International Journal of Automation and Computing 18(1), February 2021, 18-44
DOI: 10.1007/s11633-020-1257-9

A Performance Evaluation of Classic Convolutional Neural Networks for 2D and 3D Palmprint and Palm Vein Recognition

Wei Jia[1,2] Jian Gao[1,2] Wei Xia[1,2] Yang Zhao[1,2] Hai Min[1,2] Jing-Ting Lu[3]

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China

[2] Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Hefei 230009, China

[3] Institution of Industry and Equipment Technology, Hefei University of Technology, Hefei 230009, China

Abstract: Palmprint recognition and palm vein recognition are two emerging biometrics technologies. In the past two decades, many traditional methods have been proposed for palmprint recognition and palm vein recognition, and have achieved impressive results. However, the research on deep learning-based palmprint recognition and palm vein recognition is still very preliminary. In this paper, in order to investigate the problem of deep learning based 2D and 3D palmprint recognition and palm vein recognition in-depth, we conduct performance evaluation of seventeen representative and classic convolutional neural networks (CNNs) on one 3D palmprint database, five 2D palmprint databases and two palm vein databases. A lot of experiments have been carried out in the conditions of different network structures, different learning rates, and different numbers of network layers. We have also conducted experiments on both separate data mode and mixed data mode. Experimental results show that these classic CNNs can achieve promising recognition results, and the recognition performance of recently proposed CNNs is better. Particularly, among classic CNNs, one of the recently proposed classic CNNs, i.e., EfficientNet achieves the best recognition accuracy. However, the recognition performance of classic CNNs is still slightly worse than that of some traditional recognition methods.

Keywords: Performance evaluation, convolutional neural network (CNN), biometrics, palmprint, palm vein, deep learning.

1 Introduction

In the network and digital society, personal authentication is becoming a basic social service. It is well known that biometrics technology is one of the most effective solutions to personal authentication[1]. In recent years, two emerging biometrics technologies, palmprint recognition and palm vein recognition have attracted a wide range of attention[2–6]. Generally speaking, there are three subtypes of palmprint recognition technology, including 2D low resolution palmprint recognition, 3D palmprint recognition and high resolution palmprint recognition. High-resolution palmprint recognition is usually used for forensic applications. 2D low-resolution palmprint recognition and 3D palmprint recognition are mainly used for civil applications. In this paper, we only focus on civil applications of biometrics, therefore, the problem of high-resolution palmprint recognition will not be investigated.

Many effective methods have been proposed for 2D low-resolution palmprint recognition (2D low-resolution palmprint recognition will be called 2D palmprint recognition for short in the rest of this paper), 3D palmprint recognition and palm vein recognition, which can be divided into two groups, i.e., traditional methods and deep learning-based methods.

In the past decade, deep learning has become the most important technology in the field of artificial intelligence. It has brought a breakthrough in performance for many applications[7, 8], such as speech recognition, natural language processing, computer vision, image and video analysis, and multimedia. In the field of biometrics, especially in face recognition, deep learning has become the most mainstream technology[9]. However, the research on deep learning-based 2D and 3D palmprint recognition and palm vein recognition is still very preliminary[9, 10].

Convolution neural network (CNN) is one of the most important branches of deep learning technology, and has been widely used in various tasks of image processing and computer vision, such as target detection, semantic segmentation and pattern recognition. For image-based biometrics technologies, CNN is the most commonly used deep learning technique. Up to now, many classic CNNs have been proposed and impressive results have been achieved in many recognition tasks. However, the recognition performance of these classic CNNs for 2D and 3D palmprint recognition and palm vein recognition has not

Research Article
Manuscript received August 6, 2020; accepted September 25, 2020; published online December 29, 2020
Recommended by Associate Editor Hui Yu

been systematically studied. For example, existing deep learning-based palmprint recognition and palm vein recognition work only used simple networks, and did not provide an in-depth analysis. In the future, with the rapid development of CNNs, the recognition accuracy of new CNNs will be continuously improved. It can be predicted that CNNs will become one of the most important techniques for 2D and 3D palmprint recognition and palm vein recognition. Therefore, it is very important to systematically investigate the recognition performance of classic CNNs for 2D and 3D palmprint recognition and palm vein recognition. To this end, this paper evaluates the performance of classic CNNs in 2D and 3D palmprint recognition and palm vein recognition. Particularly, seventeen representative and classic CNNs are exploited for performance evaluation.

The selected CNNs are evaluated on five 2D palmprint databases, one 3D palmprint database and two palm vein databases, all of which are representative databases in the field of 2D and 3D palmprint recognition and palm vein recognition. The five 2D palmprint databases include Hong Kong Polytechnic University palmprint II database (PolyU II)[11], the blue band of the Hong Kong Polytechnic University multispectral (PolyU M_B) palmprint database[12], Hefei University of Technology (HFUT) palmprint database[13], Hefei University of Technology cross sensor (HFUT CS) palmprint database[14], and Tongji University palmprint (TJU-P) database[15]. The 3D palmprint database we used is Hong Kong Polytechnic University 3D palmprint database (PolyU 3D)[16]. Two palm vein databases include the near-infrared band of Hong Kong Polytechnic University multispectral palmprint database (PolyU M_N)[12] and Tongji University palm vein (TJU-PV) database[17].

It should be noted that the samples within the above databases are captured in two different sessions at certain time intervals. In traditional recognition methods, some samples captured in the first session are usually used as training sets, while all the samples captured in the second session are used as the test set. However, in existing deep learning-based palmprint recognition and palm vein recognition methods, the training set often contains samples from both sessions. Thus, it is easy to obtain a high recognition accuracy. If the training samples are only from the first session, and the test samples are from the second session, we call this experimental mode a separate data mode. If the training samples are from both sessions, we call this experimental mode a mixed data mode. We conduct experiments in both separate data mode and mixed data mode to observe the recognition performance of classic CNNs in these two different modes.

The main contributions of our work are as follows.

1) We briefly summarize the classic CNNs, which can help the readers to better understand the development history of CNNs for image classification tasks.

2) We evaluate the performance of the classic CNNs for 3D palmprint and palmprint recognition. To the best of our knowledge, it is the first time such an evaluation has been conducted.

3) We evaluated the performance of classic CNNs on Hefei University of Technology cross sensor palmprint database. To the best of our knowledge, it is the first time the problem of palmprint recognition across different devices using deep learning technology has been investigated.

4) We investigate the problem of the recognition performance of CNNs on both separate data mode and mixed data mode.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 briefly introduces seventeen classic CNNs. Section 4 introduces the 2D and 3D palmprint and palm vein databases used for evaluation. Extensive experiments are conducted and reported in Section 5. Section 6 offers the concluding remarks.

2 Related work

2.1 Traditional 2D palmprint recognition methods

For 2D palmprint recognition, researchers have proposed many traditional methods, which can be divided into different sub-categories, such as palm line-based, texture-based, orientation coding-based, correlation filter-based, and subspace learning-based[3].

Because palm line is the basic feature of palmprint, some methods exploiting palm line features for recognition have been proposed. Huang et al.[18] proposed the modified finite Radon transform (MFRAT) to extract principal lines, and designed a pixel-to-area algorithm to match the principal lines of two palmprints. Palma et al.[19] used a morphological top-hat filtering algorithm to extract principal lines, and proposed a dynamic matching algorithm involving a positive linear dynamical system.

The texture-based method is also very effective for pattern recognition. Some local texture descriptors were designed and used for palmprint recognition[20]. Replacing the gradient by the response of Gabor filters in the local descriptor of histogram of oriented gradients (HOG), Jia et al.[21] proposed the descriptor of histogram of oriented lines (HOL) for palmprint recognition. Later, Luo et al.[22] proposed the descriptor of local line directional pattern (LLDP) using the modulation of two orientations. Motivated by LLDP, Li and Kim[23] proposed the descriptor called the local micro-structure tetra pattern (LMTrP). To fully utilize different direction information of a pixel and explore the most discriminant direction representation, Fei et al.[24] proposed the methods of the local discriminant direction binary pattern (LDDBP), the discriminant direction binary palmprint descriptor

(DDBPD)[25], and the apparent and latent direction code (ALDC)-based descriptor[26]. Scale-invariant feature transform (SIFT) is a powerful descriptor and has been applied to palmprint recognition. Using SIFT, Wu and Zhao[27] tried to solve the problem of deformed palmprint matching.

Orientation is a robust feature of palmprint. A lot of orientation coding-based methods have been proposed. These methods have high accuracy and fast matching speed. Generally, orientation coding-based methods first detect the orientation of each pixel, and then encode the orientation number to a bit string, at last, exploit Hamming distance for matching. Jia et al.[13] summarized orientation coding-based methods. Typical orientation coding-based methods include competitive code[11], ordinal code[28], robust line orientation code (RLOC)[29], binary orientation co-occurrence vector (BOCV)[30], double-orientation code (DOC)[31], etc.

Recently, correlation-based methods have been successfully used for biometrics. Jia et al.[13] proposed to use a band-limited phase-only correlation (BLPOC) filter for palmprint recognition.

Subspace learning has been one of the important techniques for pattern recognition. Some subspace learning-based methods have been used for palmprint recognition, including principal component analysis (PCA)[32], linear discriminant analysis (LDA)[33], kernel PCA (KPCA)[34], etc. However, the recognition performance of subspace learning-based methods is sensitive to illumination changes and other image variations.

2.2 Traditional 3D palmprint recognition methods

For 3D palmprint recognition, researchers have proposed a lot of traditional methods[5,10,16,35]. Generally, 3D palmprint data preserves the depth information of a palm surface. At the same time, the original captured 3D palmprint data is a small positive or negative float, which is usually transformed into the grey-level value for practical feature extraction. In previous researches, the original 3D palmprint data is usually transformed into a curvature-based data. Two most important curvatures include the mean curvature (MC) and Gaussian curvature (GC). Their corresponding gray images are called as mean curvature image (MCI) and Gaussian curvature image (GCI)[35]. In the recognition process, researchers extracted features from MCI or GCI for 3D palmprint recognition. Besides GC and MC, researchers also tried to propose other 2D representations of 3D palmprints. Based on GC and MC, Yang et al.[36] proposed a new grey-level image representation, called surface index image (SI). Recently, Fei et al.[37] proposed a simple yet effective compact surface type (CST) to represent surface features of a 3D palmprint. Since the representations of MCI, GCI, SI and CST depict a 3D palmprint as a 2D grey-level palmprint image, those 2D palmprint recognition methods can be used for 3D palmprint recognition. Li et al.[16] extracted competitive code from MCI for 3D palmprint recognition, which is an important orientation coding method. Zhang et al.[5] proposed a blockwise statistics-based ST vector for 3D palmprint feature representation, and used collaborative representation-based classification (CRC) as the classifier. Fei et al.[38] proposed a complete binary representation (CBR) for the 3-D palmprint recognition by combining descriptors extracted from both MCI and CST. Fei et al.[39] proposed the precision direction code (PDC) to depict the 2D texture-based features, and then combined CST to form the PDCST descriptor to represent the multiple level and multiple dimensional features of 3D palmprint images.

2.3 Traditional palm vein recognition methods

For palm vein recognition, traditional methods can also be divided into the following categories: vein line-based, texture-based, orientation coding-based, and subspace learning-based.

To extract palm vein lines, Zhang et al.[40], Kang and Wu[6] proposed two typical methods. In Zhang′s method, the multiscale Gaussian matched filters were exploited to extract vein lines[40]. In Kang′s method, the normalized gradient-based maximal principal curvature (MPC) algorithm was exploited to extract vein lines[6].

Kang and Wu[6] proposed a texture-based method, in which a mutual foreground-based linear binary pattern (LBP) was exploited for texture feature extraction. Mirmohamadsadeghi and Drygajlo[41] also proposed a texture-based method, in which two texture descriptors, LBP and local derivative patterns (LDP) were used for palm vein recognition. ManMohan et al.[42] proposed a palm vein recognition method using local tetra patterns (LTP). Kang et al.[43] investigated the SIFT-based method for palm vein recognition.

Zhou and Kumar[44] presented an orientation coding-based method for palm vein recognition, named neighborhood matching Radon transform (NMRT) which is similar to the RLOC method proposed for palmprint recognition. The experimental results showed that the recognition performance of NMRT is much better than other methods such as Hessian phase, ordinal code, competitive code, and SIFT.

For subspace learning-based methods, the LDA[45], 2DLDA[46], $(2D)^2$LDA[47], and sparse representation methods[48] have been studied for palm vein recognition.

2.4 The brief development history of classic CNNs

Fig. 1 shows the chronology of the events in the development history of classic CNNs for image classification tasks. In 1998, the first CNN, LeNet, was proposed by

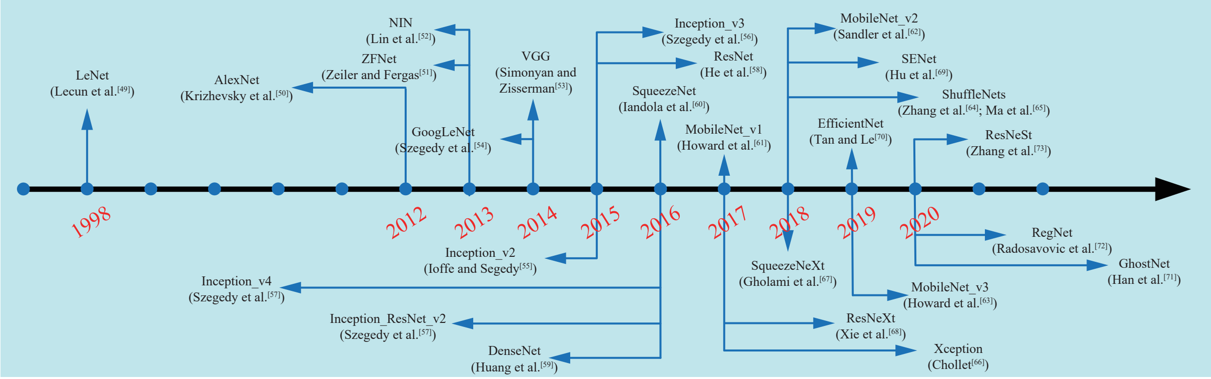


Fig. 1 Chronology of classic CNNs chronology for classification tasks

Lecun et al.[49] However, LeNet did not have a widespread impact due to various restrictions. In 2012, AlexNet was proposed by Hinton and his student Krizhevsky and won the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC 2012)[50]. AlexNet demonstrated the effectiveness of CNN in some complex tasks. As a result, excellent performance of AlexNet attracted the attention of researchers, and promoted the further development of CNNs. In 2013, ZFNet was proposed by Zeiler and Fergus[51]. Zeiler and Fergus[51] also explain the essence of each layer of the neural network through visualization technology. In 2013, network in network (NIN) was proposed, which has two important contributions including global average pooling and the use of 1×1 convolution layer[52]. In 2014, VGG was proposed by the Oxford Visual Geometry Group[53], and was the 2nd runner-up in ILSVRC 2014. Compared with AlexNet, VGG has two important improvements. The first one is using a smaller kernel size. The second one is using a deeper network. In 2014, another most important CNN is GoogLeNet (Inception_v1)[54], which was the champion of the ILSVRC 2014. Later, the subsequent versions of GoogLeNet, i.e., Inception_v2[55], Inception_v3[56] and Inception_v4[57], were successively proposed in 2016 and 2017. Inception_ResNet_v1 and Inception_ResNet_v2 were proposed in the same paper, which are improved versions of Inception_v4[57]. In 2015, ResNet was proposed by He et al.[58] and won the ILSVRC 2015. The paper of ResNet obtained the best paper of CVPR 2016. It can be said that the emergence of ResNet is an important event in the history of deep learning, because ResNet made it possible to train hundreds of layers of neural networks, and ResNet greatly improved the performance of image classification and other computer vision tasks. In 2016, DenseNet was proposed by Huang et al.[59] As the best paper of CVPR 2017, DenseNet broke away from the stereotyped thinking of deepening network structure (ResNet) and broadening network structure (Inception) to improve network performance. Considered from the point of view of features, DenseNet not only greatly reduces the amount of network parameters, but also alleviates the gradient vanishing problem to a certain extent by feature reuse and bypass settings. In the same year, SqueezeNet, the first lightweight network, was proposed by Iandoula et al.[60] to compress the number of feature maps by using a 1×1 convolution core to accelerate the network. Since then, other important lightweight networks such as MobileNets[61−63], ShuffleNets[64,65], Xception[66], SqueezeNeXt[67] were proposed in turn. In order to solve the problem of poor information circulation, MoblieNets uses the strategy of point-wise convolution, ShuffleNets uses the strategy of channel shuffle, Xception uses the strategy of modified depth-wise convolution, and SqueezeNeXt hoists the speed from the perspective of hardware. In 2017, Xie et al.[68] proposed ResNeXt combining ResNet and Inception, which does not need to design complex structural details manually. Particularly, ResNeXt uses the same topological structure for each branch, the essence of which is group convolution. In 2018, the winner of the last image classification mission, SENet was proposed by Hu et al.[69] SENet consists of squeeze and excitation, in which the former compresses the model, and the latter predicts the importance of each channel. In addition, SENet can be plugged into any network to improve the recognition performance. In 2019, EfficientNet was proposed by Google[70], which relies on AutoML and compound scaling to achieve state-of-the-art accuracy without compromising resource efficiency. In 2020, the team of Huawei Noah's Ark Lab proposed a lightweight network, i.e., GhostNet, which can achieve better recognition performance than MobileNet_v3 with similar computational cost[71]. Some members from Facebook AI Research (FAIR) developed RegNet that outperforms EfficientNet while being up to 5× faster on GPUs[72]. The work of RegNet presented a new network design paradigm, which combines the advantages of manual network design and neural network search (NAS). By stacking split-attention blocks, Zhang et al.[73] proposed a new ResNet variant, i.e., ResNeSt, which has better recognition performance than ResNet.

2.5 2D and 3D palmprint recognition and palm vein recognition methods based on deep learning

A lot of researchers have studied 2D and 3D palmprint recognition and palm vein recognition based on deep learning. Table 1 summarizes the existing CNN-based 2D palmprint recognition methods including the networks, training data configuration, and performance. ERR represents equal error rate. Jalali et al.[74] used the whole palmprint image without region of interest (ROI) extraction to train a four-layer CNN for 2D palmprint recognition. Zhao et al.[75] proposed a 2D palmprint recognition method by using a deep belief network (DBN). Minaee and Wang[76] proposed a 2D palmprint recognition method based on a deep scattering convolutional network (DSCNN). Liu and Sun[77] used AlexNet to extract

Table 1 Summary of existing 2D palmprint recognition methods based on deep learning

Reference	Publish year	Networks	Database	Training data configuration	Performance	
					Recognition rate	EER
Jalali et al.[74]	2015	4-layer CNN	PolyU H	Mixed data mode	99.98%	N/A
Zhao et al.[75]	2015	DBN	BJJU	Mixed data mode	90.63%	N/A
Minaee and Wang[76]	2016	DSCNN	PolyU M	Mixed data mode	100%	N/A
Liu and Sun[77]	2017	AlexNet	PolyU II	Not provided	N/A	0.04%
			CASIA		N/A	0.08%
			IITD		N/A	0.11%
Svoboda et al.[78]	2016	4-layer CNN	IITD	Mixed data mode	N/A	1.64%
			CASIA		N/A	1.86%
Yang et al.[79]	2018	VGG-F	PolyU II PolyU M_B	Mixed data mode	N/A	0.166 5%
Meraoumia et al.[80]	2017	PCANet	CASIA	Mixed data mode	N/A	0.299%
Zhang et al.[17]	2018	PalmRCNN	TJU-P	Mixed data mode	100%	2.74%
Zhong et al.[81]	2018	Siamese network	PolyU M	Not provided	N/A	0.281 9%
			XJTU		N/A	4.559%
Michele et al.[82]	2019	MobileNet_V2	PolyU_M_B	Mixed data mode	100%	N/A
Genovese et al.[83]	2019	PalmNet	CASIA	Mixed data mode	99.77%	0.72%
			IIDT		99.37%	0.52%
			REST		97.16%	4.50%
			TJU-P		99.83%	0.16%
Zhong et al.[81]	2018	CNN with C-LMCL	TJU-P	Mixed data mode	99.93%	0.26%
			PolyU		100%	0.125%
Matkowski et al.[85]	2020	EE-PRnet	IIDT	Mixed data mode	99.61%	0.26%
			PolyU-CF		99.77%	0.15%
			CASIA		97.65%	0.73%
Zhao and Zhang[86]	2020	DDR based on VGG-F	PolyU R	Mixed data mode	100%	0.000 4%
			PolyU G		100%	0.000 1%
			PolyU B		100%	0.000 4%
			PolyU NIR		100%	0.003 6%
			IIDT		98.70%	0.003 8%
			CASIA		99.41%	0.005 2%
Zhao and Zhang[87]	2020	JCLSR	PolyU R	Mixed data mode	100%	N/A
			PolyU G		99.99%	N/A
			PolyU B		100%	N/A
			PolyU NIR		100%	N/A
			IIDT		98.17%	N/A
			CASIA		98.94%	N/A
			GPDS		96.33%	N/A

the features of palmprint images and combined hausdorff distance for matching and recognition. Svoboda et al.[78] trained CNN with palmprint ROI and d-prime loss function, and observed that d-prime loss function has better effect than contrastive loss function. In addition, Yang et al.[79] combined the methods of deep learning and local coding. They first extracted the features of palmprint with CNN, and then used local coding to encode the extracted features. Meraoumia et al.[80] applied PCANet for palmprint recognition, which is an unsupervised convolutional deep learning network. Zhang et al.[17] proposed the method of PalmRCNN for palmprint and palm vein recognition, which is a modified version of Inception_ResNet_v1. Zhong et al.[81] applied a Siamese network for 2D palmprint recognition. Michele et al.[82] used MobileNet_v2 to extract palmprint features and then explored support vector machine (SVM) for classification. Genovese et al.[83] proposed the method of PalmNet, which is a CNN that uses a method to tune palmprint specific filters through an unsupervised procedure based on Gabor responses and principal component analysis (PCA). Zhong and Zhu[84] proposed an end-to-end method for open-set 2D palmprint recognition by applying CNN with a novel loss function, i.e., centralized large margin cosine loss (C-LMCL). In order to solve the problem of palmprint recognition in uncontrolled and uncooperative environments, Matkowski et al.[85] proposed end-to-end palmprint recognition network (EE-PRnet) consisting of two main networks, i.e., ROI localization and alignment network (ROI-LAnet) and feature extraction and recognition network (FERnet). Zhao and Zhang[86] proposed a deep discriminative representation (DDR) for palmprint recognition. DDR uses several CNNs similar to VGG-F to extract deep features from global and local palmprint images. Lastly, the collaborative representation-based classifier (CRC) is used for recognition. Zhao and Zhang[87] presented a joint constrained least-square regression (JCLSR) model with a deep local convolution feature for palmprint recognition. Zhao et al.[88] also proposed a joint deep convolutional feature representation (JDCFR) methodology for hyperspectral palmprint recognition.

Table 2 summarizes the existing CNN-based 3D palmprint recognition methods. Generally, in these methods, the CNNs were applied to different 2D representations of 3D palmprints for recognition such as MCI, GCI, and ST. Samai et al.[89] proposed to use DCTNet for 3D palmprint recognition. Chaa et al.[90] firstly used a single scale retinex (SSR) algorithm to enhance the depth image of 3D palmprint, then used PCANet for recognition.

Table 3 summarizes the existing CNN-based palm vein recognition methods. Hassan and Abdulrazzaq[91] proposed to use CNN for palm vein recognition, in which they designed a simple CNN and used the strategy of data augmentation to obtain more training data. Zhang et al.[17] released a new touchless palm vein database, and used the method of PalmRCNN for palm vein recognition. Lefkovits et al.[92] applied four CNNs for palm vein identification including AlexNet, VGG-16, ResNet-50, and SqueezeNet. Thapar et al.[93] proposed the method of PVSNet. In PVSNet, using triplet loss, a Siamese network was trained. Chantaf et al.[94] applied Inception_v3 and SmallerVggNet for palm vein recognition.

3 Selected classic CNNs for performance evaluation

For 2D and 3D palmprint recognition and palm vein recognition, we select seventeen classic CNNs for performance evaluation including AlexNet[50], VGG[53], Inception_v3[56], Inception_v4[57], ResNet[58], ResNeXt[68], Inception_ResNet_v2[57],DenseNet[59],Xception[66],MobileNet_v2[62], MobliNet_v3[63], ShunffleNet_v2[65], SENet[69], EfficientNet[70], GhostNet[71], RegNet[72] and ResNeSt[73]. The reasons for choosing these CNNs are as follows: AlexNet and VGG are representatives of early CNNs, furthermore, we hope to compare the performance between early CNNs and recent CNNs; Inception_v3 and Inception_v4 are representatives of GoogLeNet, and Inception_v3 is an improvement of Inception_v1 and Inception_v2, meanwhile, Inception_v4 is a new version of Inception_v3 with a more uniform architecture; ResNet is a very well-known CNN, and can deepen CNN to more than 100 layers, in addition, it can be well trained; Inception_ResNet_v1 and Inception_ResNet_v2 share the overall structure, but Inception_ResNet_v2 is more representative than Inception_ResNet_v1; DenseNet is the extreme version of ResNet; Xception is a new attempt for convolution order; MobileNet_v3 can be used in embedded devices, and is an

Table 2 Summary of 3D palmprint recognition methods based on deep learning

Reference	Publish year	Methodology	Database	Training data configuration	Performance	
					Recognition rate	EER
Samai et al.[89]	2018	MCI+DCTNet	PolyU 3D	Mixed data mode	99.83%	N/A
		GCI+DCTNet			99.22%	N/A
Chaa et al.[90]	2019	GCI+PCANet	PolyU 3D	Mixed data mode	98.63%	0.12%
		MCI+PCANet			98.22%	0.12%
		ST+PCANet			99.88%	0.02%
		SSR+PCANet			99.98%	0

Table 3 Summary of palm vein recognitoin methods based on deep learning

Reference	Publish year	Networks	Database	Training data configuration	Performance	
					Recognition rate	EER
Hassan and Abdulrazzag[91]	2018	A simple CNN	PolyU M_NIR	Mixed data mode	99.73%	N/A
			CASIA		98%	N/A
Zhang et al.[17]	2018	PalmRCNN	TJU-PV	Mixed data mode	100%	2.30%
Lefkovits et al.[92]	2019	AlexNet	PUT	Mixed data mode	92.16%	N/A
		VGG-16			97.33%	N/A
		ResNet-50			99.83%	N/A
		SqueezeNet			91.66%	N/A
Thapar et al.[93]	2019	PVSNet	CASIA	The first half samples are used as the training set	85.16%	3.71
			IITI		97.47%	0.93
			PolyU MS		98.78%	0.66
Chantaf et al.[94]	2020	Inception_v3	20 subjects 4 000 images	Mixed data mode	91.4%	N/A
		SmallerVggNet			93.2%	N/A

improved version of MobileNet_v1 and MobileNet_v2; ShuffleNet_v2 is a good compression network, and is a modified version of ShuffleNet_v1; SENet enhances important features to improve accuracy; EfficientNet, GhostNet, RegNet and ResNeSt are four representative CNNs proposed recently.

In this section, we briefly introduce the selected CNNs as follows.

1) AlexNet

The network structure of AlexNet is shown in Fig. 2. AlexNet is based on LeNet and uses some new techniques such as rectified linear unit (ReLU), dropout and local response normalization (LRN) for the first time[50]. Due to the limitation of hardware capability, the training of AlexNet uses distributed computing technology to distribute network on two GPUs. Each GPU stored half of the parameters. The GPUs can communicate with each other and access memory. Therefore, AlexNet is divided into upper and lower parts, each part corresponding to a single GPU. In AlexNet, data enhancement technology is used, such as random cropping and horizontal flipping of raw data, to improve the generalization of the network while reducing over-fitting problem.

2) VGG

VGG is a further improvement of AlexNet, which makes the network deeper[53]. The VGG′s structure is shown in Fig. 3. Because the size of all convolutional kernel is 3×3, the structure of VGG is neat and its topology is simple. Small convolution kernel size also brings some benefits, such as increasing the number of layers. VGG expands the number of layers of CNN to more than 10, enhancing the expressive ability of the network and facilitating subsequent modification in network structure.

3) Inception_v3

Based on Inception_v2, Inception_v3 further decomposed the convolution[56]. That is, any $n \times n$ convolution can be replaced by a $1 \times n$ convolution followed by a $n \times 1$ convolution (see Fig. 4(a)), which can reduce a lot of parameters, avoid over-fitting problems, and strengthen the nonlinear expression ability. In addition, Szegedy et al.[56] have carefully designed three types of Inception module, as shown in Fig. 4(b).

4) ResNet

As the depth of the network continues to increase, the vanishing gradient and exploding gradient problems became more and more difficult to solve. In this situation, it is hard to train the deep network. But ResNet can overcome this difficulty[58]. ResNet relies on a shortcut connection structure called residual module. Multiple residual modules are sequentially stacked to form ResNet, as shown in Fig. 5(a). Actually, the shortcut connection performs identity mapping, and its outputs are added to the output of the following layer. This simple calculation does not increase the number of parameters and computational complexity, and can improve the performance and speed up the training. The residual module actually con-

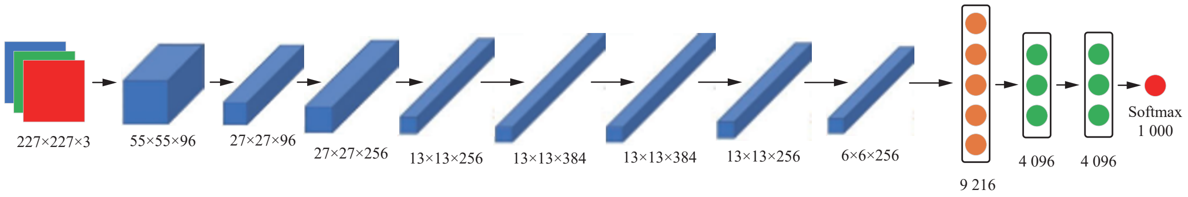


Fig. 2 Structure of AlexNet

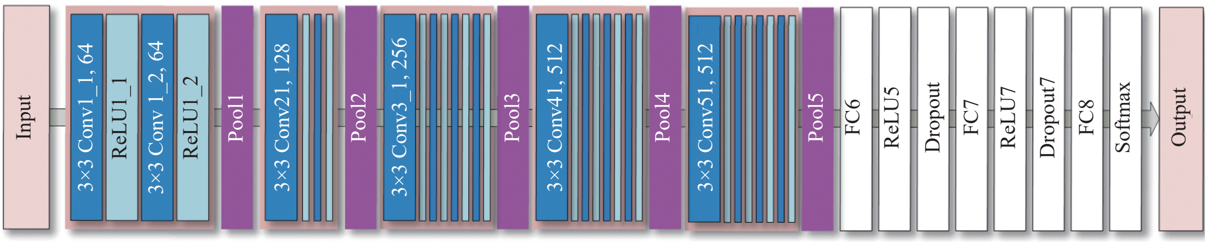
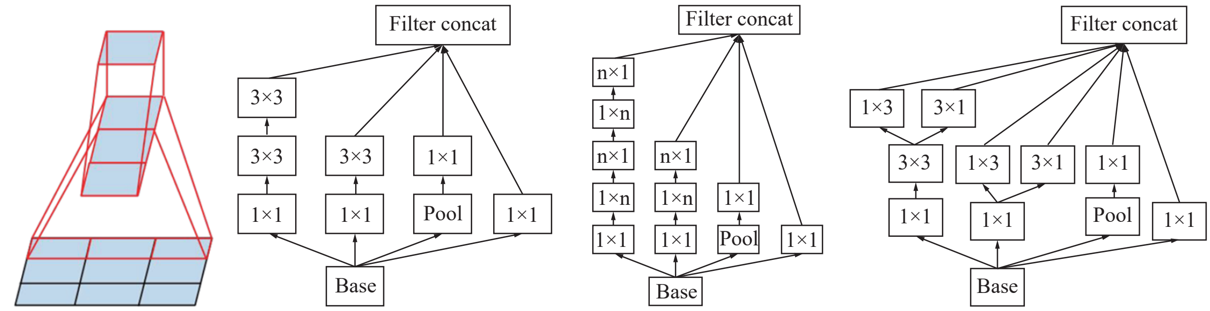


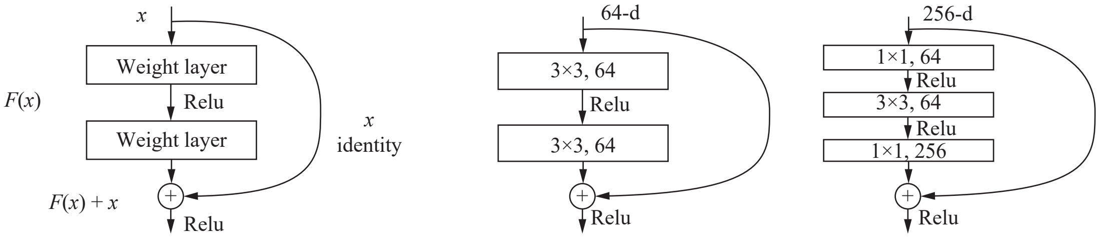
Fig. 3 Structure of VGG



(a) $1{\times}n + n{\times}1$ convolution instead of $n{\times}n$ convolution

(b) Three different Inception modules in Inception_v3

Fig. 4 Module of Inception_v3: (a) $1{\times}n$ + $n{\times}1$ convolution instead of $n{\times}n$ convolution; (b) Three different Inception modules in Inception_v3.



(a) Residual module in ResNet

(b) Two forms of residual module

Fig. 5 Module of ResNet: (a) Residual module in ResNet; (b) Two forms of residual module.

tains two types. One is the basic module, as shown in the left of Fig. 5(b), and the other is the Bottleneck Block, as shown in the right of Fig. 5(b). The bottleneck module replaces the 3×3 convolution with two 1×1 convolution, which also reduces the number of parameters and computational complexity, and increases the nonlinear expression of the network.

5) Inception_v4

Inception_v4 is an improved version of Inception_v3[57]. Compared with regular network structure such as VGG and ResNet, Inception_v4 is mainly composed of one input stem, three Inception modules and two reduction modules, each of which is designed separately. The overall structure of Inception_v4 and the structure of each module are shown in Fig. 6.

6) Inception_ResNet_v2

While designing Inception_v4, Szegedy et al.[57] introduced the residual modules into Inception_v3 and Inception_v4, respectively, resulting in Inception_ResNet_v1 and Inception_ResNet_v2. The overall structure of Inception_ResNet_v1 and Inception_ResNet_v2 is the same, and the difference is the modules in the network. Fig. 7 shows the Inception_ResNet_v2 overall structure and module structure.

7) DenseNet

It seems that DenseNet is an extreme version of ResNet[59]. DenseNet introduces short connections from any layer to all the following layers. But, in fact, DenseNet combines features by concatenating them instead of summation before features are passed to a layer, which enables the network to make better use of features. As shown in Fig. 8(a), a five-layer dense block is illustrated, in which each layer of output is connected to each subsequent layer. The dense block is continuously stacked to form DenseNet. The structure of DenseNet is depicted in Fig. 8(b).

8) Xception

Xception is another improved version of Inception_v3[66]. It is based on the assumption that spatial convolution (convolution along the horizontal and vertical directions of the feature map) and channel convolution (convolution along the direction of the feature map channel) can be performed independently to separate convolution. As shown in Fig. 9, for the feature maps in the

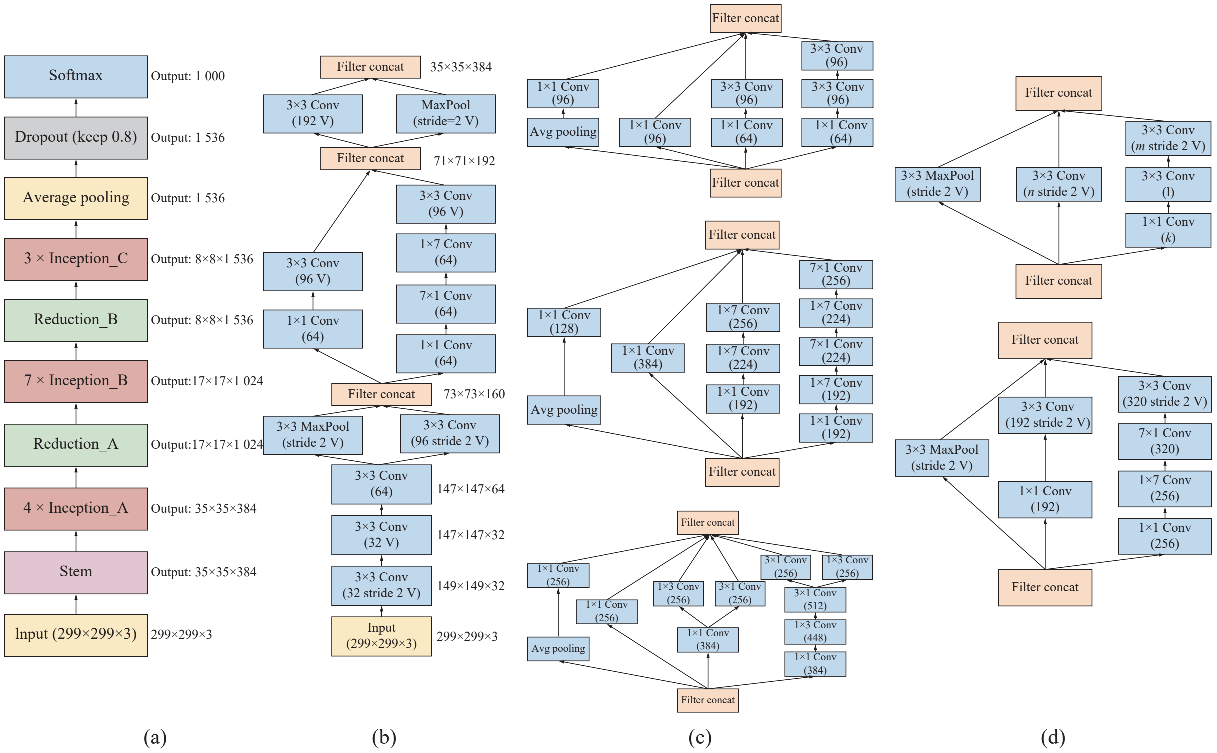


Fig. 6 Inception_v4 overall structure and module structure: (a) Inception_v4 overall structure; (b) Stem module; (c) Up: Inception_A, Mid: Inception_B, Down: Inception_C; (d) Up: Reduction_A, Down: Reduction_B.

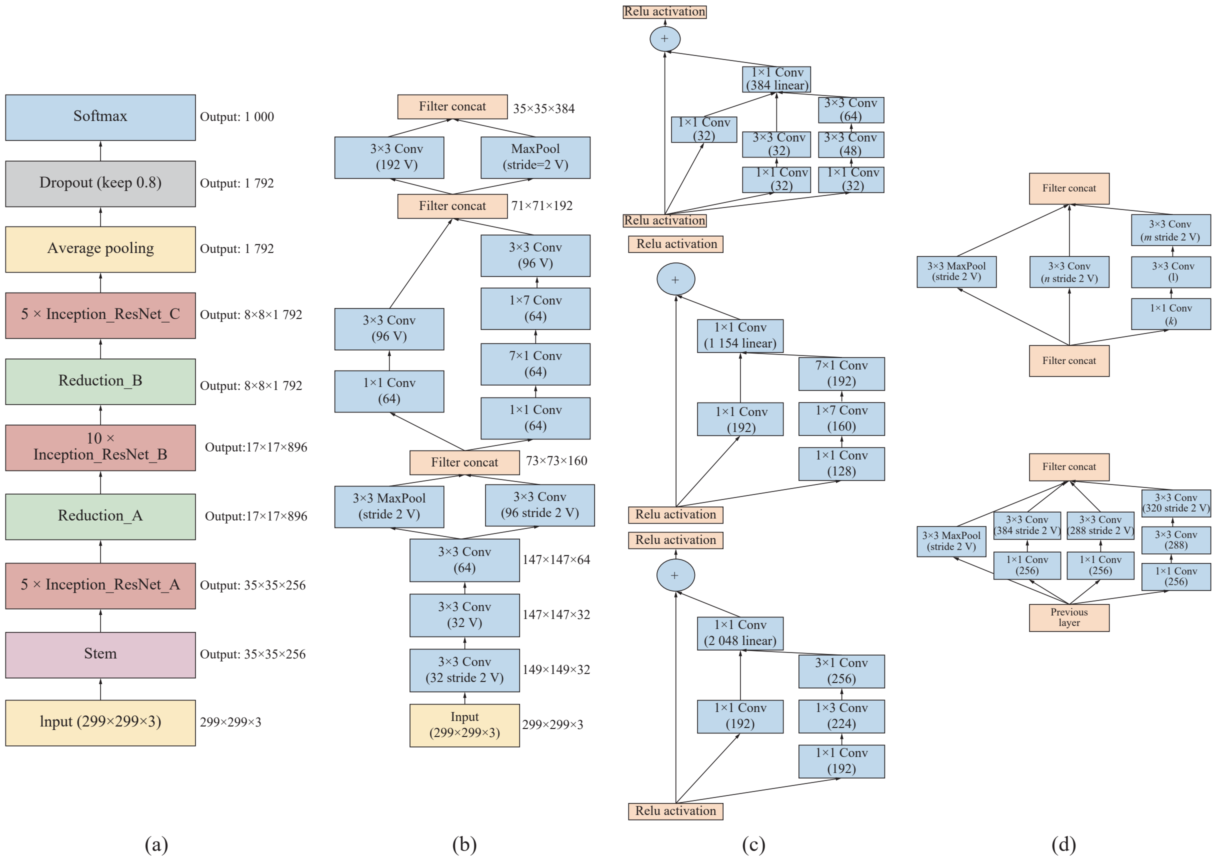
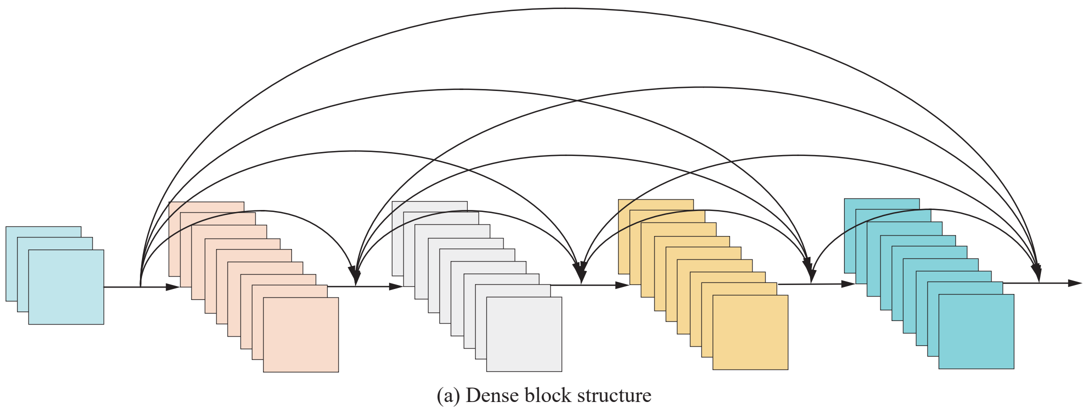
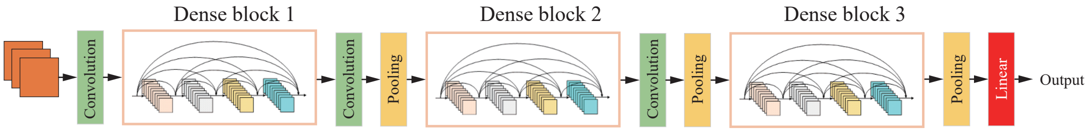


Fig. 7 Inception_ResNet_v2 overall structure and module structure: (a) Overall structure; (b) Stem module; (c) Up: Inception_ResNet_A, Middle: Inception_ResNet_B, Bottom: Inception_ResNet _C; (d) Up: Reduction_A, Bottom: Reduction_B.



(a) Dense block structure



(b) DenseNet overall structure

Fig. 8 Structure of DenseNet: (a) Dense block structure; (b) DenseNet overall structure.

previous layer, 1×1 convolutions are used to linearly combine the feature maps, and then use convolutions separately for each channel, where M is the channel of the feature maps, N is the number of convolutions (or the output channel), n is the size of convolution kernel.

In fact, the Inception module can be simplified as follows: all 1×1 convolutions in Inception can be reformulated as a large convolution, then utilize convolutions separately on every output channel, forming the extreme Inception, as shown in Fig. 10. Extreme Inception is consistent with the initial assumption and achieves the decoupling operation of convolution. This kind of extreme Inception is named Xception.

9) MobileNet_v2 & 10)MobileNet_v3

In order to meet the needs of embedded devices such as mobile phones, the research team of Google proposed a compact neural network named MobileNet in 2017. MobileNet is based on depthwise separable convolution to reduce the number of parameters. Depthwise separable convolution splits the standard convolution into two steps: depthwise convolution, which applies convolution to each channel of the feature map separately and pointwise convolution, which uses 1×1 convolutions to combine the feature, as shown in Fig. 11. In Fig. 11, M is the number of input channels, $D_{\rm K}$ is the convolution kernel size, N is the number of convolution kernels, and if the size of feature map is $D_{\rm F} \times D_{\rm F}$, then for standard convolution, the computational cost is $D_{\rm F} \times D_{\rm F} \times M \times N \times D_{\rm K} \times D_{\rm K}$ and for depthwise separable convolution, the computational cost of the depthwise convolution is $D_{\rm F} \times D_{\rm F} \times M \times D_{\rm K} \times D_{\rm K}$, and the computational cost of the pointwise convolution is $D_{\rm F} \times D_{\rm F} \times M \times N \times 1 \times 1$, so the total computational cost is $D_{\rm F} \times D_{\rm F} \times M \times D_{\rm K} \times D_{\rm K} + D_{\rm F} \times D_{\rm F} \times M \times N$. Therefore, we get a reduction in computation of:

$$\frac{D_{\rm K}\times D_{\rm K}\times M\times D_{\rm F}\times D_{\rm F}+M\times N\times D_{\rm F}\times D_{\rm F}}{D_{\rm K}\times D_{\rm K}\times M\times N\times D_{\rm F}\times D_{\rm F}}=\frac{1}{N}+\frac{1}{D_{\rm K}^2}. \tag{1}$$

For example, if a 3×3 convolution is used, the computational cost can be reduced by about 8 or 9 times. In ad-

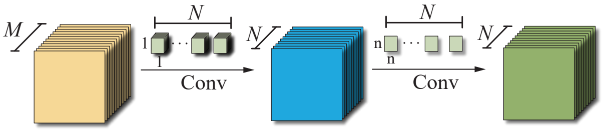
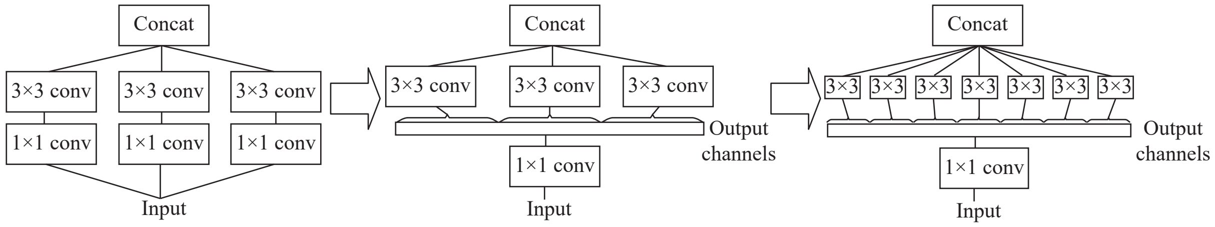


Fig. 9 Spatial convolution and channel convolution in Xception



(a) Inception module (ignoring max pooling) (b) Merging 1×1 convolution (c) Extreme inception

Fig. 10 Formation of the extreme Inception: (a) Inception module (ignoring max pooling); (b) Merging 1 × 1 convolution; (c) Extreme Inception.

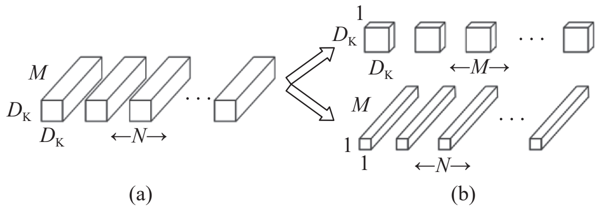
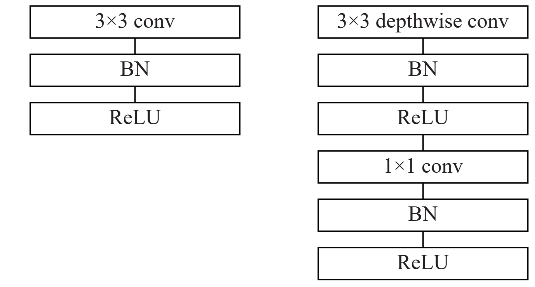


Fig. 11 Depthwise separable convolution: (a) Standard convolution filters; (b) Upper part: depthwise convolution filters, lower part: pointwise convolution filters.

dition, the batch normalization and the nonlinear activation function ReLU are added after the 3×3 convolution and 1×1 convolution, respectively, as shown in Fig. 12.

In 2018, the research team of Google continued to improve MobileNet and designed MobileNet_v2[62]. MobileNet_v2 introduces the shortcut connection in ResNet and DesNet to the network. Since the output of the depthwise separable convolutions is limited by the number of input channels and the characteristics of the bottleneck residual module, if the residual is directly introduced into the network without modification, the initial feature compression will result in too few features available in the subsequent layers. Therefore, MobileNet_v2 proposes Inverted residuals – expanding the number of features first, then extracts the features using convolution, and finally compresses the features. In addition, MobileNet_v2 cancels the ReLU at the end of the inverted residual, because ReLU sets all non-positive inputs to zeros, and adding ReLU after feature compression loses the feature. The network structure of MobileNet_v2 is shown in Fig. 13.

A year later, MobileNet_v3, which gets its model by neural architecture search (NAS), was proposed by the research team of Google[63]. The internal modules of MobileNet_v3 inherit MobileNet_v1, MobileNet_v2 and MnasNet, and networks are researched by platform-aware NAS and NetAdapt. The calculation in the final stage of the network is redesigned on MobileNet_v3 due to the extensive calculation in MobileNet_v2. In addition, a new activation function h-swish[x] is proposed to improve the accuracy of networks effectively. MobileNet_v3 includes two versions: MobileNet_v3-small and MobileNet_v3-large. MobileNet_v3-small has faster speed and its accuracy is similar to MobileNet_v2. MobileNet_v3-large has higher accuracy. Finally, the results of image classification, target detection and semantic segmentation experiments show the advantage of MobileNet_v3. The network structure of MobileNet_v3 is shown in Fig. 14.



(a) Standard convolution layer (b) Depthwise separable convolution layer

Fig. 12 Standard convolution layer and depthwise separable convolution layer: (a) Standard convolution layer; (b) Depthwise separable convolution layer.

11) SENet

Squeeze-and-Excitation (SENet) is a new image recognition structure, which was proposed by the autopilot company Momenta in 2017[69]. It enhances important features by modeling the correlation between feature channels to improve accuracy. The SENet block is a substructure that can be embedded in other classification or detection models. In the 2017 ILSVRC competition, the SENet block and ResNeXt are applied to reduce the top-5 error to 2.251% on the ImageNet dataset, which was the champion in the classification project. The network structure of the SENet block is shown in Fig. 15.

12) ResNeXt

ResNeXt is the upgraded version of ResNet[68]. In order to improve the accuracy of the model, some networks deepen and widen the network structure, resulting in increasing the number of network hyperparameters as well as the difficulty and computational cost of network design. However, ResNeXt improves the accuracy without increasing the complexity of the parameters, even reducing the number of hyperparameters. ResNeXt has three equivalent network structures, as shown in Fig. 16. The original three-layer convolution block in RseNet is replaced by a block of parallel stacking topologies. The topologies are the same, but the hyperparameters are reduced, which facilitates model migration.

13) ShuffleNet_v2

In ResNeXt, the packet convolution strategy is applied as a compromise strategy, and the pointwise convolution of the entire feature map restricts the performance of ResNeXt. Thus, an efficient strategy is to perform

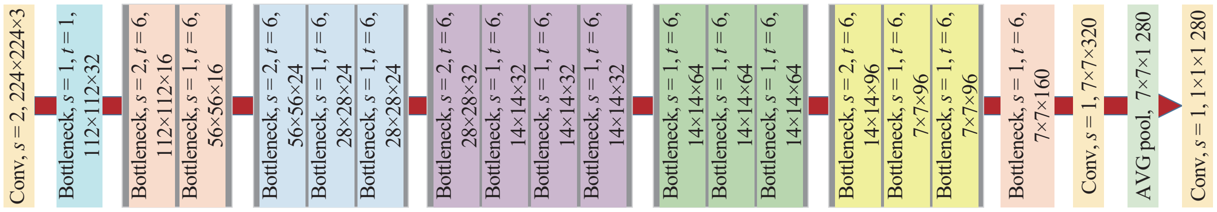
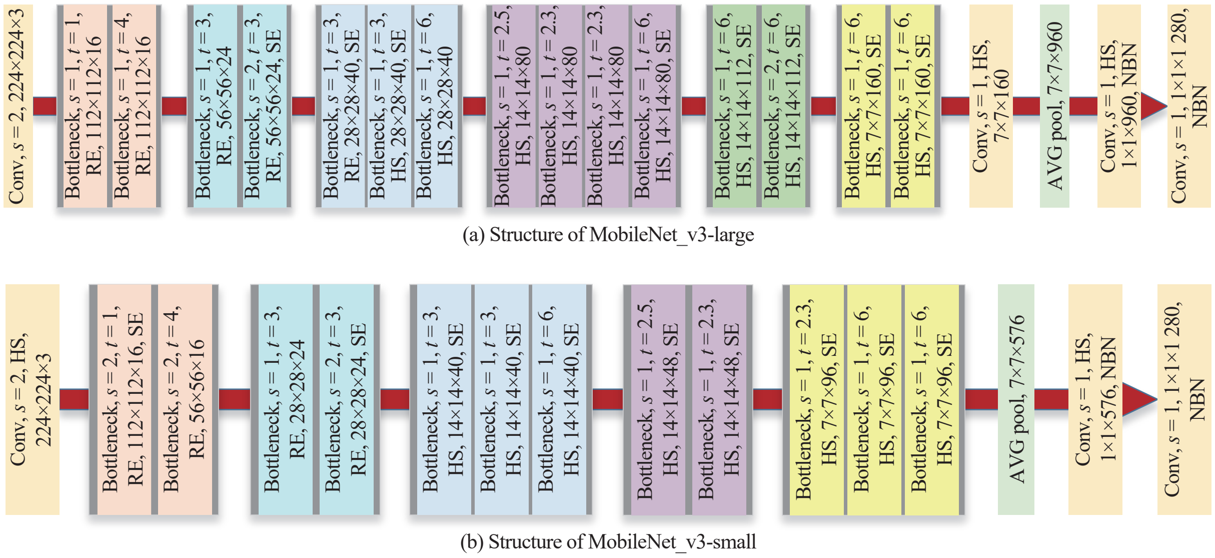

Conv, s = 2, 224×224×3
Bottleneck, s = 1, t = 1, 112×112×32
Bottleneck, s = 2, t = 6, 112×112×16
Bottleneck, s = 1, t = 6, 56×56×16
Bottleneck, s = 2, t = 6, 56×56×24
Bottleneck, s = 1, t = 6, 28×28×24
Bottleneck, s = 1, t = 6, 28×28×24
Bottleneck, s = 2, t = 6, 28×28×32
Bottleneck, s = 1, t = 6, 14×14×32
Bottleneck, s = 1, t = 6, 14×14×32
Bottleneck, s = 1, t = 6, 14×14×32
Bottleneck, s = 1, t = 6, 14×14×64
Bottleneck, s = 1, t = 6, 14×14×64
Bottleneck, s = 1, t = 6, 14×14×64
Bottleneck, s = 2, t = 6, 14×14×96
Bottleneck, s = 1, t = 6, 7×7×96
Bottleneck, s = 1, t = 6, 7×7×96
Bottleneck, s = 1, t = 6, 7×7×160
Conv, s = 1, 7×7×320
AVG pool, 7×7×1 280
Conv, s = 1, 1×1×1 280

Fig. 13 Structure of MobileNet_v2


Conv, s = 2, 224×224×3
Bottleneck, s = 1, t = 1, RE, 112×112×16
Bottleneck, s = 1, t = 4, RE, 112×112×16
Bottleneck, s = 1, t = 3, RE, 56×56×24
Bottleneck, s = 2, t = 3, RE, 56×56×24, SE
Bottleneck, s = 1, t = 3, RE, 28×28×40, SE
Bottleneck, s = 1, t = 3, HS, 28×28×40, SE
Bottleneck, s = 1, t = 6, HS, 28×28×40
Bottleneck, s = 1, t = 2.5, HS, 14×14×80
Bottleneck, s = 1, t = 2.3, HS, 14×14×80
Bottleneck, s = 1, t = 2.3, HS, 14×14×80
Bottleneck, s = 1, t = 6, HS, 14×14×80, SE
Bottleneck, s = 1, t = 6, HS, 14×14×112, SE
Bottleneck, s = 2, t = 6, HS, 14×14×112, SE
Bottleneck, s = 1, t = 6, HS, 7×7×160, SE
Bottleneck, s = 1, t = 6, HS, 7×7×160, SE
Conv, s = 1, HS, 7×7×160
AVG pool, 7×7×960
Conv, s = 1, HS, 1×1×960, NBN
Conv, s = 1, 1×1×1 280, NBN

(a) Structure of MobileNet_v3-large

Conv, s = 2, HS, 224×224×3
Bottleneck, s = 2, t = 1, RE, 112×112×16, SE
Bottleneck, s = 2, t = 4, RE, 56×56×16
Bottleneck, s = 1, t = 3, RE, 28×28×24
Bottleneck, s = 2, t = 3, RE, 28×28×24, SE
Bottleneck, s = 1, t = 3, HS, 14×14×40, SE
Bottleneck, s = 1, t = 3, HS, 14×14×40, SE
Bottleneck, s = 1, t = 6, HS, 14×14×40, SE
Bottleneck, s = 1, t = 2.5, HS, 14×14×48, SE
Bottleneck, s = 1, t = 2.3, HS, 14×14×48, SE
Bottleneck, s = 1, t = 2.3, HS, 7×7×96, SE
Bottleneck, s = 1, t = 6, HS, 7×7×96, SE
Bottleneck, s = 1, t = 6, HS, 7×7×96, SE
AVG pool, 7×7×576
Conv, s = 1, HS, 1×1×576, NBN
Conv, s = 1, 1×1×1 280, NBN

(b) Structure of MobileNet_v3-small

Fig. 14 Structure of MobileNet_v3: (a) Structure of MobileNet_v3-large; (b) Structure of MobileNet_v3-small.

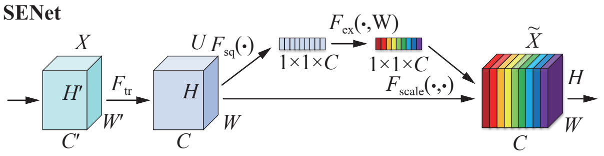


Fig. 15 Structure of SENet block

pointwise convolution within a group, but it is not conducive to information exchange between channels. To solve this problem, ShuffleNet_v1 proposed a channel shuffle operation. The structure of ShuffleNet_v1 is depicted in Fig. 17, where Fig. 17(a) does not need downsampling, and Fig. 17(b) required downsampling operation.

In ShuffleNet_v2[65], researchers found that it is unreasonable to only apply commonly-used FLOPs in the evaluation of model performance, because file IO, memory read, GPU execution efficiency also need to be considered. Taking memory consumption and GPU parallelism into account, researchers designed an efficient ShuffleNet_v2 model. This model is similar to DenseNet, but ShuffleNet_v2 has higher accuracy and faster speed. The network structure of ShuffleNet_v2 is shown in Fig. 18.

14) EfficientNet

EfficientNet was proposed in 2019[70], and is a more general idea for the optimization of current classification networks. Widening the network, deepening the network and increasing the resolution are three common network indicators, which are applied independently in most previous networks. Thus, the compound model scaling algorithm is proposed, which comprehensively optimizes the network width, network depth and resolution to improve the accuracy and the existing classification network, and the amount of model parameters and calculations are greatly reduced. EfficientNet uses the EfficientNet-b0 as the basic network to design eight network structures called b0−b7, and EfficientNet-b7 has the highest accuracy. The network structure of EfficientNet-b0 is shown in Fig. 19.

15) GhostNet

In GhostNet, Han et al.[71] proposed a novel ghost module, which can generate more feature maps with fewer parameters. Specifically, the convolution layer in the depth neural network is divided into two parts. The first part involves the common convolution, but the number of them should be strictly controlled. Given the inherent characteristic graph of the first part, then a series of

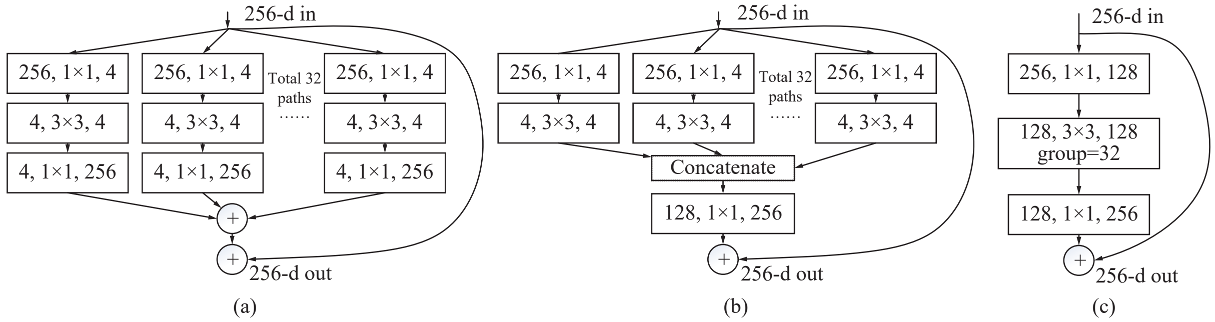


Fig. 16 Equivalent building blocks of ResNeXt: (a) Aggregated residual transformations; (b) A block equivalent to (a), implemented as early concatenation; (c) A block equivalent to (a) and (b), implemented as grouped convolutions.

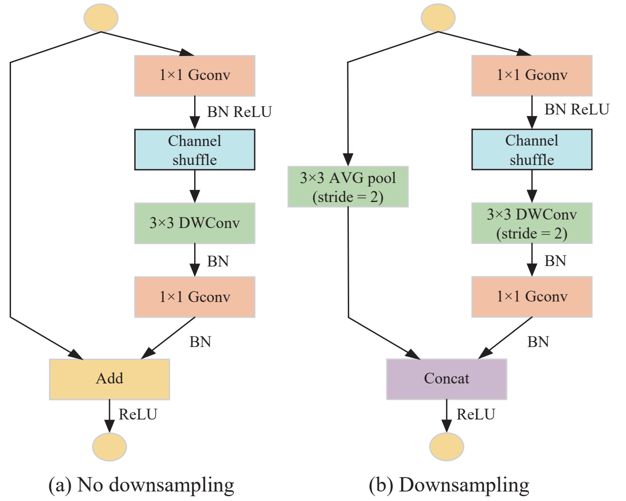


Fig. 17 Structure of ShuffleNet_v1: (a) No downsampling; (b) Downsampling.

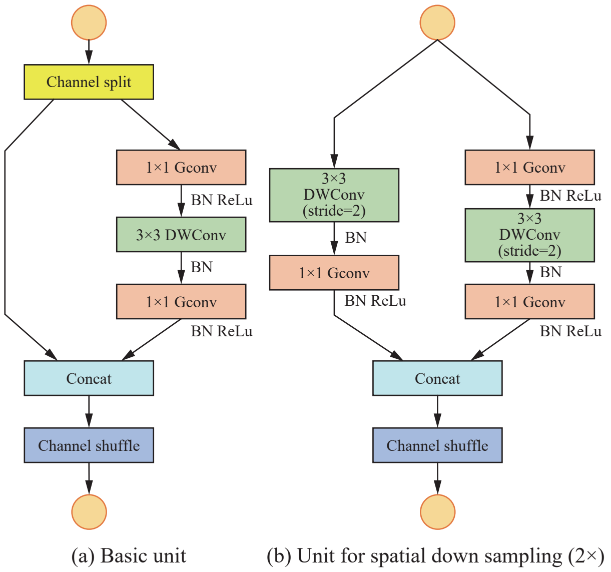


Fig. 18 Structure of ShuffleNet_v2: (a) Basic unit; (b) Unit for spatial down sampling (2×).

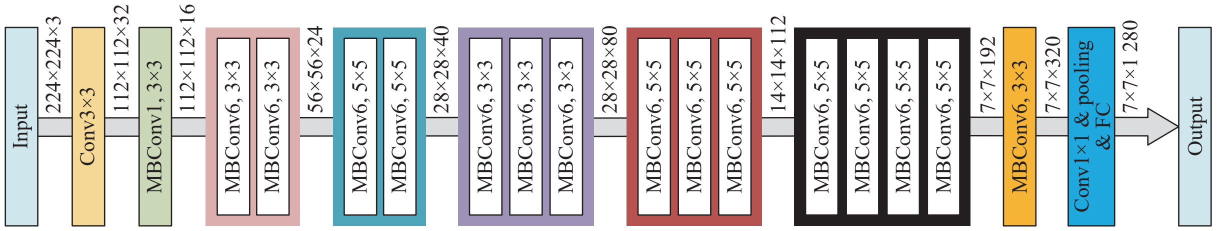


Fig. 19 Structure of EfficientNet-b0

simple linear operations are applied to generate more characteristic graphs. Compared with the conventional CNN, the total number of parameters and the computational complexity of the ghost module are the lowest without changing the size of the output characteristic map. Based on the ghost module, Han et al.[71] proposed GhostNet. Fig. 20 shows the ghost module.

16) RegNet

In RegNet, Radosavovic et al.[72] proposed a new network design paradigm, which aims to help improve the understanding of network design. Radosavovic et al.[72] focused on the design of network design space of parameterized networks. The whole process is similar to the classic manual network design, but it is promoted to the level of design space. Using this rule to search for a simple low dimensional network, i.e., RegNet. The core idea of

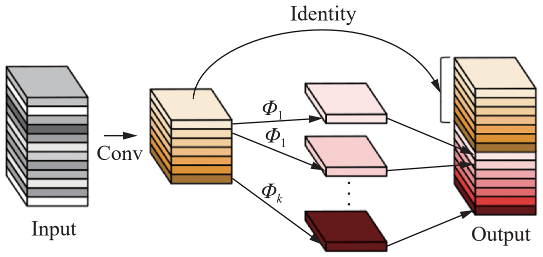


Fig. 20 Ghost module

RegNet parameterization is that the width and depth of a good network can be explained by a quantized linear function. Particularly, RegNet outperforms traditional available models and runs five times on GPUs.

17) ResNeSt

In ResNeSt, Zhang et al.[73] explored the simple architecture modification of ResNet, and incorporated feature-map split attention within the individual network blocks. More specifically, each of the blocks divides the feature-map into several groups (along the channel dimension) and finer-grained subgroups or splits, where the feature representation of each group is determined via a weighted combination of the representations of its splits (with weights chosen based on global contextual information). Zhang et al.[73] refer to the resulting unit as a split-attention block, which remains simple and modular. By stacking several split-attention blocks, a ResNet-like network is created called ResNeSt. The architecture of ResNeSt requires no more computation than existing ResNet-variants, and is easily adopted as a backbone for other vision tasks. The performance of ResNeSt is better than all existing ResNet variants, while the computational efficiency is the same, an even better speed and accuracy tradeoff is achieved than the most advanced CNN model generated by NAS. Fig. 21 shows the ResNeSt block module.

4 2D and 3D palmprint and palm vein databases used for evaluation

In this paper, five 2D palmprint image databases, one 3D palmprint database and two palm vein databases are used for performance evaluation, including PolyU II[11], PolyU M_B[12], HFUT[13], HFUT CS[14], TJU-P[15], PolyU 3D[15], PolyU M_B[12] and TJU-PV[17]. After preprocessing, the ROI sub-images were cropped. The ROI size of all databases is 128×128. The detailed descriptions of above databases are listed in Table 4. Figs. 22–25 depict some ROI images of four 2D palmprint databases. In Figs. 22–25, the three images depicted in the first row were captured in the first session. The three images depicted the second row were captured in the second session.

Fig. 26 shows three original palmprints of HFUT CS database and their corresponding ROI images. Fig. 27 shows three original 3D palmprint data of the PolyU 3D database. Fig. 28 shows four different 2D representations from one 3D palmprint including MCI, GCI, ST and CST. Figs. 29 and 30 depict some ROI images of two 2D palm vein databases. In Figs. 29 and 30, three images depicted in the first row were captured in the first session. Three images depicted the second row were captured in the second session.

PolyU II is a challenging palmprint database because the illuminations between the first session and the second session have an obvious change. HFUT CS is also a challenging palmprint database. From Fig. 25, it can be seen that there are some differences between the palmprints captured by different devices.

5 Experimental results

5.1 Experimental configuration

In this section, we introduce the default configuration

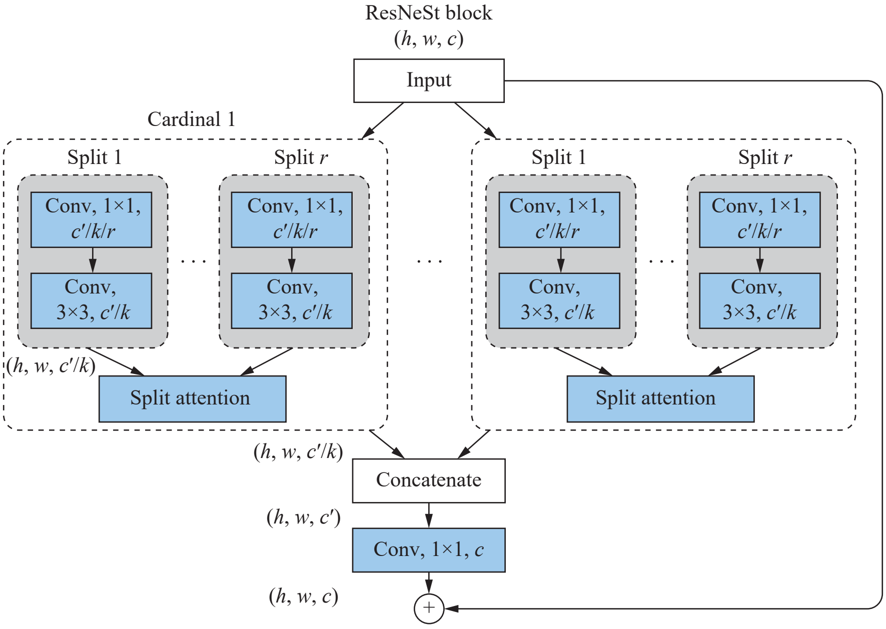


Fig. 21 ResNeSt block module

Table 4 Details of 2D palmprint, 3D palmprint and palm vein databases

Database	Type	Touch?	Individual number	Palm number	Session number	Session interval	Image number of each palm	Total image number
PolyU II	2D Palmprint	Yes	193	386	2	2 months	10×2	7752
PolyU M_B	2D Palmprint	Yes	250	500	2	9 days	6×2	6000
HFUT	2D Palmprint	Yes	400	800	2	10 days	10×2	16000
HFUT CS	2D Palmprint	No	100	200	2	10 days	10×2×3	12000
TJU-P	2D Palmprint	No	300	600	2	61 days	10×2	12000
PolyU 3D	3D Palmprint	Yes	200	400	2	1 months	10×2	8000
PolyU M_N	Palm vein	Yes	250	500	2	9 days	6×2	6000
TJU-PV	Palm vein	No	300	600	2	61 days	10×2	12000

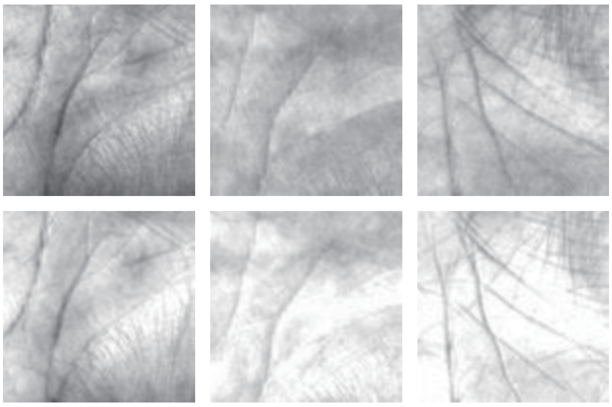


Fig. 22 Six palmprint ROI images of PolyU II database. The three images of the first row were captured in the first session. The three images of the second row were captured in the second session.

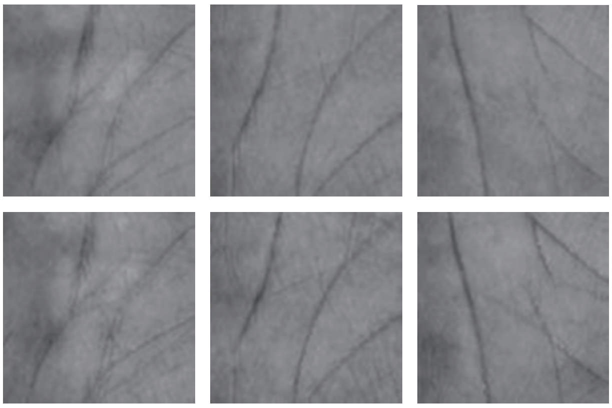


Fig. 23 Six palmprint ROI images of PolyU M_B database. The three images of the first row were captured in the first session. The three images of the second row were captured in the second session.

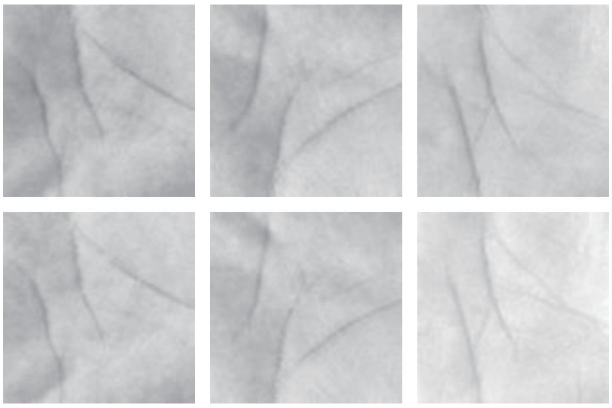


Fig. 24 Six palmprint ROI images of HFUT database. The three images of the first row were captured in the first session. The three images of the second row were captured in the second session.

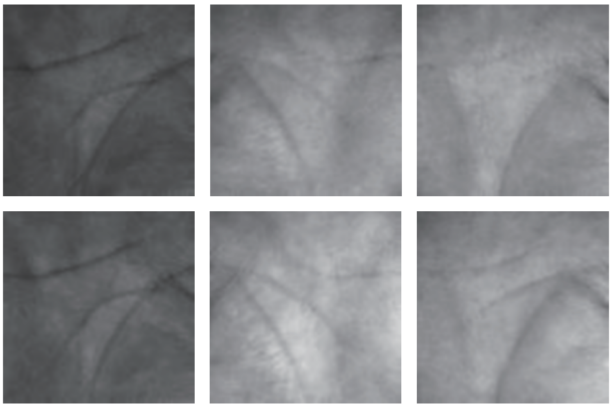


Fig. 25 Six palmprint ROI images of TJU-P database. The three images of the first row were captured in the first session. The three images of the second row were captured in the second session.

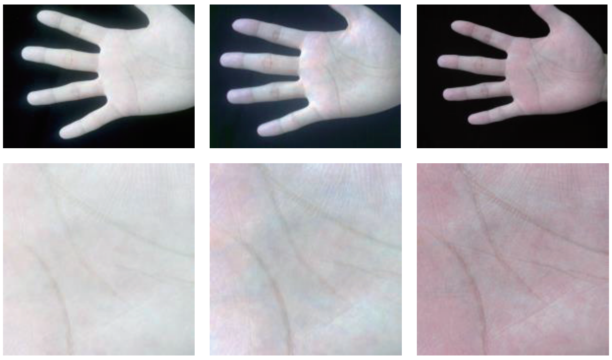


Fig. 26 Three original palmprint and ROI images of HFUT CS database. The three images of the first row are three original palmprint images. The three images of the second row are corresponding ROI images.

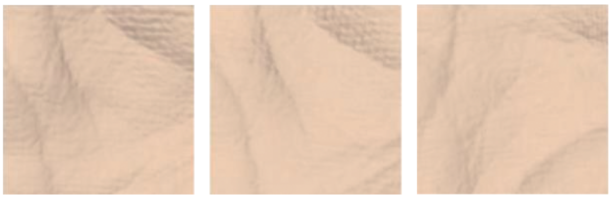


Fig. 27 Three original 3D palmprint ROI images of PolyU 3D database.

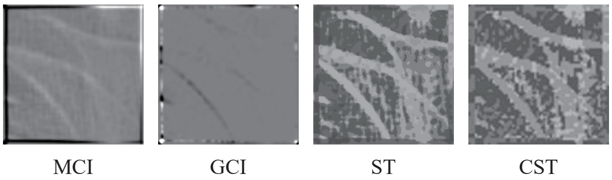


Fig. 28 Four different 2D representations of 3D palmprint ROI images of PolyU 3D database including MCI, GCI, ST, and CST.

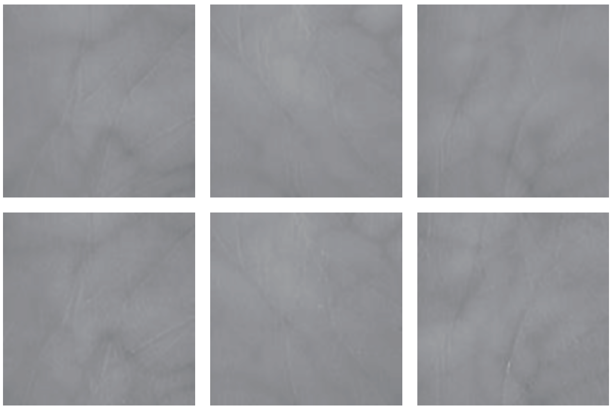


Fig. 29 Six palm vein ROI images of PolyU M_N database. Three images of the first row were captured in the first session. Three images of the second row were captured in the second session.

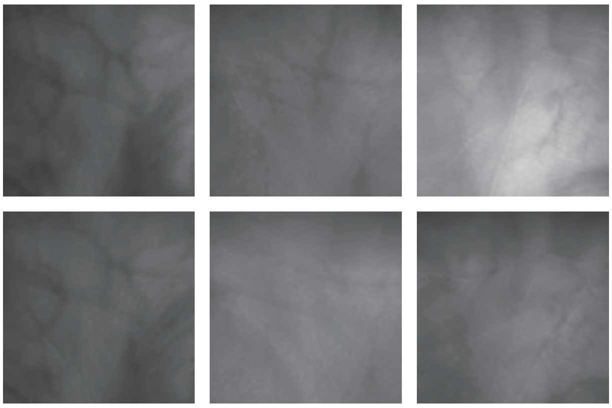


Fig. 30 Six palm vein ROI images of TJU-PV database. Three images of the first row were captured in the first session. Three images of the second row were captured in the second session.

of the experiment, including experimental hyperparameters and hardware configuration. The full names of some CNNs are too long, and it is difficult to insert them into the tables of experimental results. Thus, we provide their abbreviation names in Table 5. It should be noted that in the following experiments, the abbreviation name Res represents the ResNet-18 network; the abbreviation name Dense represents the DenseNet-121 network; the abbreviation name SE represents the SENet-154 network; the abbreviation name ResX represents the ResNeXt-101 network; the abbreviation name Efficient represents the EfficientNet-b7 network; the abbreviation name Reg represents the RegNet-Y network; and the abbreviation name ResS represents the ResNeSt-50 network.

Table 5 Full name and its abbreviated name of selected CNNs

Full name	Abb.*	Reference	Year
AlexNet	Alex	Krizhevsk et al.[50]	2012
VGG	VGG	Simonyan and Zisserman[53]	2015
Inception_v3	IV3	Szegedy et al.[56]	2016
ResNet	Res	He et al.[58]	2016
Inception_v4	IV4	Szegedy et al.[57]	2017
Inception_ResNet_v2	IResV2	Szegedy et al.[57]	2017
DenseNet	Dense	Huang et al.[59]	2017
Xception	Xec	Chollet[66]	2017
ResNeXt	ResX	Xie et al.[68]	2017
MobileNet_v2	MbV2	Howard et al.[62]	2018
ShuffleNet_v2	ShuffleV2	Ma et al.[65]	2018
SENet	SE	Hu et al.[69]	2018
MobileNet_v3	MbV3	Howard et al.[63]	2019
EfficientNet	Efficient	Tan et al.[70]	2019
GhostNet	Ghost	Han et al.[71]	2020
RegNet	Reg	Radosavovic et al.[72]	2020
ResNeSt	ResS	Zhang et al.[73]	2020

*Abb. means the abbreviated name

Since different networks need different input sizes, such as 227×227 in AlexNet, 299×299 in Inception_v3, and 224×224 in ResNet, the palmprint/palmvein ROI image needs to be upsampled to a suitable size before input into the network. In order to enhance the stability of the network, we also added a random flip operation (only during the training phase), i.e., for a training image, there is a certain probability that the image is flipped horizontally and then input into the network. We do not initialize the model parameters using the random parameter initialization method, but initialize it using the parameters of the pretrained model in the ImageNet. The palmprint/palmvein ROI image in the database is usually a grayscale image, that means the number of image channels is 1, and the input of the model is a RGB image, so the grayscale channel of the image is copied three times to form a RGB image.

The system configuration is as follows: Intel CPU i7 4.2GHz, NVIDIA GPU GTX 1080Ti (EfficientNet runs on two parallel GPUs GTX 1080Ti), 16Gb memory and Windows 10 operating system. All evaluation experiments are performed on Pytorch. The cross entropy loss function (CrossEntropyLoss in Pytorch), Adam optimizer is used by default and the batch size is 4.

5.2 Recognition performance on separate data mode

We first conduct evaluation experiments on a separate data mode, i.e., all samples captured in the first session are used for training, and all samples captured in the second session are used for test.

5.2.1 Recognition results of ResNet18 and EfficientNet under different learning rates

Learning rate is a very important hyperparameter in model training, which affects the convergence of the loss function. If the learning rate is too small, the decrease of loss along the gradient direction will be slow, and it will take time to reach the optimal solution. If the learning rate is too large, it may lead optimal solutions to be missed, and may cause severe turbulence and even vanishing gradient problems. Here, we are only looking for the initial learning rate, combined with the dynamic learning rate strategy in the actual experiment. Therefore, choosing a suitable learning rate is especially critical. In this sub-section, we select ResNet18 and EfficientNet for evaluation because ResNet18 has a high recognition rate in early networks and EfficientNet is one of representative networks proposed recently. The experimental results are listed in Tables 6 and 7. From Tables 6 and 7, it can be seen that when the learning rate is 5×10^{-5}, ResNet18 and EfficientNet achieve the best recognition rate. Thus, in the remaining experiments, we set the initializing learning rate to 5×10^{-5}. It should be noted that all our experiments have an initial learning rate of 5×10^{-5}, and 100 iterations (EfficientNet used 200 iterations since it has slow convergence) are the learning rate decline steps, where learning rate decay rate is 0.1. That is, the learning rate drops by ten times every 100 iterations, and the total number of iterations is 500.

5.2.2 Recognition results of ResNet and VGG with different numbers of layers

Some CNNs may have different versions with different numbers of layers. For example, ResNet has different versions with 18, 34 and 50 layers. Using more layers may get better recognition rates, but may have the problem of overfitting. Thus, the number of layers is also an important factor for recognition. In this sub-section, we evaluate VGG and ResNet with different numbers of layers. Since most databases have difficulty in training on VGG when the learning rate is 5×10^{-5}, we set the learning rate of VGG to 10^{-5}. The recognition rates of VGG and ResNet under different numbers of layers are shown in Table 8. In this experiment, we verify the impact of network depth on the recognition performance.

The results in Table 8 indicate that:

1) For VGG, the recognition performance of VGG-16 is slightly better than that of VGG-19, and the recognition performances of VGG-16 and VGG-19 are close.

2) For ResNet, the recognition performance of Res-18 is better than those of Res-34 and Res-50. On those chal-

Table 6 Recognition rates of ResNet18 under different learning rates

Learning rate	5×10^{-3}	10^{-3}	5×10^{-4}	10^{-4}	5×10^{-5}	10^{-5}
PolyU II	66.16%	88.39%	88.64%	96.99%	**97.66%**	96.40%
PolyU M_B	82.20%	93.33%	96.97%	99.97%	**100%**	**100%**
HFUT	54.61%	78.45%	89.55%	97.67%	**98.51%**	98.42%
HFUT CS	42.96%	56.38%	69.79%	92.85%	**95.37%**	93.73%
TJU	57.67%	82.75%	88.18%	98.38%	**99.25%**	99.18%
PolyU 3D CST	75.38%	87.44%	93.26%	96.77%	**97.58%**	93.21%
PolyU 3D ST	81.47%	88.35%	94.17%	97.90%	**99.12%**	95.67%
PolyU 3D MCI	82.27%	86.65%	93.46%	98.29%	**99.35%**	97.66%
PolyU 3D GCI	73.72%	80.20%	83.54%	89.47%	**93.65%**	90.39%
PolyU M_N	79.20%	95.07%	93.57%	99.90%	**100%**	99.97%
TJUV	63.83%	82.95%	88.27%	97.85%	**98.58%**	97.88%

Table 7 Recognition rates of EfficientNet under different learning rates

Learning rate	5×10^{-3}	10^{-3}	5×10^{-4}	10^{-4}	5×10^{-5}	10^{-5}
PolyU II	75.05%	89.67%	94.09%	96.29%	**97.39%**	92.56%
PolyU M_B	84.50%	98.53%	99.47%	99.78%	**100%**	97.23%
HFUT I	71.79%	89.19%	93.70%	97.44%	**99.41%**	91.85%
HFUT CS	77.18%	89.88%	92.96%	94.46%	**96.55%**	87.93%
TJU	67.58%	93.57%	96.98%	97.58%	**99.89%**	97.08%
PolyU 3D CST	83.56%	93.44%	96.70%	97.22%	**97.81%**	97.17%
PolyU 3D ST	84.59%	91.28%	94.77%	98.40%	**99.37%**	98.87%
PolyU 3D MCI	85.90%	89.48%	93.33%	97.66%	**99.88%**	98.35%
PolyU 3D GCI	78.67%	81.58%	88.83%	92.79%	**95.66%**	94.33%
PolyU M_N	89.70%	96.97%	98.37%	98.50%	**100%**	97.83%
TJUV	78.32%	94.32%	96.43%	97.48%	**99.00%**	92.58%

Table 8 Recognition rates of VGG and ResNet under different numbers of layers

	VGG-16	VGG-19	Res-18	Res-34	Res-50
PolyU II	96.79%	**97.43%**	**97.66%**	96.25%	93.68%
PolyU M_B	**99.47%**	99.33%	**100%**	99.93%	99.53%
HFUT	96.04%	**96.25%**	**98.51%**	98.14%	93.79%
HFUT CS	**86.55%**	82.13%	**95.37%**	91.04%	85.21%
TJU-P	**93.92%**	91.28%	**99.25%**	98.67%	95.33%
PolyU 3D CST	**94.80%**	92.10%	**97.58%**	96.55%	95.70%
PolyU 3D ST	**95.90%**	94.33%	**99.12%**	98.75%	96.30%
PolyU 3D MCI	**94.40%**	94.53%	**99.35%**	99.15%	97.40%
PolyU 3D GCI	**90.18%**	90.80%	**93.65%**	93.30%	90.65%
PolyU M_N	**97.37%**	96.10%	**100%**	99.93%	99.57%
TJU-PV	**92.33%**	90.60%	**98.58%**	98.03%	95.63%

lenging databases such as PolyU II, HFUT, HFUT CS, TJU-P, and TJU-PV, the recognition performance of Res-18 is obviously better than that of Res-50.

3) In all databases, the recognition rate of ResNet-18 is obviously better than those of VGG-16 and VGG-19.

According to the results listed in Table 8, for VGG and ResNet, we only use VGG-16 and Res-18 for evaluation in the remaining experiments.

For different CNNs, the best number of layers to obtain the best recognition rate is determined by many factors, such as network structure, data size, data type, etc. Therefore, in practical applications, a lot of experiments need to be done to determine the optimal number of network layers for different CNNs.

5.2.3 Recognition results of EfficientNet from b0 to b7

EfficientNet gets the baseline network EfficientNet-b0 by grid search, and further optimizes different parameters to get EfficientNet-b1 to b7. The recognition results of EfficientNet from b0 to b7 are listed in Table 9. It can be seen that the recognition accuracy of EfficientNet is gradually increasing from b0 to b7, and EfficientNet-b7 achieves the best recognition accuracy. In the remaining experiments of this paper, for EfficientNet, we only use EfficientNet-b7 to conduct evaluation experiments. It should be noted that although EfficientNet-b7 performs well, it converges almost twice as slowly as other networks. In fact, EfficientNet-b6 is also slow, but the of speed EfficientNet-b0 to b5 is normal.

5.2.4 Recognition results of selected CNNs on all databases

In this sub-section, we conduct the experiments using all selected CNNs on all databases. The recognition results of selected CNNs on 2D palmprint and palm vein databases are listed in Table 10. The recognition results of selected CNNs on the 3D palmprint databases are listed in Table 11. Sometimes, when the learning rate is set to 5 × 10^{-5}, AlexNet and VGG-16 are untrainable. In this time, we adjust the learning rate of AlexNet and VGG-16 to 10^{-5}. In Table 10, AlexNet and VGG-16 have two recognition rates. The former is the result under the learning rate of 5 × 10^{-5}, and the latter is the result under the learning rate of 10^{-5}. If AlexNet and VGG-16 are untrainable, we mark the result as U.

Table 9 Recognition rates of EfficientNet from b0 to b7

	b0	b1	b2	b3	b4	b5	b6	b7
PolyU II	93.42%	93.47%	93.78%	95.38%	95.86%	96.35%	96.78%	**97.39%**
PolyU M_B	99.97%	100%	100%	100%	100%	100%	100%	**100%**
HFUT	97.98%	98.14%	98.32%	98.40%	98.75%	99.10%	99.18%	**99.41%**
HFUT CS	81.99%	84.85%	93.45%	93.60%	94.56%	95.79%	96.06%	**96.55%**
TJU-P	99.57%	99.58%	98.68%	99.63%	99.78%	99.83%	99.87%	**99.89%**
PolyU 3D CST	95.93%	96.11%	96.34%	96.88%	97.35%	97.55%	97.66%	**97.81%**
PolyU 3D ST	98.62%	98.77%	98.83%	98.90%	99.03%	99.15%	99.26%	**99.37%**
PolyU 3D MCI	99.55%	99.58%	99.63%	99.67%	99.75%	99.80%	99.82%	**99.88%**
PolyU 3D GCI	93.62%	93.77%	93.89%	94.29%	94.50%	94.89%	95.17%	**95.66%**
PolyU M_N	99.27%	99.33%	99.63%	99.64%	99.67%	100%	100%	**100%**
TJU-PV	95.88%	97.00%	96.97%	97.13%	97.85%	98.13%	98.77%	**99.00%**

Table 10 Recognition results of different CNNs on 2D palmprint and palm vein databases under the separate data mode

	PolyU II	PolyU M_B	HFUT	HFUT CS	TJU-P	PolyU M_N	TJU-PV
Alex	U/81.81%	92.63%/94.36%	78.33%/86.17%	42.53%/46.49%	80.35%/81.85%	87.07%/88.47%	76.08%/74.87%
VGG-16	U/96.79%	U/99.47%	U/96.04%	73.86%/86.55%	78.38%/93.92%	U/97.37%	84.22%/92.33%
IV3	94.66%	99.23%	97.74%	85.65%	98.08%	99.50%	97.02%
Res-18	**97.66%**	**100%**	98.51%	95.37%	99.25%	**100%**	98.58%
IV4	95.22%	99.03%	97.72%	84.78 %	96.78%	99.63%	97.27%
IRes2	95.07%	99.73%	96.45%	74.26%	98.50%	99.37%	97.00%
Dense-121	96.53%	**100%**	98.05%	94.47%	99.38%	**100%**	98.55%
Xec	94.94%	97.83%	94.45%	74.94%	94.20%	98.90%	94.79%
MbV2	96.99%	99.97%	98.08%	93.97%	99.27%	**100%**	97.77%
MbV3	97.35%	**100%**	98.67%	95.20%	99.37%	**100%**	98.67%
ShuffleV2	95.41%	99.16%	97.63%	92.35%	98.38%	99.76%	97.08%
SE-154	94.07%	98.10%	95.13%	85.15%	96.77%	98.20%	96.70%
ResX-101	93.98%	98.67%	94.56%	90.26%	98.23%	97.34%	97.17%
Efficient	97.39%	**100%**	**99.41%**	96.55%	**99.89%**	**100%**	**99.00%**
Ghost	94.74%	99.90%	97.01%	83.70%	98.90%	99.60%	96.20%
Reg-Y	79.32%	93.67%	78.91%	84.36%	79.50%	90.03%	87.32%
ResS-50	93.55%	99.10%	92.16%	**99.15%**	96.48%	98.57%	94.92%

Table 11 Recognition results of different CNNs on four 2D representations of 3D palmprint databases under the separate data mode

Recognition	CST	ST	MCI	GCI
Alex	88.18%	91.27%	88.45%	83.40%
VGG-16	U/94.80%	U/95.90%	U/94.40%	U/90.18%
IV3	96.05%	98.55%	99.20%	98.55%
Res-18	97.58%	99.12%	99.35%	93.65%
IV4	95.37%	97.50%	96.47%	93.27%
IRes2	97.12%	99.00%	98.70%	92.15%
Dense-121	97.60%	98.17%	98.88%	94.97%
Xec	95.47%	96.20%	97.10%	91.27%
MbV2	96.83%	98.32%	93.17%	93.50%
MbV3	97.53%	98.47%	97.32%	94.16%
ShuffleV2	95.63%	98.10%	98.55%	92.35%
SE-154	96.65%	98.02%	97.80%	90.58%
ResX-101	96.25%	98.32%	98.72%	92.40%
Efficient	**97.81%**	**99.37%**	**99.88%**	**99.66%**
Ghost	95.75%	98.20%	98.30%	91.97%
Reg-Y	95.62%	93.73%	95.30%	87.28%
ResS-50	95.08%	98.22%	97.90%	92.83%

From Tables 10 and 11, we have the following observations:

1) EfficientNet achieves the best recognition rate on most databases. The overall recognition result of ResNet is in the second place.

2) As a representative of lightweight networks, the overall recognition performance of MobileNet_v3 is worse than that of EfficientNet, close to ResNet, but better than other CNNs. This demonstrates that MobileNet_v3 is effective.

3) The recognition performance of recently proposed CNNs is obviously better than those of early CNNs. For example, the recognition rates of AlexNet and VGG are rather low. For those early CNNs such as AlexNet and VGG, their structures are relatively simple, and the number of layers is small. Thus, the recognition performance of them is not as good as those of the recently proposed CNNs such as EfficientNet.

4) HFUT CS is a very challenging database. The recognition performances of the most CNNs on HFUT CS database are unsatisfactory. In this database, ResNeSt (ResS-50) achieves the highest recognition rate, which is 99.15%.

5) Except ResNeSt has achieved good results on HFUT CS database, several recently proposed networks, including GhostNet, RegNet, ResNeSt, etc. have not achieved very good recognition results on various databases. Maybe the network structures of GhostNet, RegNet and ResNeSt are not very suitable for palmprint recognition and palmar vein recognition.

6) Among four 2D representations of the 3D palmprint, the recognition results obtained from MCI are the best.

7) For 3D palmprint recognition, based on MCI representation, EfficientNet achieved the recognition rate of 99.88%, which is a very promising result.

5.3 Recognition performance on mixed data mode

In the mixed mode, the first image captured in the second session is added to the training data. That is, the training set of each palm contains all images captured in the first session and the first image captured in the second session. Here, we use EfficientNet to conduct experiments. For each palm, the total number of training images are the number of images captured in the first session adding one (+1). This one means the first image captured in the second session. From Table 12, it can be seen that the recognition accuracy of EfficientNet gradually increases when the number of training samples increases.

We list the recognition rates of different CNNs on mixed data mode in Tables 13 and 14. It can be seen that the recognition accuracies of all CNNs increased significantly, particularly, for 2D palmprint recognition, EfficientNet achieves 100% recognition accuracies on PolyU II, PolyU M_B, HFUT, TJU-P, PolyU M_N and TJU-PV. For 3D palmprint recognition, Res-18 achieves the best recognition results, and all CNNs achieve the best recognition results from MCI representation among four 2D representations.

This experiment proves once again that the sufficiency of data is very important to improve the recognition accuracy of deep learning. In the future, with the wide application of palmprint recognition and palmar vein recognition, the data volume of palmprint and palmar vein will increase continuously. In this way, the recognition accuracy of palmprint recognition and palm vein recognition technology based on deep learning will reach a new level.

5.4 Performance comparison with other methods

For 2D palmprint and palm vein recognition, we compare the performance of CNNs and other methods including some traditional methods and one deep learning method PalmNet[83]. Four traditional palmprint recognition methods, including competition code, sequence number, RLOC and LLDP, are selected for comparison. For CNNs, we only list the results of MobileNet_v3 and EfficientNet which have excellent performance. The performance comparison is conducted on both separate data mode and mixed data mode.

On the separate data mode, for traditional methods, four images collected in the first session are used as the training data, and all images collected in the second session are exploited as the test data. For MobileNet_v3 and EfficientNet, all images collected in the first session are used as the training data and the second session images are used as the test data (In the HFUT CS database, all images captured by the camera are used as the training data). The comparison results on separate data mode are shown in Table 15.

From Table 15, it can be seen that, on separate data mode, the performances of the traditional methods are better than those of the CNNs. As we know, traditional methods use fewer training samples. Because the features of 2D and 3D palmprint and palm vein are relatively stable, thus, hand-crafted features can well represent the palmprint, resulting in a better recognition performance of traditional methods. In addition, the classic CNNs used in this paper are designed for general image classification tasks, and are not specially designed for 2D and 3D palmprint recognition and palm vein recognition, so the accuracies of them are not satisfactory.

On the mixed data mode, for traditional methods, four images collected in the first session are used as the training data, and we add the first image captured in the second session to the training set. The remaining images collected in the second session are exploited as the test data. For MobileNet_v3 and EfficientNet, all images collected in the first session are used as the training data,

Table 12 Recognition rates on different mixed-training data amounts of EfficientNet

The number of training images	1+1	2+1	3+1	4+1	5+1	6+1	7+1	8+1	9+1	10+1
PolyU II	98.27%	99.10%	99.67%	99.95%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
PolyU M_B	99.85%	**100%**	**100%**	**100%**	**100%**	**100%**	N/A	N/A	N/A	N/A
HFUT I	98.08%	98.48%	99.47%	99.93%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
HFUT CS	83.26%	87.39%	88.72%	90.61%	92.14%	94.28%	95.06%	96.79%	97.33%	**99.57%**
TJU	96.04%	98.81%	99.75%	99.98%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
PolyU 3D CST	92.38%	92.68%	93.04%	93.62%	94.50%	95.04%	95.97%	96.48%	97.50%	**98.54%**
PolyU 3D ST	92.17%	92.90%	93.57%	94.12%	94.99%	95.77%	96.36%	97.58%	98.79%	**99.88%**
PolyU 3D MCI	91.89%	93.33%	94.17%	95.28%	96.00%	96.85%	97.88%	98.49%	99.29%	**99.94%**
PolyU 3D GCI	90.44%	91.34%	92.13%	93.37%	94.55%	95.87%	96.49%	96.90%	97.22%	**97.43%**
PolyU M_N	99.00%	99.76%	**100%**	**100%**	**100%**	**100%**	N/A	N/A	N/A	N/A
TJUV	93.44%	94.29%	98.25%	99.66%	99.97%	**100%**	**100%**	**100%**	**100%**	**100%**

Table 13 Recognition results of different CNNs on 2D palmprint and palm vein databases under the mixed data mode

	PolyU II	PolyU M_B	HFUT	HFUT CS	TJU-P	PolyU M_N	TJU-PV
Alex	97.58%	98.92%	–/98.33%	94.08%	94.67%	99.20%	94.57%
VGG-16	U/99.83%	**U/100%**	U/99.62%	96.55%	U/99.23%	**U/100%**	U/98.76%
IV3	98.49%	**100%**	99.85%	99.08%	99.67%	99.95%	98.85%
Res-18	**100%**	**100%**	**100%**	99.84%	**100%**	**100%**	99.92%
IV4	98.56%	**100%**	99.87%	99.12%	99.84%	99.98%	99.17%
IRes2	99.00%	100%	99.89%	98.77%	99.79%	100%	99.07%
Dense-121	99.86%	100%	100%	99.58%	99.88%	100%	99.74%
Xec	97.78%	99.88%	99.46%	97.97%	99.15%	99.92%	98.46%
MbV2	99.40%	**100%**	99.94%	99.36%	**100%**	**100%**	99.96%
MbV3	99.69%	**100%**	**100%**	99.49%	**100%**	**100%**	**100%**
ShuffleV2	99.96%	**100%**	**100%**	99.09%	**100%**	**100%**	**100%**
SE-154	99.46%	**100%**	99.64%	98.85%	99.73%	**100%**	99.46%
ResX-101	98.58%	99.93%	**100%**	97.82%	99.08%	99.87%	99.35%
Efficient	**100%**	**100%**	**100%**	99.57%	**100%**	**100%**	**100%**
Ghost	95.33%	99.98%	98.77%	86.31%	99.59%	99.87%	97.32%
Reg-Y	81.44%	94.88%	80.37%	87.15%	81.60%	91.48%	88.38%
ResS-50	94.56%	99.87%	94.10%	**99.76%**	97.39%	99.48%	96.01%

Table 14 Recognition results of different CNNs on four 2D representations of 3D palmprint databases under the mixed data mode

Recognition	CST	ST	MCI	GCI
Alex	95.72%	98.25%	96.42%	93.86%
VGG-16	U/95.39%	U/97.50%	U/97.72%	U/94.53%
IV3	98.61%	99.83%	**100%**	98.06%
Res-18	**99.50%**	**99.94%**	**100%**	**98.56%**
IV4	98.50%	99.11%	97.86%	95.22%
IRes2	99.28%	98.86%	99.89%	95.08%
Dense-121	99.03%	99.83%	99.92%	97.44%
Xec	98.28%	99.39%	99.14%	94.19%
MbV2	98.89%	99.89%	99.83%	97.22%
MbV3	98.64%	99.67%	99.67%	97.19%
ShuffleV2	99.06%	99.47%	99.92%	96.57%
SE-154	98.47%	99.36%	99.31%	95.42%
ResX-101	96.81%	99.07%	99.03%	93.03%
Efficient	98.54%	99.88%	99.94%	97.43%
Ghost	96.33%	99.10%	99.25%	93.08%
Reg-Y	96.75%	94.55%	96.78%	89.32%
ResS-50	96.74%	99.27%	98.21%	94.15%

and we add the first image captured in the second session to the training set. The remaining images collected in second session are exploited as the test data. The comparison results on mixed data mode are shown in Table 16.

From Table 16, it can be seen that, on mixed data mode, the performances of CNNs are nearly equal to that of the traditional methods. The scale of the 2D and 3D palmprint and palm vein databases is small. But deep learning methods rely heavily on learning from large-scale database. If there are sufficient training samples, deep learning methods can achieve better performance.

For 3D palmprint recognition, we compare the performances between CNNs and other traditional methods on the separate data mode. Table 17 lists the comparison results. It can be seen that the recognition accuracy of CNN is slightly better than traditional methods.

6 Conclusions

This paper systematically investigated the recognition performance of classic CNNs for 2D and 3D palmprint recognition and palm vein recognition. Seventeen representative and classic CNNs were exploited for performance evaluation including AlexNet, VGG, Inception_v3, Inception_v4, ResNet, ResNeXt, Inception_ResNet_v2, DenseNet, Xception, MobileNet_v2, MobliNet_v3, ShunffleNet_v2, SENet, EfficientNet, GhostNet, RegNet and ResNeSt. Five 2D palmprint image databases, one 3D palmprint database and two palm vein databases were exploited for performance evaluation, including PolyU II, PolyU M_B, HFUT, HFUT CS, TJU-P, PolyU 3D, PolyU M_B and TJU-PV. These databases are very representative. For example, PolyU II, PolyU M_B, PolyU M_N and HFUT databases were collected by the contact manner; HFUT CS, TJU-P, and TJU-PV were captured by the contactless manner. All databases were collected in two different sessions. In particular, HFUT CS is a rather challenging database because it was collected in

Table 15 2D palmprint and palm vein recognition: Performance comparison between classic CNNs and other methods under the separate data mode

	Competitive code	Ordinal code	RLOC	LLDP	PalmNet	MbV3	EfficientNet
PolyU II	**100%**	**100%**	**100%**	**100%**	**100%**	97.35%	97.39%
PolyU M_B	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
HFUT	99.64%	99.60%	**99.75%**	99.89%	**100%**	98.67%	99.41%
HFUT CS	99.45%	**99.67%**	99.36%	99.40%	92.45%	95.20%	96.55%
TJU-P	99.87%	**99.95%**	99.63%	99.50%	**100%**	99.37%	99.89%
PolyU M_N	99.97%	**100%**	**100%**	**100%**	99.02%	**100%**	**100%**
TJU-PV	99.32%	99.55%	**100%**	98.93%	99.61%	98.67%	99.00%

Table 16 2D palmprint and palm vein recognition: Performance comparison between classic CNNs and other methods under the mixed data mode

	Competitive code	Ordinal code	RLOC	LLDP	PalmNet	MbV3	EfficientNet
PolyU II	**100%**	**100%**	**100%**	**100%**	**100%**	99.40%	**100%**
PolyU M_B	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
HFUT I	99.98%	99.98%	**100%**	99.93%	**100%**	**100%**	**100%**
HFUT CS	99.96%	**100%**	**100%**	**100%**	**100%**	99.49%	99.57%
TJU	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
PolyU M_N	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
TJUV	99.87%	99.87%	**100%**	99.96%	99.91%	**100%**	**100%**

Table 17 3D palmprint recognition: Performance comparison between classic CNNs and other methods under the separate data mode

Reference	2D representation	Recognition method	Recognition rate
[16]	MCI	Competitive code	99.24%
[5]	ST	Block-wise features and collaborative representation	99.15%
[38]	MCICST	Binary representations of orientation and compact ST	99.67%
This paper	MCI	EfficientNet	99.88%

the conditions of two different sessions, contactless manner and crossing three different sensors. We conducted a lot of experiments on the above databases in the conditions of different network structures, different learning rates, different numbers of network layers. We conducted the experiments on both separate data mode and mixed data mode. And we also compared the recognition performances between the CNNs and traditional methods. According to the experimental results, we have the following observations. 1) The performances of recently proposed CNNs such as EfficientNet and MobileNet_v3 are obviously better than those of other early CNNs. Particularly, EfficientNet achieves the best recognition accuracy. 2) Learning rate is an important hyperparameter. It has an important influence on the recognition performance of CNNs. For palmprint and palm vein recognition, 5×10^{-5} is an appropriate learning rate. 3) Using more layers, VGG and ResNet did not get better recognition results. Compared with ILSVRC, the scale of palmprint and palm vein databases is small, and the model with more layers may lead to the problem of over-fitting. 4) For 3D palmprint recognition, deep learning-based methods obtained promising results. Among four 2D representations of 3D palmprints, MCI can help deep learning methods to achieve the best recognition results. 5) In separate data mode, the recognition performance of classic CNNs is not satisfactory, and is worse than those of some traditional methods on those challenging databases. On mixed data mode, CNNs can achieve good recognition accuracy. For example, CNNs achieved 100% recognition accuracies on most databases.

In this work, a lot of classic CNNS have been evaluated. However, these CNNs have been designed manually by human experts. In recent two years, NAS technology has attracted more and more attention. The core idea of NAS is to use search algorithms to find better neural network structure, so that can obtain better recognition performance. In our future work, we will try to exploit NAS technology for 2D and 3D palmprint and palm vein recognition. In our future work, we will also design special CNNs according to the characteristics of 2D and palmprint recognition and palm vein recognition. In this way, better recognition performance of deep learning for 2D and 3D palmprint recognition and palm vein recognition can be expected.

Acknowledgements

This work was supported by National Science Foundation of China (Nos. 61673157, 62076086, 61972129 and 61702154), and Key Research and Development Program in Anhui Province (Nos. 202004d07020008 and 2019 04d07020010).

Open Access

References

[1] S. G. Tong, Y. Y. Huang, Z. M. Tong. A robust face recognition method combining LBP with multi-mirror symmetry for images with various face interferences. *International Journal of Automation and Computing*, vol. 16, no. 5, pp. 671–682, 2019. DOI: 10.1007/s11633-018-1153-8.

[2] D. Zhang, W. M. Zuo, F. Yue. A comparative study of palmprint recognition algorithms. *ACM Computing Surveys*, vol. 44, no. 1, Article number 2, 2012. DOI: 10.1145/2071389.2071391.

[3] L. K. Fei, G. M. Lu, W. Jia, S. H. Teng, D. Zhang. Feature extraction methods for palmprint recognition: A survey and evaluation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 346–363, 2019. DOI: 10.1109/TSMC.2018.2795609.

[4] D. X. Zhong, X. F. Du, K. C. Zhong. Decade progress of palmprint recognition: A brief survey. *Neurocomputing*, vol. 328, pp. 16–28, 2019. DOI: 10.1016/j.neucom.2018.03.081.

[5] L. Zhang, Y. Shen, H. Y. Li, J. W. Lu. 3D palmprint identification using block-wise features and collaborative representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1730–1736, 2015. DOI: 10.1109/TPAMI.2014.2372764.

[6] W. X. Kang, Q. X. Wu. Contactless palm vein recognition using a mutual foreground-based local binary pattern. *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1974–1985, 2014. DOI: 10.1109/TIFS.2014.2361020.

[7] B. Hu, J. C. Wang. Deep learning based hand gesture recognition and UAV flight controls. *International Journal of Automation and Computing*, vol. 17, no. 1, pp. 17–29, 2020. DOI: 10.1007/s11633-019-1194-7.

[8] V. K. Ha, J. C. Ren, X. Y. Xu, S. Zhao, G. Xie, V. Masero, A. Hussain. Deep learning based single image super-resolution: A survey. *International Journal of Automation and Computing*, vol. 16, no. 4, pp. 413–426, 2019. DOI: 10.1007/s11633-019-1183-x.

[9] K. Sundararajan, D. L. Woodard. Deep learning for biometrics: A survey. *ACM Computing Surveys*, vol. 51, no. 3, Article number 65, 2018. DOI: 10.1145/3190618.

[10] L. K. Fei, B. Zhang, W. Jia, J. Wen, D. Zhang. Feature extraction for 3-D palmprint recognition: A survey. *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 3, pp. 645–656, 2020. DOI: 10.1109/TIM.2020.2964076.

[11] D. Zhang, W. K. Kong, J. You, M. Wong. Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1041–1050, 2003. DOI: 10.1109/TPAMI.2003.1227981.

[12] D. Zhang, Z. H. Guo, G. M. Lu, L. Zhang, W. M. Zuo. An online system of multispectral palmprint verification. *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 2, pp. 480–490, 2010. DOI: 10.1109/TIM.2009.2028772.

[13] W. Jia, B. Zhang, J. T. Lu, Y. H. Zhu, Y. Zhao, W. M. Zuo, H. B. Ling. Palmprint recognition based on complete direction representation. *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4483–4498, 2017. DOI: 10.1109/TIP.2017.2705424.

[14] W. Jia, R. X. Hu, J. Gui, Y. Zhao, X. M. Ren. Palmprint recognition across different devices. *Sensors*, vol. 12, no. 6, pp. 7938–7964, 2012. DOI: 10.3390/s120607938.

[15] L. Zhang, L. D. Li, A. Q. Yang, Y. Shen, M. Yang. Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach. *Pattern Recognition*, vol. 69, pp. 199–212, 2017. DOI: 10.1016/j.patcog.2017.04.016.

[16] W. Li, D. Zhang, L. Zhang, G. M. Lu, J. Q. Yan. 3-D palmprint recognition with joint line and orientation features. *IEEE Transactions on Systems, Man, and Cybernetics– Part C: Applications and Reviews*, vol. 41, no. 2, pp. 274–279, 2011. DOI: 10.1109/TSMCC.2010.2055849.

[17] L. Zhang, Z. X. Cheng, Y. Shen, D. Q. Wang. Palmprint and palmvein recognition based on DCNN and a new large-scale contactless palmvein dataset. *Symmetry*, vol. 10, no. 4, Article number 78, 2018. DOI: 10.3390/sym10040078.

[18] D. S. Huang, W. Jia, D. Zhang. Palmprint verification based on principal lines. *Pattern Recognition*, vol. 41, no. 4, pp. 1316–1328, 2008. DOI: 10.1016/j.patcog.2007.08.016.

[19] D. Palma, P. L. Montessoro, G. Giordano, F. Blanchini. Biometric palmprint verification: A dynamical system approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2676–2687, 2019. DOI: 10.1109/TSMC.2017.2771232.

[20] W. Nie, B. Zhang, S. P. Zhao. Discriminative local feature for hyperspectral hand biometrics by adjusting image acutance. *Applied Sciences*, vol. 9, no. 19, Article number 4178, 2019. DOI: 10.3390/app9194178.

[21] W. Jia, R. X. Hu, Y. K. Lei, Y. Zhao, J. Gui. Histogram of oriented lines for palmprint recognition. *IEEE Transac-*

tions on Systems, *Man*, *and Cybernetics*: *Systems*, vol. 44, no. 3, pp. 385–395, 2014. DOI: 10.1109/TSMC.2013.2258010.

[22] Y. T. Luo, L. Y. Zhao, B. Zhang, W. Jia, F. Xue, J. T. Lu, Y. H. Zhu, B. Q. Xu. Local line directional pattern for palmprint recognition. *Pattern Recognition*, vol. 50, pp. 26–44, 2016. DOI: 10.1016/j.patcog.2015.08.025.

[23] G. Li, J. Kim. Palmprint recognition with Local Microstructure Tetra Pattern. *Pattern Recognition*, vol. 61, pp. 29–46, 2017. DOI: 10.1016/j.patcog.2016.06.025.

[24] L. K. Fei, B. Zhang, Y. Xu, D. Huang, W. Jia, J. Wen. Local discriminant direction binary pattern for palmprint representation and recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 468–481, 2020. DOI: 10.1109/TCSVT.2019.2890835.

[25] L. K. Fei, B. Zhang, Y. Xu, Z. H. Guo, J. Wen, W. Jia. Learning discriminant direction binary palmprint descriptor. *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3808–3820, 2019. DOI: 10.1109/TIP.2019.2903307.

[26] L. K. Fei, B. Zhang, W. Zhang, S. H. Teng. Local apparent and latent direction extraction for palmprint recognition. *Information Sciences*, vol. 473, pp. 59–72, 2019. DOI: 10.1016/j.ins.2018.09.032.

[27] X. Q. Wu, Q. S. Zhao. Deformed palmprint matching based on stable regions. *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4978–4989, 2015. DOI: 10.1109/TIP.2015.2478386.

[28] Z. A. Sun, T. N. Tan, Y. H. Wang, S. Z. Li. Ordinal palmprint represention for personal identification. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Diego, USA, pp. 279–284, 2005. DOI: 10.1109/CVPR.2005.267.

[29] W. Jia, D. S. Huang, D. Zhang. Palmprint verification based on robust line orientation code. *Pattern Recognition*, vol. 41, no. 5, pp. 1504–1513, 2008. DOI: 10.1016/j.patcog.2007.10.011.

[30] Z. H. Guo, D. Zhang, L. Zhang, W. M. Zuo. Palmprint verification using binary orientation co-occurrence vector. *Pattern Recognition Letters*, vol. 30, no. 13, pp. 1219–1227, 2009. DOI: 10.1016/j.patrec.2009.05.010.

[31] L. K. Fei, Y. Xu, W. L. Tang, D. Zhang. Double-orientation code and nonlinear matching scheme for palmprint recognition. *Pattern Recognition*, vol. 49, pp. 89–101, 2016. DOI: 10.1016/j.patcog.2015.08.001.

[32] G. M. Lu, D. Zhang, K. Q. Wang. Palmprint recognition using eigenpalms features. *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1463–1467, 2003. DOI: 10.1016/S0167-8655(02)00386-0.

[33] X. Q. Wu, D. Zhang, K. Q. Wang. Fisherpalms based palmprint recognition. *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2829–2838, 2003. DOI: 10.1016/S0167-8655(03)00141-7.

[34] J. Yang, A. F. Frangi, J. Y. Yang, D. Zhang, Z. Jin. KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 230–244, 2005. DOI: 10.1109/TPAMI.2005.33.

[35] D. Zhang, G. M. Lu, W. Li, L. Zhang, N. Luo. Palmprint recognition using 3-D information. *IEEE Transactions on Systems Man, and Cybernetics – Part C*: *Applications and Reviews*, vol. 39, no. 5, pp. 505–519, 2009. DOI: 10.1109/TSMCC.2009.2020790.

[36] B. Yang, X. H. Wang, J. L. Yao, X. Yang, W. H. Zhu. Efficient local representations for three-dimensional palmprint recognition. *Journal of Electronic Imaging*, vol. 22, no. 4, Article number 043040, 2013. DOI: 10.1117/1.JEI.22.4.043040.

[37] L. K. Fei, S. H. Teng, J. G. Wu, Y. Xu, J. Wen, C. W. Tian. Combining enhanced competitive code with compacted ST for 3D palmprint recognition. In *Proceedings of the 4th IAPR Asian Conference on Pattern Recognition*, IEEE, Nanjing, China, pp. 483–487, 2017. DOI: 10.1109/ACPR.2017.62.

[38] L. K. Fei, G. M. Lu, W. Jia, J. Wen, D. Zhang. Complete binary representation for 3-D palmprint recognition. *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 12, pp. 2761–2771, 2018. DOI: 10.1109/TIM.2018.2830858.

[39] L. Fei, B. Zhang, Y. Xu, W. Jia, J. Wen, J. G. Wu. Precision direction and compact surface type representation for 3D palmprint identification. *Pattern Recognition*, vol. 87, pp. 237–247, 2019. DOI: 10.1016/j.patcog.2018.10.018.

[40] Y. B. Zhang, Q. Li, J. You, P. Bhattacharya. Palm vein extraction and matching for personal authentication. In *Proceedings of the 9th International Conference on Advances in Visual Information Systems*, Springer, Shanghai, China, vol. 4781, pp. 154–164, 2007. DOI: .

[41] L. Mirmohamadsadeghi, A. Drygajlo. Palm vein recognition with local texture patterns. *IET Biometrics*, vol. 3, no. 4, pp. 198–206, 2014. DOI: 10.1049/iet-bmt.2013.0041.

[42] ManMohan, J. Saxena, K. Teckchandani, P. Pandey, M. K. Dutta, C. M. Travieso, J. B. Alonso-Hernández. Palm vein recognition using local tetra patterns. In *Proceedings of the 4th International Work Conference on Bioinspired Intelligence*, IEEE, San Sebastian, Spain, pp. 151–156, 2015. DOI: 10.1109/IWOBI.2015.7160159.

[43] W. X. Kang, Y. Liu, Q. X. Wu, X. S. Yue. Contact-free palm-vein recognition based on local invariant features. *PLoS ONE*, vol. 9, no. 5, Article number e97548, 2014. DOI: 10.1371/journal.pone.0097548.

[44] Y. B. Zhou, A. Kumar. Human identification using palm-vein images. *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1259–1274, 2011. DOI: 10.1109/TIFS.2011.2158423.

[45] S. Elnasir, S. M. Shamsuddin. Proposed scheme for palm vein recognition based on linear discrimination analysis and nearest neighbour classifier. In *Proceedings of International Symposium on Biometrics and Security Technologies*, IEEE, Kuala Lumpur, Malaysia, pp. 67–72, 2014. DOI: 10.1109/ISBAST.2014.7013096.

[46] J. X. Xu. An online biometric identification system based on two dimensional Fisher linear discriminant. In *Proceedings of the 8th International Congress on Image and Signal*, IEEE, Shenyang, China, pp. 894–898, 2015. DOI: 10.1109/CISP.2015.7408004.

[47] Y. P. Lee. Palm vein recognition based on a modified $(2D)^2$ LDA. *Signal, Image and Video Processing*, vol. 9, no. 1, pp. 229–242, 2015. DOI: 10.1007/s11760-013-0425-6.

[48] B. H. Shekar, N. Harivinod. Multispectral palmprint matching based on joint sparse representation. In *Proceedings of the 4th National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, IEEE, Jodhpur, India, 2013. DOI: 10.1109/NCVPRIPG.

2013.6776243.

[49] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998. DOI: 10.1109/5.726791.

[50] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, ACM, Red Hook, USA, pp. 1097–1105, 2012.

[51] M. D. Zeiler, R. Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the 13th European Conference on Computer Vision*, Springer, Zurich, Switzerland, vol. 8689, pp. 818–833, 2014. DOI: 10.1007/978-3-319-10590-1_53.

[52] M. Lin, Q. Chen, S. Yan. Network in network. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada, 2014.

[53] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA 2015.

[54] C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Boston, USA, pp. 1–9, 2015. DOI: 10.1109/CVPR.2015.7298594.

[55] S. Ioffe, C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, pp. 448–456, 2015.

[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 2818–2826, 2016. DOI: 10.1109/CVPR.2016.308.

[57] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, San Francisco, USA, pp. 4278–4284, 2017.

[58] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[59] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Honolulu, USA, pp. 2261–2269, 2017. DOI: 10.1109/CVPR.2017.243.

[60] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. [Online], Available: https://arxiv.org/abs/1602.07360, 2017.

[61] A. G. Howard, M. L. Zhu, B. Chen, D. Kalenichenko, W. J. Wang, T. Weyand, M. Andreetto, H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. [Online], Available: https://arxiv.org/abs/1704.04861, 2017.

[62] M. Sandler, A. Howard, M. L. Zhu, A. Zhmoginov, L. C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 4510–4520, 2018. DOI: 10.1109/CVPR.2018.00474.

[63] A. Howard, M. Sandler, B. Chen. W. J. Wang, L. C. Chen, M. X. Tan, G. Chu, V. Vasudevan, Y. K. Zhu, R. M. Pang, H, Adam, Q. Le. Searching for mobileNetV3. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, South Korea, pp. 1314–1324, 2019. DOI: 10.1109/ICCV.2019.00140.

[64] X. Y. Zhang, X. Y. Zhou, M. X. Lin, J. Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 6848–6856, 2018. DOI: 10.1109/CVPR.2018.00716.

[65] N. N. Ma, X. Y. Zhang, H. T. Zheng, J. Sun. Shufflenet V2: Practical guidelines for efficient CNN architecture design. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, vol. 11218, pp. 122–138, 2018. DOI: 10.1007/978-3-030-01264-9_8.

[66] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Honolulu, USA, pp. 1800–1807, 2017. DOI: 10.1109/CVPR.2017.195.

[67] A. Gholami, K. Kwon, B. Wu, Z. Z. Tai, X. Y. Yue, P. Jin, S. C. Zhao, K. Keutzer. SqueezeNext: Hardware-aware neural network design. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, Salt Lake City, USA, pp. 1719–1728, 2018. DOI: 10.1109/CVPRW.2018.00215.

[68] S. N. Xie, R. Girshick, P. Dollár, Z. W. Tu, K. M. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Honolulu, USA, pp. 5987–5995, 2017. DOI: 10.1109/CVPR.2017.634.

[69] J. Hu, L. Shen, G. Sun. Squeeze-and-excitation networks. In *Proceedings of IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 7132–7141, 2018. DOI: 10.1109/CVPR.2018.00745.

[70] M. X. Tan, Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp. 10691–10700, 2019.

[71] K. Han, Y. H. Wang, Q. Tian, J. Y. Guo, C. J. Xu, C. Xu. GhostNet: More features from cheap operations. [Online], Available: https://arxiv.org/abs/1911.11907, 2019.

[72] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. M. He, P. Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 10428–10436, 2020. DOI: 10.1109/CVPR42600.2020.01044.

[73] H. Zhang, C. R. Wu, Z. Y. Zhang, Y. Zhu, Z. Zhang, H. B. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, A. Smola. ResNeSt: Split-attention networks. [Online], Available: https://arxiv.org/abs/2004.08955, 2020.

[74] A. Jalali, R. Mallipeddi, M. Lee. Deformation invariant and contactless palmprint recognition using convolutional neural network. In *Proceedings of the 3rd International*

Conference on Human-agent Interaction, ACM, Daegu, Korea, pp. 209–212, 2015. DOI: 10.1145/2814940.2814977.

[75] D. D. Zhao, X. Pan, X. L. Luo, X. J. Gao. Palmprint recognition based on deep learning. In *Proceedings of the 6th International Conference on Wireless, Mobile and Multi-Media*, IEEE, Beijing, China, pp. 214–216, 2015. DOI: 10.1049/cp.2015.0942.

[76] S. Minaee, Y. Wang. Palmprint recognition using deep scattering convolutional network.[Online], Available: https://arxiv.org/abs/1603.09027, 2016.

[77] D. Liu, D. M. Sun. Contactless palmprint recognition based on convolutional neural network. In *Proceedings of the 13th IEEE International Conference on Signal Processing*, IEEE, Chengdu, China, pp. 1363–1367, 2017. DOI: 10.1109/ICSP.2016.7878049.

[78] J. Svoboda, J. Masci, M. M. Bronstein. Palmprint recognition via discriminative index learning. In *Proceedings of the 23rd International Conference on Pattern Recognition*, IEEE, Cancun, Mexico, pp. 4232–4237, 2016. DOI: 10.1109/ICPR.2016.7900298.

[79] A. Q. Yang, J. X. Zhang, Q. L. Sun, Q. Zhang. Palmprint recognition based on CNN and local coding features. In *Proceedings of the 6th International Conference on Computer Science and Network Technology*, IEEE, Dalian, China, pp. 482–487, 2018. DOI: 10.1109/ICCSNT.2017.8343744.

[80] A. Meraoumia, F. Kadri, H. Bendjenna, S. Chitroub, A. Bouridane. Improving biometric identification performance using pcanet deep learning and multispectral palmprint. *Biometric Security and Privacy: Opportunities & Challenges in the Big Data Era*, R. Jiang, S. Almaadeed, A. Bouridane, P. D. Crookes, A. Beghdadi Eds., Cham, Switzerland: Springer, pp. 51–69, 2017. DOI: 10.1007/978-3-319-47301-7_3.

[81] D. Zhong, Y. Yang, X. Du. Palmprint recognition using siamese network. In *Proceedings of the 13th Chinese Conference on Biometric Recognition*, Springer, Urumqi, China, vol. 10996, pp. 48–55, 2018. DOI: 10.1007/978-3-319-97909-0_6.

[82] A. Michele, V. Colin, D. D. Santika. MobileNet convolutional neural networks and support vector machines for palmprint recognition. *Procedia Computer Science*, vol. 157, pp. 110–117, 2019. DOI: 10.1016/j.procs.2019.08.147.

[83] A. Genovese, V. Piuri, K. N. Plataniotis, F. Scotti. PalmNet: Gabor-PCA convolutional networks for touchless palmprint recognition. *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3160–3174, 2019. DOI: 10.1109/TIFS.2019.2911165.

[84] D. X. Zhong, J. S. Zhu. Centralized large margin cosine loss for open-set deep palmprint recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1559–1568, 2020. DOI: 10.1109/TCSVT.2019.2904283.

[85] W. M. Matkowski, T. T. Chai, A. W. K. Kong. Palmprint recognition in uncontrolled and uncooperative environment. *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1601–1615, 2020. DOI: 10.1109/TIFS.2019.2945183.

[86] S. P. Zhao, B. Zhang. Deep discriminative representation for generic palmprint recognition. *Pattern Recognition*, vol. 98, Article number 107071, 2020. DOI: 10.1016/j.patcog.2019.107071.

[87] S. P. Zhao, B. Zhang. Joint constrained least-square regression with deep convolutional feature for palmprint recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2020. DOI: 10.1109/TSMC.2020.3003021.

[88] S. P. Zhao, B. Zhang, C. L. P. Chen. Joint deep convolutional feature representation for hyperspectral palmprint recognition. *Information Sciences*, vol. 489, pp. 167–181, 2019. DOI: 10.1016/j.ins.2019.03.027.

[89] D. Samai, K. Bensid, A. Meraoumia, A. Taleb-Ahmed, M. Bedda. 2D and 3D palmprint recognition using deep learning method. In *Proceedings of the 3rd International Conference on Pattern Analysis and Intelligent Systems*, IEEE, Tebessa, Algeria, 2018. DOI: 10.1109/PAIS.2018.8598522.

[90] M. Chaa, Z. Akhtar, A. Attia. 3D palmprint recognition using unsupervised convolutional deep learning network and SVM classifier. *IET Image Processing*, vol. 13, no. 5, pp. 736–745, 2019. DOI: 10.1049/iet-ipr.2018.5642.

[91] N. F. Hassan, H. I. Abdulrazzaq. Pose invariant palm vein identification system using convolutional neural network. *Baghdad Science Journal*, vol. 15, no. 4, pp. 502–509, 2018. DOI: 10.21123/bsj.15.4.502-509.

[92] S. Lefkovits, L. Lefkovits, L. Szilágyi. Applications of different CNN architectures for palm vein identification. In *Proceedings of the 16th International Conference on Modeling Decisions for Artificial Intelligence*, Springer, Milan, Italy, vol. 11676, pp. 295–306, 2019. DOI: 10.1007/978-3-030-26773-5_26.

[93] D. Thapar, G. Jaswal, A. Nigam, V. Kanhangad. PVSNet: Palm vein authentication siamese network trained using triplet loss and adaptive hard mining by learning enforced domain specific features. In *Proceedings of the 5th International Conference on Identity, Security, and Behavior Analysis*, IEEE, Hyderabad, India, 2019. DOI: 10.1109/ISBA.2019.8778623.

[94] S. Chantaf, A. Hilal, R. Elsaleh. Palm vein biometric authentication using convolutional neural networks. In *Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications*,Springer,vol. 146,pp. 352–363,2020.DOI:10.1007/978-3-030-21005-2_34.

Wei Jia received the B.Sc. degree in informatics from Central China Normal University, China in 1998, the M.Sc. degree in computer science from Hefei University of Technology, China in 2004, and the Ph.D. degree in pattern recognition and intelligence system from University of Science and Technology of China, China in 2008. He has been a research associate professor in Hefei Institutes of Physical Science, Chinese Academy of Sciences, China from 2008 to 2016. He is currently an associate professor in Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, and in School of Computer Science and Information Engineering, Hefei University of Technology, China.

His research interests include computer vision, biometrics, pattern recognition, image processing and machine learning.

E-mail: jiawei@hfut.edu.cn (Corresponding author)

ORCID iD: 0000-0001-5628-6237

Jian Gao received the B. Sc. degree in mechanical design and manufacturing and automation from Hefei University of Technology, China in 2018. Now, he is currently a master student in School of Computer Science and Information Engineering, Hefei University of Technology, China.

His research interests include computer vision, biometrics recognition and deep learning.

E-mail: 787117010@qq.com

Wei Xia received the B. Sc. degree in computer science from Anhui University of Science and Technology, China in 2018. He is a master student in School of Computer Science and Information Engineering, Hefei University of Technology, China.

His research interests include biometrics, pattern recognition and image processing.

E-mail: hewelxw@mail.hfut.edu.cn

Yang Zhao received the B. Eng. and Ph.D. degrees in pattern recognition and intelligence from Department of Automation, University of Science and Technology of China, China in 2008 and 2013. From 2013 to 2015, he was a postdoctoral researcher at School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, China. Currently, he is an associate professor at School of Computer Science and Information Engineering, Hefei University of Technology, China.

His research interests include image processing and computer vision.

E-mail: yzhao@hfut.edu.cn

Hai Min received the Ph. D. degree in pattern recognition and intelligence system from University of Science and Technology of China, China in 2014. He is currently an associate professor in School of Computer Science and Information Engineering, Hefei University of Technology, China.

His research interests include pattern recognition and image segmentation.

E-mail: minhai361@aliyun.com

Jing-Ting Lu received the B. Sc., M. Sc. and Ph. D. degrees in computer science from Hefei University of Technology, China in 2004, 2009, and 2014, respectively. She is currently a lector in School of Computer and Information, Hefei University of Technology, China.

Her research interests include computer vision, biometrics, pattern recognition, image processing and machine learning.

E-mail: lujt@hfut.edu.cn

ORCID iD: 0000-0002-0210-7149