SN Computer Science (2021) 2:355
https://doi.org/10.1007/s42979-021-00743-0

ORIGINAL RESEARCH

Training Neural Networks on Top of Support Vector Machine Models for Classifying Fingerprint Images

Thanh-Nghi Do[1,2]

Received: 13 March 2021 / Accepted: 10 June 2021 / Published online: 24 June 2021

Abstract
We propose to train neural networks on top of support vector machine (SVM) classifiers learned from various visual features for efficiently classifying fingerprint images. Real datasets of fingerprint images are collected from students at the Can Tho University. The SVM algorithm learns classification models from the handcrafted features such as the scale-invariant feature transform (SIFT) and the bag-of-words (BoW) model, the histogram of oriented gradients (HoG), and the deep learning of invariant features (e.g., Inception-v3, Xception, VGG, ResNet50), extracted from fingerprint images. Followed which, neural networks are learned on top of SVM classifiers trained on these diverse visual features, making improvements of the fingerprint image classification. The empirical test results show that the proposed approach is more accurate than SVM classifiers trained on any single visual feature type. On average, a neural network trained on top of SVM-ResNet50, SVM-HoG, and SVM-SIFT-BoW improves 36.47, 12.30, and 8.74% classification accuracy against SVM-ResNet50, SVM-HoG, and SVM-SIFT-BoW, separately.

Keywords Fingerprint image classification · Visual features · Stacking classifiers

Introduction

Fingerprint images are uniqueness and durable over time, making them ideal as long-term markers of individual identity. The recognition of fingerprint images is one of the most popular and useful methods for identifying individuals. It is successfully applied to both government and civilian applications, including suspect and victim identifications, the recovery of partial fingerprints from a crime scene in forensic science, border control, employment background checks, and secure facility entrance [28, 29, 40].

Our previous work in [48] proposed to combine SVM [57] models learned from different visual features, such as SIFT-BoW [4, 34, 37, 38, 53], HoG [14], and pre-trained deep learning network such as Xception [11], for efficiently classifying fingerprint images. In this paper, we develop its extension to train neural networks on the top of SVM classifiers learned from diverse visual handcrafted feature types, SIFT-BoW, HoG, and invariant features extracted by pre-trained deep neural networks, including VGG16, VGG19 [52], ResNet50 [26], Inception-v3 [56], and Xception [11], making improvements for the fingerprint image classification. First, we collect fingerprint image datasets. And then, we study visual feature extraction techniques for fingerprint images, including two classical handcrafted features like the SIFT-BoW, the HoG, and deep features learned by popular deep networks such as VGG16, VGG19, ResNet50, and Inception-v3, Xception. SVM classifiers are learned from separately the single visual feature type to classify fingerprint images. And then, we propose to train neural networks on the top of SVM classifiers, offering a full complement of diverse visual feature types in the fingerprint image recognition. The empirical test results on real fingerprint image datasets show that training neural networks on top of SVM classifiers learned from SIFT-BoW, HoG, and deep features improves classification correctness compared to the ones trained on any single visual feature type. An example of the proposed approach's effectiveness on average is that a neural

This article is part of the topical collection "Future Data andSecurity Engineering 2020" guest edited by Tran Khanh Dang.

✉ Thanh-Nghi Do
dtnghi@cit.ctu.edu.vn

1 College of Information Technology, Can Tho University, 92000 Cantho, Vietnam

2 UMI UMMISCO 209 (IRD/UPMC), Sorbonne University, Pierre and Marie Curie University, Paris 6, France

network trained on top of SVM-ResNet50, SVM-HoG, and SVM-SIFT-BoW improves 36.47%, 12.30%, and 8.74% classification accuracy of SVM-ResNet50, SVM-HoG, and SVM-SIFT-BoW, respectively.

The remainder of this paper is organized as follows: Section 2 briefly presents the related work. Section 3 illustrates our proposal for classifying fingerprint image datasets. Section 4 shows the experimental results before conclusions and future work presented in Sect. 5.

Related Work

The review papers [28, 29, 40] summarize related works in fingerprint recognition: techniques, accomplishments, challenges, and opportunities.

The classification system of fingerprint images in Fig. 1 follows the usual framework for the classification of images. Building a system for the image classification includes three main works:

1. collecting the dataset of fingerprint images;

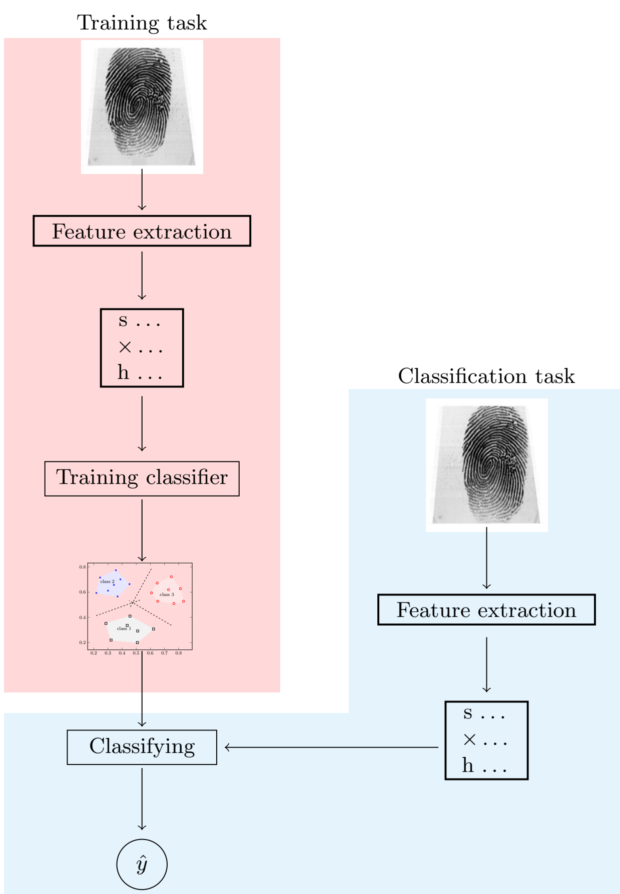


Fig. 1 Framework for classifying fingerprint images

2. extracting visual features from fingerprint images and representing them;
3. training SVM classifiers.

There are public benchmarks such as FVC datasets [20], and SD datasets [23]. The international Fingerprint Verification Competitions released FVC datasets in 2000, 2002, 2004, and 2006. The National Institute of Standards and Technology (NIST) created SD datasets in 2012-2018 to evaluate measure accuracy of biometric identification. A benchmarking fingerprint [12] consists of 40,000 synthetically generated images. Recent work [45] used a Generative Adversarial Network (GAN [24]) to synthesize 100 million fingerprint images.

As illustrated in Fig. 1, the usual approach for the fingerprint image classification includes two tasks. The training task consists of two main stages: extracting features from images and training classifiers. In the classification task, the resulting classifier receives the features extracted from a new fingerprint image to predict the class label for this new fingerprint image.

For a long time, the classical approaches commonly used minutiae (e.g., ridge ending, ridge bifurcation, etc.) as features and the matching method between minutiae of fingerprint images [8, 28, 29, 40, 41]. The proposal in [46] trains a deep convolutional neural network on texture features extracted from fingerprint images. The fingerprint image classification in [16] is performed by support vector machines (SVM [57]), random forest of oblique decision trees (RF-ODT [17]), models being trained on visual features such as the scale-invariant feature transform (SIFT [37, 38]), and the bag-of-words model (BoW [4, 34, 53]).

More recent techniques aim to train deep convolutional neural networks (CNN [33]) and fine-tuning pre-trained deep neural networks to recognize fingerprint images. The CNN-based approach illustrated in Fig. 2 benefits from the ability to learn visual features (low-level, mid-level, and high-level) from images and the softmax classifier in an unified framework. The CNN-based approach is more accurate than the classical one. Shrein [51] proposed a CNN architecture to recognize fingerprint images. FingerNet [44] fine-tunes pre-trained ResNet50 [26] to recognize fingerprint images. Do and his colleagues [19] proposed to fine-tune recent pre-trained deep learning models such as VGG16, VGG19 [52], ResNet50 [26], Inception-v3 [56], and Xception [10] for classifying fingerprint images. Militello et al. [42] showed the performance of pre-trained CNNs, including AlexNet [31], GoogLeNet [55], and ResNet [26] for the fingerprint image classification. DeepPrint network [21] learns alignment and minutiae from fingerprint images, making fixed-length fingerprint representations of only 200 bytes. Kai and Anil [5] proposed a CNN to learn an orientation field dictionary for fingerprint alignment. An most interesting work in [6] used CNNs for ridge

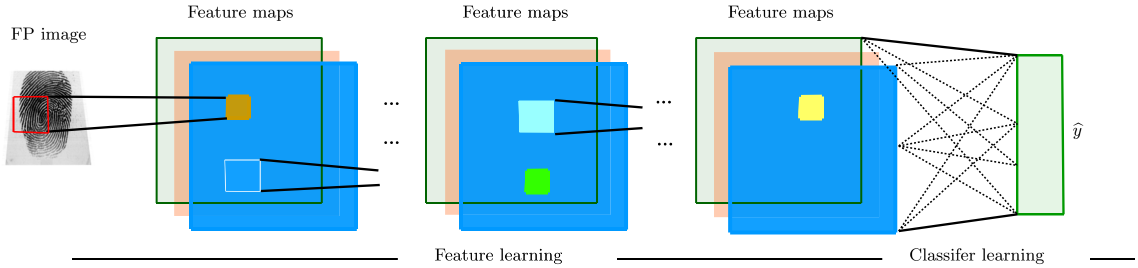


Fig. 2 Convolutional neural network (CNN) for classifying fingerprint images

Table 1 Description of fingerprint image datasets

ID	Dataset	#Datapoints	#Classes
1	FP-235	3485	235
2	FP-389	6306	389
3	FP-559	10270	559

flow estimation and minutiae descriptor extraction. MinuNet and MinuNetLatent [7] train convolutional autoencoders to detect minutiae. A survey [43] presents the state-of-the-art deep learning works on fingerprint recognition.

Proposed Methods

Datasets of Fingerprint Images

Public fingerprint databases for the performance evaluation are summarized in [2, 3]. FVC datasets [20] consist of 1800 fingerprint images from 150 individuals. SD datasets [23] contain 8871 fingerprint images of 888 individuals.

Our investigation aims at studying the classification of large fingerprint image datasets. Therefore, we start with the collection of fingerprint image datasets. In 2016, 2017, and 2018, we used Microsoft Fingerprint Reader (optical fingerprint scanner, resolution: 512 DPI, image size: 355x390, colors: 256 levels grayscale) to capture fingerprint images from our students and colleagues at the College of Information Technology, Can Tho University. And then, we obtain three real fingerprint datasets named FP-235, FP-389, and FP-559, which include fingerprint images of 235, 389, and 559 individuals, respectively. There are from 15 to 20 fingerprint images captured for each individual (class label). Three datasets are described in Table 1.

Visual Approaches for Classifying Fingerprint Images

The visual framework for the classification of images consists of two main stages. The first stage is to extract and depict visual features from images. And then, the second stage is to train SVM models for classifying images.

At the first stage, we propose to study two most popular methods to produce handcrafted features, including the scale-invariant feature transform (SIFT [37, 38]) and the bag-of-words model (BoW [4, 34, 53]), the histogram of oriented gradients (HoG [14]).

Scale-Invariant Feature Transform and Bag-of-Words

The SIFT descriptors [37, 38]) extracted from images and the bag-of-words model (BoW) are the most habitual representation for tasks of images' classification [4, 34, 53]. The SIFT method detects the appearance of the object at particular interest points in images, invariant to image scale, rotation, and also robust to changes in illumination, noise, and occlusion. Clustering SIFT descriptors is to form visual words. A fingerprint image is then represented by the frequencies of the visual words, i.e., BoW.

Histogram of Oriented Gradients

The HoG descriptors proposed by [14] are used to detect the human. The HoG technique tries to describe local object appearance and shape within an image, by computing the distribution of local intensity gradients or edge directions to. And then, it combines the distributions to form the image representation. The HoG descriptor is invariant to geometric and photometric transformations, except for object orientation.

In last years, the researchers propose deep neural networks including VGG16, VGG19 [52], ResNet50 [26], Inception-v3 [56], and Xception [11] for efficiently performing the large-scale image classification. These deep neural network models are pre-trained on ImageNet dataset [15], achieving most accurate classification results. We also propose to use them to extract deep features from fingerprint images at the first stage of the visual classification framework.

VGG16 and VGG19

The VGG network architecture proposed by [52] consists of two parts. The first one stacks several VGG block modules (convolutional and pooling layers as shown in Fig. 3) to learn visual features from images. The second one includes fully connected layers to train classifiers.

The VGG block module uses 3×3 convolutional layers stacked on top of each other to develop depth. The 2×2 max pooling layer is used to reduce volume size. Furthermore, the VGG architectures are designed, so that every convolutional layer captures the spatial information from images, consecutively increasing the access to a larger spatial context. VGG architectures to prevent the spatial feature of the image. VGG-16 and VGG-19 consist of 16 and 19 VGG block modules for the large-scale image classification of ImageNet dataset.

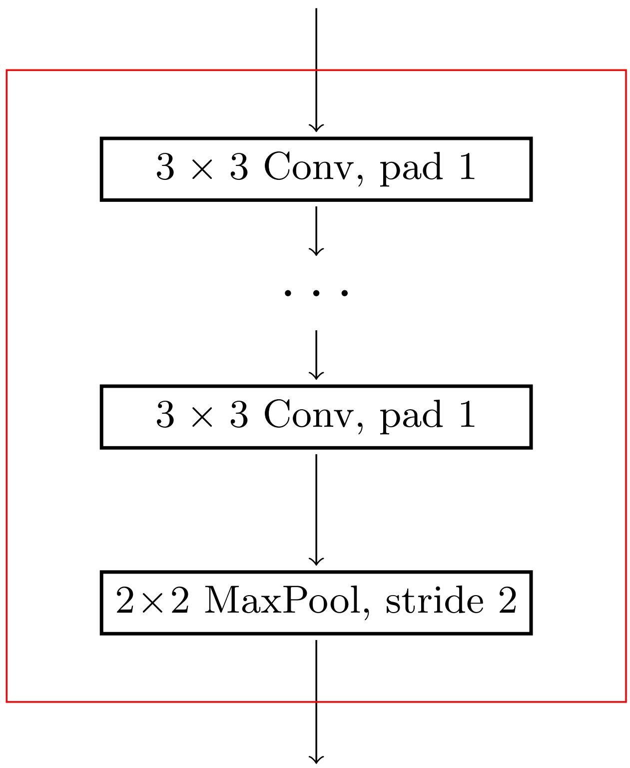


Fig. 3 VGG block module

Inception-v3

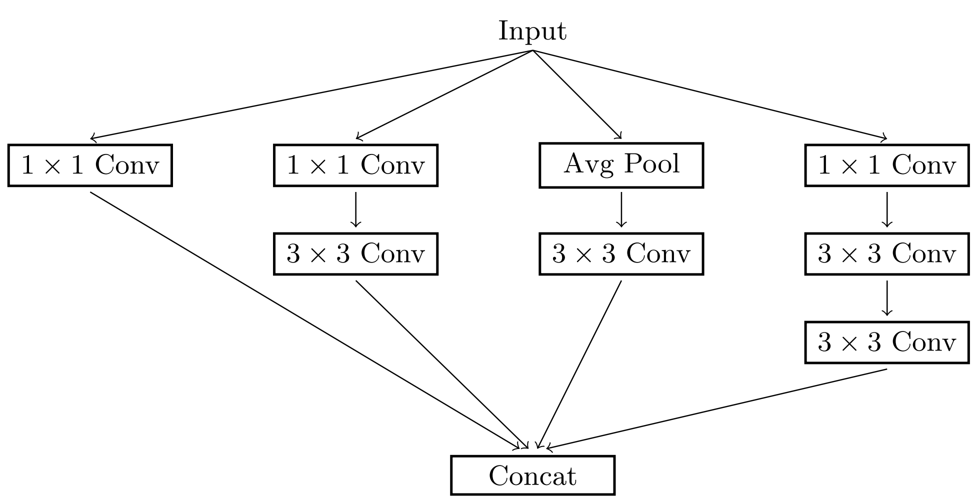
The Inception-v3 network proposed by [56] is used to deal with ImageNet dataset. Its architecture stacks 11 inception modules and global average pooling to learn multi-level features for the image classification. An inception module (in Fig. 4) consists of 1×1, 3×3 convolutions to learn invariant features from different spatial sizes, in which 1×1 convolutions are used to reduce the volume size. These Inception modules are stacked on top of each other. Maximum and average pooling layers aim to to reduce the dimension in the Inception-v3. The network architecture aims to capture the spatial information from images.

ResNet50

The ResNet50 network architecture proposed by [26] develops extremely deep networks for classifying ImageNet dataset.

The well-known failure of extremely deep networks is the vanishing gradient problem during the learning process of deep network. It is the motivation for He and his colleagues [26] to propose a new residual block (in Fig. 5), to overcome this issue. The residual block includes convolutional layers. Each convolutional layer is followed by a batch normalization layer and a ReLU activation function. Furthermore, an identity mapping is added on the output from the last

Fig. 4 Inception module



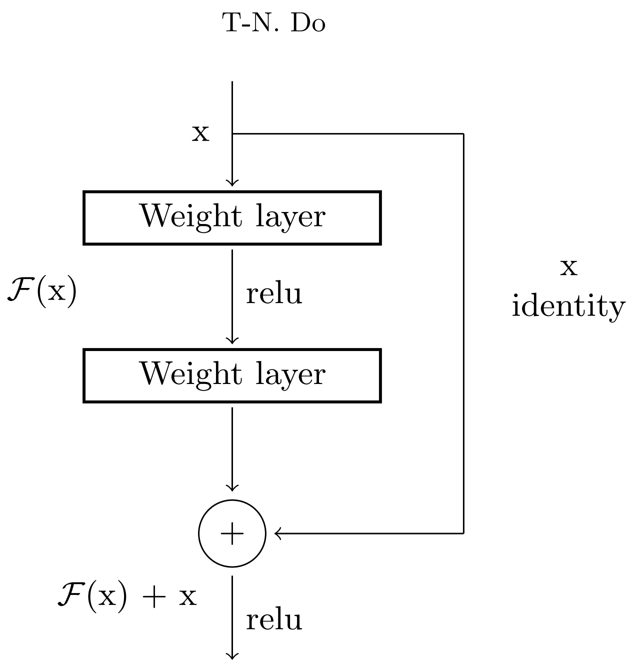


Fig. 5 Residual block

convolutional layer before the final ReLU activation function. The identity mapping allows to skip one or many layers in the network. The ResNet50 is designed with the depth of 50 layers in blocks.

Xception

The Xception network proposed by [10] is an extension of the Inception architecture. The network architecture includes Xception modules being also stacked on top of each other. The Xception module (Fig. 6) replaces the standard depthwise separable convolution (the depthwise convolution followed by a pointwise convolution) in Inception modules with depthwise separable convolutions. This new modification does not need any intermediate activation being the pointwise convolution followed by a depthwise convolution.

Pre-trained deep networks VGG16, VGG19, Inception-v3, ResNet50, and Xception are used to extract invariant visual features from fingerprint images.

Followed the first stage for the feature extraction, the Support vector machine (SVM [57]) algorithm is used to train classifiers for recognizing fingerprint images.

Support Vector Machines

For a linear binary classification problem depicted in Fig. 7, the SVM algorithm proposed by [57] tries to find the best separating plane furthest from both class +1 and class −1. To pursue this aim, the training SVM algorithm simultaneously maximizes the margin (or the distance) between the supporting planes for each class and minimizes errors.

The SVM algorithm uses various kernel functions [13] to handle non-linear classification tasks. The commonly non-linear kernel functions can be a polynomial function of degree d, a radial basis function (RBF).

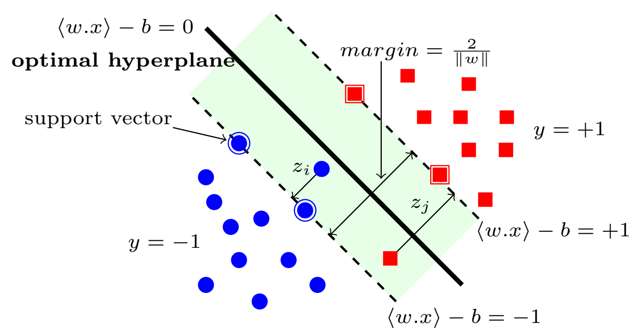
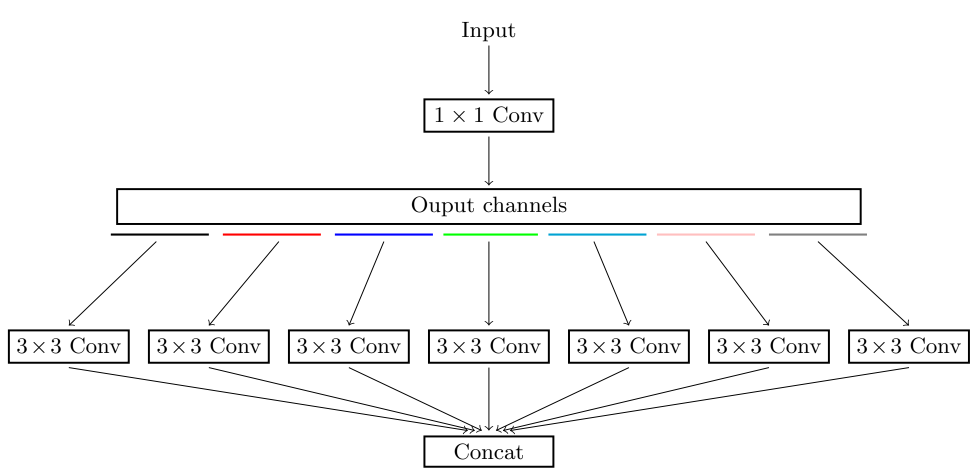


Fig. 7 Classification of the datapoints into two classes

Fig. 6 Xception module



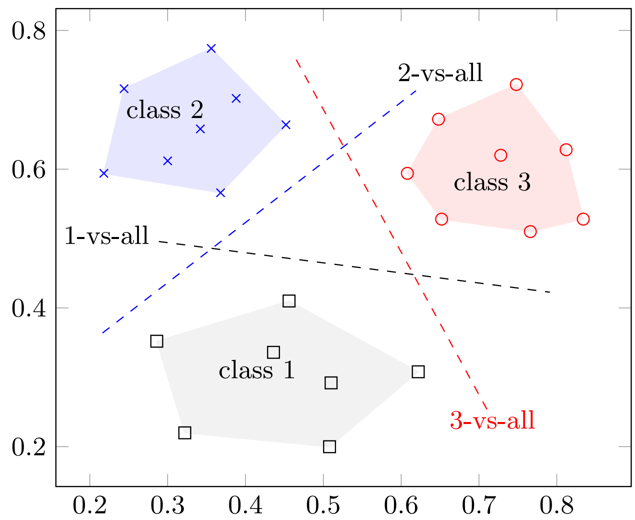


Fig. 8 Multi-class SVM (One-Versus-All)

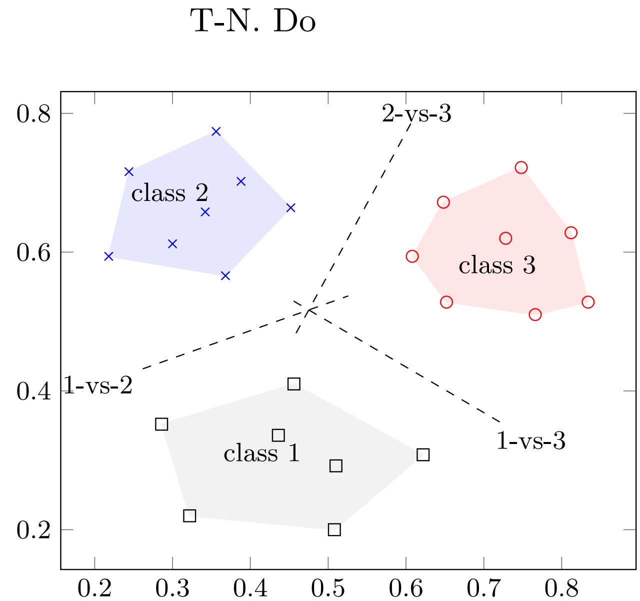


Fig. 9 Multi-class SVM (One-Versus-One)

The SVM learning algorithm can be extended for handling the multi-class problems (c classes, $c \geq 3$). The main idea is to decompose multi-class into a series of binary SVMs, including One-Versus-All [57], One-Versus-One [30]. The One-Versus-All strategy (as illustrated in Fig. 8) trains c different binary SVM models where the ith one separates the ith class from the rest. The One-Versus-One strategy (as illustrated in Fig. 9) trains $c(c-1)/2$ binary SVM models for all the binary pairwise combinations of the c classes. The class is then predicted with the largest distance vote. In practice, the One-Versus-All strategy is implemented in LIBLINEAR [22] and the One-Versus-One technique is also used in LibSVM [9].

Training Neural Networks on Top of SVM Models

Classical classification of images usually trains the SVM model on a single visual feature type extracted from fingerprint images. Therefore, the classification correctness is restricted, because any visual feature type has advantages and disadvantages. The handcrafted feature descriptors like SIFT and HoG have several advantages, including invariant to image scaling, geometric and photometric transformations, robust to noise, small distortions, and changes in illumination [14, 37, 38, 50]. As illustrated in [58, 61], local features extracted by SIFT are more robust than global features produced by HoG, under severe conditions. However, SIFT ignores global information of the image; this leads to a negative impact. Furthermore, deep neural networks are efficient techniques among feature extractors to learn invariant visual features (low-level, mid-level, and high-level features) from images [32]. However, training deep neural networks requires huge amounts of resource (data, hardware). Our investigation aims to combine the strength of SVM models learned from diverse visual feature types for improving the classification of fingerprint images. Last idea [36, 60] is to fuse deep features to enhance image classification. Nevertheless, the nature of these visual features are different. Therefore, it cannot combine visual features before training for the SVM classifier. Instead of this, we propose to train the neural network on top of SVM models learned from SIFT-BoW, HoG, and deep visual features (as illustrated in Fig. 10) to complement each other. To avoid tuning by hand weights in habitual voting scheme [18, 59] between the prediction of each visual classifier, our proposed scheme is to automatically learn weights with the simple neural network as follows:

$$\boxed{\textbf{Input} \Rightarrow \textbf{Full Connected} \Rightarrow \textbf{Sampling} \Rightarrow \textbf{Full Connected} \Rightarrow \textbf{Softmax}}$$

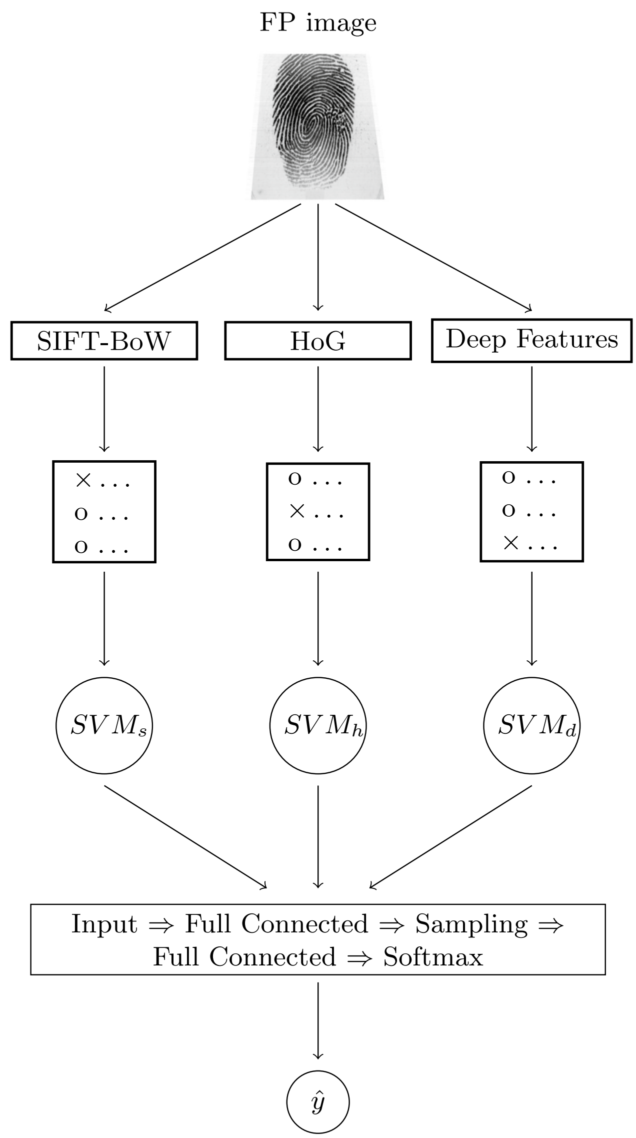


Fig. 10 Training neural networks on top of SVM models for classifying fingerprint images

However, the training process of the such neural network usually occurs the overfitting problem, making the poor performance in the classification. To address this problem, we propose to use a sampling layer in which it is to randomly drop units (along with their connections). This is a simple way to significantly reduces overfitting as illustrated in [54].

And then, the network fuses SVM models trained on SIFT-BoW, HoG, and deep visual features to classify fingerprint images.

Table 2 Hyper-parameters for training SVM models

No	Feature extraction method	γ	C
1	SIFT and BoW	0.00005	100000
2	HoG	0.025	100000
3	Deep features	0.001	100000

Experimental Results

In this section, we present experimental results of different visual approaches for classifying fingerprint images. We implement them in Python using library Keras [10] with backend Tensorflow [1], library Scikit-learn [47], and library OpenCV [27]. All experiments are conducted on a machine Linux Fedora 27, Intel(R) Core i7-4790 CPU, 3.6 GHz, 4 cores, and 32 GB main memory and the Nvidia GeForce GTX 960M 2GB GPU.

Three fingerprint image datasets are presented in Table 1. Datasets are randomly split into the trainset (80% fingerprint images) and the testset (20% fingerprint images). We use the trainset to build visual classification models. Then, results are reported on the testset using the resulting visual classification models.

Tuning Parameters

With methods for feature extractor and image representation, only handcrafted features SIFT and BoW model needs tuning the number of clusters (visual words) well known as the parameter of *k*means algorithm [39]. The number of visual words are varied from 1000 to 3000. And then, experimental results are unchanged while increasing the number of visual words over 2000. Therefore, we use 2000 visual words for the BoW model.

With SVM models, we propose to use RBF kernel functions, because it is general and efficient [35]. There is need to tune the hyper-parameter γ of RBF function $[K\langle x_i, x_j\rangle = \exp(-\gamma||x_i - x_j||^2)]$ and the cost C (a trade-off between the margin size and the errors) to obtain the best correctness. Finally, we find out best parameters' SVM in Table 2 for visual classification models.

For our proposed training neural networks on top of SVM models in Sect. 3, we tried to tune for the good neural network architecture. It consists of the number of neurons for the hidden layer (Full Connected) and the probability for the Sampling layer. The best configuration is with 128 neurons for the Full Connected and a probability of 0.5 for the Sampling layer. The number of epochs for training is set to 200.

Table 3 Overall classification accuracy for FP-235

No	Visual approach	Accuracy (%)
1	SVM-SIFT-BoW	89.38
2	SVM-HoG	82.78
3	SVM-ResNet50	57.82
4	SVM-VGG16	81.06
5	SVM-VGG19	80.92
6	SVM-Inceptionv3	84.79
7	SVM-Xception	87.39
8	SVM-ResNet50, SVM-HoG, SVM-SIFT-BoW	95.84
9	SVM-VGG16, SVM-HoG, SVM-SIFT-BoW	96.27
10	SVM-VGG19, SVM-HoG, SVM-SIFT-BoW	95.84
11	SVM-Inceptionv3, SVM-HoG, SVM-SIFT-BoW	**96.70**
12	SVM-Xception, SVM-HoG, SVM-SIFT-BoW	**96.70**

Table 4 Overall classification accuracy for FP-389

No	Visual approach	Accuracy (%)
1	SVM-SIFT-BoW	87.19
2	SVM-HoG	86.01
3	SVM-ResNet50	62.50
4	SVM-VGG16	85.46
5	SVM-VGG19	85.52
6	SVM-Inceptionv3	85.46
7	SVM-Xception	87.75
8	SVM-ResNet50, SVM-HoG, SVM-SIFT-BoW	96.35
9	SVM-VGG16, SVM-HoG, SVM-SIFT-BoW	*97.28*
10	SVM-VGG19, SVM-HoG, SVM-SIFT-BoW	97.22
11	SVM-Inceptionv3, SVM-HoG, SVM-SIFT-BoW	**97.71**
12	SVM-Xception, SVM-HoG, SVM-SIFT-BoW	97.22

Table 5 Overall classification accuracy for FP-559

No	Visual approach	Accuracy (%)
1	SVM-SIFT-BoW	85.55
2	SVM-HoG	82.65
3	SVM-ResNet50	58.61
4	SVM-VGG16	84.06
5	SVM-VGG19	84.60
6	SVM-Inceptionv3	83.94
7	SVM-Xception	85.89
8	SVM-ResNet50, SVM-HoG, SVM-SIFT-BoW	96.14
9	SVM-VGG16, SVM-HoG, SVM-SIFT-BoW	96.64
10	SVM-VGG19, SVM-HoG, SVM-SIFT-BoW	*96.72*
11	SVM-Inceptionv3, SVM-HoG, SVM-SIFT-BoW	**97.18**
12	SVM-Xception, SVM-HoG, SVM-SIFT-BoW	*96.72*

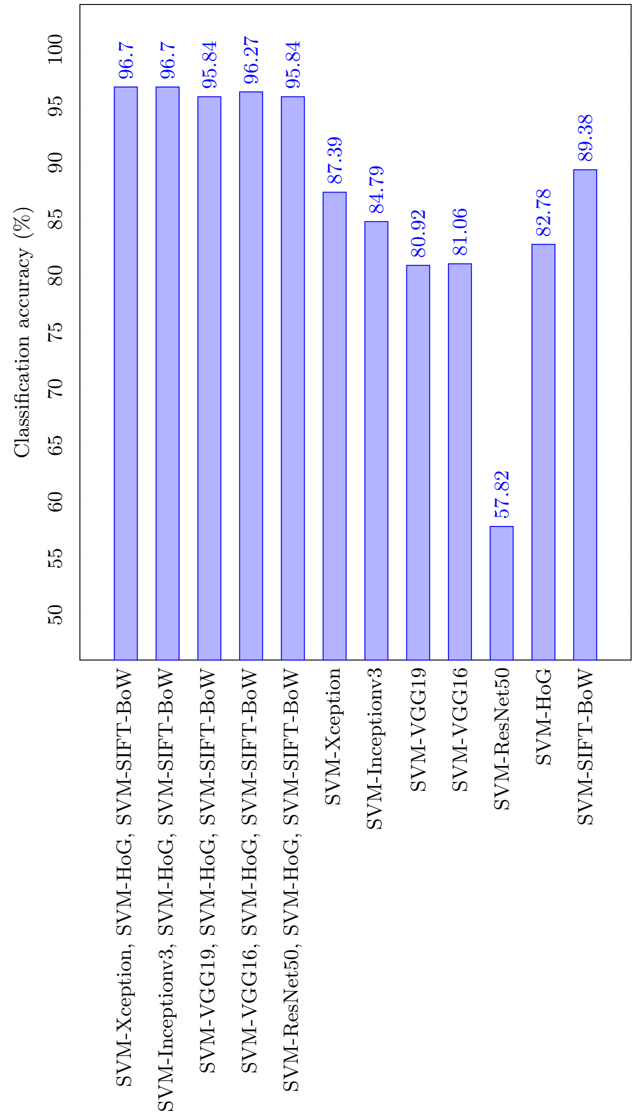


Fig. 11 Overall classification accuracy for FP-235

Classification Results

We obtain the classification accuracy of visual approaches in Tables 3, 4, 5 and Figs. 11, 12, 13. In tables, the highest accuracy is bold-faced and the second one is in italic. The SVM model learned from visual features extracted by the feature extraction method (feature extractor) is denoted by SVM-Feature-Extractor.

In the comparison among visual classification approaches, we can see that the SVM models trained on the single type of features are not suited for classifying fingerprint images. SVMs using deep features extracted by the ResNet50 (denoted by SVM-ResNet50) give lowest correctness. The SVM models learned from features performed by SIFT-BoW or Xception give competitive classification results compared to among SVM classifiers trained on the other single feature type.

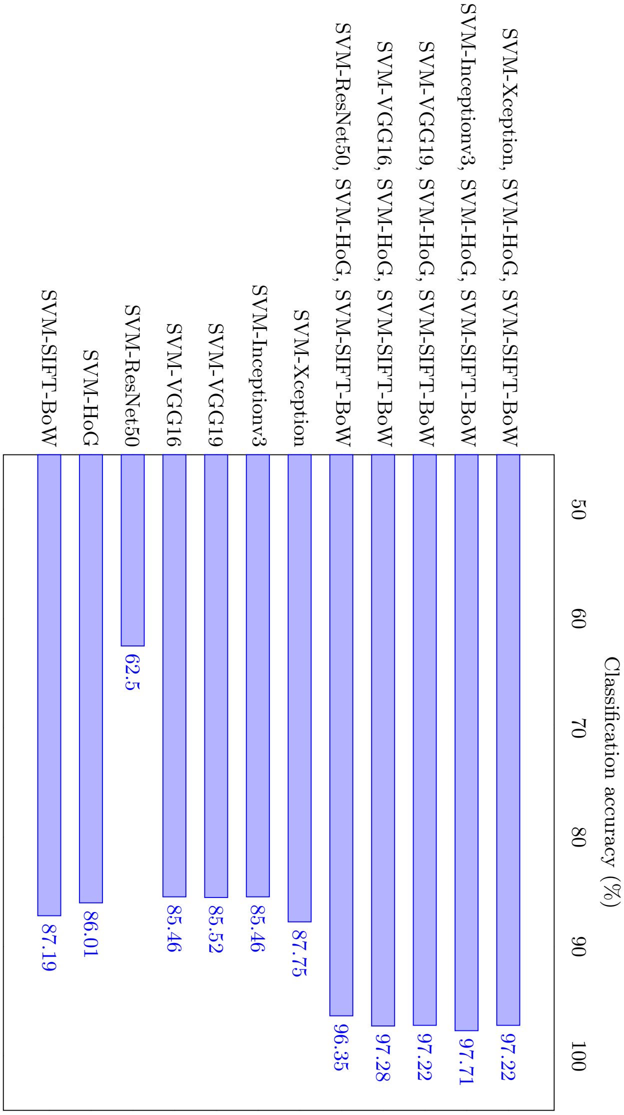


Fig. 12 Overall classification accuracy for FP-389

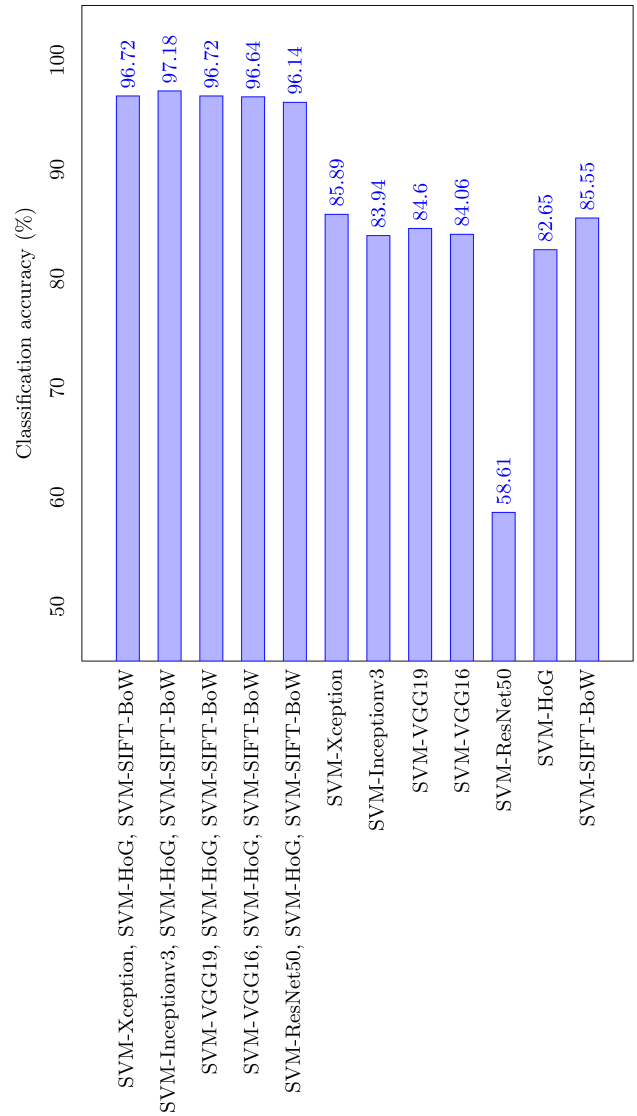


Fig. 13 Overall classification accuracy for FP-559

As mentioned in Sect. 3, we are interested in the effectiveness of training the neural network on top of SVM models learned from SIFT-BoW, HoG, and deep visual features to complement each other.

Our proposed training neural networks on top of SVM models achieve most classification accuracy. On average, SVM-ResNet50, SVM-HoG, and SVM-SIFT-BoW achieve 59.64%, 83.81%, and 87.37% classification accuracy but training neural networks on top of SVM-SIFT-BoW, SVM-HoG, SVM-ResNet50 (denoted by the triplet <SVM-SIFT-BoW, SVM-HoG,SVM-ResNet50>) gives 96.11% accuracy. It means that <SVM-SIFT-BoW, SVM-HoG,SVM-ResNet50> improves 36.47%, 12.30%, and 8.74% classification accuracy of SVM-SIFT-BoW, SVM-HoG, and SVM-ResNet50, respectively.

The same effectiveness of training neural networks on top of other SVM models, the superiority of <SVM-VGG16, SVM-HoG, SVM-SIFT-BoW> against SVM-VGG16, SVM-HoG, and SVM-SIFT-BoW are 13.20%, 12.91%, and 9.36%, accuracy.

The improvements of the triplet <SVM-VGG19, SVM-HoG, and SVM-SIFT-BoW> over each single visual classifier are 12.91%, 12.78%, and 9.22%.

The triplet <SVM-Inceptionv3, SVM-HoG, SVM-SIFT-BoW> makes 12.47%, 13.38%, 9.82% classification accuracy better than SVM-Inceptionv3, SVM-HoG, and SVM-SIFT-BoW.

The triplet <SVM-Xception, SVM-HoG, SVM-SIFT-BoW> also improves 9.64%, 12.84%, and 9.28% classification accuracy of SVM-Xception, SVM-HoG, and SVM-SIFT-BoW, respectively.

Table 6 Additional training time for FP-235

No	Visual approach	Additional train time (min)
8	SVM-ResNet50, SVM-HoG, SVM-SIFT-BoW	1.78
9	SVM-VGG16, SVM-HoG, SVM-SIFT-BoW	13.65
10	SVM-VGG19, SVM-HoG, SVM-SIFT-BoW	7.9
11	SVM-Inceptionv3, SVM-HoG, SVM-SIFT-BoW	1.85
12	SVM-Xception, SVM-HoG, SVM-SIFT-BoW	1.78

Table 7 Additional training time for FP-389

No	Visual approach	Additional train time (min)
8	SVM-ResNet50, SVM-HoG, SVM-SIFT-BoW	8.82
9	SVM-VGG16, SVM-HoG, SVM-SIFT-BoW	75.77
10	SVM-VGG19, SVM-HoG, SVM-SIFT-BoW	42.18
11	SVM-Inceptionv3, SVM-HoG, SVM-SIFT-BoW	8.86
12	SVM-Xception, SVM-HoG, SVM-SIFT-BoW	8.60

Table 8 Additional training time for FP-559

No	Visual approach	Additional train time (min)
8	SVM-ResNet50, SVM-HoG, SVM-SIFT-BoW	23.25
9	SVM-VGG16, SVM-HoG, SVM-SIFT-BoW	225.77
10	SVM-VGG19, SVM-HoG, SVM-SIFT-BoW	99.14
11	SVM-Inceptionv3, SVM-HoG, SVM-SIFT-BoW	23.99
12	SVM-Xception, SVM-HoG, SVM-SIFT-BoW	23.53

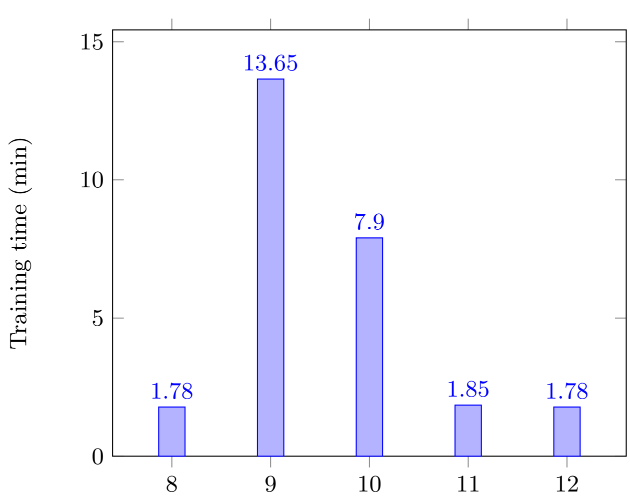


Fig. 14 Additional training time for FP-235

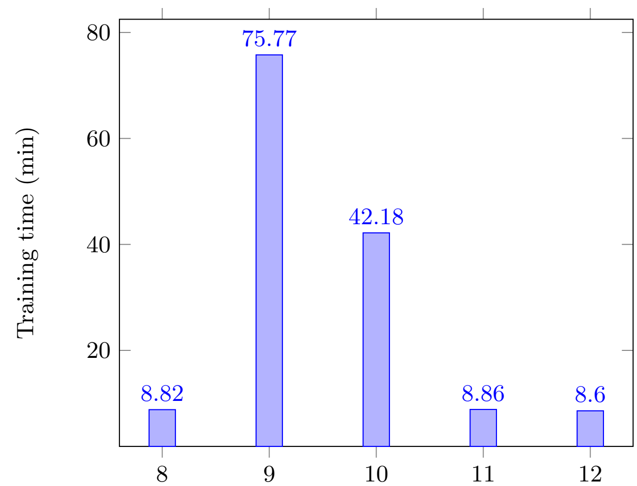


Fig. 15 Additional training time for FP-389

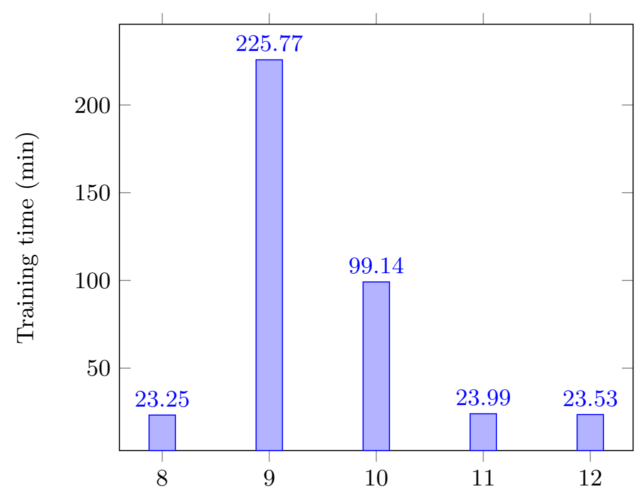


Fig. 16 Additional training time for FP-559

These empirical results illustrate that our proposed method improves the classification correctness against the SVM models trained on the single type of visual features.

However, training the neural network on top of SVM models learned takes additional training time, as presented in Tables 6, 7, 8 and Figs. 14, 15, 16. Due to the large number of features (200,704 features) extracted from VGG networks, triplets using VGG models have longest training time.

Conclusions and Future Work

We have presented the new proposal, training neural networks on top of SVM classifiers learned from different visual features for improving the fingerprint image classification. We collect three real fingerprint image datasets from our students and colleagues. Visual approaches train SVM classifiers on diverse types of visual features, including two most popular handcrafted features such as SIFT-BoW, HoG, and deep features learned by recent deep neural networks, including VGG16, VGG19, ResNet50, Inceptionv3, and Xception, to classify fingerprint images. The empirical test results on real fingerprint image datasets show that the SVM model separately trained on any single visual feature type is not suited for recognizing fingerprint images. Training neural networks on top of SVM classifiers offers a full complement of visual feature types, making improvements of the classification correctness given by any single one. An example of the proposed approach's effectiveness on average is that a neural network trained on top of SVM-ResNet50, SVM-HoG, and SVM-SIFT-BoW improves 36.47%, 12.30%, and 8.74% classification accuracy of SVM-ResNet50, SVM-HoG, and SVM-SIFT-BoW, respectively.

In the future, we will propose a new deep neural network architecture to serve the fingerprint image classification with feature extractors. We will focus on the transfer learning approach [25, 49, 62] and efficient visual models to improve fingerprint image classification results. We intend to provide more empirical test on large benchmarks.

Acknowledgements This work has received support from the College of Information Technology, Can Tho University. The author would like to thank very much the Big Data and Mobile Computing Laboratory.

References

1. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. TensorFlow: Large-scale machine learning on heterogeneous systems 2015; https://www.tensorflow.org/, software available from tensorflow.org.
2. Alonso-Fernandez F, Fiérrez J. Fingerprint databases and evaluation. In: Li SZ, Jain AK, editors. Encyclopedia of Biometrics. 2nd ed. USA: Springer; 2015. p. 599–606.
3. Alonso-Fernandez F, Fiérrez J, Ortega-Garcia J. Quality measures in biometric systems. IEEE Secur Priv. 2012;10(6):52–62.
4. Bosch A, Zisserman A, Munoz X. Scene classification via pLSA. In: Proceedings of the European Conference on Computer Vision. 2006;517–530.
5. Cao K, Jain AK. Fingerprint indexing and matching: An integrated approach. In: 2017 IEEE International Joint Conference on Biometrics (IJCB). 2017;437–445. https://doi.org/10.1109/BTAS.2017.8272728.
6. Cao K, Jain AK. Automated latent fingerprint recognition. IEEE Trans Pattern Anal Mach Intell. 2019;41(4):788–800. https://doi.org/10.1109/TPAMI.2018.2818162.
7. Cao K, Nguyen DL, Tymoszek C, Jain AK. End-to-end latent fingerprint search. IEEE Trans Inf Forensics Secur. 2020;15:880–94. https://doi.org/10.1109/TIFS.2019.2930487.
8. Cappelli R, Ferrara M, Maltoni D. Large-scale fingerprint identification on gpu. Information Sciences. 2015;306:1–20. https://doi.org/10.1016/j.ins.2015.02.016. https://www.sciencedirect.com/science/article/pii/S0020025515001097.
9. Chang CC, Lin CJ. LIBSVM : a library for support vector machines. ACM Trans Intell Syst Technol. 2011;2(27):1–27.
10. Chollet F, et al. Keras. https://keras.io (2015).
11. Chollet F. Xception: deep learning with depthwise separable convolutions. CoRR arXiv:abs/1610.02357 2016.
12. Chugh T, Arora SS, Jain AK, Paulter NG. Benchmarking fingerprint minutiae extractors. In: 2017 International Conference of the Biometrics Special Interest Group (BIOSIG). 2017;1–8. https://doi.org/10.23919/BIOSIG.2017.8053498.
13. Cristianini N, Shawe-Taylor J. An introduction to support vector machines: and other Kernel-based learning methods. New York: Cambridge University Press; 2000.
14. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01. 2005;886–893. IEEE Computer Society.
15. Deng J, Berg AC, Li K, Li F. What does classifying more than 10, 000 image categories tell us? In: Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V. 2010;71–84.
16. Do T, Lenca P, Lallich S. Classifying many-class high-dimensional fingerprint datasets using random forest of oblique decision trees. Vietnam J Comput Sci. 2015;2(1):3–12.
17. Do T, Lenca P, Lallich S, Pham N. Classifying very-high-dimensional data with random forests of oblique decision trees. In: Guillet, F., Ritschard, G., Zighed, D.A., Briand, H. (eds.) Advances in Knowledge Discovery and Management [Best of EGC 2009, Strasbourg, France]. Studies in Computational Intelligence, vol. 292, 2009;39–55. Springer.
18. Do T, Pham T, Pham N, Nguyen H, Tabia K, Benferhat S. Stacking of svms for classifying intangible cultural heritage images. In: Advanced Computational Methods for Knowledge Engineering—Proc. of the 6th Intl Conf. on Computer Science, Applied Mathematics and Applications, ICCSAMA 2019. Advances in Intelligent Systems and Computing, vol. 1121, 2019;186–196. Springer.
19. Do T, Pham T, Tran-Nguyen M. Fine-tuning deep network models for classifying fingerprint images. In: 12th International Conference on Knowledge and Systems Engineering, KSE 2020, Can Tho City, Vietnam, November 12-14, 2020. 2020;79–84. IEEE.
20. Dorizzi B, Cappelli R, Ferrara M, Maio D, Maltoni D, Houmani N, Garcia-Salicetti S, Mayoue A. Fingerprint and on-line signature verification competitions at icb 2009. In: Tistarelli M, Nixon MS, editors. Advances in Biometrics. Berlin Heidelberg: Springer; 2009. p. 725–32.
21. Engelsma JJ, Cao K, Jain AK. Learning a fixed-length fingerprint representation. IEEE Trans Pattern Anal Mach Intell. 2021;43(6):1981–97. https://doi.org/10.1109/TPAMI.2019.2961349.
22. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: a library for large linear classification. J Mach Learn Res. 2008;9(4):1871–4.
23. Fiumara G, Flanagan P, Grantham J, Bandini B, Ko K, Libert J. National institute of standards and technology special database 300:uncompressed plain and rolled images from fingerprint cards. Tech. rep., National Institute of Standards and Technology 2018; https://doi.org/10.6028/NIST.TN.1993.
24. 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. 2014;2672–2680.

25. Goodfellow IJ, Bengio Y, Courville AC. Deep Learning. Adaptive computation and machine learning: MIT Press; 2016.
26. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. CoRR arXiv:abs/1512.03385. 2015.
27. Itseez: Open source computer vision library. https://github.com/itseez/opencv 2015.
28. Jain AK, Feng J, Nandakumar K. Fingerprint matching. IEEE Comput. 2010;43(2):36–44.
29. Jain AK, Nandakumar K, Ross A. 50 years of biometric research: accomplishments, challenges, and opportunities. Pattern Recogn Lett. 2016;79:80–105.
30. Kreßel UHG. Pairwise classification and support vector machines. In: Schölkopf B, Burges CJC, Smola AJ, editors. Advances in Kernel Methods. Cambridge: MIT Press; 1999. p. 255–68.
31. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Commun ACM. 2017;60(6):84–90. https://doi.org/10.1145/3065386.
32. LeCun Y. Self-supervised learning: Could machines learn like humans? November 2018.
33. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86:2278–324.
34. Li F, Perona P. A bayesian hierarchical model for learning natural scene categories. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego. 2005;524–531.
35. Lin C. A practical guide to support vector classification 2003.
36. Liu D, Liu Y, Li S, Li W, Wang L. Fusion of handcrafted and deep features for medical image classification. J Phys: Conference Series. 2019. https://doi.org/10.1088/1742-6596/1345/2/022052.
37. Lowe D. Object recognition from local scale invariant features. In: Proceedings of the 7th International Conference on Computer Vision. 1999;1150–1157.
38. Lowe D. Distinctive image features from scale invariant keypoints. Int J Comput Vision 2004;91–110.
39. MacQueen J. Some methods for classification and analysis of multivariate observations. Berkeley Symposium on Mathematical Statistics and Probability, University of California Press. 1967;1:281–97.
40. Maltoni D. Fingerprint Recognition, Overview, pp. 664–668. Springer US, Boston 2015.
41. Maltoni D, Maio D, Jain AK, Prabhakar S. Handbook of Fingerprint Recognition. 2nd ed. Springer; 2009.
42. Militello C, Rundo L, Vitabile S, Conti V. Fingerprint classification based on deep learning approaches: experimental findings and comparisons. Symmetry. 2021. https://doi.org/10.3390/sym13050750.
43. Minaee S, Abdolrashidi A, Su H, Bennamoun M, Zhang D. Biometrics recognition using deep learning: a survey. arXiv e-prints arXiv:1912.00271 Nov 2019.
44. Minaee S, Azimi E, Abdolrashidi A. Fingernet: Pushing the limits of fingerprint recognition using convolutional neural network. CoRR arXiv:abs/1907.12956 2019.
45. Mistry V, Engelsma JJ, Jain AK. Fingerprint synthesis: search with 100 million prints. In: 2020 IEEE International Joint Conference on Biometrics (IJCB). 2020;1–10. https://doi.org/10.1109/IJCB48548.2020.9304885.
46. Pandya B, Cosma G, Alani AA, Taherkhani A, Bharadi V, McGinnity T. Fingerprint classification using a deep convolutional neural network. In: 2018 4th International Conference on Information Management (ICIM). 2018;86–91. https://doi.org/10.1109/INFOMAN.2018.8392815.
47. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
48. Pham T, Tran-Nguyen M, Tran M, Do T. Combining support vector machines for classifying fingerprint images. In: Future Data and Security Engineering - 7th International Conference, FDSE 2020, Quy Nhon, Vietnam, November 25-27, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12466, 2020;399–410. Springer.
49. Razavian AS, Azizpour H, Sullivan J, Carlsson S. CNN features off-the-shelf: An astounding baseline for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014. 2014;512–519. IEEE Computer Society.
50. Sharma V, Gool LV. Image-level classification in hyperspectral images using feature descriptors, with application to face recognition. CoRR arXiv:abs/1605.03428 2016.
51. Shrein JM. Fingerprint classification using convolutional neural networks and ridge orientation images. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). Nov 2017;1–8.
52. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. CoRR arXiv:abs/1409.1556 2014.
53. Sivic J, Zisserman A. Video google: a text retrieval approach to object matching in videos. In: 9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France. 2003;1470–1477.
54. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.
55. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015;1–9. https://doi.org/10.1109/CVPR.2015.7298594
56. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. CoRR arXiv:abs/1512.00567 2015.
57. Vapnik V. The Nature of Statistical Learning Theory. Springer-Verlag; 1995.
58. Wallraven C. Graf: Recognition with local features: the kernel recipe. Proceedings Ninth IEEE International Conference on Computer Vision. 2003;1:257–64. https://doi.org/10.1109/ICCV.2003.1238351.
59. Wolpert D. Stacked generalization. Neural Netw. 1992;5:241–59.
60. Wu K, Liu X, Shao J, Hong R, Yang T. Retrieving images by multiple samples via fusing deep features. In: Chen, E., Gong, Y., Tie, Y. (eds.) Advances in Multimedia Information Processing - PCM 2016 - 17th Pacific-Rim Conference on Multimedia, Xi'an, China, September 15-16, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9916, 2016;221–230. Springer. https://doi.org/10.1007/978-3-319-48890-5_22.
61. Xu Y, Yu X, Wang T, Xu Z. Pooling region learning of visual word for image classification using bag-of-visual-words model. PLOS ONE. 2020;15(6):1–19. https://doi.org/10.1371/journal.pone.0234144.
62. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec. 2014;3320–3328.