Developing a Student Monitoring System for Online Classrooms Based on Face Recognition Approaches

Trong-Nghia Pham[1,2](✉), Nam-Phong Nguyen[1,2], Nguyen-Minh-Quan Dinh[1,2], and Thanh Le[1,2]

[1] Faculty of Information Technology, University of Science,
Ho Chi Minh city, Vietnam
{ptnghia,lnthanh}@fit.hcmus.edu.vn

[2] Vietnam National University, Ho Chi Minh city, Vietnam

Abstract. One of the primary activities that lectures usually do is to take a roll call. This activity not only helps lecturers determine the participation of students but also detect strangers in the classroom. When the number of students increases, lectures take more time to monitor and check students' attendance. We propose a student monitoring system based on facial recognition approaches to tackle that problem. With the recent development of deep learning techniques, many new approaches have made remarkable progress in face recognition. However, most of those approaches only focus on improving accuracy, while a practical end-to-end face recognition system demands good accuracy and reasonable runtime. We make adjustments and apply CenterFace for the face detection task and ArcFace for extracting embedding features from images to achieve high efficiency in both accuracy and speed. In addition, our proposed system is designed to be lightweight and scalable, capable of running in various environments, especially in a web browser. The results show that the system takes an average of 0.22 s to register a new face and 4.3 s for identifying a face in a database of 500 samples. Experiments also indicate that the system was less likely to misrecognize faces in most of our tests.

Keywords: Face recognition · Student monitoring system · CenterFace · FaceNet · ArcFace

1 Introduction

The online classroom has been on the trend in recent years, especially when the Covid pandemic has prevented students from going to school. It offers many advantages such as we can study from anywhere, the number of participants can be extended up to hundreds without limited space like a traditional classroom. Besides the benefits that online classroom offers, there are issues that lecturers have to deal with, like strangers appearing in the classroom, checking students'

N. T. Nguyen et al. (Eds.): ICCCI 2022, LNAI 13501, pp. 555–567, 2022.
https://doi.org/10.1007/978-3-031-16014-1_44

attendance becomes more challenging. We propose an end-to-end face recognition system to tackle those problems and improve the online teaching/studying experience.

Face recognition is a branch of study in the field of biometrics along with fingerprint recognition, iris recognition. Although face recognition is less reliable than fingerprint or iris recognition, it does not require specialized device or hardware. In addition, face recognition has a richer data source and requires less interaction to perform. That is why a face recognition system is easier and cost less to deploy.

Many current face recognition systems need high processing power hardware to perform [18,25]. However, those systems are deployed on a single machine and could encounter overload when there are many students to identify. Therefore, we aim to develop a face recognition system that can be deployed across machines that do not have powerful hardware. Furthermore, in real life scenario where only a fraction of probe sample identities is enrolled in the database, the system should be able to reject or ignore those that correspond to unknown identities. Moreover, the ability to quickly identify and export the results are other objectives of the system. We review the state-of-the-art approaches to choose the most suitable models for the proposed system. We also make adjustments to meet the reality of the context.

To sum up, the contributions of this paper are as follows:

- We analyze the main components of an end-to-end face recognition system and compare them under different conditions.
- We point out the effect of different components on the overall system's performance.
- We propose a face recognition system that has a good trade-off between accuracy and execution time can perform on hardware that does not have high processing power.

We organize the paper into six sections. The first section introduces an overview of the motivation and the requirements of the problem. The following section describes the related work and the components of a typical end-to-end face recognition system. The architecture and core elements applied to our system are introduced in Sect. 3. Section 4 describes parameters, datasets, and settings for evaluation. In Sect. 5, we present and analyze the results obtained. Finally, the conclusion and the future work are given.

2 Related Work

In this section, we describe the primary components of an end-to-end face recognition system and review some promising approaches.

A facial recognition system has three basic components (shown in Fig. 1): face detection, feature extraction, and face recognition. There are many factors such as pose, age, glasses, hairstyle, facial expression, and lighting conditions which may have significant impact on discrimination ability of a facial recognition system. Many developed methods have focused on addressing these challenges in order to enhance robustness of facial recognition systems. However, they require high processing time, consume a lot of memory, and are relatively complex. Therefore, we conduct research and evaluate different methods to check the suitability of the system we intend to build.

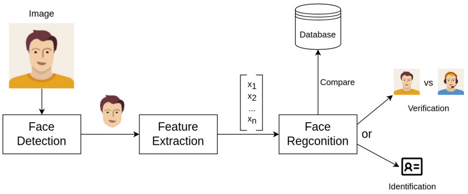


Fig. 1. Three basic steps of face recognition

The purpose of face detection is to assess whether or not the input image contains human faces. Some classics face detection methods such as Haar Cascade [20], Histograms of Oriented Gradients (HOG) [4] can perform fast. However, these methods do not archive good result when it comes to different light angles, face poses. There are many current methods have been developed that have good performance like Single-Shot Detector (SSD) [12] with ResNet 10 as the foundation, Multi-task Cascaded Convolutional Networks (MTCNN) [24], RetinaFace [5], CenterFace [23].

Feature extraction's primary role is to extract the characteristics of the face found in the detection step. The major aspects of the face image, such as the eyes, nose, and mouth, as well as their geometric distribution is represented by a collection of features vector. Some well-known methods for carrying out this work are linear discriminant analysis (LDA) [10], principal component analysis (PCA) [15] and local binary sampling method (LBP) [14]. With the recent development of deep learning techniques, many state-of-the-art methods such as FaceNet [19], CosFace [21], ArcFace [6] have achieved high performance while still having reasonable execution time.

Face recognition step compares the features extracted from the feature extraction step to known faces stored in databases. Face recognition has two general applications, one is identification and the other is verification. In the identification process, an input face is compared to a collection of faces in order to find the most likely match. In the verification process, an input face is compared to a known face in the database to determine if it should be accepted or rejected. Some of classic methods to solve this task are support vector machine (SVM) [3], softmax classifier [8], and k-nearest neighbors (K-NN) [2].

3 The Proposed Student Monitoring System

In this section, we introduce architecture of our face recognition system and core components that match the system's criteria including high accuracy, reasonable runtime, scalability.

3.1 Face Detection

Through surveying and analyzing many face detection methods, we selected three methods: MTCNN [24], RetinaFace [5] and CenterFace [23]. The reason for choosing these two methods is high accuracy and can run on a single CPU core in real-time for VGA resolution images. MTCNN is a three-stage algorithm used to detect the bounding box along with five landmark points on the face. RetinaFace is designed based on the feature pyramids with independent context modules. Following the context modules, a multi-task loss is calculated for each anchor. CenterFace is a one-stage, anchor-free approach for predicting facial box and landmark position in real-time with high accuracy.

MTCNN is composed of three stages corresponding to three convolutional neural networks which are P-Net, R-Net and O-Net. Before being fed into P-Net, the input image is scaled down to several sizes. Each scaled image is an input to this network. This operation aims to find many faces of different sizes in the input image. For each scaled image, a 12 × 12 kernel slides over its surface to find the face with a stride of 2 pixels. After each convolution layer, the PReLU activation function is applied. Output is the coordinates, the probability that a face exists and does not exist in each frame. After collecting all outputs, the model discards all frames with low confidence and merges high-overlapped frames into an unique frame using NMS (non-maximum suppression). Because some frames may grow out of the image boundary when we convert them to square, it is necessary to buffer them to get enough input value. Then, all frames are converted to 24 × 24 size and fed into the R-net. After each convolution layer, the PReLU activation function is applied. One output is the coordinates of the more precise frames and the confidence of that frame.

O-Net and R-NET are structurally similar, differing only in-depth. The results of R-Net after employing NMS are resized to 48 × 48, then fed into O-Net as its input. O-Net outputs not only the coordinates of the bounding boxes, but also the coordinates of the five landmarks on the face.

RentinaFace is a single-stage face detector that uses a multi-task learning strategy to predict face score, face box, five facial landmarks, and dense facial landmark at the same time.

The loss function for an anchor i is presented follow:

$$L = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel} \quad (1)$$

This multi-task loss function consists of four parts.

- The first part is face classification loss, $L_{cls}(p_i, p_i^*)$, where p_i is the predicted probability of anchor i being a face and p_i^* is 1 for the positive anchor and 0 for the negative anchor. The classification loss L_{cls} is the softmax loss for binary classes.
- The second part is face box regression loss, $\lambda_1 p_i^* L_{box}(t_i, t_i^*)$, where $t_i = \{t_x, t_y, t_w, t_h\}_i$ and $t_i^* = \{t_x^*, t_y^*, t_w^*, t_h^*\}_i$ represents the predicted and groundtruth box location corresponding with the positive anchor and $L_{box}(t_i, t_i^*) = R(t_i - t_i^*)$, where R is the robust loss function (smooth-L1) defined in [7].
- Facial landmark regression loss, L_{pts}, represent the predicted five facial landmarks and groundtruth associated with the positive anchor.
- The last part is dense regression loss, L_{pixel} (refer to Eq. 2).

$$L_{pixel} = \frac{1}{W \times H} \sum_{i}^{W} \sum_{j}^{H} \| R(D_{P_{ST}}, P_{cam}, P_{ill})_{i,j} - I_{i,j}^* \|_1 \qquad (2)$$

where W and H are the width and height of the anchor crop $I_{i,j}^*$, respectively.

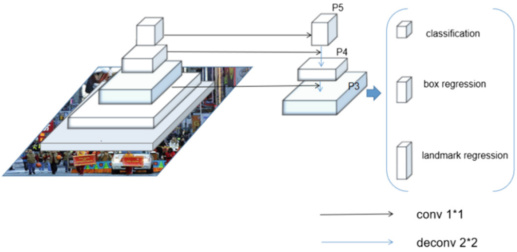


Fig. 2. The Architecture of CenterFace

One of the severe downsides of the anchor-based method is to require a large dense of anchors to attain a good recall rate to improve the overlap between anchor boxes and ground truth. As a result, the models implemented using this approach are rather heavy and sluggish. CenterFace is a lightweight and effective anchor-free face detection method, thereby avoiding the downsides of the anchor-based method.

For the subsequent detection, CenterFace used MobileNetV2 [17] as the backbone and Feature Pyramid Network (FPN) [11] as the neck. In general, FPN builds a feature pyramid from a single scale input using a top-down architecture with lateral connections. CenterFace represents the face by using the face box's center point. The box size and placement of the face are then regressed to image features at the central location. As a result, just one layer of the pyramid is employed for face detection and alignment. The architecture of CenterFace is shown in Fig. 2.

3.2 Feature Extraction

We focus on FaceNet [19] and Arcface [6] because of the good capabilities. FaceNet is an algorithm introduced in 2015 by Google that uses deep learning to extract features on human faces. FaceNet takes an image of a person's face and returns a vector containing 128-dimensional important features.

In FaceNet, the CNN network helps encode the input image into a 128-dimensional vector and then input the triplet error function to evaluate the distance. To use the triplet loss function, three images are required, of which one is selected as the landmark. The landmark photo (A) must be fixed first of the three. The remaining two images include an image labeled Negative (N) (object different from the original image subject) and an image labeled Positive (P) (same object as the original image). The objective of the error function is to minimize the distance between two images if it is negative and maximize the distance when the two images are positive. The loss function is as follows:

$$L(A, P, N) \ = \sum_{i=0}^{n} \max(|| \ f(A_i) \ \ - f(P_i)||_2^2 - || \ f(A_i) \ \ - f(N_i)||_2^2 + \alpha, \ 0) \quad (3)$$

where n is the number of triplets; f is the embedding function; α is a margin between positive and negative pairs.

The selection of three images dramatically affects the quality of FaceNet model. If a good triplet is selected, the model converges quickly, and the prediction results are more accurate. Furthermore, hard triplet makes the training model smarter because the resulting vector is a vector representing each image. These vectors can distinguish negatives (similar to positives). As a result, images with the same label are closer together in Euclidean space. However, the triplet loss has some drawbacks:

- Combinatorial explosion in the number of face triplets especially for large-scale datasets, creating an increase in iteration steps.
- For effective model training, semi-hard sample mining is a difficult task.

To avoid these problems, some methods use the solfmax-loss. However, the softmax loss function does not explicitly optimise the feature embedding to enforce higher similarity for intraclass samples and diversity for inter-class samples, which results in a performance gap for deep face recognition under large intra-class appearance variations. To boost the discriminative ability of the face recognition model and stabilize the training process, Arcface introduces an additive angular margin loss, is presented as follows:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{s\left(\cos\left(\theta_{y_i} + m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i} + m\right)\right)} + \sum_{j=1, j \neq y_i}^{n} e^{s \cos \theta_j}} \quad (4)$$

where m is an angular margin penalty, s is feature scale, N is the batch size, n is the number of classes (identities), and y_i is the ground-truth label of sample x_i.

After the last convolutional layer, the BN - Dropout - FC - BN structure is applied to get the final 512-D embedding feature.

3.3 Classifiers

There are two strategies to build a classifier in a face recognition system. The first approach is to use closed set classifier such as support vector machine (SVM), Bayesian classification. However, when an unknown object is added, the closed set classifiers misclassify it as a known class. To solve this problem, we can periodically retrain classifiers, but it still takes more time and resource. The second approach is to use open set classifiers based on the distance between two feature vectors. Then we can set a threshold to classify unknown objects.

Cosine Similarity: is a method of measuring the similarity between two non-zero vectors in an inner product space. It is calculated by taking the cosine of the angle formed by two vectors. Therefore, it depends on vectors' orientation, not magnitude. The formula for cosine similarity is:

$$cosine\_similarity(x, y) = \frac{x \cdot y}{\|x\|\|y\|} \tag{5}$$

Based on cosine similarity, the cosine distance between two vectors can be easily inferred as follows:

$$cosine\_distance(x, y) = 1 - cosine\_similarity(x, y) = 1 - \frac{x \cdot y}{\|x\|\|y\|} \tag{6}$$

where x, y are feature embeddings extracted from face images in our system.

3.4 The Architecture System

Attendance and student monitoring system should have a high accuracy rate while maintaining a reasonable runtime. In addition, the system need to be able to detect unknown identities in classroom.

To register, student faces are collected then use CenterFace to detect and extact face region from original images. Detected faces go through Arcface model to extract features. The feature vectors are labeled by the system and stored in the database (Fig. 3). The lecturer only need to do this register step once during the first lecture. Whenever the lecturer want to check student's attendance, the system detects the face from photo captured from student's camera. The detected face then go through the feature extraction step, return a 512-dimensional vector. Finally, the system calculates and compares the cosine distance between the feature vector with other feature vectors stored in the database to determine face's identity. If the face matches the known identity, the system makes a record on student's attendance. Otherwise, the system informs the lecturer about unknown identities (Fig. 4).

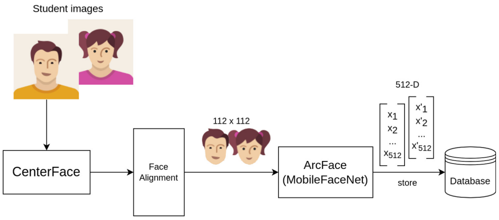


Fig. 3. An illustration of the student registration

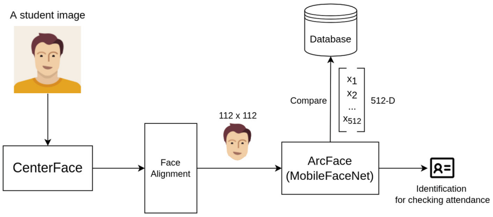


Fig. 4. The architecture of the attendance checking

We trained Arcface model from scratch using MobileFaceNet [1] as backbone on the MS1MV2 dataset with more than 5.8 million photos of nearly 85,000 identities. The input is a 112 × 112 pixels color image normalized on all three color channels.

In real-life scenarios, the face obtained after the detection step can have different angles, which impact the system's accuracy. The work in [22] shows that moderate alignment can boost the recognition accuracy. Therefore, We apply 2D affine transformation to align face in prior to feature extraction step.

4 Datasets and Experiment Settings

4.1 Datasets

Face datasets have grown in size and variety in recent years, and the testing scene has been approaching the real-world unconstrained condition. We evaluate the proposed face recognition system on several well-known datasets:

- Labeled Faces in the Wild (LFW) [9]: contains 13,233 images of 5,749 individuals collected from the web; this is one of the most popular benchmark datasets for face verification with 6,000 pairs.
- AgeDB-30: The subset of AgeDB [13] contains 12,240 images of 570 famous people. There are 6,000 pairs in this dataset; the age difference of each pair's faces is equal to 30 years.
- Celebrities in Frontal-Profile in the Wild (CFPW) [16] includes frontal and profile images of 500 celebrities' faces with ten frontal and four profile images per person. This dataset has two verification protocols: one compares just

frontal faces (CFP-FF), and the other compares frontal and profile faces (CFP-FP) with 7,000 pairs for each protocol.

For face identification task, we generate four mini versions of LFW, AgeDB-30, CFP-FF and CFP-FP to simulate the student attendance check process. For each dataset, we randomly select 500 individuals with two images apiece, then we divide them into two different classrooms (250 individuals each) and register their first image in database. The remaining 500 images are used as data to evaluate our system's performance when taking a roll call.

Furthermore, we also evaluate our system in an online class of 17 students on the Google Meet platform in unconstrained environments. We focus on testing our system with various poses, light conditions, glasses, and mask.

4.2 Experiment Settings

Experimental configuration includes computers equipped with Intel Core i5 8400. To make experiments closer to real-life scenarios, we conduct experiments on CPU only because most clients' machines do not support GPU. The validation method utilized is k-fold cross-validation, with $k = 10$. The integrated librares have OpenCV and Scikit-learn to support preprocessing images. We use MySQL database to store and manage embedding vectors.

5 Experiments and Result Analysis

Face detection is the initial step of an end-to-end face recognition system, and serves as input towards face alignment and feature extraction. The quality of detection bounding box has a direct impact on the performance of the subsequent steps. As shown in Table 1, the system that uses MTCNN detector achieves lowest accuracy compare to CenterFace and RentinaFace detector. The results indicate that a robust face detector can boost face recognition accuracy.

Table 1. Accuracy of verification task of our system with different face detectors and feature extraction models on LFW dataset

	ArcFace	Facenet
MTCNN	0.955	0.948
CenterFace	**0.996**	**0.992**
RentinaFace	0.995	0.984

Figure 5 shows average time our system takes to perform verification task with a pair of 250 × 250 pixels input images. Overall, systems use Arcface as feature extraction model take less time than Facenet to perform. There is a significant gap in execution time when we apply different face detectors. The system

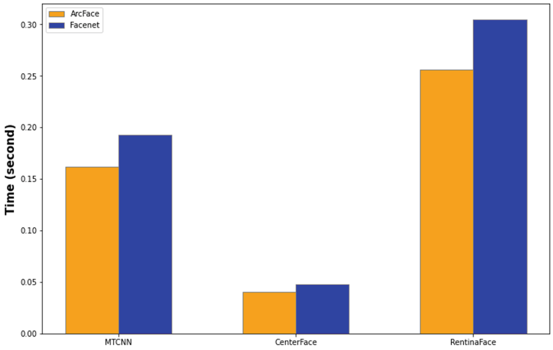
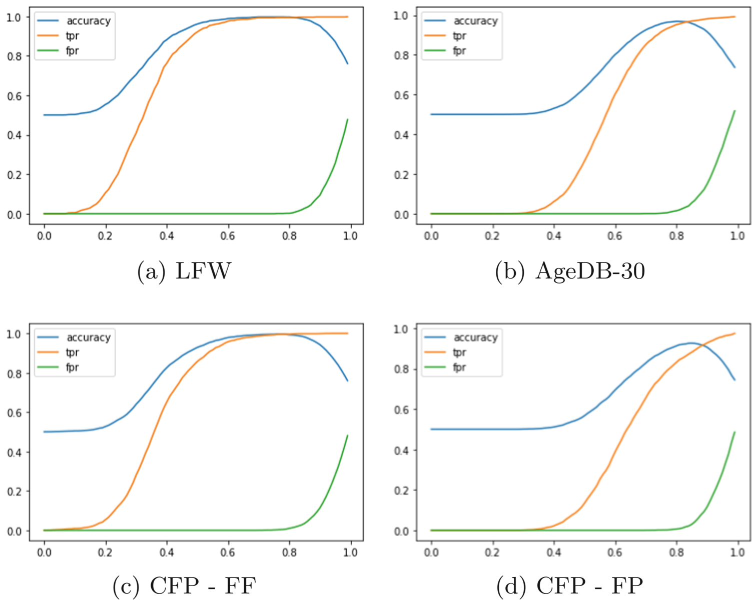


Fig. 5. Average time to verify a pair of face

that uses CenterFace achieves impressive results compares to other detectors. Our final system applies CenterFace to detect face and Arcface to extract embedding vector because it achieves not only high accuracy but also fast execution time.

In the identification task, the proposed system takes an average of 0.22 s to register a new face and 4.3 s to identify a face in a database of 500 samples. The highest accuracy rate our proposed system achieve on LFW, AgeDB-30, CFP-FF, CFP-FP is 0.996, 0.968, 0.995, 0.925 respectively. As shown in Fig. 6, we recommend the threshold should be in range from 0.6 to 0.75 so that the system has best performance. If we set the threshold too small, the system tends to be too sensitive and it is pointless if the system can not recognize most registered member. On the other hand, if the threshold is too high, the system is likely to misrecognize. We also notice that our system is more sensitive to data from cfp-fp when the profile and the frontal images are mixed.

Figure 7 shows the results of the identification task performed by the proposed system under various settings. We do not show all of the volunteers due to paper length constraints and we have to blur their face because of the policy on privacy protection. As a result, instead of all 17, we only show 6 of them in Fig. 7. We use cosine distance in these tests and set the threshold to 0.6. In Fig. 7, the red highlights represent students that our system labels Unidentified because the value of cosine distance is greater than the threshold. On the other hand, the green highlights represent students that are correctly identified by the system. In the Different light conditions and Without glasses test, our system correctly identifies all 17 students. Only 2 out of 17 students are labeled Unidentified in the Do not look at camera test as shown in Fig. 7. With the threshold set to 0.6, our system can only successfully identify 4 students in the Mask test. To overcome this problem, we try to increase the threshold to 0.7 and the system can now correctly identify 12 out of 17 students.



(a) LFW (b) AgeDB-30

(c) CFP - FF (d) CFP - FP

Fig. 6. The accuracy, the true positive rate, the false positive rate of the identification task under different thresholds

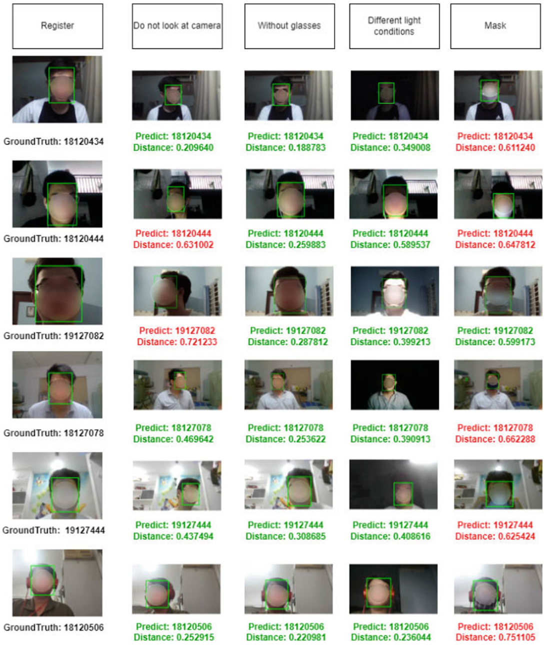


Fig. 7. Results of the identification task in an online classroom

6 Conclusion

The face identification is one of the highly applicable problems. We focus on deep learning approaches to solve this problem and apply them to the student monitoring system in the classroom. The implementation of the system helps lecturers track students and detect intruders, especially in online classes. In this work, we propose the student tracking system which has good accuracy and real-time execution time. The cores of the system are CenterFace for detecting faces and ArcFace for extracting features. To improve the ability to identify unknown objects, we also adjusted and added measures such as cosine distance. Experimental results show that the accuracy and time of the system meet the requirements for practical implementation. Moreover, some issues need to be further considered, such as improving the ability to recognize when parts of the face are obscured and preventing anti-spoofing attacks. They will be interesting challenges to continue working on in the future.

Acknowledgements. This research is funded by the University of Science, VNU-HCM, Vietnam under grant number CNTT 2021-14 and Advanced Program in Computer Science.

References

1. Chen, S., Liu, Y., Gao, X., Han, Z.: MobileFaceNets: efficient CNNs for accurate real-time face verification on mobile devices. In: Zhou, J., Wang, Y., Sun, Z., Jia, Z., Feng, J., Shan, S., Ubul, K., Guo, Z. (eds.) CCBR 2018. LNCS, vol. 10996, pp. 428–438. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97909-0_46
2. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: Concepts and algorithms. J. Mach. Learn. Res. **10**(3) (2009)
3. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893. IEEE (2005)
5. 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)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
7. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
8. Gupta, P., Saxena, N., Sharma, M., Tripathi, J.: Deep neural network for human face recognition. Int. J. Eng. Manuf. (IJEM) **8**(1), 63–71 (2018)
9. 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)
10. Jolicoeur, P.: Fisher's linear discriminant function, pp. 303–308. Springer, US, Boston, MA (1999)

11. 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, pp. 2117–2125 (2017)
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
13. Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., Zafeiriou, S.: Agedb: the first manually collected, in-the-wild age database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, vol. 2, p. 5 (2017)
14. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recogn. **29**(1), 51–59 (1996)
15. Pearson, K.: Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh Dublin Philosophical Mag. J. Sci. **2**(11), 559–572 (1901)
16. S. Sengupta, J.C. Cheng, C.C.V.P.R.C.D.J.: Frontal to profile face verification in the wild. In: IEEE Conference on Applications of Computer Vision, February 2016
17. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv 2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
18. Sardar, S., Babu, K.A.: Hardware implementation of real-time, high performance, RCE-NN based face recognition system. In: 2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems, pp. 174–179. IEEE (2014)
19. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
20. 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, vol. 1, p. I. IEEE (2001)
21. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5265–5274 (2018)
22. Wei, H., Lu, P., Wei, Y.: Balanced alignment for face recognition: a joint learning approach. arXiv preprint arXiv:2003.10168 (2020)
23. 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 (2020)
24. 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)
25. Zhang, Y., Cao, W., Wang, L.: Implementation of high performance hardware architecture of face recognition algorithm based on local binary pattern on FPGA. In: 2015 IEEE 11th International Conference on ASIC (ASICON), pp. 1–4. IEEE (2015)