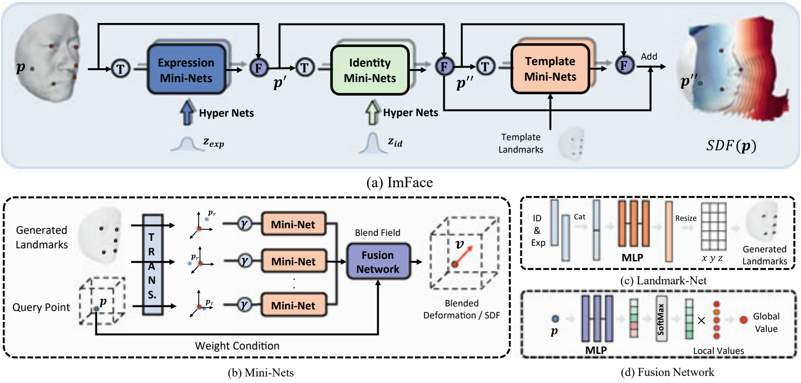
15.4.1.1 Disentangled INRs Network

The fundamental idea of INRs is to train a neural network to fit a continuous function f, which implicitly represents surfaces through level-sets. The function can be defined in various formats, e.g., occupancies [62], Signed Distance Function (SDF) [61], or Unsigned Distance Function (UDF) [102]. ImFace exploits a deep SDF conditioned on the latent embeddings of both expression and identity for comprehensive face representations. It outputs the signed distance s from a query point:

$$f : (\mathbf{p}, \mathbf{z}_{exp}, \mathbf{z}_{id}) \in \mathbb{R}^3 \times \mathbb{R}^{d_{exp}} \times \mathbb{R}^{d_{id}} \mapsto s \in \mathbb{R}, \tag{15.7}$$

where $\mathbf{p} \in \mathbb{R}^3$ is the coordinate of the query point in 3D space, $\mathbf{z}_{exp}$ and $\mathbf{z}_{id}$ denote the expression and identity embeddings, respectively.

The goal of ImFace is to learn a neural network to parameterize f, making it satisfy the genuine facial shape priors. As shown in Fig. 15.13, the proposed network for Imface is composed of three sub-networks (Mini-Nets), which explicitly disentangles the learning process of face shape morphs, ensuring that inter-individual differences and fine-grained deformations can be accurately learned. In particular, the first two Mini-Nets learn separate deformation fields associated with expression and identity-variation respectively, and the Template Mini-Nets automatically learn a signed distance field of template face shape. Along with the network design, an improved auto-decoder embedding learning strategy is introduced, which extends the latent space of expressions to allow higher deformation variety.



(a) ImFace

(b) Mini-Nets

(c) Landmark-Net

(d) Fusion Network

Fig. 15.13 ImFace overview. The network consists of three sub-networks (Mini-Nets) to explicitly disentangle shape morphs into separate deformation fields, where the Expression and Identity Mini-Nets are associated with expression and identity deformations, respectively, and the Template Mini-Nets learn the SDF of a template face space. From Zheng et al. [73], with permission

All fields above are implemented by a shared Mini-Nets architecture, where the entire facial deformation or geometry is further decomposed into semantically meaningful parts and encoded by a set of local field functions, so that rich facial details can be well captured. A lightweight module conditioned on the query point position, i.e., Fusion Network, is stacked at the end of Mini-Nets to adaptively blend the local fields. As such, an elaborate Neural Blend-Field is achieved. The three core components of ImFace work for different purposes and correspondingly their structures are slightly changed. We briefly describe them as follows:

Expression Mini-Nets (ExpNet) The facial deformations introduced by expressions are represented by ExpNet $\mathcal{E}$, which learns an observation-to-canonical warping for every face scan:

$$\mathcal{E} : (\mathbf{p}, \mathbf{z}_{exp}, l) \mapsto \mathbf{p}' \in \mathbb{R}^3, \tag{15.8}$$

$l \in \mathbb{R}^{k\times3}$ denotes k 3D landmarks on a observed face generated by a Landmark-Net $\eta : (\mathbf{z}_{exp}, \mathbf{z}_{id}) \mapsto l$, introduced to localize the query point $\mathbf{p}$ in the Neural Blend-Field. A point $\mathbf{p}$ in the observation space is deformed by $\mathcal{E}$ to a new point $\mathbf{p}'$ in the person-specific canonical space, which represents faces with a neutral expression.

Identity Mini-Nets (IDNet) To model shape morphing among individuals, the IDNet $\mathcal{I}$ further warps the canonical space to a template shape space shared by all faces:

$$\mathcal{I} : (\mathbf{p}', \mathbf{z}_{id}, l') \mapsto (\mathbf{p}'', \delta) \in \mathbb{R}^3 \times \mathbb{R}, \tag{15.9}$$

where $l' \in \mathbb{R}^{k\times3}$ denotes k landmarks on the canonical face generated by another Landmark-Net conditioned only on the identity embedding $\eta' : \mathbf{z}_{id} \mapsto l'$, and $\mathbf{p}''$ is the deformed point in the template space. To cope with the possible non-existent correspondences generated during preprocessing, $\mathcal{I}$ additionally predicts a residual term $\delta \in \mathbb{R}$ to correct the predicted SDF value s_0, similar to [69].

Template Mini-Nets (TempNet) TempNet $\mathcal{T}$ learns a signed distance field of the shared template face:

$$\mathcal{T} : (\mathbf{p}'', l'') \mapsto s_0 \in \mathbb{R}, \tag{15.10}$$

where $l'' \in \mathbb{R}^{k\times3}$ denotes k landmarks on the template face, which is averaged on the whole training set, and s_0 denotes uncorrected SDF value. The final SDF value of a query point is calculated via $s = s_0 + \delta$, and the ImFace model can be ultimately formulated as

$$f(\mathbf{p}) = \mathcal{T}(\mathcal{I}_{\mathbf{p}''}(\mathcal{E}(\mathbf{p}, \mathbf{z}_{exp}), \mathbf{z}_{id})) + \mathcal{I}_{\delta}(\mathcal{E}(\mathbf{p}, \mathbf{z}_{exp}), \mathbf{z}_{id}). \tag{15.11}$$

15.4.1.2 Neural Blend-Field

The Mini-Nets have a common architecture shared across the sub-networks $\mathcal{E}$, $\mathcal{I}$, and $\mathcal{T}$. It is specifically designed to learn a continuous field function $\boldsymbol{\psi} : \mathbf{x} \in \mathbb{R}^3 \mapsto v$ to produce a Neural Blend-Field for sophisticated face representations. In particular, to overcome the limited shape expressivity of a single network, a face space is decomposed into a set of semantically meaningful local regions, and their deformations or signed distance fields are individually learned before blending. Such design is inspired by the recent INRs study [103] on the human body, which introduces linear blend skinning algorithm [74] to enable the network to learn from separate body parts transformation. In order to better represent detailed facial surface, the constant transformation term in the original linear blend skinning algorithm is replaced with $\boldsymbol{\psi}_n(\mathbf{x} - l_n)$, and Neural Blend-Field is defined as

$$v = \boldsymbol{\psi}(\mathbf{x}) = \sum_{n=1}^{k} w_n(\mathbf{x}) \boldsymbol{\psi}_n(\mathbf{x} - l_n), \tag{15.12}$$

where l_n is a parameter that describes the n-th local region, $w_n(\mathbf{x})$ is the n-th blend weight, and $\boldsymbol{\psi}_n(\mathbf{x} - l_n)$ is the corresponding local field. By such, the blending is performed on a series of local fields, rather than calculating a weighted average of the output values v (such as deformation) of some fixed positions, leading to a more powerful representation capability in handling complicated local features.

Five landmarks located at outer eye corners, mouth corners, and nose tip are utilized to describe the local regions $(l_n \in \mathbb{R}^3)_{n=1}^5$, and each is assigned a tiny MLP with sinusoidal activations to generate the local field, denoted as $\boldsymbol{\psi}_n$. To well capture high-frequency local variations, sinusoidal positional encoding γ on the coordinate $\mathbf{x} - l_n$ is leveraged. At the end of Mini-Nets, a lightweight Fusion Network is introduced, which is implemented by a 3-layer MLP with softmax to predict the blend weights $(w_n \in \mathbb{R}^+)_{n=1}^5$, conditioned on the absolute coordinate of input $\mathbf{x}$.

Deformation Formulation The deformations is formulated with a SE(3) field $(\boldsymbol{\omega}, \mathbf{v}) \in \mathbb{R}^6$, where $\boldsymbol{\omega} \in so(3)$ is a rotate vector representing the screw axis and the angle of rotation. The deformed coordinates $\mathbf{x}'$ can be calculated by $e^{\boldsymbol{\omega}}\mathbf{x} + \mathbf{t}$, where the rotation matrix $e^{\boldsymbol{\omega}}$ (exponential map form of Rodrigues' formula) is written as

$$e^{\boldsymbol{\omega}} = \mathbf{I} + \frac{\sin \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|}\boldsymbol{\omega}^\wedge + \frac{1 - \cos \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2}(\boldsymbol{\omega}^\wedge)^2, \tag{15.13}$$

and the translation $\mathbf{t}$ is formulated as

$$\mathbf{t} = \left[\mathbf{I} + \frac{1 - \cos \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2}\boldsymbol{\omega}^\wedge + \frac{\|\boldsymbol{\omega}\| - \sin \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^3}(\boldsymbol{\omega}^\wedge)^2\right]\mathbf{v}, \tag{15.14}$$

where $\boldsymbol{\omega}^\wedge$ denotes the skew-symmetric matrix of vector $\boldsymbol{\omega}$.

Hyper Nets To obtain a more compact and expressive latent space, a meta-learning approach [104] is further introduced. In particular, a Hyper Net ϕ_n is implemented by an MLP and predicts the instance-specific parameters for ExpNet $\mathcal{E}$ and IDNet $\mathcal{I}$. It takes a latent code $\mathbf{z}$ as input and generates the parameters for the neurons in a Mini-Net $\boldsymbol{\psi}_n$, so that the learned face representations possess higher variety.

15.4.1.3 Improved Expression Embedding Learning

The auto-decoder framework proposed by [61] has been widely adopted in INRs to jointly learn embeddings and network parameters. Face modeling, ImFace improves the learning strategy by treating each non-neutral face scan as a unique expression and generating a specific embedding for it. In this way, the latent space is significantly extended, which enables ExpNet to represent diverse and fine-grained details. Nevertheless, there exists a potential failure mode in that the identity properties are tangled into expression space again, and IDNet $\mathcal{I}$ collapses to an identity mapping. To tackle this challenge, the ExpNet $\mathcal{E}$ is suppressed when the current training sample is a neutral face, written as

$$\mathcal{E}(\mathbf{p}_{nu}, \mathbf{z}_{exp}, l) \equiv \mathbf{p}_{nu}, \tag{15.15}$$

where $\mathbf{p}_{nu}$ denotes a point from neutral face. By applying such learning strategy, IDNet and TempNet jointly learn shape representations on neutral faces, and ExpNet focuses only on expression deformations. Moreover, expression annotations are no longer required during training.

15.4.1.4 Loss Functions

ImFace is trained with a series of loss functions to learn plausible face shape representations as well as dense correspondence.

Reconstruction Loss The basic SDF loss is applied to learn implicit fields:

$$\mathcal{L}^i_{sdf} = \lambda_1 \sum_{\mathbf{p}\in\Omega_i} |f(\mathbf{p}) - \bar{s}| + \lambda_2 \sum_{\mathbf{p}\in\Omega_i} (1 - \langle \nabla f(\mathbf{p}), \bar{\mathbf{n}} \rangle), \tag{15.16}$$

where $\bar{s}$ and $\bar{\mathbf{n}}$ denote the ground-truth SDF values and the field gradients, respectively, Ω_i is the sampling space of the face scan i, and λ indicates the trade-off parameter.

Eikonal Loss To obtain reasonable fields throughout the network, multiple Eikonal losses are used to enforce the L-2 norm of spatial gradients to be unit:

$$\mathcal{L}^i_{eik} = \lambda_3 \sum_{\mathbf{p}\in\Omega_i} \left(| \|\nabla f(\mathbf{p})\| - 1 | + | \|\nabla \mathcal{T}(\mathcal{I}(\mathbf{p}'))\| - 1 | \right), \tag{15.17}$$

where $\mathcal{L}^i_{eik}$ enables the network to satisfy Eikonal constraint [63] in the observation space and canonical space simultaneously, which also contributes to a reasonable correspondence along face deformations at all network stages.

Embedding Loss It regularizes the embeddings with a zero-mean Gaussian prior:

$$\mathcal{L}^i_{emb} = \lambda_4 \left(\|\mathbf{z}_{exp}\|^2 + \|\mathbf{z}_{id}\|^2 \right). \tag{15.18}$$

Landmark Generation Loss The l_1-loss is used to learn the Landmark-Net η, η', and parameters l'':

$$\mathcal{L}^i_{lmk_g} = \lambda_5 \sum_{n=1}^{k} \left(|l_n - \bar{l}^i_n| + |l'_n - \bar{l'}_n| \right), \tag{15.19}$$

where $\bar{l}^i$ denotes the k labeled landmarks on sample i, $\bar{l'}$ denotes the landmarks on the corresponding neutral face.

Landmark Consistency Loss This loss is exploited to guide the deformed 64 landmarks to be located at the corresponding positions on the ground-truth neutral and template faces for better correspondence performance:

$$\mathcal{L}^i_{lmk_c} = \lambda_6 \sum_{n=1}^{64} \left(|\mathcal{E}(l_n) - \bar{l'}_n| + |\mathcal{I}(\mathcal{E}(l_n)) - \bar{l''}_n| \right). \tag{15.20}$$

Residual Constraint To avoid the situation that the residual item δ learns too much template face information and further downgrades the morphable model, δ is penalized by

$$\mathcal{L}^i_{res} = \lambda_7 \sum_{\mathbf{p} \in \Omega_i} |\delta(\mathbf{p})|. \tag{15.21}$$

The total training loss is calculated on all face samples indexed by i, finally formulated as

$$\mathcal{L} = \sum_i (\mathcal{L}^i_{sdf} + \mathcal{L}^i_{eik} + \mathcal{L}^i_{emb} + \mathcal{L}^i_{lmk_g} + \mathcal{L}^i_{lmk_c} + \mathcal{L}^i_{res}). \tag{15.22}$$

In the testing phase, for each 3D face indexed by j, the following objective is minimized to obtain its latent embeddings and the reconstructed 3D face:

$$\arg\min_{\mathbf{z}_{exp}, \mathbf{z}_{id}} \sum_j (\mathcal{L}^j_{sdf} + \mathcal{L}^j_{eik} + \mathcal{L}^j_{emb}). \tag{15.23}$$

15.4.2 Experiments

Extensive experiments are performed on the FaceScape [48] database for both subjective and objective evaluations on ImFace.

15.4.2.1 Reconstruction

Qualitative Evaluation In the testing phase, ImFace is used to fit face scans by optimizing Eq. (15.23). Figure 15.14 visualizes the reconstruction results achieved by different models, where each column corresponds to a test person with a non-neutral expression. The results also include the unseen expressions during learning. In particular, i3DMM [72] is the first deep implicit model for the human head, but it is less capable of capturing complicated deformations and fine-grained shape details under a relatively intricate circumstance, resulting in artifacts on the reconstructed faces. FLAME [47] is able to well present the

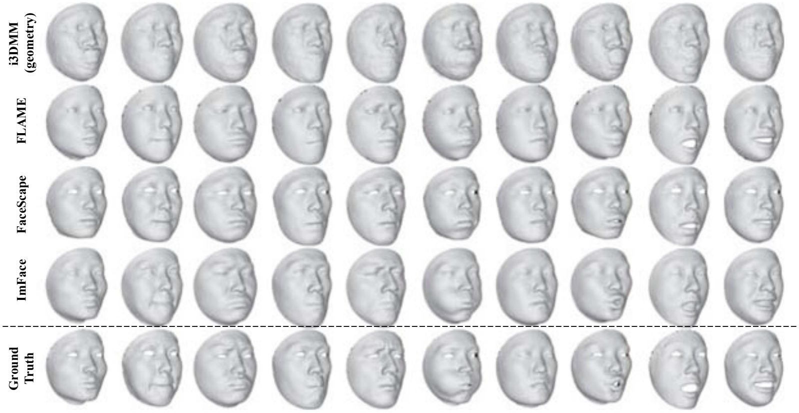


Fig. 15.14 Reconstruction comparison with i3DMM [72], FLAME [47], and FaceScape [48]. From Zheng et al. [73], with permission

Table 15.8 Quantitative comparison with the state-of-the-art methods (†Lower is better; ¶Higher is better)

Metrics	Chamfer $(mm)^{\dagger}$	F-score@0.001 ¶
i3DMM [72]	1.635	42.26
FLAME [47]	0.971	64.73
FaceScape [48]	0.929	67.09
ImFace	**0.625**	**91.11**

identity characteristics, but is not so competent at representing nonlinear deformations, that it delivers stiff facial expressions. FaceScape [48] performs more favorably than FLAME mainly due to high-quality training scans and that test faces are included by training set, but it still cannot describe expression morphs precisely. Comparatively, ImFace reconstructs faces with more accurate identity and expression properties, and it is able to capture subtle and rich nonlinear facial muscle deformations such as frowns and pouts, which is achieved by fewer latent parameters.

Quantitative Evaluation Symmetric Chamfer distance and F-score are used as metrics, and the threshold of F-score is set to 0.001 for a strict comparison. The results are shown in Table 15.8. As we can see, ImFace exceeds the compared counterparts by a large margin under both metrics, which clearly validates its effectiveness.

15.4.2.2 Correspondence

In contrast to existing methods that generally requires accurate face registration, correspondences can be automatically learned in INRs models, and the corresponding training critic

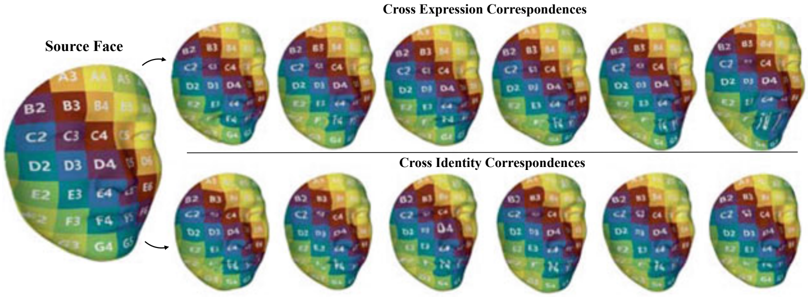

Cross Expression Correspondences
Source Face
Cross Identity Correspondences

Fig. 15.15 Correspondence results. The leftmost source face is morphed into multiple expressions (upper row) and identities (lower row). From Zheng et al. [73], with permission

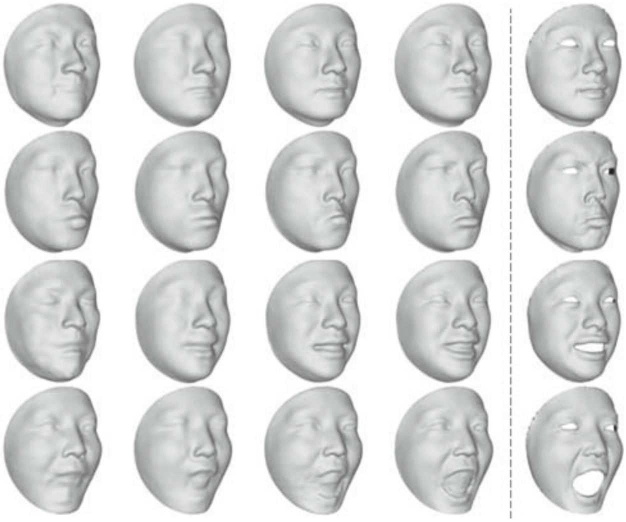
is specially designed to enhance such a feature. This evaluation aims to check this point. Given two 3D faces, ImFace is used to fit them and deform the densely sampled points to the template space, so that point-to-point correspondences can be determined by nearest neighbor search. Figure 15.15 visualizes the correspondences generated by ImFace, where color patterns are manually painted on the shapes to better check the quality. It can be inspected that tiny internal texture dispersion indeed occurs around mouth corners, it is mainly because facial shape changes drastically in these local areas under different expressions. Nevertheless, ImFace is able to establish pleasing overall correspondences across various expressions and identities.

15.4.2.3 Ablation Study

ImFace is built on the following core ideas: disentangled deformation fields, Neural Blend-Field, and improved expression embedding learning. Therefore, ablation studies are performed to experimentally verify the credit of the corresponding architecture designs and learning strategy.

On Disentangled Deformation Fields To highlight the disentangled deformation learning process, a compared network that contains only one deformation field to learn face shape morphs universally is built. Accordingly, $\mathbf{z}_{exp}$ and $\mathbf{z}_{id}$ are concatenated as the input of the hyper net. Figure 15.16a provides a demonstration. In spite of some fine-grained details brought by other designs, there exists a chaos on the reconstructed faces, especially for the ones with dramatic expressions. The quantitative results in Table 15.9 also indicate the significance of decoupled deformation learning.

On Neural Blend-Field The Neural Blend-Field in $\mathcal{E}$, $\mathcal{I}$, $\mathcal{T}$ is replaced with vanilla MLPs of the same amount of parameters, which directly predict the global deformations or SDF values of an entire face. As shown in Fig. 15.16b, a visible blur appears due to the limited capability in learning high-frequency fine details. The quantitative evaluation results



(a) w/o dist. (b) w/o blend (c) w/o extend. (d) ImFace (e) GT

Fig. 15.16 Qualitative ablation study results. From Zheng et al. [73], with permission

Table 15.9 Quantitative ablation study results. From Zheng et al. [73], with permission

Metrics	Chamfer (mm) †	F-score@0.001 ¶
Ours w/o dist.	0.772	82.70
Ours w/o blend	0.767	82.37
Ours w/o extend.	0.705	86.98
ImFace	**0.625**	**91.11**

in Table 15.9 also validate the necessity of Neural Blend-Field in learning sophisticated representations.

On Improved Embedding Learning Strategy This strategy is introduced to learn diverse and fine-grained facial deformations. As shown in Fig. 15.16c, when restricting the number of expression embeddings to be the same with expression categories, the generated expressions tend to be average. Moreover, for exaggerated expressions, such as mouth stretch, the compared model can hardly converge to a reasonable state.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 62176012, 62202031).

References

1. Turk, M., Pentland, A.: Eigenfaces for recognition. JOCN **3**(1), 71–86 (1991)
2. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. TPAMI **19**(7), 711–720 (1997)
3. Wiskott, L., Fellous, J.-M., Kuiger, N., Von der Malsburg, C.: Face recognition by elastic bunch graph matching. TPAMI **19**(7), 775–779 (1997)
4. Ahonen, T., Hadid, A., Pietikainen, M.: Face recognition with local binary patterns. In: ECCV (2004)
5. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. TPAMI **31**(2), 210–227 (2009)
6. Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: CVPR (2005)
7. Savran, A., Alyuz, N., Dibeklioglu, H., Celiktutan, O., Gokberk, B., Sankur, B., Akarun, L.: Bosphorus database for 3d face analysis. In: BioID (2008)
8. Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.: A 3d facial expression database for facial behavior research. In: FG (2006)
9. Yin, L., Chen, X., Sun, Y., Worm, T., Reale, M.: A high-resolution 3d dynamic facial expression database. In: FG (2008)
10. Kakadiaris, I., Passalis, G., Toderici, G., Murtuza, M., Lu, Y., Karampatziakis, N., Theoharis, T.: Three dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. TPAMI **29**(4), 640–649 (2007)
11. Alyuz, N., Gokberk, B., Akarun, L.: Regional registration for expression resistant 3-d face recognition. TIFS **5**(3), 425–440 (2010)
12. Huang, D., Ardabilian, M., Wang, Y., Chen, L.: 3-D face recognition using elbp-based facial description and local feature hybrid matching. TIFS **7**(5), 1551–1565 (2012)
13. Drira, H., Ben Amor, B., Srivastava, A., Daoudi, M., Slama, R.: 3D face recognition under expressions, occlusions, and pose variations. TPAMI **35**(9), 2270–2283 (2013)
14. Mian, A., Bennamoun, M., Owens, R.: An efficient multimodal 2d–3d hybrid approach to automatic face recognition. TPAMI **29**(11), 1927–1943 (2007)
15. Mian, A., Bennamoun, M., Owens, R.: Keypoint detection and local feature matching for textured 3d face recognition. IJCV **79**(1), 1–12 (2008)
16. Huang, D., Ben Soltana, W., Ardabilian, M., Wang, Y., Chen, L.: Textured 3d face recognition using biological vision-based facial representation and optimized weighted sum fusion. In: CVPRW (2011)
17. Huang, D., Ardabilian, M., Wang, Y., Chen, L.: Oriented gradient maps based automatic asymmetric 3d-2d face recognition. In: ICB (2012)
18. Chu, B., Romdhani, S., Chen, L.: 3D-aided face recognition robust to expression and pose variations. In: CVPR (2014)
19. Smeets, D., Claes, P., Hermans, J., Vandermeulen, D., Suetens, P.: A comparative study of 3-D face recognition under expression variations. TSMC Part C (Appl. Rev.) **42**(5), 710–727 (2011)
20. Drira, H., Amor, B.B., Srivastava, A., Daoudi, M., Slama, R.: 3D face recognition under expressions, occlusions, and pose variations. TPAMI **35**(9), 2270–2283 (2013)

21. Soltanpour, S., Boufama, B., Wu, Q.J.: A survey of local feature methods for 3D face recognition. Pattern Recogn. **72**, 391–406 (2017)
22. Zhou, S., Xiao, S.: 3D face recognition: a survey. HCIS **8**(1), 1–27 (2018)
23. Sandbach, G., Zafeiriou, S., Pantic, M., Yin, L.: Static and dynamic 3D facial expression recognition: A comprehensive survey. IVC **30**(10), 683–697 (2012)
24. Gilani, S.Z., Mian, A.: Learning from millions of 3D scans for large-scale 3D face recognition. In: CVPR (2018)
25. Kim, D., Hernandez, M., Choi, J., Medioni, G.: Deep 3D face identification. In: IJCB (2017)
26. Richardson, E., Sela, M., Kimmel, R.: 3D face reconstruction by learning from synthetic data. In: 3DV (2016)
27. Gilani, S.Z., Mian, A.: Towards large-scale 3D face recognition. In: DICTA (2016)
28. Oyedotun, O.K., Demisse, G., El Rahman Shabayek, A., Aouada, D., Ottersten, B.: Facial expression recognition via joint deep learning of rgb-depth map latent representations. In: ICCV workshop (2017)
29. Li, H., Sun, J., Xu, Z., Chen, L.: Multimodal 2D+ 3D facial expression recognition with deep fusion convolutional neural network. TMM **19**(12), 2816–2831 (2017)
30. Amberg, B., Romdhani, S., Vetter, T.: Optimal step nonrigid ICP algorithms for surface registration. In: CVPR (2007)
31. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3D classification and segmentation. In: CVPR (2017)
32. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NIPS (2017)
33. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on X-transformed points. In: NIPS (2018)
34. Bhople, A.R., Shrivastava, A.M., Prakash, S.: Point cloud based deep convolutional neural network for 3D face recognition. In: MTA (2020)
35. Chen, Z., Huang, D., Wang, Y., Chen, L.: Fast and light manifold cnn based 3D facial expression recognition across pose variations. In: ACM MM (2018)
36. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. In: ICCV workshops (2015)
37. Boscaini, D., Masci, J., Rodolá , E., Bronstein, M.: Learning shape correspondence with anisotropic convolutional neural networks. In: NIPS (2016)
38. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: CVPR (2017)
39. Xia, L., Chen, C., Aggarwal, J.K.: View invariant human action recognition using histograms of 3d joints. In: CVPRW (2012)
40. Xia, L., Chen, C., Aggarwal, J.K.: Human detection using depth information by Kinect. In: CVPRW (2011)
41. Tao, D., Jin, L., Yang, Z., Li, X.: Rank preserving sparse learning for Kinect based scene classification. IEEE Trans. Cybern. **43**(5), 1406–1417 (2013)
42. Min, R., Choi, J., Medioni, G., Dugelay, J.-L.: Real-time 3D face identification from a depth camera. In: ICPR (2012)
43. Zhang, J., Huang, D., Wang, Y., Sun, J.: Lock3dface: A large-scale database of low-cost kinect 3d faces. In: ICB (2016)
44. Mu, G., Huang, D., Hu, G., Sun, J., Wang, Y.: Led3d: A lightweight and efficient deep approach to recognizing low-quality 3d faces. In: CVPR (2019)
45. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: A 3d facial expression database for visual computing. IEEE TVCG **20**(4), 413–425 (2013)

46. James, B., Roussos, A., Ponniah, A., Dunaway, D., Zafeiriou, S.: Large scale 3d morphable models. IJCV **126**(2), 233–254 (2018)
47. Li, T., Bolkart, T., Black, M., Li, H., Romero, J.: Learning a model of facial shape and expression from 4D scans. ACM TOG **36**(6), **194**(17), 1–194 (2017)
48. Yang, H., Zhu, H., Wang, Y., Huang, M., Shen, Q., Yang, R., Cao, X.: Facescape: A large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In: CVPR (2020)
49. Volker, B., Thomas, V.: A morphable model for the synthesis of 3D faces. In: SIGGRAPH (1999)
50. Patel, A., Smith, W.: 3d morphable face models revisited. In: CVPR (2009)
51. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter T.: A 3d face model for pose and illumination invariant face recognition. In: AVSS (2009)
52. Vlasic, D., Brand, M., Pfister, H., Popovic, J.: Face transfer with multilinear models. In: ACM SIGGRAPH Courses (2006)
53. Brunton, A., Bolkart, T., Wuhrer, S.: Multilinear wavelets: A statistical shape space for human faces. In: ECCV (2014)
54. Bolkart, T., Wuhrer, S.: A groupwise multilinear correspondence optimization for 3d faces. In: ICCV (2015)
55. Tran, L., Liu, X.: Nonlinear 3d face morphable model. In: CVPR (2018)
56. Tran, L., Liu, F., Liu, X.: Towards high-fidelity nonlinear 3d face morphable model. In: CVPR (2019)
57. Bagautdinov, T., Wu, C., Saragih, J., Fua, P., Sheikh, Y.: Modeling facial geometry using compositional vaes. In: CVPR (2018)
58. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3D faces using convolutional mesh autoencoders. In: ECCV (2018)
59. Bouritsas, G., Bokhnyak, S., Ploumpis, S., Bronstein, M., Zafeiriou, S.: Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In: ICCV (2019)
60. Chen, Z., Kim, T.: Learning Feature Aggregation for Deep 3D Morphable Models. In: CVPR (2021)
61. Park, J., Florence, P., Straub, J., Newcombe, Ri., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: CVPR (2019)
62. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: CVPR (2019)
63. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit Geometric Regularization for Learning Shapes. In: ICML (2020)
64. Lipman, Y.: Phase Transitions, Distance Functions, and Implicit Neural Representations. In: ICML (2021)
65. Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In: NeurIPS (2019)
66. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In: NeurIPS (2021)
67. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W., Funkhouser, T.: Learning shape templates with structured implicit functions. In: ICCV (2019)
68. Zhang, J., Yao, Y., Quan, L.: Learning signed distance field for multi-view surface reconstruction. In: ICCV (2021)
69. Deng, Y., Yang, J., Tong, X.: Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In: CVPR (2021)
70. Liu, F., Liu, X.: Learning implicit functions for topology-varying dense 3D shape correspondence. In: NeurIPS (2020)

71. Zheng, Z., Yu, T., Dai, Q., Liu, Y.: Deep implicit templates for 3D shape representation. In: CVPR (2021)
72. Yenamandra, T., Tewari, A., Bernard, F., Seidel, H., Elgharib, Mo., Cremers, D., Theobalt, C.: i3DMM: Deep implicit 3D morphable model of human heads. In: CVPR (2021)
73. Zheng, M., Yang, H., Huang, D., Chen, L.: ImFace: A nonlinear 3D morphable face model with implicit neural representations. In: CVPR (2022)
74. Lewis, J.P., Cordner, M., Fong, N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In: SIGGRAPH (2000)
75. Tola, E., Lepetit, V., Fua, P.: DAISY: An efficient dense descriptor applied to wide-baseline stereo. TPAMI **32**(5), 815–830 (2009)
76. Huang, D., Zhu, C., Wang, Y., Chen, L.: HSOG: A novel local image descriptor based on histograms of the second-order gradients. TIP **23**(11), 4680–4695 (2014)
77. Szegedy, C., Liu, W., Jia, Y., Sermanet, Pi., Reed, S., Anguelov, Dr., Erhan, D., Vanhoucke, Vi., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
78. Koenderink, J., Van, D., Andrea, J.: Surface shape and curvature scales. IVC **10**(8), 557–564 (1992)
79. Yang, X., Huang, D., Wang, Y., Chen, L.: Automatic 3D facial expression recognition using geometric scattering representation. In: AFGR (2015)
80. Zhen, Q., Huang, D., Wang, Y., Chen, L.: Muscular movement model-based automatic 3D/4D facial expression recognition. In: TMM (2016)
81. Qin, Y., Han, X., Yu, H., Yu, Y., Zhang, J.: Fast and exact discrete geodesic computation based on triangle-oriented wavefront propagation. TOG **35**(4), 1–13 (2016)
82. Wang, J., Yin, L., Wei, X., Sun, Y.: 3D facial expression recognition based on primitive surface feature distribution. In: CVPR (2006)
83. Soyel, H., Demirel, H.: Facial expression recognition using 3D facial feature distances. In: ICIAR (2007)
84. Tang, H., Huang, T.S.: 3D facial expression recognition based on automatically selected features. In: CVPR Workshops (2008)
85. Gong, B., Wang, Y., Liu, J., Tang, X.: Automatic facial expression recognition on a single 3D face by exploring shape deformation. In: ACMMM (2009)
86. Berretti, S., Del Bimbo, A., Pala, P., Amor, B.B., Daoudi, M.: A set of selected SIFT features for 3D facial expression recognition. In: ICPR (2010)
87. Lemaire, P., Ben Amor, B., Ardabilian, M., Chen, L., Daoudi, M.: Fully automatic 3D facial expression recognition using a region-based approach. In: Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding (2011)
88. Lemaire, P., Ardabilian, M., Chen, L., Daoudi, M.: Fully automatic 3D facial expression recognition using differential mean curvature maps and histograms of oriented gradients. In: FG (2013)
89. Li, H., Chen, L., Huang, D., Wang, Y., Morvan, J.M.: 3D facial expression recognition via multiple kernel learning of multi-scale local normal patterns. In: ICPR (2012)
90. Zeng, W., Li, H., Chen, L., Morvan, J.M., Gu, X.D.: An automatic 3D expression recognition framework based on sparse representation of conformal images. In: FG (2013)
91. Azazi, A., Lutfi, S.L., Venkat, I., Fernández-Martínez, F.: Towards a robust affect recognition: Automatic facial expression recognition in 3D faces. In: Expert Systems with Applications (2015)
92. Li, H., Ding, H., Huang, D., Wang, Y., Zhao, X., Morvan, J.M., Chen, L.: An efficient multimodal 2D+ 3D feature-based approach to automatic facial expression recognition. CVIU **140**, 83–92 (2015)
93. Li, H., Sun, J., Wang, D., Xu, Z., Chen, L.: Deep Representation of Facial Geometric and Photometric Attributes for Automatic 3D Facial Expression Recognition. arXiv:1511.03015 (2015)

94. Mori, N., Suzuki, T., Kakuno, S.: Noise of acoustic Doppler velocimeter data in bubbly flows. J. Eng. Mech. **133**(1), 122–125 (2007)
95. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV (1998)
96. Cui, J., Zhang, H., Han, H., Shan, S., Chen, X: Improving 2d face recognition via discriminative face depth estimation. In: ICB (2018)
97. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
98. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
99. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
100. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: CVPR (2018)
101. Gilani, S.Z., Mian, A.: Learning from millions of 3D scans for large-scale 3D face recognition. In: CVPR (2018)
102. Chibane, J., Mir, A., Pons-Moll, G.: Neural unsigned distance fields for implicit function learning. In: NeurIPS (2020)
103. Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Qi., Zhou, X., Bao, H.: Animatable neural radiance fields for modeling dynamic human bodies. In: ICCV (2021)
104. Sitzmann, V., Martel Julien, N.P., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: NeurIPS (2020)

Appendix A: Code and Data

A

This third edition, while inheriting the title of Handbook of Face Recognition, will be composed of entirely new content describing the latest face recognition methodologies and technologies in the deep neural network framework. The book presents a unified resource of theory, algorithms, and implementations to bring students, researchers, and practitioners to all aspects of face recognition. The book not only presents the latest developments in methods and algorithms but also provides code and data to allow for hands-on learning and developing reproducible face recognition algorithms and systems by deep learning programming. The code and data will be released in the Github and will be updated subsequently to keep the materials up to date. The main face processing modules include

1. Face Detection
2. Facial Landmark Localization
3. Facial Attribute Analysis
4. Face Presentation Attack Detection
5. Face Feature Embedding
6. Video-based Face Recognition
7. Face Recognition with Synthetic Data
8. Uncertainty-aware Face Recognition
9. Reducing Bias in Face Recognition
10. Adversarial Attacks on Face Recognition
11. Heterogeneous Face Recognition
12. 3D Face Recognition

S. Z. Li et al. (eds.), *Handbook of Face Recognition*,
https://doi.org/10.1007/978-3-031-43567-6