

2D and 3D Palmprint and Palm Vein Recognition Based on Neural Architecture Search

Wei Jia^{1,2} Wei Xia^{1,2} Yang Zhao^{1,2} Hai Min^{1,2} Yan-Xiang Chen^{1,2}

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

²Anhui Province Key Laboratory of Industry Safety and Emergency 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. In recent years, in the field of artificial intelligence, deep learning has gradually become the mainstream recognition technology because of its excellent recognition performance. Some researchers have tried to use convolutional neural networks (CNNs) for palmprint recognition and palm vein recognition. However, the architectures of these CNNs have mostly been developed manually by human experts, which is a time-consuming and error-prone process. In order to overcome some shortcomings of manually designed CNN, neural architecture search (NAS) technology has become an important research direction of deep learning. The significance of NAS is to solve the deep learning model's parameter adjustment problem, which is a cross-study combining optimization and machine learning. NAS technology represents the future development direction of deep learning. However, up to now, NAS technology has not been well studied for palmprint recognition and palm vein recognition. In this paper, in order to investigate the problem of NAS-based 2D and 3D palmprint recognition and palm vein recognition in-depth, we conduct a performance evaluation of twenty representative NAS methods on five 2D palmprint databases, two palm vein databases, and one 3D palmprint database. Experimental results show that some NAS methods can achieve promising recognition results. Remarkably, among different evaluated NAS methods, ProxylessNAS achieves the best recognition performance.

Keywords: Performance evaluation, neural architecture search, biometrics, palmprint, palm vein, deep learning.

Citation: W. Jia, W. Xia, Y. Zhao, H. Min, Y. X. Chen. 2D and 3D palmprint and palm vein recognition based on neural architecture search. *International Journal of Automation and Computing*, vol.18, no.3, pp.377–409, 2021. <http://doi.org/10.1007/s11633-021-1292-1>

1 Introduction

In the digital and intelligent society, more and more application scenarios need to authenticate people's identity effectively. Biometric technology is considered to be one of the most effective solutions for personal authentication. The so-called biometrics refers to the technology that uses the human body's physical or behavioral characteristics to identify individuals through image processing, computer vision, pattern recognition and other techniques. Generally speaking, face recognition, fingerprint recognition and iris recognition are the three most successful biometric technologies and have been widely used. However, different biometric technologies have their advantages and disadvantages. In other words, there is no one biometric technology that can meet the needs of all applications of personal authentication. Therefore, academic and industrial circles are developing different bio-

metric technologies to meet the application requirements of different scenarios.

In recent years, palmprint recognition and palm vein recognition have become two new biometric recognition technologies, which have attracted great attention^[1-4]. Palmprint recognition refers to the technology conducting personal authentication based on the palm skin images of human hands. According to the resolution and data type of palmprint image, palmprint recognition technology can be divided into 2D palmprint recognition and 3D palmprint recognition. Furthermore, 2D palmprint recognition can be further divided into low-resolution palmprint recognition and high-resolution palmprint recognition. High-resolution palmprint recognition is generally used for forensics purposes, while low-resolution palmprint recognition and 3D palmprint recognition are mainly used for civilian purposes. Palm vein recognition refers to the technology using the palm vein images captured under near-infrared light for personal authentication. Palm vein recognition is also mainly used for civilian purposes. Since palmprint and palm vein are both collected from the palm, and their recognition methods are similar to some extent, some researchers study them simultaneously. In this paper, we only pay attention to civil-

Research Article

Manuscript received January 7, 2021; accepted March 5, 2021; published online April 13, 2021

Recommended by Associate Editor Hui Yu

Colored figures are available in the online version at <https://link.springer.com/journal/11633>

© The author (s) 2021

ian use of biometrics technology, so we mainly study 2D low-resolution palmprint recognition, 3D palmprint recognition, and palm vein recognition. In the rest of this paper, for the sake of convenience, we will write 2D low-resolution palmprint recognition as 2D palmprint recognition.

Researchers have proposed many effective methods for 2D and 3D palmprint recognition and palm vein recognition, which can be divided into two groups, i.e., traditional methods and deep learning-based methods. Generally, traditional methods are based on hand-crafted features and traditional machine learning techniques. Different from traditional methods, deep learning can automatically learn features from images, videos or texts. The highly flexible architecture of deep learning can learn directly from the original data, and the prediction accuracy will be improved after more data are obtained. Nowadays, deep learning has become one of the most important technologies in the field of artificial intelligence. In recent years, the explosive progress made in computer vision, speech recognition, natural language processing, robotics and other fields almost all depend on deep learning technology^[5–8].

In the field of biometrics, especially in face recognition, deep learning has become the most mainstream technology^[9]. Convolutional neural network (CNN) is one of the most important branches in deep learning. For image-based biometrics technologies, CNN is the most commonly used deep learning technique^[9]. Until now, a lot of classic CNNs have been proposed, and have achieved impressive results for many recognition tasks. The success of these CNNs is mainly attributed to the automation of the feature engineering process: A layered feature extractor learns from data in an end-to-end manner. With this success, there is a growing demand for architecture engineering, and more and more complex neural architectures are designed in a manual manner. That is, currently employed architectures have mostly been developed manually by human experts, which is a time-consuming and error-prone process. In order to overcome some shortcomings of manually designed CNN, neural architecture search (NAS) technology has become an important research direction of deep learning^[10–14]. The core idea of NAS is to use a search algorithm to find the neural network structure needed to solve the problems. The significance of NAS is to solve the parameter adjustment problem of the deep learning model, which is a cross-study combining optimization and machine learning. The concept of NAS was first proposed by Zoph and Le^[15] at *International Conference on Machine Learning (ICML)* in 2017, and has become a fundamental and active research direction of deep learning.

With the continuous improvement of deep learning network architecture and the increasing amount of data, the recognition accuracy of deep learning in different biometrics tasks is also increasing. For example, in the field

of face recognition, the recognition accuracy of deep learning has far exceeded the traditional hand-crafted algorithms; thus, deep learning has successfully promoted the large-scale application of face recognition technology. However, in the fields of 2D and 3D palmprint recognition and palm vein recognition, the related research based on deep learning is still preliminary. A lot of researchers have used some classic CNNs or manually designed CNNs for 2D and 3D palmprint recognition and palm vein recognition. Nevertheless, up to now, NAS technology has not been well studied for 2D and 3D palmprint recognition and palm vein recognition. Because NAS technology represents the future development direction of deep learning, it is vital to systematically investigate the recognition performance of NAS methods for 2D and 3D palmprint recognition and palm vein recognition. To this end, we conduct the performance evaluation of NAS methods on 2D and 3D palmprint recognition and palm vein recognition in this paper. Particularly, twenty representative NAS methods are selected and exploited for performance evaluation.

The selected NAS methods are evaluated on five 2D palmprint databases, one 3D palmprint database and two palm vein databases. They are all representative databases in 2D and 3D palmprint recognition and palm vein recognition. Five 2D palmprint databases include Hong Kong Polytechnic University palmprint database II (PolyU II)^[16], the blue band of the Hong Kong Polytechnic University Multispectral (PolyU M_B) palmprint database^[17], Hefei University of Technology (HFUT) palmprint database^[18], Hefei University of Technology Cross Sensor (HFUT CS) palmprint database^[19], and Tongji University palmprint (TJU-P) database^[20]. The 3D palmprint database is Hong Kong Polytechnic University 3D palmprint database (PolyU 3D)^[21]. Two palm vein databases include the near-infrared band of Hong Kong Polytechnic University Multispectral palmprint database (PolyU M_N)^[17] and Tongji University palm vein (TJU-PV) database^[22].

It should be noted that the samples within the above databases are captured in two different sessions at certain time intervals. If the training samples are only from the first session, and the test samples are from the second session, we call this experimental mode the “separate data mode”. If the training samples are from both sessions, we call this experimental mode the “mixed data mode”. 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. Therefore, the experiments of those traditional recognition methods were usually conducted in the “separate data mode”. However, in existing deep learning-based palmprint recognition and palm vein recognition methods, the experiments were usually conducted in the “mixed data mode”. Thus, it is easy to obtain a high recognition accuracy. In this paper, we will

conduct experiments in both “separate data mode” and “mixed data mode” to observe the recognition performance of representative NAS methods in these two different modes.

The main contributions of our work are as follows.

1) We briefly summarize some important NAS methods, which can help the readers to better understand the development history of NAS technology.

2) We conduct a performance evaluation of representative NAS methods for 2D and 3D palmprint and palm vein recognition. To the best of our knowledge, it is the first time such an evaluation has been conducted. Particularly in the field of biometrics, this is also the first work to evaluate the recognition performance of representative NAS methods.

3) We evaluated the performance of representative NAS methods on Hefei University of Technology cross sensor palmprint database. It is the first time that the problem of palmprint recognition across different devices using NAS technology has been investigated.

4) We investigate the problem of the recognition performance of NAS methods 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 NAS technology. Section 4 introduces the selected NAS methods in detail. Section 5 introduces the 2D and 3D palmprint and palm vein databases used for evaluation. Extensive experiments are conducted and reported in Section 6. Section 7 offers the concluding remarks.

2 Related work

2.1 Traditional 2D palmprint recognition methods

For 2D palmprint recognition, researchers have proposed many traditional methods. Kong et al.^[1], Zhang et al.^[2], Fei et al.^[3], Zhong et al.^[4] have published several survey papers on the traditional palmprint recognition methods. As shown in Fig. 1, these traditional methods can be classified into different subcategories, such as palm line-based, texture-based, orientation coding-based, correlation filter-based, and subspace learning-based. The palm line is the primary feature of the palmprint. Therefore, some researchers tried to extract the palm lines for palmprint recognition. However, due to the complexity of palmprint images, it is still difficult to extract palm lines

accurately. Palmprint images contain obvious texture features. Therefore, researchers have proposed many texture-based palmprint recognition methods. Texture-based methods usually exploited sparse descriptor, dense descriptor, or other texture representations, such as the Gabor feature and wavelet feature, for palmprint recognition. Thus, texture-based methods can be further divided into three subtypes, i.e., Gabor and wavelet-based methods, dense texture descriptor-based methods, and sparse texture descriptor-based methods. Notably, some dense texture descriptors have achieved promising recognition results. As we know, palmprint contains many palm lines, and these lines have their orientations. Orientation features are insensitive to some variations such as illumination changes; thus, orientation is a robust feature of palmprint. A lot of orientation coding-based methods have been proposed, which 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, exploited Hamming distance for matching. Recently, correlation methods have been successfully used for biometrics, which also has high accuracy and fast matching speed. Subspace learning has been one of the important techniques for pattern recognition. Some subspace learning-based methods have been used for palmprint recognition. 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

Fei et al.^[23] surveyed the papers on traditional 3D palmprint recognition methods. Generally, 3D palmprint data preserves the depth information of a palm surface. The original captured 3D palmprint data is a small positive or negative float. For practical feature extraction, the original 3D palmprint data is usually transformed into the grey-level value. To this end, the original 3D palmprint data is usually transformed into a curvature-based data to facilitate the design of recognition algorithms. The two most important curvatures include the mean curvature (MC) and Gaussian curvature (GC), and their corresponding images are mean curvature image (MCI) and Gaussian curvature image (GCI). Based on GC and MC, two new grey-level image representations have been proposed, including surface type (ST) and compact ST (CST). Since the representations of MCI, GCI, ST and CST depict a 3D palmprint as a 2D grey-

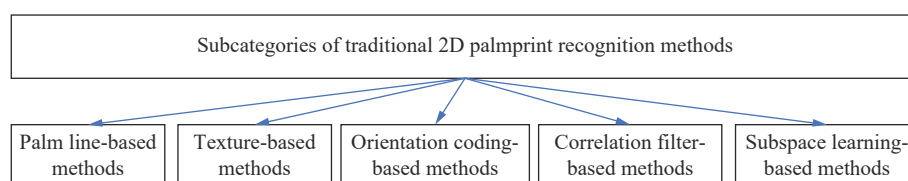


Fig. 1 Subcategories of traditional 2D palmprint recognition methods

level palmprint image, those 2D palmprint recognition methods can be used for 3D palmprint recognition. Fig. 2 shows the examples of original 3D palmprint, 3D palmprint region of interest (ROI), and four 2D representations, including MCI, GCI, ST and CST.

2.3 Traditional palm vein recognition methods

As shown in Fig. 3, the traditional palm vein recognition methods can be divided into the following subcategories: structure-based, texture-based, orientation coding-based, and subspace learning-based. Structure-based methods usually first perform an image segmentation algorithm or line detection algorithm, and then extract the structure features of palm vein for recognition such as lines, skeletons, points, minutiae and graph. Thus, structure-based methods can be further divided into three subtypes, i.e., line/skeleton-based methods, points/minutiae-based methods, and graph-based methods. Texture-based methods, orientation coding-based methods, and subspace learning-based methods used for palm vein recognition are similar to those used for palmprint recognition.

2.4 2D and 3D palmprint recognition and palm vein recognition methods based on deep learning

Many researchers have studied 2D and 3D palmprint recognition and palm vein recognition based on deep learning.

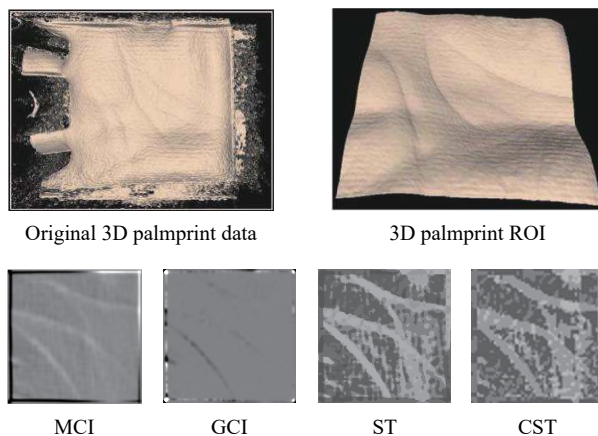


Fig. 2 Samples of original 3D palmprint, 3D palmprint ROI, and four 2D representations including MCI, GCI, ST and CST

Some representative 2D palmprint recognition methods based on deep learning are as follows. Zhang et al.^[22] proposed the method of PalmRCNN for palmprint recognition, which is a modified version of Inception-ResNet-V1. Genovese et al.^[24] proposed the method of PalmNet, 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^[25] 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 an uncontrolled and uncooperative environment, Matkowski et al.^[26] 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^[27] 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, and uses the collaborative representation-based classifier (CRC) for recognition. Zhao and Zhang^[28] presented a joint constrained least-square regression (JCLSR) model with a deep local convolution feature for palmprint recognition. Zhao et al.^[29] also proposed a joint deep convolutional feature representation (JDCFR) methodology for hyperspectral palmprint recognition. Liu and Kumar^[30] proposed a generalizable deep learning-based framework for the contactless palmprint recognition, in which the network is based on a fully convolutional network that generates deeply learned residual features.

Some representative palm vein recognition algorithms based on deep learning are as follows. Zhang et al.^[22] released a new touchless palm vein database and used the method of PalmRCNN for palm vein recognition. Lefkivits et al.^[31] applied four CNNs for palm vein identification, including AlexNet, VGG-16, ResNet-50, and SqueezeNet. Thapar et al.^[32] proposed the method of PVSNet, where a Siamese network was trained using triplet loss. Chantaf et al.^[33] applied Inception-V3 and SmallerVGGNet for palm vein recognition. Stanuch et al.^[34] proposed a contact-free multispectral palm vein recognition system using a designed CNN, whose architecture comprises of ten different layers, including five convolutional, four max pooling, and one dense layers.

In our previous work^[35], we systematically investigated the recognition performance of classic CNNs for 2D and 3D palmprint recognition and palm vein recognition. Seventeen representative and classic CNNs were ex-

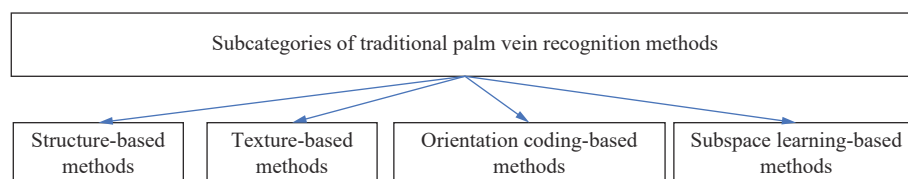


Fig. 3 Subcategories of traditional 2D palm vein recognition methods

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. We have also conducted experiments on both separate data mode and mixed data mode. Among different classic CNNs, EfficientNet achieves the best recognition accuracy.

3 A brief introduction of NAS technology

NAS is the sub-field of automated machine learning (AutoML). The goal of NAS is to design a network architecture with the best performance with the least human intervention and limited computing resources. The papers of [15] and [36] are considered as the pioneering work of NAS. In [15], the network structure obtained by the reinforcement learning (RL) achieves very promising accuracy in image classification tasks, which shows the idea of automation network architecture design is feasible. The development of NAS technology is very rapid. At the same time, NAS technology is being widely used in various tasks, such as classification, object detection, semantic segmentation, language modeling, and data augmentation, etc.

Despite the short development history of NAS, there have been many papers published, including five survey papers[10–14]. In 2019, Elsken et al.[10] provided an overview of existing NAS methods and categorized them according to three dimensions: search space, search strategy, and performance estimation strategy. Wistuba et al.[11] provided a formalism that unifies the landscape of existing NAS methods. This formalism can be used to critically examine the different approaches and understand the benefits of the different components that contribute to the design and success of NAS. Wistuba et al.[11] also highlighted some popular misconception pitfalls in the current trends of NAS technology. Ren et al.[12] provided a new perspective of NAS technology: Starting with an overview of the characteristics of the

earliest NAS algorithms, a summary of the problems in these early NAS algorithms, and then giving solutions for subsequent related research work. Ren et al.[12] also conducted a detailed and comprehensive analysis, comparison and summary of existing NAS works and gave possible future research directions. Hu and Yu[13] surveyed NAS technology from a technical view. By summarizing the previous NAS approaches, Hu and Yu[13] drew a picture of NAS from different aspects, including problem definition, search approaches, progress towards practical applications and possible future directions. He et al.[14] compared the performance and efficiency of existing NAS algorithms on the CIFAR-10 and ImageNet datasets and provided an in-depth discussion of different research directions on NAS, including one/two-stage NAS, one-shot NAS, and joint hyperparameter and architecture optimization.

Almost all NAS methods are organized around three components: search space, optimization method, and evaluation method. Fig. 4 shows an abstract illustration of the NAS methods.

1) Search space. Search space is a set of possible neural network architectures. It adopts different design concepts according to different application scenarios, including computer vision tasks and language modeling tasks. From the above point of view, NAS is not completely out of artificial design; instead, it is more based on the design of the network structure of the search and reconstruction, and the number of architectures searched will usually reach a very large order of magnitude.

2) Optimization method. The optimization method teaches the search space how to search better. A good optimization method often plays a key role. Although there are many optimization methods, the starting point of their research is to obtain a better network architecture. In addition, most optimization methods are introduced on the basis of traditional optimization methods, such as reinforcement learning, evolutionary search, gradient-based optimization, Bayesian optimization, etc.

3) Evaluation method. The evaluation method especially evaluates the network structure. The general evalu-

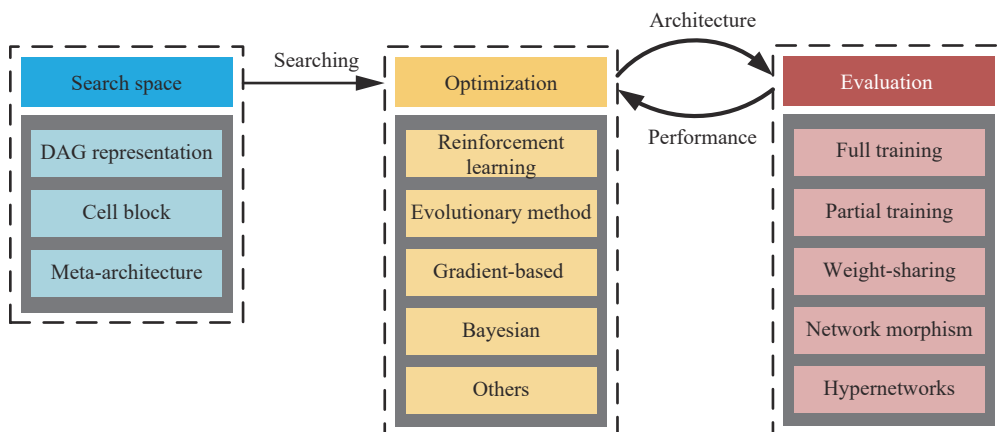


Fig. 4 Abstract illustration of neural architecture search methods

ation methods include a full training mode, partial training mode and NAS specific evaluation method. The full training mode is a time-consuming method, which usually requires thorough training for all the searched models, while partial training mode usually stops the training process early, saving cost and time. Among the specific evaluation methods of NAS, network morphism, weight sharing and hypernetworks are often used as heuristic quality assessment methods. In general, partial training mode is typically an order-of-magnitude cheaper than full training mode, while NAS specific evaluation methods are 2–3 orders-of-magnitude cheaper than full training mode.

Many NAS methods have been proposed. According to the survey papers of NAS and the new development of NAS, we selected a lot of important NAS methods and listed them in Tables 1–3 according to the publishing year. It can be seen that most NAS papers have been published after 2017 and at some top conferences of artificial intelligence such as CVPR, ICML, ICCV, ECCV, ICLR, and NeurIPS. In fact, the number of papers on NAS is increasing rapidly. More comprehensive and newest NAS papers can be found on the websites <https://github.com/D-X-Y/Awesome-AutoDL> and <https://www.automl.org>.

4 Selected representative NAS methods for performance evaluation

The classification task is one of the important applications of NAS technology. As can be seen from Tables 1–3, most NAS methods are dedicated to finding a robust classification model. Fig. 5 shows the chronology of representative NAS methods for the classification task. These methods play an important role in the development history of NAS. Here, we briefly introduce them by year of publication.

In 2017, Zoph and Le^[15] published the first paper that proposed the concept of NAS. Their work expresses the network structure as a variable-length string. They learn a good network structure through reinforcement learning (RL), specifically generate a description of the neural network model by using recurrent neural network (RNN), and train the RNN to maximize the accuracy of the generated neural network model.

In 2018, Zoph et al.^[44] proposed the method of NASNet. NASNet improves the search space from searching hyperparameters to searching block cell structure, and its accuracy can reach state of the art (SOTA). Moreover, in this method, Zoph et al.^[44] proposed to search on the proxy dataset, the small datasets (such as CIFAR-10), and then migrated to large datasets (such as ImageNet). Brock et al.^[41] proposed the method of SMASH. SMASH uses an auxiliary network to initialize parameters of different networks and avoid retraining again, which greatly reduces the training time. Liu et al.^[39] proposed the method of PNASNet to learn the structure of the CNN, which is more effective than the NAS methods based on reinforcement learning and evolutionary algorithm. Particularly, a sequential model-based optimization strategy is used in PNASNet. Luo et al.^[40] proposed the method of NAONet, which is a new method for optimizing network architecture, mapping the architecture to a continuous vector space. NAONet uses performance predictors and encoders to perform gradient optimization in continuous space to find a new coding structure with higher accuracy and decode it into a network by decoder.

In 2019, Xie et al.^[62] proposed the method of SNAS. SNAS directly optimizes the objective function of the NAS task, and puts forward the expectation of optimizing the network loss function and network forward delay, so as to automatically generate hardware-friendly sparse

Table 1 List of important NAS papers published in 2017 and 2018

Reference	Full name	Abbreviation	Year	Source	Optimization method	Application
Zoph and Le ^[15]	Neural architecture search	NAS	2017	ICML	Reinforcement learning	Classification
Baker et al. ^[36]	MetaQNN	MetaQNN	2017	ICML	Reinforcement learning	Classification
Shin et al. ^[37]	Differentiable neural network architecture Search	DAS	2018	ICLR	Gradient based	Classification
Kandasamy et al. ^[38]	Neural architecture search	NASBOT	2018	NeurIPS	Other	Classification
Liu et al. ^[39]	Progressive neural architecture search	PNASNet	2018	ECCV	Performance prediction	Classification
Luo et al. ^[40]	Neural architecture optimization	NAONet	2018	NeurIPS	Gradient based	Classification
Brock et al. ^[41]	One-shot model architecture search	SMASH	2018	ICLR	Gradient based	Classification
Bender et al. ^[42]	Understanding and simplifying one-shot architecture search	One-shot	2018	ICML	Gradient based	Classification
Zhong et al. ^[43]	Block-wise neural network architecture	Block-QNN	2018	CVPR	Reinforcement learning	Classification
Zoph et al. ^[44]	NASNet architecture	NASNet	2018	CVPR	Reinforcement learning	Classification
Yang et al. ^[45]	Platform-aware neural network adaptation	NetAdapt	2018	ECCV	Other	Classification

Table 2 List of important NAS papers published in 2019

Reference	Full name of methods	Abbreviation	Year	Source	Optimization method	Application
Ying et al. ^[46]	NAS-Bench-101	NAS-Bench-101	2019	ICML	Other	Classification
Tan and Le ^[47]	EfficientNet	EfficientNet	2019	ICML	Reinforcement learning	Classification
Chu et al. ^[48]	Fairness of weight sharing neural architecture search	FairNAS	2019	arXiv	Evolutionary algorithm	Classification
Ho et al. ^[49]	Population based augmentation	PBA	2019	ICML	Evolutionary algorithm	Augmentation
Cubuk et al. ^[50]	AutoAugment	AutoAugment	2019	CVPR	Reinforcement learning	Augmentation
Real et al. ^[51]	AmoebaNet	AmoebaNet	2019	AAAI	Evolutionary algorithm	Classification
Nekrasov et al. ^[52]	–	–	2019	CVPR	Reinforcement learning	Semantic segmentation
Chu et al. ^[53]	Bridging the gap between stability and scalability in weight-sharing neural architecture search	ScarletNAS	2019	arXiv	Evolutionary algorithm	Classification
Pérez-Rúa et al. ^[54]	Multimodal fusion architecture search	MFAS	2019	CVPR	Evolutionary algorithm	Classification
Tan et al. ^[55]	Mobile neural architecture search	MNASNet	2019	CVPR	Reinforcement learning	Classification
Liu et al. ^[56]	Auto-Deeplab	Auto-Deeplab	2019	CVPR	Gradient based	Semantic segmentation
Chen et al. ^[57]	Reinforced evolutionary neural architecture search	RENAS	2019	CVPR	Gradient based	Classification
Wu et al. ^[58]	Facebook-Berkeley-Nets	FBNet-V1	2019	CVPR	Gradient based	Classification
Li et al. ^[59]	Dongfeng networks	DF	2019	CVPR	Evolutionary algorithm	Classification
Dong and Yang ^[60]	Differentiable architecture sampler	GDAS	2019	CVPR	Gradient based	Classification
Pham et al. ^[61]	Efficient neural architecture search	ENAS	2019	CVPR	Reinforcement learning	Classification
Xie et al. ^[62]	Stochastic neural architecture search	SNAS	2019	ICLR	Gradient based	Classification
Elsken et al. ^[63]	LEMONADE	LEMONADE	2019	ICLR	Evolutionary algorithm	Classification
Ghiasi et al. ^[64]	Feature pyramid architecture with neural architecture search	NAS-FPN	2019	CVPR	Reinforcement learning	Object detection
Zhang et al. ^[65]	Graph hypernetworks for neural architecture search	GHN	2019	ICLR	Gradient based	Classification
Cai et al. ^[66]	Direct neural architecture search	ProxylesNAS	2019	ICLR	Reinforcement learning Gradient based	Classification
Liu et al. ^[67]	Differentiable architecture search	DARTS	2019	ICLR	Gradient based	Classification
Nayman et al. ^[68]	Neural architecture search with expert advice	XNAS	2019	NeurIPS	Gradient based	Classification
Peng et al. ^[69]	Neural architecture transformation	NATS	2019	NeurIPS	Gradient based	Object detection
Hu et al. ^[70]	Petridish	Petridish	2019	NeurIPS	Gradient based	Classification
Dong and Yang ^[71]	Transformable architecture search	TAS	2019	NeurIPS	Gradient based	Classification
Chen et al. ^[72]	Neural architecture search on object detection	DetNAS	2019	NeurIPS	Other	Object detection
Wortsman et al. ^[73]	Discovering neural wirings	DNW	2019	NeurIPS	Gradient based	Classification
Dong and Yang ^[74]	Neural architecture search for generative adversarial networks	AutoGAN	2019	ICCV	Reinforcement learning	GAN
Dong and Yang ^[75]	Self-evaluated template network	SETN	2019	ICCV	Gradient based	Classification
Xiong et al. ^[76]	Resource constrained neural network architecture search	RCNet	2019	ICCV	Evolutionary algorithm	Classification
Howard et al. ^[77]	MobileNet-V3	MobileNet-V3	2019	ICCV	Evolutionary algorithm	Classification
Zheng et al. ^[78]	Multinomial distribution learning for effective neural architecture search	MdeNAS	2019	ICCV	Other	Classification
Pasunuru and Bansal ^[79]	Continual architecture search	CAS	2019	ACL	Reinforcement learning	Video captioning
Jiang et al. ^[80]	Improved differentiable architecture search	I-DARTS	2019	EMNLP	Gradient based	NLP
Li and Talwalkar ^[81]	Random search and reproducibility for neural architecture search	Random search WS	2019	UAI	Gradient based	Classification

network. Real et al.^[51] proposed the method of AmoebaNet. The search space of AmoebaNet adopts the search space of NASNet^[44], and its network structure is similar to the structure of Inception^[110]. In SNAS, the algorithm

of ageing evolution is used to achieve better results. Pham et al.^[61] proposed the method of ENAS, which is an economical and automatic model design method. By forcing all sub-models to share weights, the shortcomings of

Table 3 List of important NAS papers published in 2020

Reference	Full name	Abbreviation	Year	Source	Optimization method	Application
Chu et al. ^[82]	Mobile GPU-aware neural architecture search	MoGA	2020	ICASSP	Evolutionary algorithm	Classification
Chen et al. ^[83]	Searching for faster real-time semantic segmentation	FasterSeg	2020	ICLR	Gradient based	Semantic segmentation
Xu et al. ^[84]	Partially-connected differentiable architecture search	PC-DARTS	2020	ICLR	Gradient based	Classification
Mei et al. ^[85]	Atomic blocks for neural architecture search	AtomNAS	2020	ICLR	Other	Classification
Dong and Yang ^[86]	NAS-Bench-201	NAS-Bench-201	2020	ICLR	Other	Classification
Tan et al. ^[87]	EfficientDet	EfficientDet	2020	CVPR	Reinforcement learning	Object detection
Fang et al. ^[88]	Densely connected search space for more flexible neural architecture search	DenseNAS	2020	CVPR	Gradient based	Classification
Zhang et al. ^[89]	Gradient-based sampling NAS-random sampling NAS	GDAS-NSAS	2020	CVPR	Gradient based	Classification
Li et al. ^[90]	Distill neural architecture	DNA	2020	CVPR	Gradient based	Classification
Guo et al. ^[91]	Robust architectures network	RobNet	2020	CVPR	Gradient based	Classification
Gao et al. ^[92]	Adversarial neural architecture search for GANs	AdversarialNAS	2020	CVPR	Gradient based	GAN
Wan et al. ^[93]	Differentiable neural architecture search for spatial and channel dimensions	FBNet-V2	2020	CVPR	Gradient based	Classification
Bender et al. ^[94]	TuNAS	TuNAS	2020	CVPR	Reinforcement learning	Classification
Li et al. ^[95]	Sequential greedy architecture search	SGAS	2020	CVPR	Gradient based	Classification
Zheng et al. ^[96]	Budgeted performance estimation	BPE	2020	CVPR	Other	Classification
Phan et al. ^[97]	Binary neural network	BNN	2020	CVPR	Evolutionary algorithm	Classification
He et al. ^[98]	Efficient neural architecture search via mixed-level reformulation	MiLeNAS	2020	CVPR	Gradient based	Classification
Dai et al. ^[99]	Data adapted pruning for efficient neural architecture search	DA-NAS	2020	ECCV	Gradient based	Classification
Tian et al. ^[100]	Efficient and effective GAN architecture search	E ² GAN	2020	ECCV	Reinforcement learning	GAN
Chu et al. ^[101]	Fair differentiable architecture search	FairDARTS	2020	ECCV	Gradient based	Classification
Hu et al. ^[102]	Three-freedom neural architecture search	TF-NAS	2020	ECCV	Gradient based	Classification
Hu et al. ^[103]	Angle-based search space shrinking	ABS	2020	ECCV	Other	Classification
Yu et al. ^[104]	Barrier penalty neural architecture search	BP-NAS	2020	ECCV	Other	Classification
Wang et al. ^[105]	Attention cell search for video classification	AttentionNAS	2020	ECCV	Other	Video classification
Bulat et al. ^[106]	Binary architecture search	BATS	2020	ECCV	Other	Classification
Yu et al. ^[107]	Neural architecture search with big single-stage models	BigNAS	2020	ECCV	Gradient based	Classification
Guo et al. ^[108]	Single path one-shot neural architecture search with uniform sampling	Single-Path-SuperNet	2020	ECCV	Evolutionary algorithm	Classification
Liu et al. ^[109]	Unsupervised neural architecture search	UnNAS	2020	ECCV	Gradient based	Classification

huge and time-consuming NAS computing power are overcome, and the GPU computing time is reduced by

more than 1 000 times. Cai et al.^[66] proposed a NAS method without proxy tasks called ProxylessNAS. Proxy-

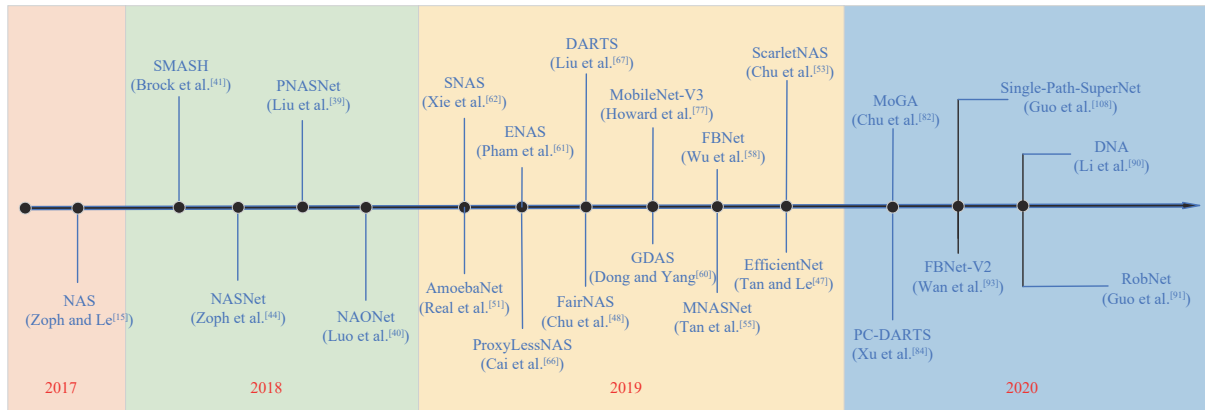


Fig. 5 Chronology of representative NAS methods for the classification task

lessNAS can directly search structures of large-scale target tasks, which can solve large GPU memory consumption problems and long computation time of the NAS method. Liu et al.^[67] proposed the method of DARTS for effective structure search. Instead of searching in the discrete set of candidate structures, the search space of DARTS is relaxed to the continuous domain. The optimization can be carried out with the effect of verification set by means of gradient descent. The method of FairNAS is proposed by Chu et al.^[48] FairNAS is the inheritance and development of the one-shot in the NAS community. FairNAS thinks that fair sampling and training methods can exert the potential of each module. Therefore, FairNAS proposes to meet the strict fairness. This constraint is that every single iteration of the hypernetwork makes the parameters of each optional operation module of each layer be trained. Dong and Yang^[60] proposed the method of GDAS, which uses the gradient descent method to realize the effective network structure search. GDAS treats the search space as a directed acyclic graph, uses a differentiable sampler to test the sample structure, and optimizes the sampler by training the verification loss of the sampled structure. Howard et al.^[77] proposed the method of MobileNet-V3, a new lightweight network structure based on MobileNet-V2^[111]. It is searched by MNASNet^[55] and NetAdapt^[45]. MobileNet-V3 contains the MobileNet-V3-large version and the MobileNet-V3-small version to cope with resource consumption scenarios. Moreover, MobileNet-V3 has been successfully used in target detection and semantic segmentation tasks. Wu et al.^[58] proposed a search framework for differentiable neural structures called DNAS, which uses a gradient-based method to optimize the convolution network structure and avoids exhaustive and independent training structure. FB-Nets are the family of network structures generated by the DNAS search framework, surpasses the manually designed and automatically generated state of the art model. Tan et al.^[55] proposed an automatic mobile NAS method called MNASNet, which explicitly incorporated model delay into the main target. The search can identify a model that achieves a good trade-off between precision and delay. Chu et al.^[53] pro-

posed the method of ScarletNAS with scalability function and solved the fairness problem of scalable hypernetwork training in one-shot routes through linear equivalent transformation. Tan and Le^[47] proposed the method of EfficientNet. They used NAS to search a baseline network with accuracy and flops simultaneously, and do the balance of depth, width and resolution, and get a group of better EfficientNets.

In 2020, Chu et al.^[82] designed the mobile terminal GPU sensitive model from a practical standpoint, which is called MoGA. The method of PC-DARTS was proposed by Xu et al.^[84] PC-DARTS is an extension of DARTS, which reduces the memory consumption of computing time in the process of network search through partial channel connection. Guo et al.^[108] constructed a simplified hypernetwork called Single-Path-SuperNet, which was trained according to the uniform path sampling method. All substructures (and their weights) are fully and equally trained. Based on the trained hypernetwork, the optimal substructure can be quickly searched by an evolutionary algorithm, in which no fine-tuning of any substructure is required. Based on the idea of knowledge distillation, Li et al.^[90] proposed the distill neural architecture (DNA), which introduces a teacher model to guide the direction of the network structure search. Using the supervision information from different depths of teacher model, the original end-to-end network search space is divided into blocks in-depth to realize the weight sharing training of independent blocks of network search space, which significantly reduces the interference caused by weight sharing. Wan et al.^[93] proposed FBNet-V2, which takes both memory and efficiency into account. FBNet-V2 uses a masking mechanism for feature graph reuse and effective shape propagation to obtain better accuracy. Guo et al.^[91] studied the model of neural network structure and then proposed the method of RobNet, which can resist the attack from the perspective of neural network structure. To obtain a large number of networks needed for research, they used a one-shot neural network structure search to train a super-net and then fine-tuned the sub-networks sampled from it.

For 2D and 3D palmprint recognition and palm vein recognition, we select twenty representative NAS methods for performance evaluation including NASNet^[44], SMASH^[41], PNASNet^[39], NAONet^[40], SNAS^[62], AmoebaNet^[51], ENAS^[61], ProxylessNSA^[66], DARTS^[67], FairNAS^[48], GDAS^[60], FBNet-V1^[58], MNASNet^[55], ScarletNAS^[53], MoGA^[82], PC-DARTS^[84], Single-Path-SuperNet^[108], DNA^[90], FBNet-V2^[93] and RobNet^[91]. There are two main reasons for choosing these NAS methods. One reason is that these NAS methods are often used as reference methods for performance comparison in many literature pieces. The second reason is that the results of these NAS approaches are outstanding. It's worth noting that MobileNet-V3^[77] and EfficientNet^[47] are network architectures based on the existing NAS methods and then elaborately designed manually. Therefore, MobileNet-V3 and EfficientNet can be viewed as semi-NAS methods; thus, they are not included in our selection. In our previous work^[35], we have classified MobileNet-V3 and EfficientNet as traditional classic CNNs, and we have evaluated their recognition performance.

In this section, we introduce the selected NAS methods in detail as follows.

1) NASNet

NASNet^[44] was designed to make the training structure transferable. The best structure of NASNet is found on CIFAR-10 dataset, and then stacked several times and then applied to the ImageNet dataset. In addition, a new regularization technique called schedule drop path is proposed in this method. Fig. 6(a) shows the controller model architecture for recursively constructing one block of a convolutional cell. Each block requires selecting 5 discrete parameters, each of which corresponds to the output of a softmax layer. Fig. 6(b) shows the architecture of the best convolutional cells with $B = 5$ blocks identified with CIFAR-10. NASNet has different versions including NASNet-A, NASNet-B, NASNet-C and NASNet-mobile. The recognition performance of NASNet-A is the best, and the NASNet-mobile is light-weight network. In this paper, NASNet-A and NASNet-mobile are used for performance evaluation.

2) SMASH

SMASH^[41] trains an auxiliary model-hypernet to train the candidate models in the search process, which dynamically generates the weights of the main model with variable structure. Although the weights generated are worse than those obtained by free learning of fixed network structure, the relative performance of different networks in early training provides meaningful guidance for the performance under an optimal state. At the same time, a network representation mechanism based on memory back is developed to define various network structures.

3) PNASNet

PNASNet^[39] can learn a CNN, which matches the previous SOTA, and requires five times less model evalu-

ation during architecture search. The starting point of this work is the structured search space proposed by NASNet, in which the task of the search algorithm is to search for a suitable convolution cell rather than a complete CNN. A cell contains B blocks, one of which is a combination operator (such as addition) applied to two inputs (tensors), each of which can be transformed (e.g., using convolution) before combining. Then, according to the size of the training set and the running time of the CNN, the cell structure is stacked for a certain number of times. This modular design also allows us to migrate the architecture from one dataset to another easily. Fig. 7(a) shows the best cell structure found by PNASNet, consisting of 5 blocks, and Fig. 7(b) shows the construction of CNNs from cells on CIFAR-10 and ImageNet.

4) NAONet

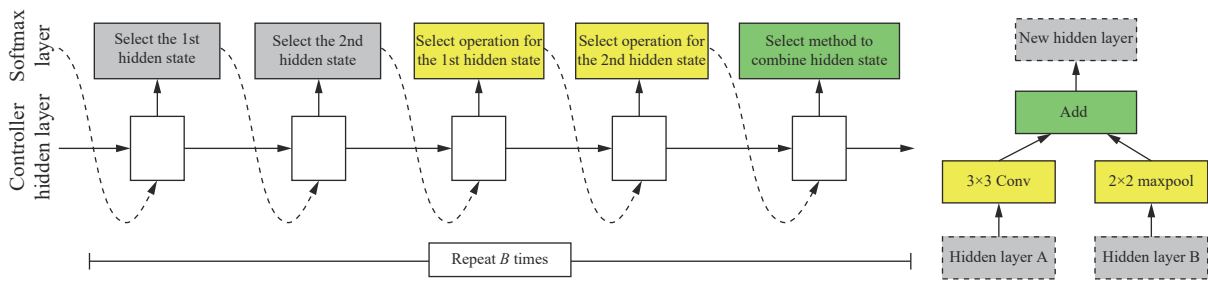
Fig. 8 shows the general framework of NAONet^[40]. NAONet consists of three parts: encoder, predictor and decoder. Experimental results showed that the architecture found by NAONet performs well in both CIFAR-10 image classification task and penn treebank (PTB) language modeling task, and is better than or equal to the best previous architecture search methods with significantly reduced computing resources.

5) SNAS

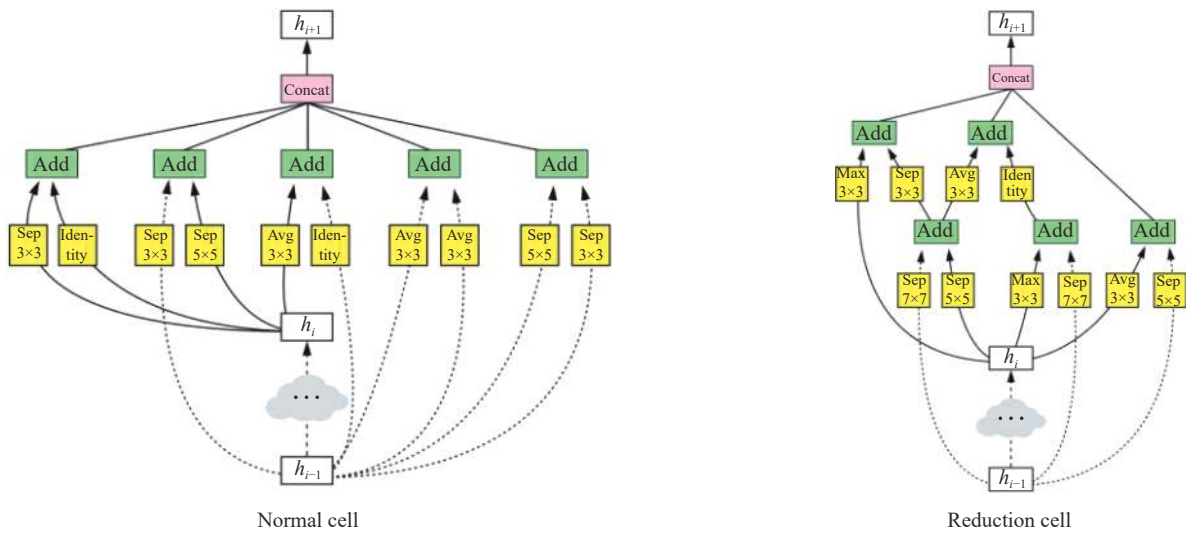
Compared with ENAS, the search optimization of SNAS is differentiable and the search efficiency is higher^[62]. Compared with other differentiable methods such as DARTS and so on, SNAS directly optimizes the objective function of NAS tasks, and the search structure is more robust and efficient for multitasking. In addition, based on the advantage that SNAS keeps the advantage of stochasticity, Xie et al.^[62] further proposed to optimize both the expectation of network loss function and the expectation of network forward delay to generate hardware friendly sparse network automatically.

6) AmoebaNet

AmoebaNet^[51] improves the tournament selection method in genetic algorithm. This method is changed into an age-based selection method, namely the ageing evolution algorithm, which makes the genetic algorithm prefer young individuals. Experiments show that the algorithm has a faster search speed than reinforcement learning and random search on the same hardware conditions. The ageing evolution algorithm has the following six steps: i) P neural network structures are randomly initialized and added to the queue to form a population for training; ii) The population was sampled, and S neural networks were selected; iii) S neural networks are obtained by sampling, and the neural network with the highest accuracy is selected as the parent; iv) The network is trained and added to the population, i.e., the rightmost side of the queue; v) Removing the "oldest" neural network in the population is actually the leftmost element of the queue; vi) Go back to Step ii) and cycle a certain number of times.



(a) Controller model architecture for recursively constructing one block of a convolutional cell



(b) Architecture of the best convolutional cells with $B = 5$ blocks identified with CIFAR-10

Fig. 6 A diagram of the top performing normal cell and reduction cell in NASNet^[44]: (a) Controller model architecture for recursively constructing one block of a convolutional cell; (b) Architecture of the best convolutional cells with $B = 5$ blocks identified with CIFAR-10.

7) ENAS

ENAS^[61] is an economical and automatic model design method beyond NASNet^[44]. By forcing all sub-models to share weights, the efficiency of NAS is improved, and the shortcomings of huge computing cost and time-consuming of NAS are overcome. The computing time of GPU is reduced by more than 1000 times. On the CIFAR-10 dataset, the test error reaches 2.89%, which is similar to NASNet (2.65% test error). Fig. 9 shows the network architecture of ENAS.

8) ProxylessNSA

ProxylessNAS^[66] is the first NAS algorithm to search large design space directly on large-scale ImageNet datasets without any proxy and customize CNN architecture for hardware for the first time. Cai et al.^[66] combine the idea of model compression (pruning and quantifying) with NAS, reduce the computing cost (GPU time, GPU memory) of NAS to the same scale as the conventional training, and reserve rich search space, and directly incorporate the hardware performance (delay, energy consumption) of neural network structure into the optimization objective. Fig. 10 shows the efficient models optimized for different hardware. Figs. 10(a)–10(c) show the

GPU model, the CPU model and the mobile model found by ProxylessNAS. GPU prefers shallow and wide model with early pooling, while CPU prefers deep and narrow model with late pooling. Pooling layers prefer large and wide kernel while early layers prefer small kernel, late layers prefer large kernel. In this paper, the GPU model and mobile model of ProxylessNAS are used for evaluation.

9) DARTS

Most network search algorithms use enhanced learning or evolutionary algorithms to search for structures. The search space of such algorithms is discrete, and the search time is too time consuming. The search space of DARTS is continuous, and the search process is completed by using a gradient descent algorithm on the verification set. The computational cost of DARTS is several orders of magnitude smaller than that of ordinary network search algorithms, but the result obtained by searching can still be equal to that of the previous SOTA algorithm. Meanwhile, its generalization ability is also very good. It can be used not only for searching CNN structure, but also for searching RNN structure. Fig. 11 shows an overview of DARTS.

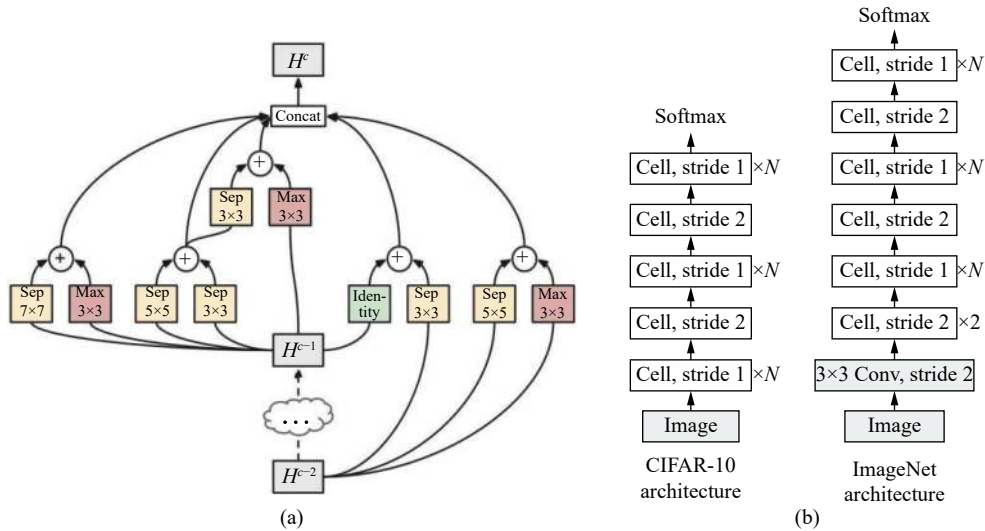


Fig. 7 Cell structure of PNASNet^[39]: (a) The best cell structure found by PNASNet; (b) Employing a similar strategy as [44] when constructing CNNs from cells on CIFAR-10 and ImageNet.

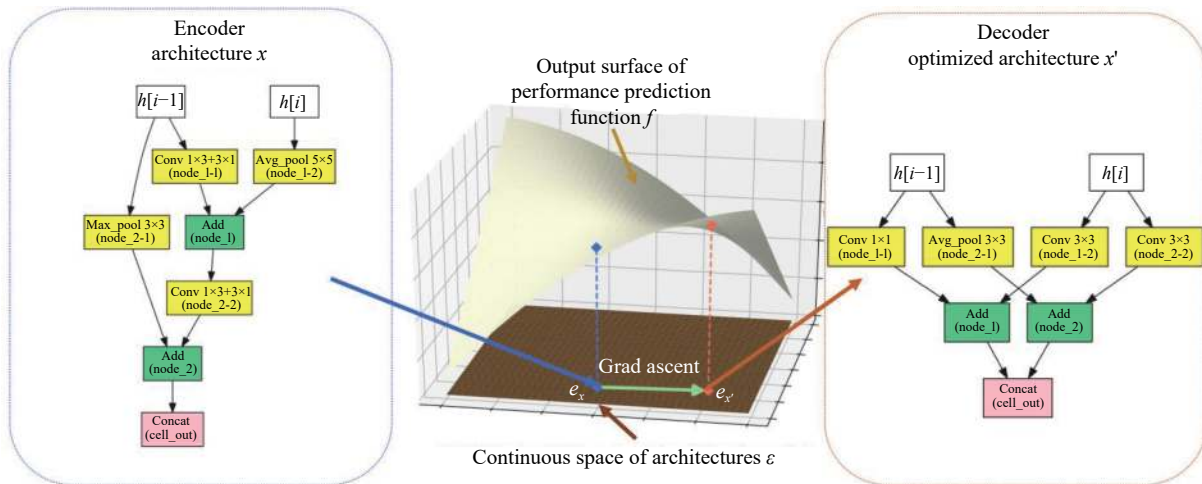


Fig. 8 General framework of NAONet^[40]

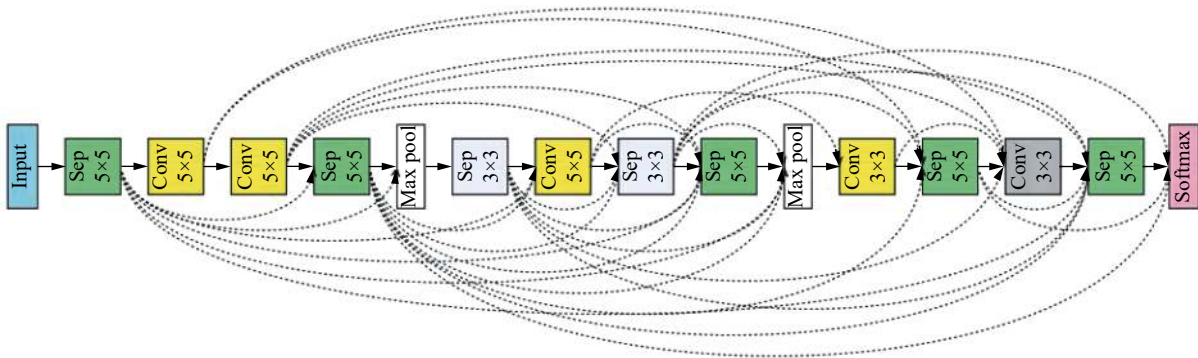


Fig. 9 ENAS's discovered network from the macro search space for image classification^[61]

10) FairNAS

FairNAS^[48] is a one-shot method in the field of NAS, and it advocates that weights can be shared. It trains a super-net from the beginning to the end (only one HYPERNET is trained completely, which is also the meaning of one-shot). Each model is a sampling model of HYPERNET. The advantage of this is that it does not need

time-consuming training for each model to know its representation ability. Therefore, it is famous for greatly improving the efficiency of NAS and has become the mainstream of NAS. Nevertheless, the premise of one-shot is that weight sharing is effective, and the model ability can be verified quickly and accurately in this way. This kind of situation is a little like the Matthew effect. If the con-

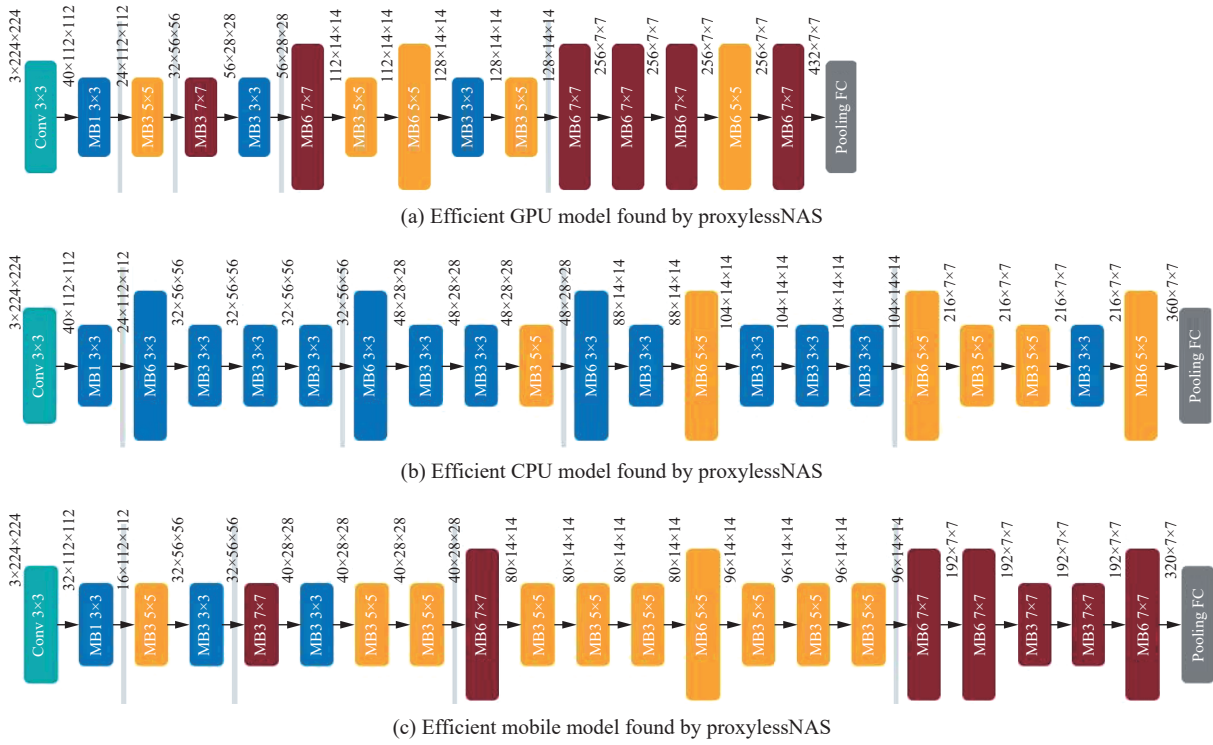


Fig. 10 Efficient models optimized for different hardware of the ProxylessNAS method [66]: (a) GPU model found by ProxylessNAS; (b) CPU model found by ProxylessNAS; (c) Mobile model found by ProxylessNAS.

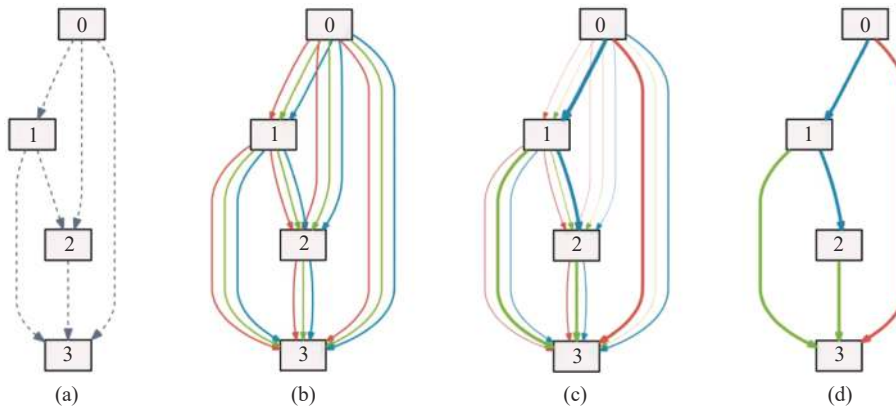


Fig. 11 An overview of DARTS[67]: (a) Operations on the edges are initially unknown; (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge; (c) Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem; (d) Inducing the final architecture from the learned mixing probabilities.

ditions are not good, they will fall into a circular dilemma. FairNAS believes that fair sampling and training methods can give full play to the potential of each module. Finally, after completing the training, the sampling model can quickly use the weights in the hypernet to get a relatively stable performance index on the verification set. This fair algorithm can almost completely maintain the ranking of the models, and the models sampled from the super-net and the models trained separately will eventually have almost the same ranking. FairNAS has three versions, including FairNAS-A, FairNAS-B and FairNAS-C, which have different search architectures by different search space. Fig. 12 shows architectures of FairNAS-A, B and C.

11) GDAS

GDAS[60] uses the gradient descent to search the network structure effectively, and the search space is represented by a directed acyclic graph (DAG). This DAG may have millions of sub-graphs, each of which is a neural network structure. In order to avoid traversing so many sub-graphs, Dong and Yang[60] use a differentiable sampler to sample the sample structure and optimizes the sampler by training the verification loss of the sampled structure. GDAS can search a robust neural network structure in 4 hours on a V100GPU. GDAS is similar to DARTS, but there are two differences between them: i) How to make the search space differentiable? DARTS transforms the weight softmax of operations into a probability after joint

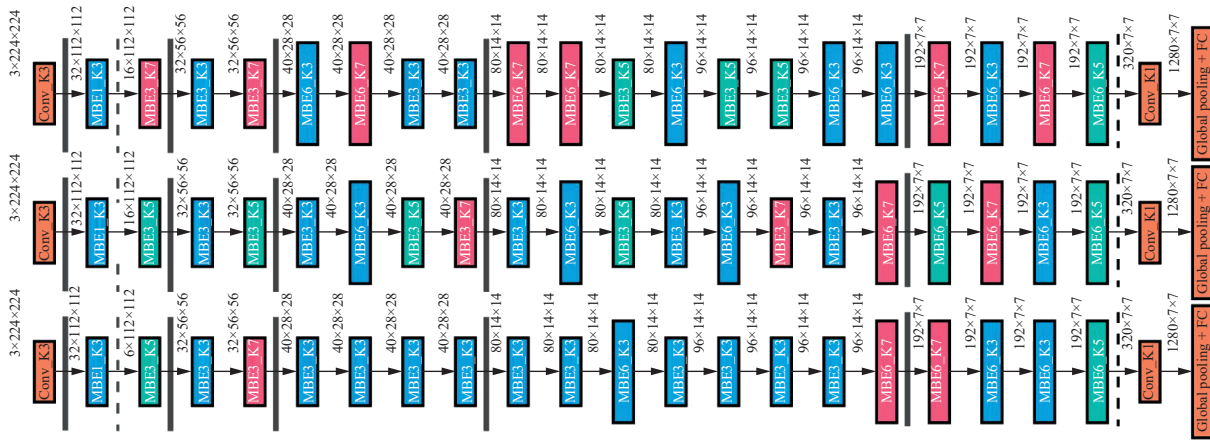


Fig. 12 Architectures of FairNAS-A, B and C (from top to bottom)[48]

optimization and takes the operation with the maximum probability of connection between nodes. Dong and Yang^[60] used the Gumbel max trick to select the transformation function between nodes with argmax function in forward propagation, and use softmax function to differentiate one hot vector in backward propagation and use gradient backpropagation. ii) DARTS jointly searches all operations, which will lead to antagonism between operations, and the weight that may be generated will offset each other, which makes the optimization difficult. Besides, the joint search of normal cell and reduction cell greatly increases the search space. In ^[60], the reduction cell is fixed, and the normal cell is searched. It only takes 4 hours on a V100GPU, and only the function between the sampled nodes is updated each time. Fig. 13 shows the search space of a neural cell using DAG. Different operations (colored arrows) transform one node (square) to its intermediate features (little circles). Meanwhile, each node is the sum of the intermediate features transformed from the previous nodes. As indicated by the solid connections, the neural cell in the proposed GDAS is a sampled sub-graph of this DAG. Specifically, among the intermediate features between every two nodes, GDAS samples one feature in a differentiable way.

12) FBNet-V1

In the method of FBNet-V1^[58], a differentiable neural architecture search (DNAS) is used to find the hardware related light-weight convolution network. The DNAS method represents the whole search space as a hypernetwork, transforms the search for the optimal network structure into finding the optimal candidate block distribution, trains the block distribution by gradient descent, and can select different blocks for each layer of the network. In order to better estimate the network delay, the actual delay of each candidate block is measured and recorded in advance, which can be accumulated directly according to the network structure and the corresponding delay. Fig. 14 shows the visualization of some of the searched architectures.

13) MNASNet

MNASNet^[55] is an automatic mobile neural network

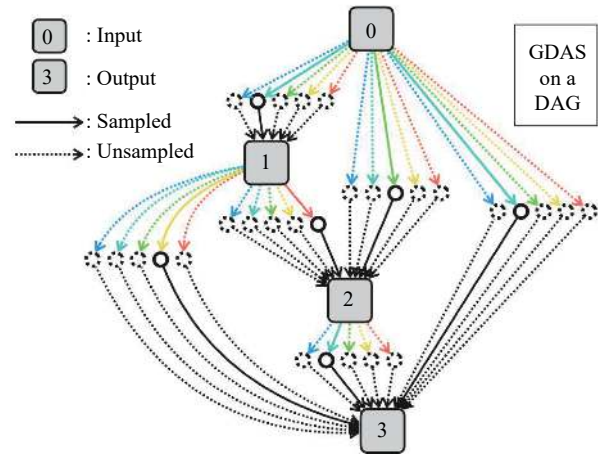


Fig. 13 Using a DAG to represent the search space of a neural cell^[60]

search method. MNASNet explicitly takes the model operation delay time as one of the main optimization objectives to search for a network model structure that can balance the latency and accuracy. In the previous work, the latency was measured indirectly by an inaccurate method, such as flops (floating-point operations per second). MNASNet can implement models on mobile devices to directly measure the inference delay in the real world. In addition, a hierarchical search space method is proposed to determine the network structure. The first inspiration of ^[55] comes from the fact that although there are similar flops (575M VS. 564M) in MobileNet and NASNet, the delay time is quite different (113ms VS. 183ms). Secondly, Tan et al.^[55] observed that the previous automation methods mainly search for several types of units and then stack the same units repeatedly through the network. This simple search mechanism will limit the diversity of layers. The first inspiration gave birth to the idea of multi-objective optimization of operation delay time and precision; the second inspiration gave birth to the method of hierarchical decomposition of search space. Allow layers to be architecturally different but still strike the right balance between flexibility and search space

size. Fig. 15 shows an overview of MNASNet.

14) ScarletNAS

In the method of ScarletNAS^[53], an automatic neural network search with a scalability function is proposed. The problem of the fairness of scalable hypernetwork training in one shot route is solved by a linear equivalent transformation. ScarletNAS uses conv1×1 (without bias/relu) + conv to replace identity + conv in training super net, which solves the problem of convergence in training scalable network. The introduction of conv1×1 (without bias/relu) + conv is a linear transformation, which is equivalent to identity + conv. In ImageNet 1K classification task, it achieves 76.9% top-1 accuracy and is currently SOTA of < 390M flops level. ScaletNAS has three different versions, including ScaletNAS-A, ScaletNAS-B and ScaletNAS-C, and their network complexity is gradually reduced. ScaletNAS-A usually gets the better result. Fig. 16 shows architectures of ScaletNAS-A, B and C.

15) MoGA

MoGA^[82] considers the use of mobile GPU in real scenes, and the model can directly serve mobile visual products. The first novel point of MoGA is mobile GPU aware (MoGA), which is to design the mobile GPU sensitive model from the perspective of practical use. The second point of view of MoGA comes from the analysis of MobileNet trilogy. From MobileNet-V1 to MobileNet-V3, the accuracy is constantly improving, but the number of model parameters is increasing. Therefore, the optimization of the model parameters is an aspect worth studying. In addition to the business indicator top-1 accuracy, the

running time of the model at the device side is regarded as the key indicator to measure the model, not the multiplier and adder, so the multiplier and adder are eliminated from the target. Also, the previous methods tried to compress the parameters, which is very disadvantageous to multi-objective optimization. On the Pareto boundary, which does not harm others but benefits oneself, one must give up to get something. It is considered that the parameter quantity is the representation of the model capability, so the model with high parameters but with low delay can be obtained by encouraging the increase of parameter quantity instead of increasing the search range. MoGA has three versions including MoGA-A, MoGA-B and MoGA-C, which have different search layers. Fig. 17 shows architectures of MoGA-A, B and C.

16) PC-DARTS

PC-DARTS^[84] is an effective channel sampling method in which only a part of the channel is sampled into the core of the multi-choice operation. Channel sampling can alleviate the “overfitting” phenomenon of hypernetwork, and greatly reduce its memory consumption so that the speed and stability of structure search can be improved by increasing the batch size in the training process. However, channel sampling will lead to the inconsistency of the edge selection of the hypernetwork, which increases the disturbance caused by random approximation. In order to solve this problem, an edge regularization method is proposed, which uses a set of additional edge weight parameters to reduce the uncertainty in search. After these two improvements, the search speed of the method is faster, the performance is more stable, and the

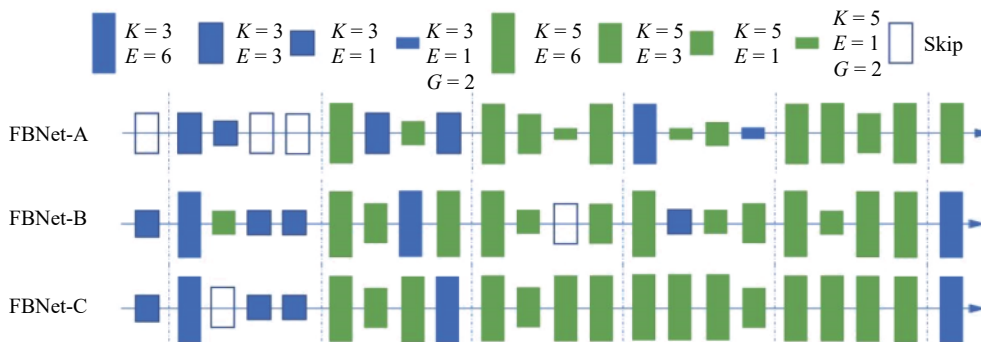


Fig. 14 Visualization of some of the searched architectures of FBNet-V1^[58]

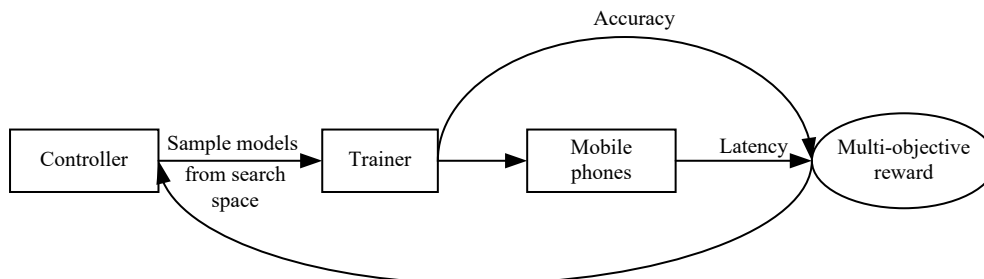


Fig. 15 An overview of MNASNet

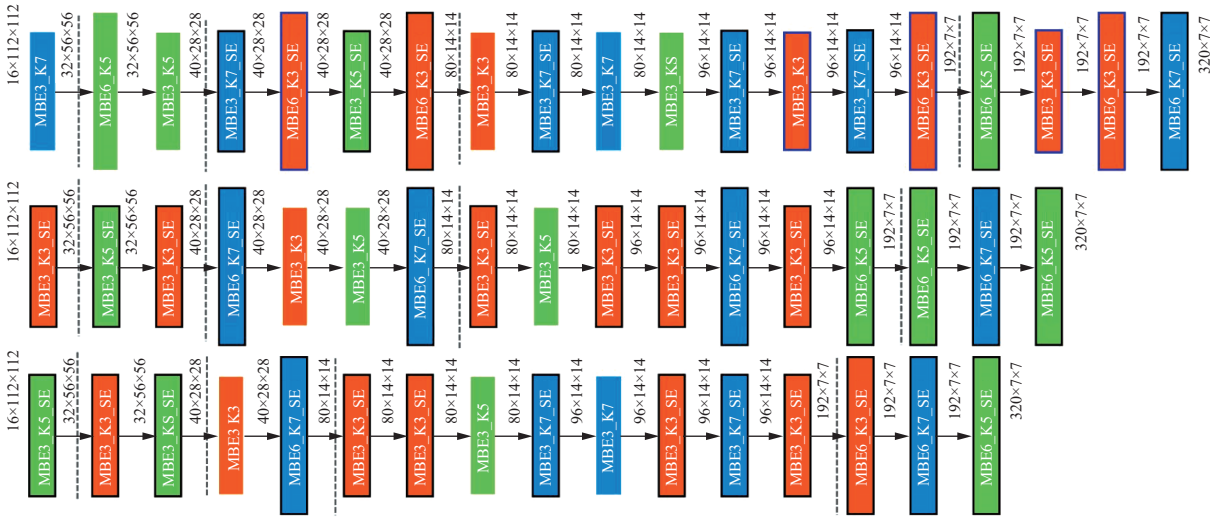


Fig. 16 Architectures of ScaletNAS-A, B and C (from top to bottom)^[53]

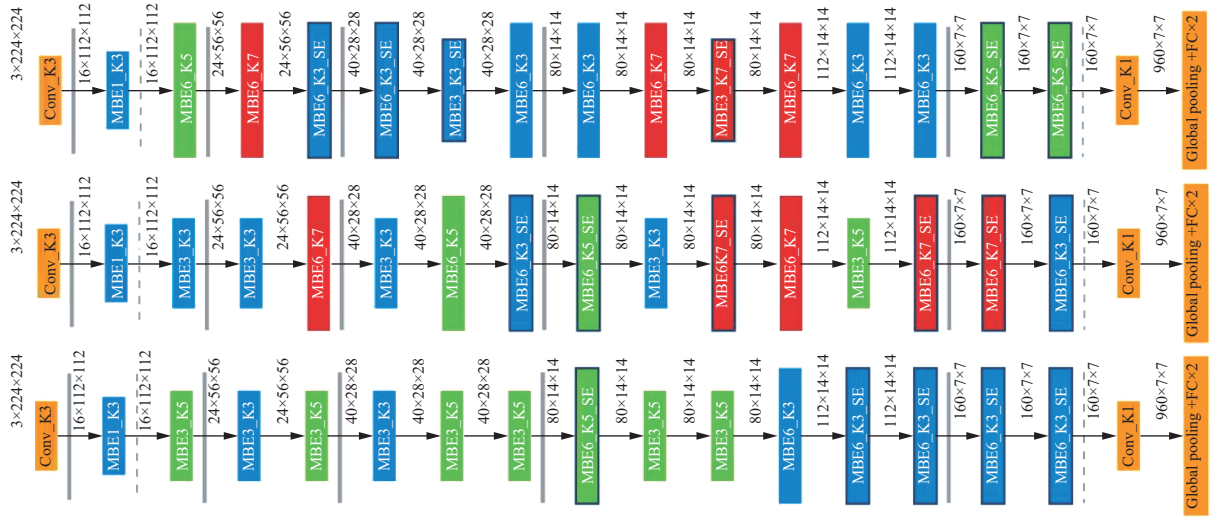


Fig. 17 Architectures of MoGA-A, B and C (from top to bottom)^[82]

accuracy is higher. Fig. 18 shows the general framework of PC-DARTS.

17) Single-Path-SuperNet

One-shot is a powerful neural architecture search framework, but its training is relatively complex, and it isn't easy to obtain competitive results on large data sets (such as ImageNet). Guo et al.[108] proposed a single path one-shot model called Single-Path-SuperNet to solve the main challenges in the training process. The core idea of Single-Path-SuperNet is to construct a simplified super network, which is trained according to the uniform path sampling method. All substructures (and their weights) are fully and equally trained. Based on the trained hyper-network, the optimal substructure can be quickly searched by an evolutionary algorithm, in which no fine-tuning of any substructure is required.

18) DNA

DNA^[90] is a method to solve the two problems: efficiency and effectiveness. It differs from the existing neural network search algorithm such as RL, DARTS, One-

Shot, etc. Based on the idea of knowledge distillation, Li et al.[90] introduce a teacher model to guide the direction of the network structure search. Using the supervision information from different depths of teacher model, the original end-to-end network search space is divided into blocks in-depth to realize the weight sharing training of independent blocks of network search space, which greatly reduces the interference caused by weight sharing. At the same time, it ensures the accuracy of the evaluation of candidate sub-models without sacrificing the efficiency of weight sharing. The algorithm can traverse all candidate structures in the search space. DNA has four versions, i.e., DNA-a, DNA-b, DNA-c and DNA-d, which are searched by different parameters. Fig. 19 shows an illustration of DNA. The teacher's previous feature map is used as input for both teacher and student block. Each cell of the super net is trained independently to mimic the behavior of the corresponding teacher block by minimizing the l2-distance between their output feature

maps. The dotted lines indicate randomly sampled paths in a cell.

19) FBNet-V2

There have been some classic NAS before the FBNet-V2 method was proposed, such as DARTS, ProxylessNAS and so on. However, all of them have their own defects, such as: i) Memory loss limits the size of search space; ii) Memory loss is linearly related to the number of operations per layer; iii) ProxylessNAS can effectively reduce memory loss by using binary training method, but in large search space, the memory loss is not limited, and the convergence rate is very slow. FBNet-V2^[93] can greatly expand the search space without increasing memory loss and can maintain high-speed search in large search space. The main contributions of FBNet-V2 are as follows: i) A NAS with both memory and efficiency is proposed; ii) A masking mechanism and an effective shape propagation for feature map reuse are proposed; iii) The precision of the network is very high.

20) RobNet

In order to improve the robustness of the deep neural network, the existing work focuses on the study of confrontation learning algorithm or loss function to enhance the network robustness. From the perspective of neural network structure, RobNet^[91] studies the model of neural network structure, which can resist attack. In order to obtain a large number of networks needed in this study, one-shot neural network structure search is used to train a supernet, and then the sub-networks sampled from it are adjusted against fine-tuning. The sampling network structure and its robustness accuracy provide a rich basis for the research.

5 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 exploited for performance evaluation, including PolyU II,

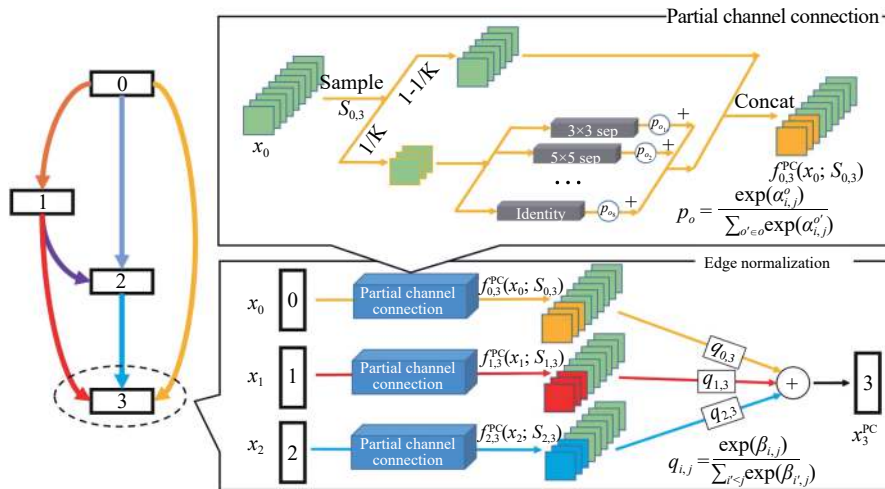


Fig. 18 Overall framework of PC-DART: The upper part is the partial channel connection and the lower part is the edge regularization^[84].

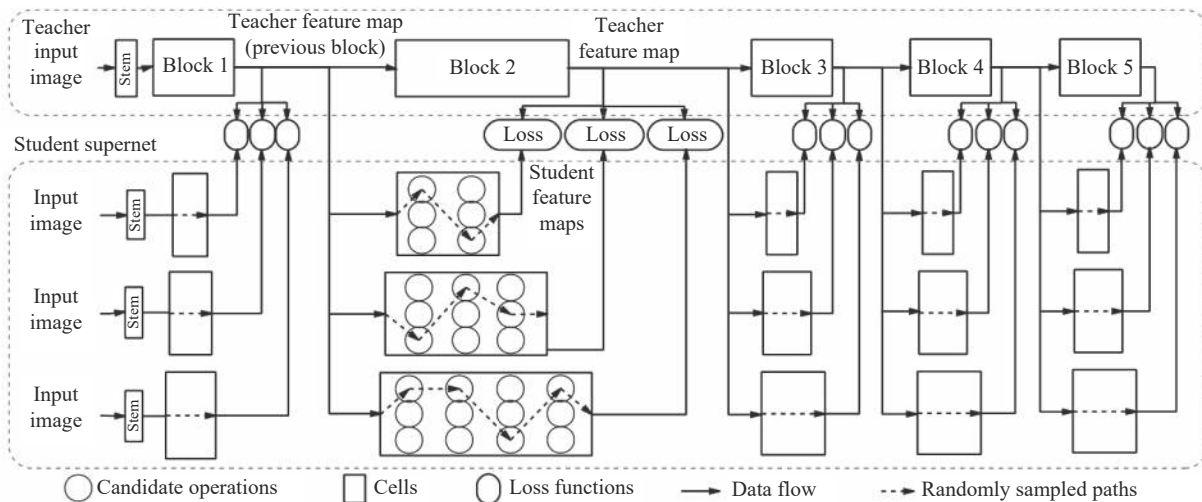


Fig. 19 Illustration of the DNA method^[90]

PolyU M_B, HFUT, HFUT CS, TJU-P, PolyU 3D, PolyU M_N and TJU-PV. After preprocessing, the ROI sub-images were cropped. The ROI size of all databases is 128×128 . The detailed descriptions of the above databases are listed in Table 4. Figs. 20–23 depict some ROI images of four 2D palmprint databases. In Figs. 20–23, the three images depicted in the first row were captured in the first session. The three images depicted in the second row were captured in the second session. Fig. 24 shows three original palmprints of the HFUT CS database and their corresponding ROI images. Figs. 25 and 26 depict some ROI images of two 2D palm vein databases. In Figs. 25 and 26, the three images depicted in the first row were captured in the first session. The three images depicted in the second row were captured in the second session. Fig. 1 shows one original 3D palmprint data, and four different 2D representations from one 3D palmprint, including MCI, GCI, ST and CST.

PolyU II is a challenging palmprint database because the illumination between the first and second sessions has a notorious change. HFUT CS is also a challenging palmprint database. From Fig. 24, it can be seen that there are some differences between the palmprints captured by different devices.

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	7 752
PolyU M_B	2D palmprint	Yes	250	500	2	9 days	6×2	6 000
HFUT	2D palmprint	Yes	400	800	2	10 days	10×2	16 000
HFUT CS	2D palmprint	No	100	200	2	10 days	$10 \times 2 \times 3$	12 000
TJU-P	2D palmprint	No	300	600	2	61 days	10×2	12 000
PolyU 3D	3D palmprint	Yes	200	400	2	1 months	10×2	8 000
PolyU M_N	Palm vein	Yes	250	500	2	9 days	6×2	6 000
TJU-PV	Palm vein	No	300	600	2	61 days	10×2	12 000

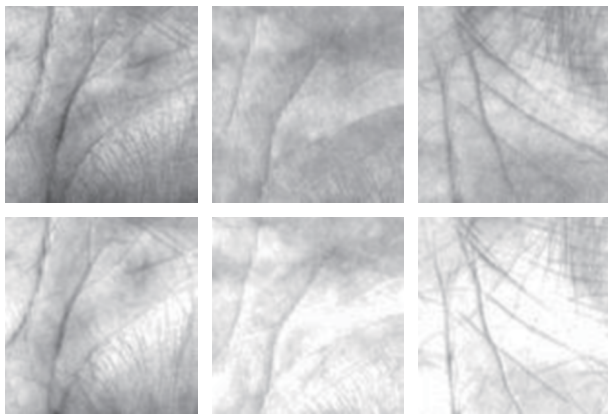


Fig. 20 Six palmprint ROI images of PolyU II database. The three images in the first row were captured in the first session. The three images in the second row were captured in the second session.

6 Experimental results

6.1 Experimental configuration

We selected twenty NAS methods for performance evaluation. As some of the selected methods have different versions, we selected various versions for evaluation, including (NASNet-A, NASNet-mobile), (ProxylessNAS, ProxylessNAS-mobile), (FairNAS-A, B, and C), (ScaletNAS-A, B and C), (MoGA-A, B and C) and (DNA-a, b, c and d).

Here, we introduce the default configuration of the experiment, including experimental hyperparameters and hardware configuration. Since different networks need different input sizes, the palmprint and palm vein ROI image need to be up-sampled to a suitable size before inputting into the network. In order to enhance the stability of the network, we also added a random flip operation (only during the training phase); that is, for a training image, there is a certain probability that the image is flipped horizontally and then input into the network. We did not initialize the model parameters using the random parameter initialization method; instead, we initialized it us-

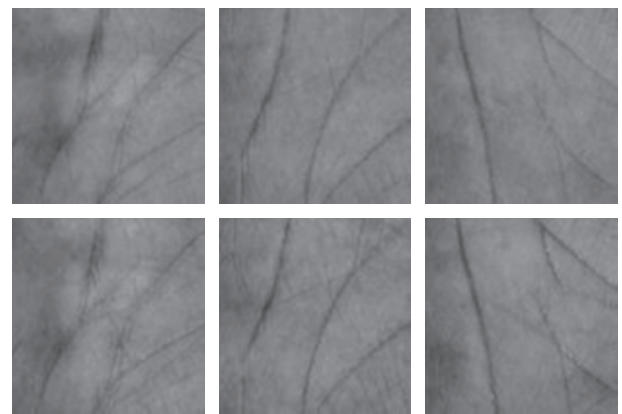


Fig. 21 Six palmprint ROI images of the PolyU M_B database. The three images in the first row were captured in the first session. The three images in the second row were captured in the second session.

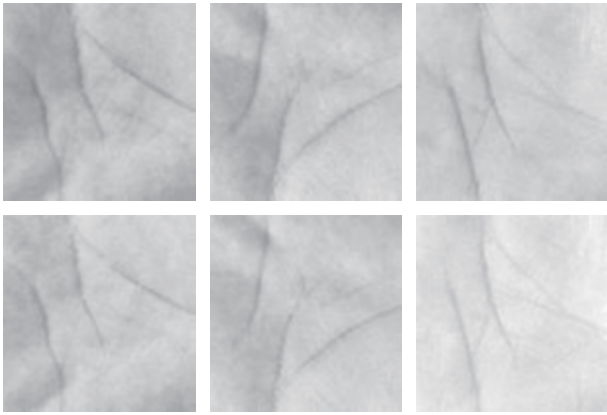


Fig. 22 Six palmprint ROI images of the HFUT database. The three images in the first row were captured in the first session. The three images in the second row were captured in the second session.

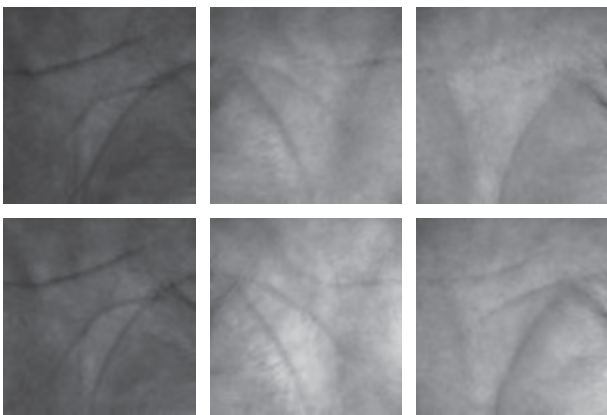


Fig. 23 Six palmprint ROI images of the TJU-P database. The three images in the first row were captured in the first session. The three images in the second row were captured in the second session.



Fig. 24 Three original palmprint and ROI images of the HFUT CS database. The three images in the first row are original palmprint images. The three images in the second row are the corresponding ROI images.

ing the parameters of the pretrained model in the ImageNet or CIFAR dataset. It is worth noting that when an official model is trained on the ImageNet dataset, we prefer to use the pre-trained model. If not, we will use the pre-trained model of the CIFAR dataset. The palmprint

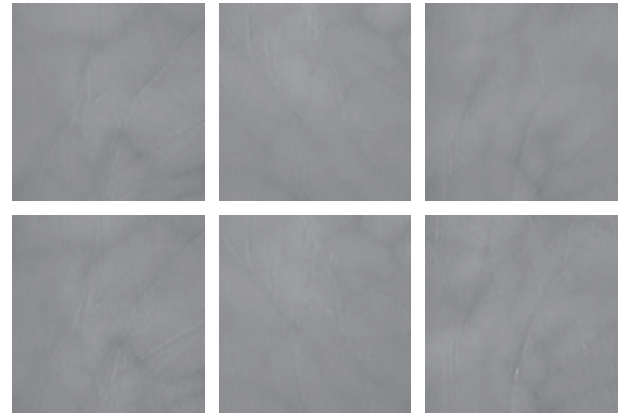


Fig. 25 Six palm vein ROI images of the PolyU M_N database. The three images in the first row were captured in the first session. The three images in the second row were captured in the second session.

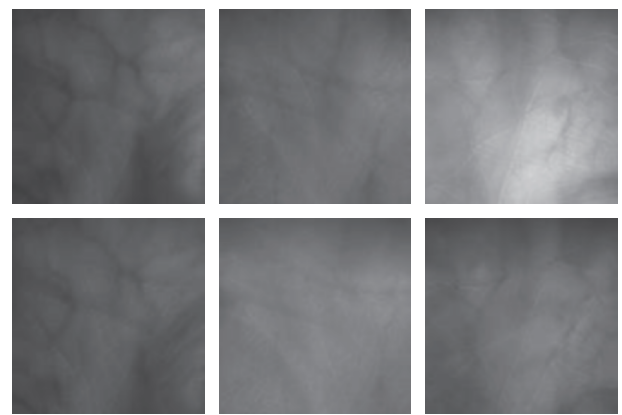


Fig. 26 Six palm vein ROI images of the TJU-PV database. The three images in the first row were captured in the first session. The three images in the second row were captured in the second session.

and palm vein ROI image in the database is usually a grayscale image, which means that the number of image channels is 1. The input of the model is an RGB image, so the grayscale channel of the image is copied three times to form an RGB image with 3 channels.

The system configuration is as follows: Intel CPU i7-8700 3.20GHz, NVIDIA GPU GTX 2080, 16GB memory and Windows 10 operating system. All evaluation experiments were performed on Pytorch. The cross-entropy loss function, Adam optimizer, was used by default. The batch size was set to 4, and the learning rate to 5×10^{-5} .

6.2 Performance measures

In this paper, both identification and verification experiments are conducted.

Identification is a one-to-many comparison, which answers the question of “who the person is?”. In this paper, the close-set identification is conducted. That is, we know all enrollments exist in the training set. In order to obtain identification accuracy, the rank 1 identification rate

is used, in which a test image will be matched with all templates in the training set, and the label of the most similar template will be assigned to this test image. For the sake of simplicity, we define the rank 1 identification rate as the accuracy recognition rate (ARR).

Verification is a one-to-one comparison, which answers the question of “whether the person is whom they claim to be”. In the verification experiments, the statistical value of equal error rate (EER) is adopted to evaluate the performance of different methods.

6.3 Recognition performance on separate data mode

We first conduct evaluation experiments on separate data mode. That is, in all databases, all samples captured in the first session are used for training, and all samples captured in the second session are used for the test. We conduct the experiments using all selected NAS methods on all databases. The values of ARR and EER of the selected NAS methods obtained from 2D palmprint and palm vein databases are listed in Tables 5 and 6, respectively. The values of ARR and EER obtained from four 2D representations (CST, ST, MCI and GCI) of 3D palmprint databases are listed in Table 7.

From Tables 5–7, we have the following observations:

1) ProxylessNAS achieves the best recognition results on most databases. In the palm vein databases PolyU M_N and FairNAS-A, it achieves the best recognition results, whose ARR and EER are 100% and 0.0001%, respectively.

2) ProxylessNAS was proposed in 2019, but its recognition performance is better than those methods proposed in 2020. This shows that the recognition performance of the latest NAS recognition methods is not necessarily better than that of the old NAS methods. For example, the recognition performance of NASNet proposed in 2016 is better than that of some methods proposed in 2019 and 2020, such as DARTS, ENAS, and RobNet, etc.

3) PolyU II is a challenging database because the samples captured in the first session and the second session have some noticeable variations, such as illumination change. In this database, the highest ARR is obtained by the proxylessNAS method, which is 98.63%. However, this is an unsatisfactory recognition result. It also shows that it is necessary to further study the new NAS methods to improve the results of 2D palmprint recognition.

4) HFUT CS is a cross-sensors database and is also a challenging database. In this database, the highest ARR is obtained by the proxylessNAS method, which is 99.78%. It is a promising result, and it also shows that cross-sensor palmprint recognition based on NAS technology is worthy of attention in the future.

5) For 3D palmprint recognition, the method of ProxylessNAS achieves 100% ARR and 0.0057% EER on

MCI representation, which is a very encouraging result. This result shows that NAS technology is up-and-coming in 3D palmprint recognition, which deserves further study. Among four 2D representations of 3D palmprint, MCI is most suitable for 3D palmprint recognition based on NAS technology.

6.4 Recognition performance on mixed data mode

In this section, we conduct evaluation experiments on the mixed data 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. The values of ARR and EER of the selected NAS methods obtained from 2D palmprint and palm vein databases are listed in Tables 8 and 9, respectively. The values of ARR and EER obtained from 3D palmprint databases are listed in Table 10.

From Tables 8–10, we have the following observations:

1) The recognition results of all the NAS methods obtained in “mixed data mode” is better than that of “separate data mode”. Although we only added one image of the second session to the training set, the recognition results of all NAS methods are still greatly improved. This experiment shows that the prediction accuracy of deep learning-based methods will be improved after more data are obtained. We can infer that if the network can be trained by the data collected from multiple stages, the recognition results of all methods can be significantly improved.

2) In the “mixed data mode”, the three methods with the best recognition performance are ProxylessNAS, ProxylessNAS-Mobile and Scarlet-A. The ProxylessNAS method achieves the best recognition results in most databases. On all databases, the ARR of ProxylessNAS is 100%, and the EER is very low.

3) In the “mixed data mode”, the convergence speed of the neural network is usually faster, and the number of epochs trained is reduced by nearly half.

6.5 Performance comparison with other methods

For 2D and 3D palmprint and palm vein recognition, we compare the performance between NAS methods and other methods, including four traditional methods and four deep learning methods.

For NAS methods, we select ProxylessNAS and FairNAS-A for performance comparison. Among different NAS method, the overall performance of ProxylessNAS is the best, and FairNAS-A achieves the best recognition performance in the PolyU M_N database.

Four traditional and representative palmprint recogni-

Table 5 ARR (%) of different NAS methods on different 2D palmprint and palm vein databases under the separate data mode

	PolyU II	PolyU M_B	HFUT I	HFUT CS	TJU-P	PolyU M_N	TJU-PV
NASNet	94.46	99.10	94.47	97.73	97.68	99.47	94.25
NASNet-Mobile	93.04	97.97	93.09	99.02	97.77	99.23	94.77
SMASH	87.50	95.68	92.44	81.29	93.16	96.29	91.03
PNASNet	95.62	99.37	97.91	99.48	99.25	99.63	97.58
NAONet	79.99	94.56	88.53	71.21	85.33	92.72	84.07
SNAS	88.77	96.17	90.66	94.03	93.78	95.55	92.12
AmoebaNet	72.58	93.93	81.85	85.17	77.75	95.47	85.33
ENAS	90.77	98.48	90.35	86.75	91.63	97.46	90.28
ProxylessNAS	98.63	100	99.68	99.78	99.75	99.89	99.67
ProxylessNAS-Mobile	98.56	99.86	98.98	99.35	99.30	99.86	99.53
DARTS	91.55	98.75	93.74	92.07	97.47	98.36	96.18
FairNAS-A	96.53	100	98.85	99.39	99.55	100	99.23
FairNAS-B	97.71	99.95	98.59	97.19	99.22	99.90	99.10
FairNAS-C	97.09	99.93	98.86	96.88	99.28	99.97	99.27
GDAS	67.45	87.30	84.00	83.81	87.83	88.58	89.50
FBNet-V1	90.31	97.80	93.34	96.83	90.08	96.45	89.78
MNASNet	94.05	99.87	95.38	98.81	96.07	99.47	95.62
Scarlet-A	98.18	99.70	98.74	98.70	99.69	99.70	99.56
Scarlet-B	98.41	99.93	98.66	98.01	99.58	99.83	99.01
Scarlet-C	97.25	99.95	98.68	97.59	99.42	99.93	98.95
MoGA-A	94.19	99.43	95.85	96.54	97.22	99.83	96.27
MoGA-B	95.33	99.60	95.77	98.29	97.98	99.77	97.03
MoGA-C	94.09	98.83	95.99	97.00	97.28	99.50	96.89
PC-DARTS	76.39	94.30	83.45	67.88	82.48	87.83	86.15
Single-Path-SuperNet	92.39	99.30	91.35	93.55	94.05	99.20	94.68
DNA-a	93.50	98.90	92.20	94.96	94.30	98.33	93.57
DNA-b	94.53	99.33	92.33	98.4	94.47	98.50	96.02
DNA-c	94.76	99.27	92.97	98.55	94.88	98.67	93.58
DNA-d	95.27	99.43	93.41	99.19	95.14	99.37	94.45
FBNet-V2	92.60	99.77	95.04	97.21	98.52	97.53	94.17
RobNet	83.86	95.57	88.89	91.27	90.48	94.71	91.95

tion methods are selected for performance comparison, including competitive code (CompC)^[16], ordinal code (OrdinalC)^[112], robust line orientation code (RLOC)^[113] and local line directional pattern (LLDP)^[114].

Four deep learning methods are selected for performance comparison including PalmNet^[24], ResNet^[115], MobileNet-V3, and EfficientNet. PalmNet is a deep learning method specially designed for palmprint recognition, which has excellent recognition performance. ResNet is a very famous CNN and is also a typical representative of manually designed CNN. As we have mentioned above, MobileNet-V3 and EfficientNet are two semi-NAS methods and have excellent recognition performance. The recognition performance of MobileNet-V3 and EfficientNet has been evaluated in our previous work.

6.5.1 Performance comparison with other methods in the separate data mode

In the “separate data mode”, for all databases, traditional methods use four images captured in the first session as the training data and uses the images collected in the second session as the test data. For deep learning-based methods including PalmNet, ResNet, MobileNet-V3, EfficientNet. ProxylessNAS and FairNAS-A, all images collected in the first session are used as the training data, and the remaining second session images are used as test data. The comparison results on 2D palmprint and palm vein databases are listed in Table 11, and the comparison results on 3D palmprint database are listed in Table 12.

From Tables 11 and 12, we have the following obser-

Table 6 EER (%) of different NAS methods on different 2D palmprint and palm vein databases under the separate data mode

	PolyU	PolyU M_B	HFUT I	HFUT CS	TJU-P	PolyU M_N	TJU-PV
NASNet	2.439 3	0.055 8	2.932 9	0.265 7	0.250 5	0.103 8	2.638 6
NASNet-Mobile	2.656 0	0.143 1	2.930 8	0.156 3	0.090 7	0.114 5	2.650 3
SMASH	6.644 3	1.657 3	3.117 0	8.587 6	2.629 1	1.830 2	3.117 6
PNASNet	1.074 6	0.072 7	0.457 5	0.108 0	0.023 5	0.020 6	0.177 1
NAONet	10.284 4	2.640 6	5.735 1	16.776 5	7.847 3	3.678 4	7.042 1
SNAS	5.159 2	1.305 2	4.398 8	2.590 5	2.536 4	1.445 3	3.062 8
AmoebaNet	15.908 2	2.685 0	8.780 4	7.589 4	11.317 1	1.896 6	7.568 0
ENAS	4.836 4	0.105 7	4.540 7	6.334 0	3.872 1	0.158 2	4.209 3
ProxylessNAS	0.172 8	3.34×10⁻⁵	0.040 7	0.021 7	0.018 7	0.001 1	0.070 4
ProxylessNAS-Mobile	0.118 2	0.011 0	0.112 0	0.075 3	0.059 3	0.001 2	0.069 0
DARTS	3.574 5	0.106 9	2.034 8	3.448 3	0.246 8	0.132 3	1.334 7
FairNAS-A	1.454 2	0.000 1	0.105 7	0.048 2	0.013 2	0.000 1	0.064 7
FairNAS-B	0.187 6	0.000 3	0.215 2	0.469 3	0.040 8	0.022 2	0.181 9
FairNAS-C	0.606 5	0.000 8	0.258 4	1.356 7	0.013 9	0.001 5	0.072 0
GDAS	17.577 1	6.072 5	7.418 6	8.434 4	6.419 3	5.364 4	6.478 3
FBNet-V1	4.733 9	0.195 1	2.802 9	1.378 2	4.138 2	1.789 0	5.343 4
MNASNet	2.499 2	0.000 4	1.506 3	0.203 4	1.281 8	0.019 5	1.371 1
Scarlet-A	0.132 3	0.002 7	0.171 9	0.129 7	0.002 5	0.021 1	0.071 9
Scarlet-B	0.114 6	0.000 9	0.183 8	0.423 8	0.017 1	0.019 2	0.102 3
Scarlet-C	0.705 6	0.000 1	0.238 6	0.351 8	0.016 6	0.000 8	0.080 9
MoGA-A	2.159 2	0.020 3	1.792 4	1.447 9	0.234 3	0.000 5	1.456 3
MoGA-B	1.106 8	0.018 1	1.581 0	0.209 3	0.177 8	0.018 9	0.198 3
MoGA-C	2.565 7	0.123 9	1.520 4	0.564 2	0.184 6	0.021 0	1.237 6
PC-DARTS	12.749 0	2.633 4	7.546 1	20.590 4	8.424 2	6.568 2	6.445 8
Single-Path-SuperNet	3.746 5	0.021 7	3.282 2	2.109 8	2.528 9	0.069 7	2.469 7
DNA-a	2.261 9	0.068 5	3.130 3	2.447 1	2.578 2	0.282 1	2.319 8
DNA-b	2.847 4	0.037 8	3.260 3	0.213 1	2.520 3	0.232 2	1.475 6
DNA-c	2.304 2	0.039 1	3.023 9	0.220 0	2.420 5	0.176 8	2.223 9
DNA-d	1.760 3	0.052 9	3.951 8	0.151 8	1.345 7	0.086 1	2.214 9
FBNet-V2	3.769 9	0.001 3	1.665 6	0.229 4	0.072 4	0.292 9	2.156 0
RobNet	9.879 7	1.396 1	5.943 9	3.690 2	4.024 5	2.089 5	3.578 9

variations:

1) In the PolyU II palmprint database, four traditional methods (CompC, OrdinalC, RLOC, and LLDP) and one manually designed deep learning method (PalmNet) achieve better recognition performance than the NAS methods. In the PolyU M_B palmprint database, all methods can achieve 100% ARR, and the method of ProxylessNAS achieves the lowest EER, which is $3.34 \times 10^{-5}\%$. In the HFUT I palmprint database, the method of PalmNet achieves the highest ARR (100%), and the method of ProxylessNAS achieves the lowest EER (0.0407%). In the HFUT CS palmprint database, the method of PalmNet achieves the highest ARR (100%), and the method of EfficientNet achieves the low-

est EER (0.0217%). In the PolyU M_N palm vein database, most methods can achieve 100% ARR, and RLOC and FairNAS-A achieve the lowest EER (0.0001%). In the TJU-PV palm vein database, the method of RLOC achieves the highest ARR (100%), and the method of FairNAS-A achieves the lowest EER (0.0647%).

2) For 2D palmprint and palm vein recognition, the overall recognition performance of NAS methods is close to that of the traditional methods. On some databases, traditional recognition methods have better recognition performance. On other databases, NAS methods have better recognition performance.

3) For 2D palmprint and palm vein recognition, the overall recognition performance of the NAS methods is

Table 7 ARR (%) and EER (%) of different NAS methods on four 2D representations of 3D palmprint databases under the separate data mode

	CST		ST		MCI		GCI	
	ARR	EER	ARR	EER	ARR	EER	ARR	EER
NASNet	97.62	1.087 6	97.88	1.067 4	97.50	1.569 8	90.76	4.998 0
NASNet-Mobile	97.78	1.003 4	98.40	0.492 7	98.98	0.354 5	91.65	4.367 4
SMASH	96.76	1.789 4	97.44	1.566 5	96.36	1.891 3	93.28	3.000 1
PNASNet	97.88	1.002 1	99.95	0.001 8	99.25	0.039 3	97.30	1.528 9
NAONet	73.76	15.443 8	68.54	18.445 0	79.33	10.775 6	69.38	16.298 0
SNAS	95.67	2.587 9	94.78	2.890 5	93.25	3.265 5	91.55	4.388 1
AmoebaNet	88.05	5.689 0	92.70	3.762 0	91.88	4.197 8	86.55	6.834 5
ENAS	97.33	1.335 7	97.67	1.302 5	93.54	3.778 0	92.10	3.018 6
ProxylessNAS	99.90	0.002 8	99.92	0.002 0	100	0.005 7	99.45	0.119 3
ProxylessNAS-Mobile	99.08	0.054 4	99.62	0.015 8	99.92	0.020 4	98.97	0.125 6
DARTS	97.17	1.156 4	96.07	1.870 3	95.73	2.665 4	91.25	4.403 5
FairNAS-A	99.58	0.043 6	99.40	0.049 2	99.25	0.058 7	94.05	3.279 0
FairNAS-B	99.62	0.033 6	99.90	1.322 5	99.52	0.023 4	94.53	2.467 4
FairNAS-C	98.25	0.106 9	98.98	0.063 2	99.75	0.003 7	96.27	1.734 5
GDAS	93.12	3.978 8	85.20	7.588 7	83.75	8.665 5	82.02	8.779 8
FBNet-V1	96.03	1.378 9	96.60	1.223 7	95.40	2.787 7	89.76	5.667 8
MNASNet	98.75	0.188 2	99.12	0.043 0	97.95	1.209 2	95.23	2.748 9
Scarlet-A	99.30	0.053 0	99.78	0.002 3	99.88	0.002 6	97.20	1.201 8
Scarlet-B	98.97	0.086 4	99.00	0.045 6	98.90	0.175 1	98.47	0.115 9
Scarlet-C	97.78	1.001 0	99.22	0.074 7	98.97	0.105 5	93.56	3.276 6
MoGA-A	98.90	0.072 0	97.60	0.083 7	99.35	0.145 1	94.30	2.987 6
MoGA-B	99.12	0.020 8	99.00	0.062 4	98.52	0.701 6	94.15	2.991 5
MoGA-C	97.88	1.000 3	98.20	0.108 9	96.70	1.589 0	91.67	4.287 6
PC-DARTS	91.45	4.865 8	92.48	3.564 4	90.67	4.171 7	90.22	4.778 0
Single-Path-SuperNet	94.75	2.889 0	94.53	2.906 7	89.55	5.389 0	89.85	5.772 3
DNA-a	93.53	3.676 1	94.64	2.287 6	92.07	3.880 1	86.45	6.899 0
DNA-b	93.88	3.548 8	95.58	2.056 7	93.03	3.380 9	87.07	6.786 5
DNA-c	95.50	2.865 0	96.47	1.447 9	93.53	3.000 2	87.86	6.398 7
DNA-d	94.33	2.991 4	96.62	1.387 5	94.67	2.855 0	88.33	5.686 8
FBNet-V2	97.05	1.787 7	98.65	0.123 1	97.47	1.010 2	90.33	4.355 0
RobNet	91.44	4.356 9	90.46	4.139 0	87.35	6.890 3	86.89	6.774 3

close to the deep learning-based method, i.e., PalmNet, a method specially designed for palmprint recognition. In the databases of PolyU II, PolyU M_B, HFUT I and TJU-P, PalmNet has better recognition performance than the NAS methods. However, in the database of HFUT CS, the recognition performance of PalmNet is very poor.

4) For 2D palmprint and palm vein recognition, the overall recognition performance of the NAS methods is notoriously better than that of one representative manually designed CNN method, i.e., ResNet.

5) For 2D palmprint and palm vein recognition, the overall recognition performance of one pure NAS method,

i.e., ProxylessNAS, is slightly better than EfficientNet and MobileNet-V3.

6) For 3D palmprint recognition, the method of ProxylessNAS achieves 100% ARR on the MCI representation, which is better than other methods.

6.5.2 Performance comparison with other methods in the mixed data mode

In 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 ProxylessNAS and FairNAS-A, all the images

Table 8 ARR (%) of different NAS methods on different 2D palmprint and palm vein databases under the mixed data mode

	PolyU	PolyU M_B	HFUT I	HFUT CS	TJU-P	PolyU M_N	TJU-PV
NASNet	98.26	100	99.13	100	100	100	98.55
NASNet-Mobile	98.11	100	98.11	100	100	100	98.68
SMASH	93.76	99.35	97.77	87.77	98.77	99.93	96.93
PNASNet	99.63	100	100	100	100	100	100
NAONet	86.35	98.78	93.08	78.95	91.22	97.88	89.89
SNAS	94.58	99.76	96.02	99.44	98.37	99.90	97.68
AmoebaNet	78.43	98.56	86.34	90.17	84.50	99.79	90.77
ENAS	96.98	100	96.01	92.37	96.68	100	95.87
ProxylessNAS	100	100	100	100	100	100	100
ProxylessNAS-Mobile	100	100	100	100	100	100	100
DARTS	97.25	100	98.67	97.23	100	100	99.97
FairNAS-A	99.76	100	100	100	100	100	100
FairNAS-B	100	100	100	100	100	100	100
FairNAS-C	100	100	100	99.95	100	100	100
GDAS	74.88	94.21	89.67	89.02	93.32	93.43	94.78
FBNet-V1	96.75	100	98.34	99.33	96.77	99.86	95.49
MNASNet	98.87	100	99.22	100	99.58	100	99.80
Scarlet-A	100	100	100	100	100	100	100
Scarlet-B	100	100	100	100	100	100	100
Scarlet-C	100	100	100	100	100	100	100
MoGA-A	98.66	100	99.45	99.86	100	100	100
MoGA-B	99.41	100	99.76	100	100	100	100
MoGA-C	98.38	100	99.86	100	100	100	99.98
PC-DARTS	83.75	98.80	89.62	73.46	88.40	93.45	92.35
Single-Path-SuperNet	97.60	100	96.47	98.90	99.08	100	98.98
DNA-a	98.22	100	98.36	99.88	99.07	100	98.12
DNA-b	99.26	100	98.55	100	99.14	100	99.85
DNA-c	99.31	100	98.67	100	99.54	100	98.54
DNA-d	99.79	100	98.90	100	99.87	100	98.69
FBNet-V2	97.44	100	99.66	100	100	100	98.72
RobNet	90.25	99.10	94.33	96.94	95.60	98.86	96.80

collected in the first session are used as the training data. Then 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. The comparison results on the 2D palmprint and palm vein databases are listed in Table 13, and the comparison results on the 3D palmprint database are listed in Table 14.

From Tables 13 and 14, we have the following observations:

1) In the mixed data mode, all methods have achieved outstanding recognition performance. Almost all methods can obtain 100% ARR and very low EER. That is to say, for various methods, it is very easy to obtain high recognition performance by using the mixed data mode.

2) In the mixed data mode, for 2D palmprint and

palm vein recognition, the overall recognition performance of NAS methods is close to that of traditional methods and other deep learning-based methods.

3) In the mixed data mode, for 3D palmprint recognition, the method of ProxylessNAS achieves the best performance on the MCI representation.

7 Conclusions

This paper systematically investigated the recognition performance of representative NAS methods for 2D and 3D palmprint recognition and palm vein recognition. Twenty representative NAS methods were exploited for performance evaluation, including NASNet, SMASH, PNASNet, NAONet, SNAS, AmoebaNet, ENAS, Proxy-

Table 9 EER (%) of different NAS methods on different 2D palmprint and palm vein databases under the mixed data mode

	PolyU	PolyU M_B	HFUT I	HFUT CS	TJU-P	PolyU M_N	TJU-PV
NASNet	0.132 5	0.001 3	0.857 3	0.010 1	0.007 8	0.006 0	0.154 3
NASNet-Mobile	0.147 8	0.001 8	0.147 6	0.009 6	0.008 4	0.005 4	0.144 8
SMASH	2.672 1	0.047 9	0.976 0	6.172 5	0.132 1	0.014 4	1.455 0
PNASNet	0.032 2	0.002 5	0.008 8	0.004 6	0.003 3	0.002 1	0.007 7
NAONet	6.643 8	0.578 8	2.402 1	12.445 2	3.276 8	0.709 4	5.233 6
SNAS	2.287 6	0.038 6	1.489 3	0.037 8	0.176 8	0.017 7	0.744 6
AmoebaNet	11.654 8	0.723 6	7.487 6	4.527 6	9.346 2	0.023 3	4.557 3
ENAS	1.368 5	0.002 6	1.587 6	3.542 6	1.376 5	0.003 7	1.663 0
ProxylessNAS	0.011 0	2.26×10^{-5}	0.006 5	0.000 8	0.001 0	0.000 4	0.002 0
ProxylessNAS-Mobile	0.013 5	0.000 1	0.009 3	0.001 8	0.002 3	0.000 8	0.003 6
DARTS	0.873 4	0.005 2	0.112 5	0.812 8	0.009 1	0.007 7	0.013 3
FairNAS-A	0.045 2	5.58×10^{-5}	0.008 6	0.001 2	0.001 6	0.000 1	0.006 4
FairNAS-B	0.019 7	0.000 1	0.010 3	0.003 5	0.001 9	0.001 8	0.007 8
FairNAS-C	0.012 5	0.000 4	0.012 2	0.020 2	0.002 8	0.001 4	0.004 5
GDAS	13.255 8	2.126 0	5.301 2	5.098 2	2.576 8	2.651 0	2.276 5
FBNet-V1	1.422 3	0.000 7	0.132 9	0.043 7	1.342 5	0.018 9	1.789 0
MNASNet	0.107 5	0.000 3	0.817 0	0.008 8	0.037 6	0.003 4	0.020 8
Scarlet-A	0.009 8	0.001 8	0.008 6	0.003 4	0.006 8	0.002 4	0.005 3
Scarlet-B	0.010 2	0.002 3	0.009 1	0.007 8	0.007 3	0.005 6	0.006 4
Scarlet-C	0.009 3	0.003 5	0.009 8	0.006 6	0.007 0	0.000 7	0.005 8
MoGA-A	0.176 5	0.003 8	0.042 2	0.023 2	0.002 4	0.000 3	0.009 3
MoGA-B	0.042 2	0.003 3	0.040 8	0.007 5	0.003 7	0.005 5	0.007 7
MoGA-C	0.221 0	0.001 7	0.038 6	0.005 6	0.004 2	0.003 9	0.009 8
PC-DARTS	9.768 3	0.556 4	5.567 3	14.679 2	5.980 6	2.648 1	3.547 0
Single-Path-SuperNet	0.746 0	0.004 4	1.442 6	0.095 6	0.056 7	0.005 5	0.138 6
DNA-a	0.187 8	0.007 4	0.177 7	0.011 0	0.056 4	0.003 5	0.174 5
DNA-b	0.038 5	0.006 5	0.174 4	0.006 4	0.049 8	0.002 8	0.020 0
DNA-c	0.032 2	0.005 8	0.165 9	0.005 1	0.042 2	0.002 4	0.153 7
DNA-d	0.028 0	0.003 4	0.132 2	0.004 8	0.035 7	0.001 6	0.142 2
FBNet-V2	0.821 2	0.000 8	0.037 6	0.003 5	0.002 5	0.007 4	0.144 4
RobNet	4.587 0	0.055 2	2.255 4	1.236 9	1.665 4	0.156 5	1.498 7

lessNSA, DARTS, FairNAS, GDAS, FBNet-V1, MNASNet, ScarletNAS, MoGA, PC-DARTS, Single-Path-SuperNet, DNA, FBNet-V2 and RobNet. 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 in a contactless manner. All databases were collected in two different sessions. In particular, HFUT CS is a rather challenging database because it was collected in the conditions of two different sessions, contactless manner and crossing three different sensors.

We conducted evaluation experiments on both separate data mode and mixed data mode. Experimental results showed that, among different NAS methods, ProxylessNAS achieved the best recognition accuracy. In other words, ProxylessNAS is a very suitable NAS method for 2D and 3D palmprint recognition and palm vein recognition.

In the “separate data mode”, for 2D palmprint recognition and palm vein recognition, the overall recognition performance of ProxylessNAS is close to that of traditional methods including CompC, OrdinalC, RLOC and LLDP. It is close to the deep learning-based method, i.e., PalmNet, which is a method specially designed for palmprint recognition. The overall recognition performance of ProxylessNAS is considerably better than that of

Table 10 ARR (%) and EER (%) of different NAS methods on four 2D representations of 3D palmprint databases under the mixed data mode

	CST		ST		MCI		GCI	
	ARR	EER	ARR	EER	ARR	EER	ARR	EER
NASNet	99.87	0.024 4	99.93	0.011 8	99.46	0.038 6	98.32	0.122 1
NASNet-Mobile	99.91	0.021 3	100	0.010 1	99.58	0.034 9	98.87	0.116 8
SMASH	99.38	0.035 2	99.87	0.013 5	99.14	0.031 1	98.25	0.128 8
PNASNet	100	0.100 7	100	0.002 2	100	0.008 7	100	0.101 8
NAONet	79.46	10.376 5	74.55	12.332 0	86.29	6.866 9	76.87	11.000 5
SNAS	98.76	0.155 3	97.42	0.636 0	97.18	0.911 4	96.33	1.458 2
AmoebaNet	95.65	1.762 1	97.77	0.789 0	97.21	0.827 8	92.46	3.464 0
ENAS	99.55	0.033 5	99.93	0.011 0	98.67	0.143 2	97.88	0.812 6
ProxylessNAS	100	0.000 6	100	0.000 4	100	0.001 5	100	0.007 6
ProxylessNAS-Mobile	100	0.001 4	100	0.001 2	100	0.003 2	100	0.008 7
DARTS	99.45	0.034 2	99.11	0.041 2	98.79	0.122 5	98.90	0.115 5
FairNAS-A	100	0.002 8	100	0.003 0	100	0.004 1	98.75	0.111 9
FairNAS-B	100	0.001 9	100	0.005 8	100	0.001 0	99.00	0.103 4
FairNAS-C	99.98	0.009 7	100	0.008 5	100	0.005 0	99.65	0.010 8
GDAS	98.33	0.165 0	91.26	3.284 4	90.59	4.344 5	88.43	5.987 0
FBNet-V1	99.22	0.037 5	99.64	0.032 2	99.69	0.021 8	98.19	0.127 6
MNASNet	100	0.008 5	100	0.003 4	100	0.010 1	99.56	0.018 3
Scarlet-A	100	0.007 4	100	0.001 8	100	0.002 0	99.66	0.009 8
Scarlet-B	100	0.008 3	100	0.006 8	100	0.009 5	100	0.007 6
Scarlet-C	99.87	0.011 0	100	0.007 7	100	0.008 5	98.13	0.151 3
MoGA-A	100	0.005 6	99.76	0.016 2	100	0.007 8	98.78	0.111 4
MoGA-B	100	0.004 8	100	0.005 5	100	0.009 8	98.44	0.113 0
MoGA-C	99.94	0.010 2	100	0.007 3	99.67	0.022 5	97.35	0.755 4
PC-DARTS	96.31	1.544 3	97.44	0.856 4	96.10	1.412 2	95.87	1.634 7
Single-Path-SuperNet	99.66	0.032 4	99.35	0.023 6	95.87	1.676 0	95.95	1.622 0
DNA-a	98.55	0.159 0	99.12	0.032 0	97.38	0.732 6	91.30	3.262 5
DNA-b	98.97	0.135 3	99.74	0.027 7	98.11	0.173 5	92.33	3.587 0
DNA-c	99.54	0.110 8	99.90	0.015 5	99.62	0.027 0	92.98	3.321 8
DNA-d	99.02	0.125 4	99.99	0.010 6	99.15	0.112 8	93.47	2.553 8
FBNet-V2	100	0.101 4	100	0.009 2	100	0.006 8	95.09	1.794 4
RobNet	97.34	0.821 1	96.55	1.452 1	93.29	2.544 9	93.06	2.571 8

one representative manually designed CNN method, i.e., ResNet, and is slightly better than EfficientNet and MobileNet-V3. For 3D palmprint recognition, ProxylessNAS achieved 100% ARR on the MCI representation, which is better than other methods.

In the “mixed data mode”, almost all methods can obtain 100% ARR and very low EER. For 2D palmprint and palm vein recognition, the overall recognition performance of ProxylessNAS is close to that of the traditional methods and other deep learning-based methods. For 3D palmprint recognition, ProxylessNAS achieves the best performance on the MCI representation.

In this work, it is the first time to conduct a performance evaluation of representative NAS methods for 2D and 3D palmprint and palm vein recognition. Experimental results showed the NAS is an up-and-coming technology for 2D and 3D palmprint and palm vein recognition. In our future work, based on NAS technology, we will try to design new methods to further improve the recognition performance of 2D and 3D palmprint recognition and palm vein recognition.

Acknowledgements

This work was supported by National Science Founda-

tion of China (Nos. 62076086, 61673157, 61972129, 61972127 and 61702154), and Key Research and Development Program in Anhui Province (Nos. 202004d07020008 and 201904d07020010).

Table 11 2D palmprint and palm vein recognition: ARR (%) and EER (%) performance comparison between NAS and other methods under the separate data mode

	PolyU II		PolyU M_B		HFUT I		HFUT CS		TJU-P		PolyU M_N		TJU-PV	
	ARR	EER	ARR	EER	ARR	EER	ARR	EER	ARR	EER	ARR	EER	ARR	EER
CompC	100	0.025 9	100	0.004 4	99.64	0.287 4	99.45	0.601 6	99.87	0.250 0	99.97	0.100 0	99.32	0.650 0
OrdinalC	100	0.051 8	100	0.005 9	99.60	0.425 5	99.67	0.585 1	99.95	0.267 1	100	0.003 4	99.55	0.594 6
RLOC	100	0.026 8	100	0.033 4	99.75	0.254 1	99.36	0.702 7	99.63	0.452 9	100	0.000 1	100	0.581 4
LLDP	100	0.052 1	100	0.003 0	99.89	0.501 1	99.40	0.199 4	99.50	0.516 6	100	0.031 4	98.93	1.077 1
PalmNet	100	0.838 0	100	0.004 7	100	0.065 5	92.45	4.097 4	100	0.011 5	99.02	1.400 9	99.61	0.328 1
ResNet	97.66	0.533 3	100	0.000 2	98.51	0.275 4	95.20	0.667 7	99.25	0.026 3	100	0.000 7	98.58	0.314 7
MobileNet-V3	97.35	0.574 1	100	0.001 1	98.67	0.173 4	96.55	0.432 6	99.37	0.055 2	100	0.000 3	98.67	0.267 8
EfficientNet	97.39	0.355 2	100	0.000 2	99.41	0.050 7	95.31	0.585 8	99.89	0.002 2	100	0.000 2	99.00	0.077 4
ProxylessNAS	98.63	0.172 8	100	3.34×10⁻⁵	99.68	0.040 7	99.78	0.021 7	99.75	0.018 7	99.89	0.001 1	99.67	0.070 4
FairNAS-A	96.53	1.454 2	100	0.000 1	98.85	0.105 7	99.39	0.048 2	99.55	0.013 2	100	0.000 1	99.23	0.064 7

Table 12 3D palmprint recognition: ARR (%) and EER (%) performance comparison between NAS and other methods under the separate data mode

	CST		ST		MCI		GCI	
	ARR	EER	ARR	EER	ARR	EER	ARR	EER
CompC	97.75	1.133 5	98.50	0.775 9	93.00	4.089 2	71.23	19.044 0
OrdinalC	98.68	0.920 1	99.03	0.638 1	91.43	3.721 5	97.95	1.554 8
RLOC	96.98	1.373 3	98.60	0.935 7	96.45	2.046 9	96.70	1.716 0
LLDP	91.05	3.517 7	96.50	1.617 3	89.10	4.726 6	93.80	2.938 6
PalmNet	97.28	1.195 7	98.70	0.892 8	96.90	1.815 7	96.92	1.690 1
ResNet	97.58	0.548 8	99.12	0.073 4	99.35	0.056 6	93.65	1.902 5
MobileNet-V3	97.53	0.602 5	98.47	0.321 6	97.32	0.712 5	94.16	1.712 2
EfficientNet	97.81	0.441 8	99.37	0.051 0	99.88	0.004 9	99.66	0.041 1
ProxylessNAS	99.90	0.002 8	99.92	0.002 0	100	0.005 7	99.45	0.119 3
FairNAS-A	99.58	0.043 6	99.40	0.049 2	99.25	0.058 7	94.05	3.279 0

Table 13 2D palmprint and palm vein recognition: ARR (%) and EER (%) performance comparison between NAS method and other methods under the mixed data mode

	PolyU II		PolyU M_B		HFUT I		HFUT CS		TJU-P		PolyU M_N		TJU-PV	
	ARR	EER	ARR	EER	ARR	EER	ARR	EER	ARR	EER	ARR	EER	ARR	EER
CompC	100	0.000 4	100	8.02×10 ⁻⁵	99.98	0.016 4	99.96	0.538 3	100	0.000 9	100	0.001 5	99.87	0.149 7
OrdinalC	100	0.004 0	100	0.003 1	99.98	0.038 5	100	0.291 1	100	1.55×10⁻⁵	100	0	99.87	0.040 6
RLOC	100	7.48×10⁻⁵	100	0.000 3	100	0.040 5	100	0.270 8	100	0.017 9	100	0	100	0.131 4
LLDP	100	0.005 8	100	0	99.93	0.041 7	100	0.332 4	100	0.018 5	100	0.001 0	99.96	0.167 1
PalmNet	100	0.006 5	100	5.32×10 ⁻⁵	100	0.027 8	100	0.333 0	100	0.018 6	100	0.001 2	99.91	0.166 8
ResNet	100	0.003 2	100	0.000 1	100	0.008 8	99.84	0.032 2	100	0.002 5	100	0.000 7	99.92	0.011 0
MobileNet-V3	99.69	0.053 4	100	0.000 3	100	0.008 2	99.49	0.072 5	100	0.002 2	100	0.000 2	100	0.009 4
EfficientNet	100	0.013 0	100	6.34×10 ⁻⁵	100	0.007 2	99.57	0.068 8	100	0.001 2	100	0.000 1	100	0.008 5
ProxylessNAS	100	0.011 0	100	2.26×10 ⁻⁵	100	0.006 5	100	0.000 8	100	0.001 0	100	0.000 4	100	0.002 0
FairNAS-A	99.76	0.045 2	100	5.58×10 ⁻⁵	100	0.008 6	100	0.001 2	100	0.001 6	100	0.000 1	100	0.006 4

Table 14 3D palmprint recognition: ARR (%) and EER (%) performance comparison between NAS and other methods under the mixed data mode

	CST		ST		MCI		GCI	
	RR	EER	ARR	EER	ARR	EER	ARR	EER
CompC	99.17	0.502 0	99.47	0.384 7	97.61	2.156 4	72.64	18.631 5
OrdinalC	99.75	0.338 8	99.81	0.165 9	97.86	1.422 0	97.97	1.446 2
RLOC	99.31	0.403 6	99.83	0.281 9	99.33	0.718 7	96.89	1.718 8
LLDP	95.86	1.952 1	98.14	0.867 9	95.84	2.437 6	94.58	2.758 8
PalmNet	99.28	0.434 9	99.89	0.190 4	99.39	0.579 1	97.33	1.493 1
ResNet	99.50	0.103 3	99.94	0.002 6	100	0.000 6	98.56	0.285 6
MobileNet-V3	98.64	0.255 8	99.67	0.032 8	99.67	0.038 5	97.19	0.833 5
EfficientNet	98.54	0.301 5	99.88	0.004 5	99.94	0.002 3	97.43	0.802 1
ProxylessNAS	100	0.000 6	100	0.000 4	100	0.001 5	100	0.007 6
FairNAS-A	100	0.002 8	100	0.003 0	100	0.004 1	98.75	0.111 9

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- [1] A. Kong, D. Zhang, M. Kamel. A survey of palmprint recognition. *Pattern Recognition*, vol.42, no.7, pp.1408–1418, 2009. DOI: [10.1016/j.patcog.2009.01.018](https://doi.org/10.1016/j.patcog.2009.01.018).
- [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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1016/j.neucom.2018.03.081).
- [5] 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](https://doi.org/10.1007/s11633-019-1194-7).
- [6] 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](https://doi.org/10.1007/s11633-019-1183-x).
- [7] C. L. Li, X. H. Wu, N. Zhao, X. C. Cao, J. Tang. Fusing two-stream convolutional neural networks for RGB-T object tracking. *Neurocomputing*, vol.281, pp.78–85, 2018. DOI: [10.1016/j.neucom.2017.11.068](https://doi.org/10.1016/j.neucom.2017.11.068).
- [8] C. L. Li, X. Y. Liang, Y. J. Lu, N. Zhao, J. Tang. RGB-T object tracking: Benchmark and baseline. *Pattern Recognition*, vol.96, Article number 106977, 2019. DOI: [10.1016/j.patcog.2019.106977](https://doi.org/10.1016/j.patcog.2019.106977).
- [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](https://doi.org/10.1145/3190618).
- [10] T. Elsken, J. H. Metzen, F. Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, vol.20, no.55, pp.1–21, 2019.
- [11] M. Wistuba, A. Rawat, T. Pedapati. A survey on neural architecture search. [Online], Available: <https://arxiv.org/abs/1905.01392>, 2019.
- [12] P. Z. Ren, Y. Xiao, X. J. Chang, P. Y. Huang, Z. H. Li, X. J. Chen, X. Wang. A comprehensive survey of neural architecture search: Challenges and solutions. [Online], Available: <https://arxiv.org/abs/2006.02903>, 2020.
- [13] Y. Q. Hu, Y. Yu. A technical view on neural architecture search. *International Journal of Machine Learning and Cybernetics*, vol.11, no.4, pp.795–811, 2020. DOI: [10.1007/s13042-020-01062-1](https://doi.org/10.1007/s13042-020-01062-1).
- [14] X. He, K. Y. Zhao, X. W. Chu. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, vol.212, Article number 106622, 2021. DOI: [10.1016/j.knosys.2020.106622](https://doi.org/10.1016/j.knosys.2020.106622).
- [15] B. Zoph, Q. V. Le. Neural architecture search with reinforcement learning. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.
- [16] 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](https://doi.org/10.1109/TPAMI.2003.1227981).
- [17] 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](https://doi.org/10.1109/TIM.2009.2028772).
- [18] 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](https://doi.org/10.1109/TIP.2017.2705424).
- [19] 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](https://doi.org/10.3390/s120607938).
- [20] 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](https://doi.org/10.1016/j.patcog.2017.04.016).
- [21] 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](https://doi.org/10.1109/TSMCC.2010.2055849).
- [22] 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](https://doi.org/10.3390/sym10040078).
- [23] 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](https://doi.org/10.1109/TIM.2020.2964076).
- [24] 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](https://doi.org/10.1109/TIFS.2019.2911165).
- [25] 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](https://doi.org/10.1109/TCSVT.2019.2904283).
- [26] 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, 2019. DOI: [10.1109/TIFS.2019.2945183](https://doi.org/10.1109/TIFS.2019.2945183).
- [27] 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](https://doi.org/10.1016/j.patcog.2019.107071).
- [28] 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*, 2020. DOI: [10.1109/TSMC.2020.3003021](https://doi.org/10.1109/TSMC.2020.3003021).
- [29] 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](https://doi.org/10.1016/j.ins.2019.03.027).
- [30] Y. Liu, A. Kumar. Contactless palmprint identification using deeply learned residual features. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 2, pp. 172–181, 2020. DOI: [10.1109/TBIOM.2020.2967073](https://doi.org/10.1109/TBIOM.2020.2967073).
- [31] 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](https://doi.org/10.1007/978-3-030-26773-5_26).
- [32] D. Thapar, G. Jaswal, A. Nigam, V. Kanhangad. PVS-Net: Palm vein authentication siamese network trained using triplet loss and adaptive hard mining by learning enforced domain specific features. In *Proceedings of the 5th IEEE International Conference on Identity, Security, and Behavior Analysis*, IEEE, Hyderabad, India, pp. 1–8, 2019. DOI: [10.1109/ISBA.2019.8778623](https://doi.org/10.1109/ISBA.2019.8778623).
- [33] 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, Maghreb, Tunisia, vol. 146, pp. 352–363, 2020. DOI: [10.1007/978-3-030-21005-2_34](https://doi.org/10.1007/978-3-030-21005-2_34).
- [34] M. Stanuch, M. Wodzinski, A. Skalski. Contact-free multispectral identity verification system using palm veins and deep neural network. *Sensors*, vol. 20, no. 19, Article number 5695, 2020. DOI: [10.3390/s20195695](https://doi.org/10.3390/s20195695).
- [35] W. Jia, J. Gao, W. Xia, Y. Zhao, H. Min, J. T. Lu. A performance evaluation of classic convolutional neural networks for 2D and 3D palmprint and palm vein recognition. *International Journal of Automation and Computing*, vol. 18, no. 1, pp. 18–44, 2021. DOI: [10.1007/s11633-020-1257-9](https://doi.org/10.1007/s11633-020-1257-9).
- [36] B. Baker, O. Gupta, N. Naik, R. Raskar. Designing neural network architectures using reinforcement learning. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.
- [37] R. Shin, C. Packer, D. Song. Differentiable neural network architecture search. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, 2018.
- [38] K. Kandasamy, W. Neiswanger, J. Schneider, B. Póczos, E. P. Xing. Neural architecture search with Bayesian optimisation and optimal transport. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 2016–2025, 2018.
- [39] C. X. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L. J. Li, L. Fei-Fei, A. Yuille, J. Huang, K. Murphy. Progressive neural architecture search. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, vol. 11205, pp. 19–35, 2018. DOI: [10.1007/978-3-030-01246-5_2](https://doi.org/10.1007/978-3-030-01246-5_2).
- [40] R. Q. Luo, F. Tian, T. Qin, E. H. Chen, T. Y. Liu. Neural architecture optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 7816–7827, 2018.
- [41] A. Brock, T. Lim, J. M. Ritchie, N. Weston. SMASH: One-shot model architecture search through hyperNetworks. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, 2018.
- [42] G. Bender, P. J. Kindermans, B. Zoph, V. Vasudevan, Q.

- Le. Understanding and simplifying one-shot architecture search. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp.883–893, 2018.
- [43] Z. Zhong, J. J. Yan, W. Wu, J. Shao, C. L. Liu. Practical block-wise neural network architecture generation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp.2423–2432, 2018. DOI: [10.1109/CVPR.2018.00257](https://doi.org/10.1109/CVPR.2018.00257).
- [44] B. Zoph, V. Vasudevan, J. Shlens, Q. V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp.8697–8710, 2018. DOI: [10.1109/CVPR.2018.00907](https://doi.org/10.1109/CVPR.2018.00907).
- [45] T. J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, H. Adam. NetAdapt: Platform-aware neural network adaptation for mobile applications. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, vol.11214, pp.289–304, 2018. DOI: [10.1007/978-3-030-01249-6_18](https://doi.org/10.1007/978-3-030-01249-6_18).
- [46] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, F. Hutter. NAS-Bench-101: Towards reproducible neural architecture search. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp.12334–12348, 2019.
- [47] M. X. Tan, Q. V. 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.
- [48] X. X. Chu, B. Zhang, R. J. Xu, J. X. Li. FairNAS: Rethinking evaluation fairness of weight sharing neural architecture search. [Online], Available: <https://arxiv.org/abs/1907.01845>, 2019.
- [49] D. Ho, E. Liang, I. Stoica, P. Abbeel, X. Chen. Population based augmentation: Efficient learning of augmentation policy schedules. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp.4843–4856, 2019.
- [50] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q. V. Le. AutoAugment: Learning augmentation strategies from data. In *Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.113–123, 2019. DOI: [10.1109/CVPR.2019.00020](https://doi.org/10.1109/CVPR.2019.00020).
- [51] E. Real, A. Aggarwal, Y. Huang, Q. V. Le. Regularized evolution for image classifier architecture search. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 31st Innovative Applications of Artificial Intelligence Conference and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, AIAA, Honolulu, USA, pp.4780–4789, 2019. DOI: [10.1609/aaai.v33i01.33014780](https://doi.org/10.1609/aaai.v33i01.33014780).
- [52] V. Nekrasov, H. Chen, C. H. Shen, I. Reid. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.9118–9127, 2019. DOI: [10.1109/CVPR.2019.00934](https://doi.org/10.1109/CVPR.2019.00934).
- [53] X. X. Chu, B. Zhang, J. X. Li, Q. Y. Li, R. J. Xu. ScarletNAS: Bridging the gap between scalability and fairness in neural architecture search. [Online], Available: <https://arxiv.org/abs/1908.06022>, 2019.
- [54] J. M. Pérez-Rúa, V. Vielzeuf, S. Pateux, M. Baccouche, F. Jurie. MFAS: Multimodal fusion architecture search. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.6959–6968, 2019. DOI: [10.1109/CVPR.2019.00713](https://doi.org/10.1109/CVPR.2019.00713).
- [55] M. X. Tan, B. Chen, R. M. Pang, V. Vasudevan, M. Sandler, A. Howard, Q. V. Le. MnasNet: Platform-aware neural architecture search for mobile. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.2815–2823, 2019. DOI: [10.1109/CVPR.2019.00293](https://doi.org/10.1109/CVPR.2019.00293).
- [56] C. X. Liu, L. C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, L. Fei-Fei. Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.82–92, 2019. DOI: [10.1109/CVPR.2019.00017](https://doi.org/10.1109/CVPR.2019.00017).
- [57] Y. K. Chen, G. F. Meng, Q. Zhang, S. M. Xiang, C. Huang, L. S. Mu, X. G. Wang. RENAS: Reinforced evolutionary neural architecture search. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.4782–4791, 2019. DOI: [10.1109/CVPR.2019.00492](https://doi.org/10.1109/CVPR.2019.00492).
- [58] B. C. Wu, X. L. Dai, P. Z. Zhang, Y. H. Wang, F. Sun, Y. M. Wu, Y. D. Tian, P. Vajda, Y. Q. Jia, K. Keutzer. FBNet: Hardware-aware efficient ConvNet design via differentiable neural architecture search. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.10726–10734, 2019. DOI: [10.1109/CVPR.2019.01099](https://doi.org/10.1109/CVPR.2019.01099).
- [59] X. Li, Y. M. Zhou, Z. Pan, J. S. Feng. Partial order pruning: For best speed/accuracy trade-off in neural architecture search. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.9137–9145, 2019. DOI: [10.1109/CVPR.2019.00936](https://doi.org/10.1109/CVPR.2019.00936).
- [60] X. Y. Dong, Y. Yang. Searching for a robust neural architecture in four GPU hours. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.1761–1770, 2019. DOI: [10.1109/CVPR.2019.00186](https://doi.org/10.1109/CVPR.2019.00186).
- [61] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, J. Dean. Efficient neural architecture search via parameter sharing. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp.6522–6531, 2018.
- [62] S. R. Xie, H. H. Zheng, C. X. Liu, L. Lin. SNAS: Stochastic neural architecture search. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA, 2019.
- [63] T. Elsken, J. H. Metzen, F. Hutter. Efficient multi-objective neural architecture search via Lamarckian evolution. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA, 2019.
- [64] G. Ghiasi, T. Y. Lin, Q. V. Le. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp.7029–7038, 2019. DOI: [10.1109/CVPR.2019.00720](https://doi.org/10.1109/CVPR.2019.00720).
- [65] C. Zhang, M. Y. Ren, R. Urtasun. Graph HyperNetworks for neural architecture search. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA, 2019.
- [66] H. Cai, L. Zhu, S. Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *Pro-*

- ceedings of the 7th International Conference on Learning Representations, New Orleans, USA, 2019.
- [67] H. X. Liu, K. Simonyan, Y. M. Yang. DARTS: Differentiable architecture search. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA, 2019.
- [68] N. Nayman, A. Noy, T. Ridnik, I. Friedman, R. Jin, L. Zelnik-Manor. XNAS: Neural architecture search with expert advice. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, pp.1975–1985, 2019.
- [69] J. R. Peng, M. Sun, Z. X. Zhang, T. N. Tan, J. J. Yan. Efficient neural architecture transformation search in channel-level for object detection. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, pp.14290–14299, 2019.
- [70] H. Z. Hu, J. Langford, R. Caruana, S. Mukherjee, E. Horvitz, D. Dey. Efficient forward architecture search. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, pp.10122–10131, 2019.
- [71] X. Y. Dong, Y. Yang. Network pruning via transformable architecture search. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, pp.759–770, 2019.
- [72] Y. K. Chen, T. Yang, X. Y. Zhang, G. F. Meng, X. Y. Xiao, J. Sun. DetNAS: backbone search for object detection. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, pp.6638–6648, 2019.
- [73] M. Wortsman, A. Farhadi, M. Rastegari. Discovering neural wirings. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, pp.2680–2690, 2019.
- [74] X. Y. Gong, S. Y. Chang, Y. F. Jiang, Z. Y. Wang. AutoGAN: Neural architecture search for generative adversarial networks. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp.3223–3233, 2019. DOI: [10.1109/ICCV.2019.00332](https://doi.org/10.1109/ICCV.2019.00332).
- [75] X. Y. Dong, Y. Yang. One-shot neural architecture search via self-evaluated template network. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp.3680–3689, 2019. DOI: [10.1109/ICCV.2019.00378](https://doi.org/10.1109/ICCV.2019.00378).
- [76] Y. Y. Xiong, R. Mehta, V. Singh. Resource constrained neural network architecture search: Will a submodularity assumption help? In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp.1901–1910, 2019. DOI: [10.1109/ICCV.2019.00199](https://doi.org/10.1109/ICCV.2019.00199).
- [77] 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, Korea, pp.1314–1324, 2019. DOI: [10.1109/ICCV.2019.00140](https://doi.org/10.1109/ICCV.2019.00140).
- [78] X. W. Zheng, R. R. Ji, L. Tang, B. C. Zhang, J. Z. Liu, Q. Tian. Multinomial distribution learning for effective neural architecture search. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp.1304–1313, 2019. DOI: [10.1109/ICCV.2019.00139](https://doi.org/10.1109/ICCV.2019.00139).
- [79] R. Pasunuru, M. Bansal. Continual and multi-task architecture search. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL, Florence, Italy, pp.1911–1922, 2019. DOI: [10.18653/v1/P19-1185](https://doi.org/10.18653/v1/P19-1185).
- [80] Y. F. Jiang, C. Hu, T. Xiao, C. L. Zhang, J. B. Zhu. Improved differentiable architecture search for language modeling and named entity recognition. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, Hong Kong, China, pp.3585–3590, 2019. DOI: [10.18653/v1/D19-1367](https://doi.org/10.18653/v1/D19-1367).
- [81] L. Li, A. Talwalkar. Random search and reproducibility for neural architecture search. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, Tel Aviv, Israel, pp.367–377, 2019.
- [82] X. X. Chu, B. Zhang, R. J. Xu. MoGA: Searching beyond Mobilenetv3. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Barcelona, Spain, pp.4042–4046, 2020. DOI: [10.1109/ICASSP40776.2020.9054428](https://doi.org/10.1109/ICASSP40776.2020.9054428).
- [83] W. Y. Chen, X. Y. Gong, X. M. Liu, Q. Zhang, Y. Li, Z. Y. Wang. FasterSeg: Searching for faster real-time semantic segmentation. [Online], Available: <https://arxiv.org/abs/1912.10917>, 2020.
- [84] Y. H. Xu, L. X. Xie, X. P. Zhang, X. Chen, G. J. Qi, Q. Tian, H. K. Xiong. PC-DARTS: Partial channel connections for memory-efficient architecture search. [Online], Available: <https://arxiv.org/abs/1907.05737>, 2019.
- [85] J. R. Mei, Y. W. Li, X. C. Lian, X. J. Jin, L. J. Yang, A. Yuille, J. C. Yang. AtomNAS: Fine-grained end-to-end neural architecture search. [Online], Available: <https://arxiv.org/abs/1912.09640>, 2020.
- [86] X. Y. Dong, Y. Yang. NAS-Bench-201: Extending the scope of reproducible neural architecture search. [Online], Available: <https://arxiv.org/abs/2001.00326>, 2020.
- [87] M. X. Tan, R. M. Pang, Q. V. Le. EfficientDet: Scalable and efficient object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.10778–10787, 2020. DOI: [10.1109/CVPR42600.2020.01079](https://doi.org/10.1109/CVPR42600.2020.01079).
- [88] J. M. Fang, Y. Z. Sun, Q. Zhang, Y. Li, W. Y. Liu, X. G. Wang. Densely connected search space for more flexible neural architecture search. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.10625–10634, 2020. DOI: [10.1109/CVPR42600.2020.01064](https://doi.org/10.1109/CVPR42600.2020.01064).
- [89] M. Zhang, H. Q. Li, S. R. Pan, X. J. Chang, S. Su. Overcoming multi-model forgetting in one-shot NAS with diversity maximization. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.7806–7815, 2020. DOI: [10.1109/CVPR42600.2020.00783](https://doi.org/10.1109/CVPR42600.2020.00783).
- [90] C. L. Li, J. F. Peng, L. C. Yuan, G. R. Wang, X. D. Liang, L. Lin, X. J. Chang. Block-wisely supervised neural architecture search with knowledge distillation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.1986–1995, 2020. DOI: [10.1109/CVPR42600.2020.00206](https://doi.org/10.1109/CVPR42600.2020.00206).
- [91] M. H. Guo, Y. Z. Yang, R. Xu, Z. W. Liu, D. H. Lin. When NAS meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern*

- Recognition, IEEE, Seattle, USA, pp.628–637, 2020. DOI: [10.1109/CVPR42600.2020.00071](https://doi.org/10.1109/CVPR42600.2020.00071).
- [92] C. Gao, Y. P. Chen, S. Liu, Z. X. Tan, S. C. Yan. AdversarialNAS: Adversarial neural architecture search for GANs. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.5679–5688, 2020. DOI: [10.1109/CVPR42600.2020.00572](https://doi.org/10.1109/CVPR42600.2020.00572).
- [93] A. Wan, X. L. Dai, P. Z. Zhang, Z. J. He, Y. D. Tian, S. Xie, B. C. Wu, M. Yu, T. Xu, K. Chen, P. Vajda, J. E. Gonzalez. FBNetV2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.12962–12971, 2020. DOI: [10.1109/CVPR42600.2020.01298](https://doi.org/10.1109/CVPR42600.2020.01298).
- [94] G. Bender, H. X. Liu, B. Chen, G. Chu, S. Y. Cheng, P. J. Kindermans, Q. V. Le. Can weight sharing outperform random architecture search? An investigation with TuNAS. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.14311–14320, 2020. DOI: [10.1109/CVPR42600.2020.01433](https://doi.org/10.1109/CVPR42600.2020.01433).
- [95] G. H. Li, G. C. Qian, I. C. Delgadillo, M. Müller, A. Thabet, B. Ghanem. SGAS: Sequential greedy architecture search. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.1617–1627, 2020. DOI: [10.1109/CVPR42600.2020.00169](https://doi.org/10.1109/CVPR42600.2020.00169).
- [96] X. W. Zheng, R. R. Ji, Q. Wang, Q. X. Ye, Z. G. Li, Y. H. Tian, Q. Tian. Rethinking performance estimation in neural architecture search. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.11353–11362, 2020. DOI: [10.1109/CVPR42600.2020.01137](https://doi.org/10.1109/CVPR42600.2020.01137).
- [97] H. Phan, Z. C. Liu, D. Huynh, M. Savvides, K. T. Cheng, Z. Q. Shen. Binarizing MobileNet via evolution-based searching. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.13417–13426, 2020. DOI: [10.1109/CVPR42600.2020.01343](https://doi.org/10.1109/CVPR42600.2020.01343).
- [98] C. Y. He, H. S. Ye, L. Shen, T. Zhang. MiLeNAS: Efficient neural architecture search via mixed-level reformulation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp.11990–11999, 2020. DOI: [10.1109/CVPR42600.2020.01201](https://doi.org/10.1109/CVPR42600.2020.01201).
- [99] X. Y. Dai, D. D. Chen, M. C. Liu, Y. P. Chen, L. Yuan. DA-NAS: Data adapted pruning for efficient neural architecture search. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12372, pp.584–600, 2020. DOI: [10.1007/978-3-030-58583-9_35](https://doi.org/10.1007/978-3-030-58583-9_35).
- [100] Y. Tian, Q. Wang, Z. W. Huang, W. Li, D. X. Dai, M. H. Yang, J. Wang, O. Fink. Off-policy reinforcement learning for efficient and effective GAN architecture search. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12352, pp.175–192, 2020. DOI: [10.1007/978-3-030-58571-6_11](https://doi.org/10.1007/978-3-030-58571-6_11).
- [101] X. X. Chu, T. B. Zhou, B. Zhang, J. X. Li. Fair DARTS: Eliminating unfair advantages in differentiable architecture search. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12360, pp.465–480, 2020. DOI: [10.1007/978-3-030-58555-6_28](https://doi.org/10.1007/978-3-030-58555-6_28).
- [102] Y. B. Hu, X. Wu, R. He. TF-NAS: Rethinking three search freedoms of latency-constrained differentiable neural architecture search. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12360, pp.123–139, 2020. DOI: [10.1007/978-3-030-58555-6_8](https://doi.org/10.1007/978-3-030-58555-6_8).
- [103] Y. M. Hu, Y. D. Yang, Z. C. Guo, R. S. Wan, X. Y. Zhang, Y. C. Wei, Q. Y. Gu, J. Sun. Angle-based search space shrinking for neural architecture search. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12364, pp.119–134, 2020. DOI: [10.1007/978-3-030-58529-7_8](https://doi.org/10.1007/978-3-030-58529-7_8).
- [104] H. B. Yu, Q. Han, J. B. Li, J. P. Shi, G. L. Cheng, B. Fan. Search what you want: Barrier panelty NAS for mixed precision quantization. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12354, pp.1–16, 2020. DOI: [10.1007/978-3-030-58545-7_1](https://doi.org/10.1007/978-3-030-58545-7_1).
- [105] X. F. Wang, X. H. Xiong, M. Neumann, A. J. Piergiiovanni, M. S. Ryoo, A. Angelova, K. M. Kitani, W. Hua. AttentionNAS: Spatiotemporal attention cell search for video classification. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12353, pp.449–465, 2020. DOI: [10.1007/978-3-030-58598-3_27](https://doi.org/10.1007/978-3-030-58598-3_27).
- [106] A. Bulat, B. Martinez, G. Tzimiropoulos. BATS: Binary Architecture search. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12368, pp.309–325, 2020. DOI: [10.1007/978-3-030-58592-1_19](https://doi.org/10.1007/978-3-030-58592-1_19).
- [107] J. H. Yu, P. C. Jin, H. X. Liu, G. Bender, P. J. Kindermans, M. X. Tan, T. Huang, X. D. Song, R. M. Pang, Q. Le. BigNAS: Scaling up neural architecture search with big single-stage models. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12352, pp.702–717, 2020. DOI: [10.1007/978-3-030-58571-6_41](https://doi.org/10.1007/978-3-030-58571-6_41).
- [108] Z. C. Guo, X. Y. Zhang, H. Y. Mu, W. Heng, Z. C. Liu, Y. C. Wei, J. Sun. Single path one-shot neural architecture search with uniform sampling. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12361, pp.544–560, 2020. DOI: [10.1007/978-3-030-58517-4_32](https://doi.org/10.1007/978-3-030-58517-4_32).
- [109] C. X. Liu, P. Dollár, K. M. He, R. Girshick, A. Yuille, S. N. Xie. Are labels necessary for neural architecture search? In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, vol.12349, pp.798–813, 2020. DOI: [10.1007/978-3-030-58548-8_46](https://doi.org/10.1007/978-3-030-58548-8_46).
- [110] 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.
- [111] M. Sandler, A. Howard, M. L. Zhu, A. Zhmoginov, L. C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp.4510–4520, 2018. DOI: [10.1109/CVPR.2018.00474](https://doi.org/10.1109/CVPR.2018.00474).
- [112] Z. N. Sun, T. N. Tan, Y. H. Wang, S. Z. Li. Ordinal palmprint representation 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](https://doi.org/10.1109/CVPR.2005.267).
- [113] W. Jia, D. S. Huang, D. Zhang. Palmprint verification based on robust line orientation code. *Pattern Recogni-*

tion, vol.41, no.5, pp.1504–1513, 2008. DOI: [10.1016/j.patcog.2007.10.011](https://doi.org/10.1016/j.patcog.2007.10.011).

- [114] 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](https://doi.org/10.1016/j.patcog.2015.08.025).
- [115] 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](https://doi.org/10.1109/CVPR.2017.634).



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 Sciences, 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



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. degree in automation from University of Science and Technology of China, China in 2008, and the Ph.D. degree in pattern recognition and intelligence system from University of Science and Technology of China, China in 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 the 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



Yan-Xiang Chen received the B.Sc. and the M.Sc. degree in electronic information engineering from Hefei University of Technology, China in 1993 and 1996, and the Ph.D. degree in signal and information processing from University of Science and Technology of China, China in 2004. She has been a visiting scholar in University of Illinois at Urbana-Champaign, USA from

2006 to 2008, and in National University of Singapore, Singapore from 2012 to 2013. She is currently a professor in School of Computer Science and Information Engineering, Hefei University of Technology, China.

Her research interests include audio-visual signal processing, saliency and machine learning.

E-mail: chenyx@hfut.edu.cn