Gesture-ProxylessNAS: A Lightweight Network for Mid-Air Gesture Recognition Based on UWB Radar

Lihong Qiao, Zhixin Li, Bin Xiao, Yucheng Shu, Weisheng Li, *Member, IEEE*, and Xinbo Gao, *Senior Member, IEEE*

***Abstract*—Hand gesture recognition with radar sensors is essential because they can detect gestures despite environmental factors like lighting, dust, and complex backgrounds. Considering the complexity of a system, it is challenging to design CNNs on CPU devices and realize the carry-on mid-air gesture recognition. We propose a mid-air gesture recognition method based on a novel discriminant feature, and it be used as part of a measurement system of hand movements using an ultrawideband (UWB) radar. The Gesture-ProxylessNAS (GPNAS) is presented to enhance the adaptability of model search and overcome the challenge of the network's computational complexity. In order to fully extract local spatial discriminant features and prevent information loss, local binary pattern (LBP) encoders are utilized to extract local spatial information. In the meantime, multilayer ShuffleNet with depthwise separable convolution is used to gradually leverage high-level spatial features. The GPNAS module revisits the multilayer ShuffleNet's design spaces using an optimization problem, greatly reducing the network's parameters and computational complexity. According to experimental verification on real UWB hand gestures, the proposed framework provides more satisfactory recognition performance and efficiency with a deeper network structure and fewer parameters. The proposed hand gesture recognition system can recognize gestures with a promising accuracy of 96.52% on the UWB-gestures public dataset.**

***Index Terms*—LBP encoder, mid-air gesture recognition, ProxylessNAS, UWB radar.**

Manuscript received 2 March 2023; revised 10 April 2023 and 30 April 2023; accepted 7 May 2023. Date of publication 10 May 2023; date of current version 14 June 2023. This work was supported in part by the National Key Research and Development Project under Grant 2019YFE0110800, in part by the National Natural Science Foundation of China under Grant 62276040 and Grant 61976031, in part by the National Major Scientific Research Instrument Development Project of China under Grant 62027827, in part by the Scientific and Technological Key Research Program of Chongqing Municipal Education Commission under Grant KJQN202200624, and in part of 2022 National College Students Innovation and Entrepreneurship Training Program Project under Grant 202210617018. *(Corresponding author: Bin Xiao.)*

Lihong Qiao, Yucheng Shu, and Xinbo Gao are with the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: qiaolh@cqupt.edu.cn; shuyc@cqupt.edu.cn; gaoxb@cqupt.edu.cn).

Zhixin Li is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: 1530215918@qq.com).

Bin Xiao is with the School of computer science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: xiaobin@cqupt.edu.cn).

Weisheng Li is with the Chongqing University of Posts and Telecommunications, Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Telecommunications and Posts, Chongqing 400065, China (e-mail: liws@cqupt.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3274830

I. Introduction

MOBILE devices, including tablets and phones, have become essential communication and entertainment tools in daily lives as a result of advancements in electronic technology and artificial intelligence. Generally, users can only interact with these devices through touch or keyboards. One of the present methods of interaction for mobile devices is gesture interaction, which takes into account the convenience of boosting human–machine contact. The existence of comparable hand gestures for several frequently used actions is thought to be useful. Additionally, the operation times of some complex instructions can be greatly shortened. Therefore, hand gesture recognition is becoming a promising technique for human–machine interaction, with numerous applications in a wide range of industries.

Many contemporary studies on gesture recognition include a variety of sensors, including cameras [2], [3], audio, WiFi, radio frequency identification (RFID), Bluetooth, Doppler radar [4], and ultrawideband (UWB) radar [5], [6], [7]. Variations in different brightness, contrast, and exposure levels have an impact on the accuracy of gesture detection in different ways for vision-based systems. On the other hand, because it may reduce the negative impacts of background disturbances when UWB radar is utilized for mid-air gesture sensing, the mid-air gesture interaction method based on UWB radar is the most promising area for study. UWB technology sends and receives impulse signals to locate moving objects. It benefits from having great antimultipath capabilities, excellent precision, and powerful penetration. It eliminates the privacy issues that have plagued conventional antenna technology for a very long time.

The main procedure is selecting features with high discrimination since the gesture is shorter and the spectrum is more noise-sensitive. As model parameters and FLOPs increase, it becomes more challenging to achieve quick inference speed on mobile devices. To extract and recognize hand motions, conventional machine learning [8], [9], [10], [11], [12] requires predetermined features. The significant features for recognition are typically unidentified [13] since the majority of features are arbitrarily defined [14]. Contrarily, deep learning algorithms, particularly convolutional neural networks (CNN), do not require predetermined features but discover key features via training that have a significant impact on gesture recognition precision [15]. A potential method for hand gesture recognition involves creating and training a convolutional neural network

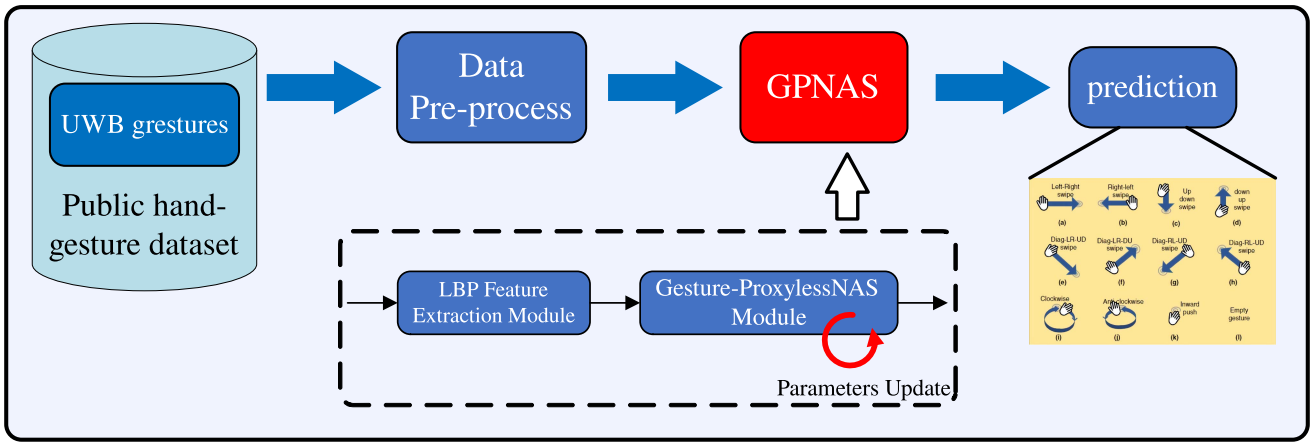

Fig. 1. Structure of our gesture recognition method. The method is divided into three main parts: a public UWB-gestures dataset (the source of the dataset), data preprocessing and GPNAS, the most crucial gesture recognition network. The LBP feature extraction module and the Gesture-ProxylessNAS Module are the two main components of GPNAS. Additionally, the Gesture-ProxylessNAS training approach updates the Gesture-ProxylessNAS Module's parameters.

as it enhances gesture categorization effectiveness and does not require a predetermined set of features.

The industry's application procedure for gesture recognition considers the latency and recognition accuracy index of hardware devices. ShuffleNet [16], [17], MobileNet [18], [19], GhostNet [20], and other lightweight convolutional neural networks are always used as benchmarks.

In particular, we concentrate on the following essential questions: How to increase network visibility without increasing latency to ensure more effective feature presentation? What features improve the precision of CPU-based lightweight models? How to design a network architecture to achieve both high recognition performance and lightweight performance?

Based on these problems, we focus on finely designing a lightweight network architecture on the framework. We introduce the multi-layer ShuffleNet with depthwise separable convolution module to improve the recognition performance, and the Gesture-ProxylessNAS training approach to compress the model and improve its real-time performance. Specifically, we utilize the efficient network architecture design of ShuffleNet [16], [17], which achieves an efficient and lightweight network architecture through group convolution and channel shuffling. We introduce the design of the network architecture and form the basic network architecture of gesture recognition.

However, in the later design of the framework, as the number of convolution layers increases, the accuracy of gesture recognition increases, but this leads to larger model sizes. How to compress the model while minimizing accuracy has become a pressing challenge. Finally, we specially designed a neural architecture search method for the gesture recognition algorithm, Gesture-ProxylessNAS, which is improved by ProxylessNAS [21].

What is more, after visualizing the UWB data, we find that each piece of data (image format) has obvious textural features, and through the ablation experiment, we find that local binary pattern (LBP) encoder is very helpful for improving the performance of gesture recognition network.

We propose a simple gesture recognition neural network called GPNAS, based on Gesture-ProxylessNAS. The UWB radar's extracted input signals for the GPNAS under consideration are gesture signals. For the purpose of obtaining local spatial discriminant features, we employ LBP encoders. The Gesture-ProxylessNAS Module reevaluates the design spaces of multilayer ShuffleNet and combines them with a proxyless neural architecture search unit, which substitutes the traditional convolution and significantly lowers the network's parameters and computational complexity. It encapsulates an optimization network by figuring out the best path to take. In order to extract the spectral-spatial feature, the proposed GPNAS framework uses lightweight multiscale attention structures. This results in fewer parameters, lower computing costs, and a deeper network structure. The experimental results further illustrate that GPNAS is capable of producing a recognition result that is more pleasing while using fewer parameters and less processing time. This article's overall summary and scope are presented in Fig. 1. The contributions of this article are summarized as follows:

1) We present a UWB radar-based system for recognizing mid-air hand gesture. Based on the LBP (local binary pattern) encoders, which extracts local spatial discriminant features.
2) The multilayer ShuffleNet with depthwise separable convolution is presented with low computational cost.
3) The Gesture-ProxylessNAS Module employs a ProxylessNAS network by selecting the optimal network, resulting in a significant reduction in time cost by approximately one-third with great precision.

The rest of this article is organized as follows. In Section II, we provide a brief overview of the research relevant to our work. In Section III, we introduce the UWB-gestures dataset. We present the proposed method in Section IV. In Section V, we perform the recognition experiments on the UWB gesture signal. Finally, Section VI is our discussion for the work and Section VII is the conclusion.

II. Related Work

Radar-based gesture recognition has garnered a great deal of interest recently. Seong et al. [22] developed a CNN model to recognize digits written in mid-air using hand gestures after collecting the gesture signals using three impulse radio ultra wide-band (IR-UWB) radar sensors. Using two-antenna doppler radar, Skaria et al. [13] were able to record these gestures and map the two beat signals into the three input channels of a DCNN for gesture recognition. This radar can generate the in-phase and quadrature Doppler components of the beat signals. Hendy et al. [23] introduced five more neural network models, including FNN, 2D-CNN, 3D-CNN, 2D-CNN-LSTM, and 3D-CNN-LSTM. They did this by utilizing two alternative data representations for the obtained gesture signals using a single UWB radar. Skaria et al. [24] examined four deep-learning recognition methods, including FCNN, k-NN, SVM, and LSTM, using a 3-D tensor made up of a range-Doppler frame sequence to describe gesture signature. However, the majority of these efforts primarily focused on hardware-based signal decoding and gathering. These works employ machine learning techniques or simple convolution layer stacking CNN models in the following recognition algorithms. The algorithms that identify motions could be further improved by these researchers. These approaches still have great potential for further development in terms of recognition algorithms. Furthermore, using a commercial frequency-modulated continuous wave (FMCW) multi-input–multi-output (MIMO) millimeter-wave radar, Xia et al. [25] developed a lightweight multi-channel convolutional neural network to represent and learn the range-Doppler (RD). Additionally, RD is a multidimensional angular property of learned gesture. Li et al. [26] proposed a unique approach for the identification of sign language and hand gestures based on the cumulative distribution density feature retrieved from the spectrogram of UWB radar signals. In order to achieve HGR, Wang et al. [24] introduced a novel model named CMFF-HGR, which fuses multidimensional features of gesture signals. Franceschini et al. [27] proposed an ultrasound system for person identification that exploits hand gestures. The system works as a sonar, measuring the ultrasonic pressure waves scattered by the subject's hand, and analyzing its Doppler information.

However, they must carefully take into account multispatial aspects to avoid any loss of information. Additionally, the present deep learning-based algorithms have a drawback: the model parameters are excessively huge, resulting in high time consumption and poor recognition accuracy.

III. Source of Data and Data Preprocessing

A. Introduction of Dataset

The data used in this article are mainly derived from a public UWB-gestures dataset [1]. To construct the UWB-Gestures dataset, the authors used three UWB radars with eight participants to capture 12 different hand gestures. Each participant was asked to perform 12 gestures, with each gesture repeated 30 times. Each kind of gesture has 2400 pieces of data. During the capture process, participants stood in an area 1.5 m away from the three UWB radars facing them.

B. Data Preprocessing

In the UWB-Gestures dataset, the authors conducted preprocessing to mitigate the impact of noise and clutter. Specifically, they used a method based on loopback filtering to estimate and remove clutter. This method extracts the current clutter term using the previously estimated clutter and the current received radar signal, and uses a weighting factor to control the learning rate of the filter. The authors applied this method to the raw data, resulting in clutter-removed data.

To input the UWB-Gestures dataset into our UWB gesture recognition method, we record the UWB signals of each gesture using MATLAB software and stored them as MAT files. After that, we convert these MAT files into CSV format and transform the raw radar data matrix into images.

IV. Methodology

The gesture have different features as its period is shorter and the spectrum is more noise-sensitive. To enhance the gesture recognition results, selecting features with strong discriminant is the key procedure. Besides, our focus has switched from a manually constructed architecture to an architecture that adaptive conducts a systematic search as a result of advancements in GPU hardware. Taking these qualities into account, we design a lightweight architecture named Gesture-ProxylessNAS, which is shown in Fig. 1. In the proposed GPNAS, we use the local binary pattern encoders to extract discriminant spatial knowledge and depress the noise. In the Gesture-ProxylessNAS Module, we reexamine the design spaces of the multilayer ShuffleNet and combine it with a proxyless neural architecture search unit. This unit replaces the standard convolution and dramatically reduces the parameters and computational complexity of the network. The overall network structure is shown in Fig. 1.

A. Local Binary Encoder

By carefully comparing the features of different classes of images in this UWB-gestures dataset, we find that the significant differences between other classes of images are mainly due to the different detected gesture movement trends. Namely, the textural feature of different classes of images are very distinguishable. Instead of the standard input of the gesture features, we utilize the local binary encoder of the gesture image through a random binary convolution layer to extract the textural feature of the detected images.

Local binary pattern (LBP) is a simple yet very powerful hand-designed descriptor used to describe the regional textural feature of an image [28]. The traditional LBP operator operates on image patches of different sizes, such as (3×3), (5×5), etc. Each operation takes a pixel as the central point, and then the pixel value of the center point is compared with the pixel value of the surrounding neighbor, and the neighbor with the high pixel value is assigned a value of 1. Reads all the neighbor assignments in a fixed order and maps them to a decimal number as the eigenvalue

of the center pixel. This process can be expressed as follows by a formula [28]:

$$LBP(x_c, y_c) = \sum_{n=1}^{L-1} t(P_n, P_c) * 2^n \tag{1}$$

$$t(P_n, P_c) = \begin{cases} 1, & \text{if } P_c \leq P_n \\ 0, & \text{else} \end{cases} \tag{2}$$

where (x_c, y_c) denotes the coordinates of the center pixel, P_c denotes the pixel value of the center pixel, and P_n denotes the pixel value of the nth neighboring pixel.

To get the local binary encoder of the gesture image, we utilize the random binary convolution layer [29]. This method combines the convolution operation and extraction operation of LBP features, which means that the extraction of LBP features is realized with convolution operations. In this method, eight sparse convolution kernels are used to convolute the input image separately to obtain eight differential maps, and then these differential maps are activated by the nonlinear activation function to generate 8-bit maps. To implement backpropagation with the random binary convolution layer, we can use the sigmoid function or the ReLU function in this nonlinear activation function. Finally, a linear combination of learnable weights produces the response output of the final random binary convolution layer. It is expressed as follows [28]:

$$X_{l+1}^t = \sum_{i=1}^{m} \sigma \left(\sum_s b_i^s * X_l^s \right) * V_{l,i}^t \tag{3}$$

where X_l^s denotes the input image of the convolution layer. X_{l+1}^t denotes the output image of the convolution layer. $\sum_s b_i^s$ are the sparse convolutional kernels. σ is a nonlinear activation function, e.g., sigmoid, ReLU, etc. $V_{l,i}^t$ is the learnable weights of linear combination.

In detail, we first use two random binary convolution layers to extract the LBP features of the hand-gesture image. At the same time, we use a structure like residual connections to extract features from gesture images of varying classes and dimensions. As shown in Fig. 2, the radar data in the form of images are fed into the convolutional neural network with a layer of ordinary convolution layer and two layers of local binary coded convolution layer. The features of the two output parts are then combined.

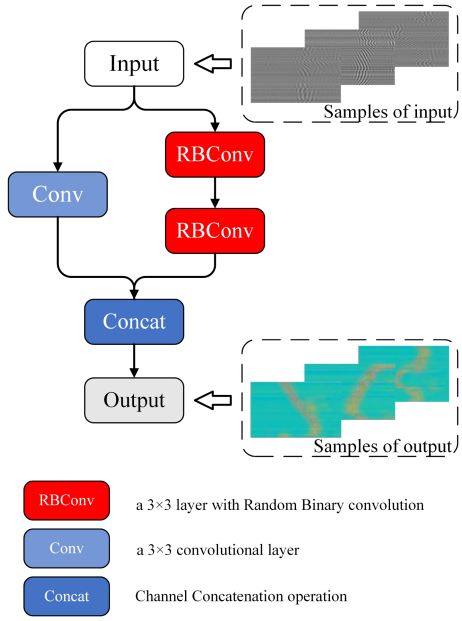

Fig. 2. Local binary encoder of the gesture image.

B. Gesture-ProxylessNAS Module

After applying the local binary encoder to the UWB-gestures dataset, we introduce the Gesture-ProxylessNAS module. We first utilize the multilayer ShuffleNet with depthwise separable convolution. To further improve network structure, we further use Gesture-ProxylessNAS module on the multilayer ShuffleNet Module. The module can directly learn the architectures with lower memory consumption and facilitate usage on target hardware platforms. The total structure of the Gesture-ProxylessNAS module is shown in Fig. 3. Before the backbone network, we use the residual structure of the random binary convolutions as the feature extraction module. After the backbone network, we add depthwise separable convolution layers to further enhance the representation capability of the network.

1) Multilayer ShuffleNet With Depthwise Separable Convolution: ShuffleNet v1 utilizes group convolutions on 1×1 convolution (also called pointwise convolution in [18]) rather than 3×3 convolution, which is conducive to improving representation capability and reducing computational cost simultaneously [16]. In addition, it employs channel shuffle operation to solve the problem of blocking the information flow between channel groups.

Motivated by ShuffleNet, we adopt the ShuffleNet unit [16] as our network's backbone unit. It is a residual block that applies pointwise group convolutions, 3×3 depthwise convolution, and channel shuffle operation in its residual branch. Using pointwise group convolution and channel shuffle operation from a ShuffleNet unit reduces the channel numbers and the computational cost. Followed by the ShuffleNet unit, 3×3 depthwise convolution enables the exchange of information between channels appropriately in the case of the computational economy and the second pointwise group convolution is used to restore the original channel dimension.

Four layers of depthwise separable convolution are added to improve the representation capability of the structure described above. The differences between these four layers of depthwise separable convolution are mainly in the input and output dimensions, and whether SE blocks are used. Compared with ordinary convolution, depthwise separable convolution is primarily divided into two steps: depthwise convolution and pointwise convolution, which makes the number of parameters and the operation cost relatively low. Intuitively, this involves splitting the ordinary convolution into two convolutions. The first convolution of $(h, w, input\_channel)$ is used to reduce the model parameters, and the second convolution of

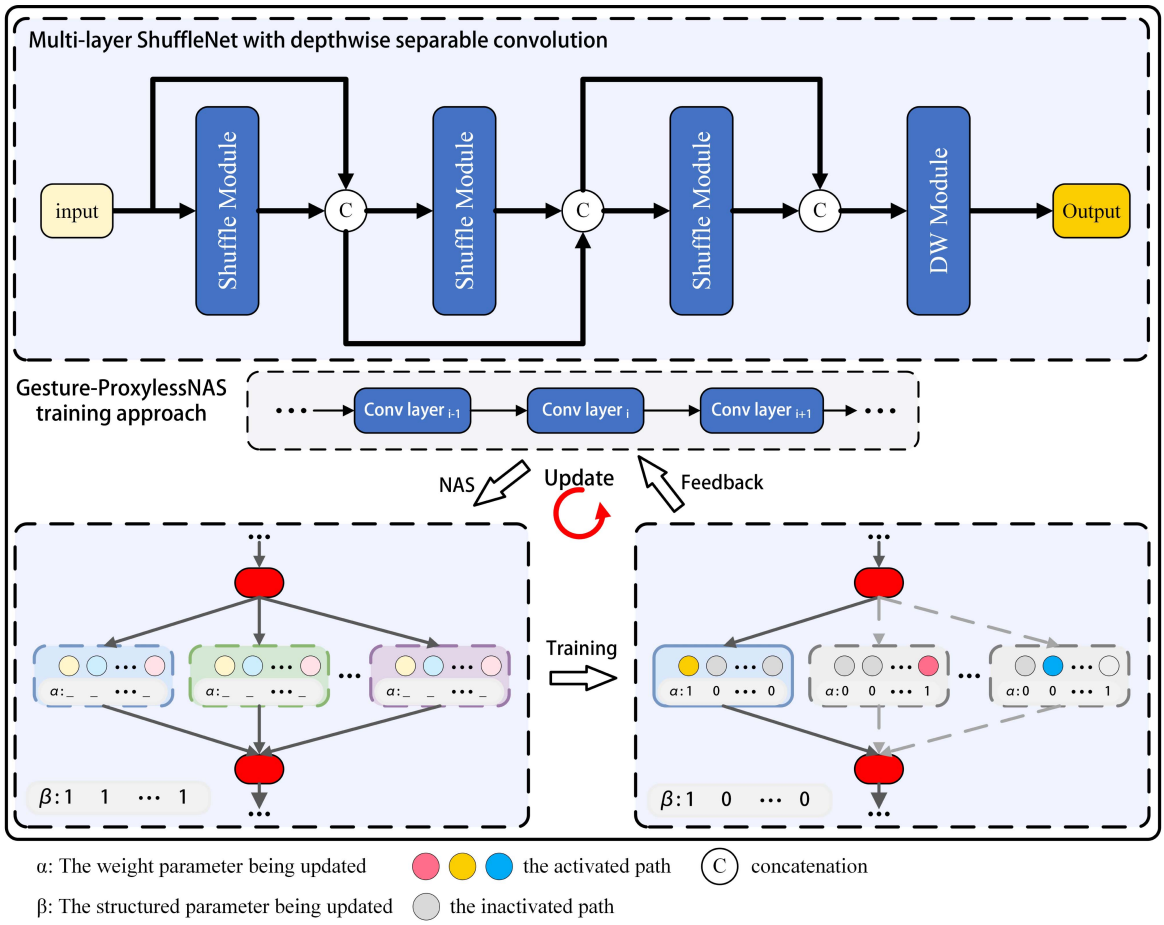

Fig. 3. Structure of the Gesture-ProxylessNAS module. The upper part is the network structure of the Gesture-ProxylessNAS module. The bottom part is a schematic diagram of the Gesture-ProxylessNAS training approach in the Gesture-ProxylessNAS module.

$(h, w, output\_channel)$ is used to increase the model's channel depth. The whole structure of multilayer ShuffleNet with depthwise separable convolution is shown in Fig. 4.

2) Gesture-ProxylessNAS Training Approach: To optimize the efficiency of the designed neural network structure, we utilize ProxylessNAS to further design effective neural network structures of multilayer ShuffleNet with depthwise separable convolution.

Neural architecture search (NAS) is an optimization algorithm for neural networks that seeks to optimize both the network's structural parameters and weight parameters [31]. NAS can be summarized as the optimization of neural network structural parameters and weight parameters. Structural parameters determine the neural network's architecture, while weight parameters determine the contribution of each structure to the output. NAS needs to filter out a relatively efficient set of parameters from a certain number of structural parameters and weight parameters through a series of selections and optimizations so that the neural network can perform better.

The NAS algorithm begins by constructing a reasonable search space composed of all structural parameters and weight parameters, forming parameterized neural network architecture. The design of the initialized search space is essential. Too large search space will complicate the subsequent optimization process, leading to high NAS training costs or even convergence issues. On the other hand, if the search space is too small, there may be no optimal set of parameters, and the network architecture will not improve significantly even after NAS training. The initialized search space must be optimized and evaluated by testing it against a suitable dataset, which corresponds to different groups of parameters. After a certain number of searches, theoretically, we can obtain a better set of parameters. However, naively traversing directly in the search space causes GPU memory explosion. Therefore, algorithms such as reinforcement learning [30], [31], evolutionary learning [32], Bayesian optimization [33], and so on can be employed to accelerate the optimization process. However, there is still room for optimization and improvement in the NAS search space described above. Unlike traditional NAS algorithms [34], [35] that directly search for large-scale task methods that result in prohibitive computational requirements, ProxylessNAS [21] addresses the issues of high memory consumption and computational cost, enabling it to directly learn the architectures for large-scale image tasks.

To enable a direct tradeoff between width and depth, we initiate an overparameterized network and allow the ProxylessNAS automatically search for a more optimal network structure and relevant parameters after training. The visualization

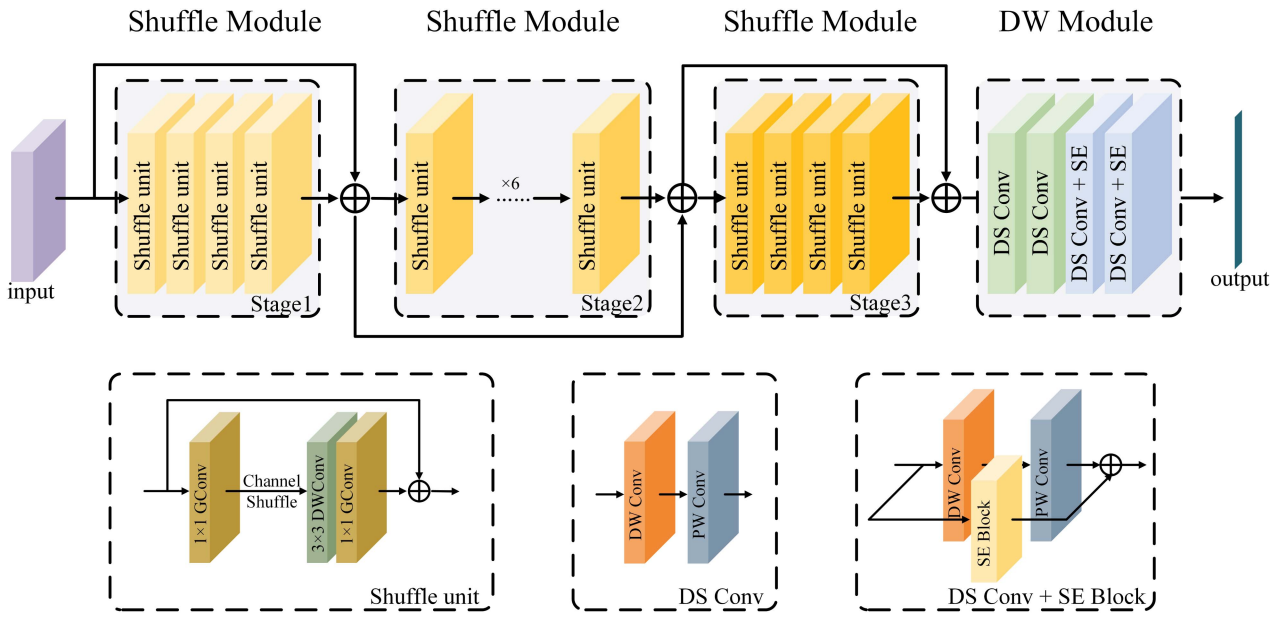

Fig. 4. Total structure of multilayer ShuffleNet with depthwise separable convolution. The network mainly consists of three shuffle modules and one depthwise module (DW module). The difference between shuffle modules is only the size of the convolution kernel and the depthwise module contains two depthwise separable convolution layers (DS Conv) and depthwise separable convolution layers with squeeze-and-excitation module (DS Conv + SE).

of the Gesture-ProxylessNAS module process is shown in Fig. 3.

We define the NAS training process as a path optimization selection problem. Each set of structural parameters and weight parameters represents a path in the search space, and the goal of NAS is to identify the optimal path. As shown in Fig. 3, precisely, for each convolution layer, the structure parameters α represent the candidate structures in that convolution layer, such as each circle node in the figure; these structures include 3×3 convolution, 5×5 convolution, pooling, etc. And the weight parameters β represent the weight coefficients in the operation process in each convolution layer.

However, the complexity of finding the optimal path in the given N candidate paths of the initialized hyperparameter network needs to be lowered, making most NAS methods achieve higher performance and bring enormous training time costs.

To reduce memory usage and computational costs, we break down the problem of choosing a path from N candidate paths into multiple binary choice problems. Thus, we can transform the real-valued path into binary gates

$$g = \text{binarize}\,(P_1, \ldots, P_N) = \begin{cases} [1, 0, \ldots, 0] & \text{with prob } P_1 \\ [0, \ldots, 0, 1] & \text{with prob } P_N. \end{cases} \tag{4}$$

Based on the binary gate g, the output of the parameter training process is given as

$$m_{\Theta}^{\text{Binary}}(x) = \sum_{i=1}^{N} g_i\, o_i(x) \begin{cases} o_1(x) & \text{with prob } P_1 \\ \ldots & \\ o_N(x) & \text{with prob } P_N. \end{cases} \tag{5}$$

As illustrated in [31] and Fig. 3, the path selection problem is defined as a binary selection problem to reduce the algorithmic complexity of network optimization. This approach ensures that only one of the paths is active during the parameter update, which simplifies the optimization process. In the optimization process of the parameter network, the two training processes mainly include weight parameters and structure parameters. When updating the structure parameters and weight parameters in each path optimization, we first freeze the structure parameter and sample binary gates according to (5) for each batch of input data. The weight parameters can be updated directly by the gradient descent method. We further use gradient-based optimization to learn structure parameters and weight parameters. The specific derivation of the gradient-based optimization method [31] is as follows:

$$\frac{\partial L}{\partial \alpha_i} = \sum_{j=1}^{N} \frac{\partial L}{\partial p_j} \frac{\partial p_j}{\partial \alpha_i} \approx \sum_{j=1}^{N} \frac{\partial L}{\partial g_j} \frac{\partial p_j}{\partial \alpha_i} \tag{6}$$

$$\sum_{j=1}^{N} \frac{\partial L}{\partial g_j} \frac{\partial p_j}{\partial \alpha_i} = \sum_{j=1}^{N} \frac{\partial L}{\partial g_j} \frac{\partial \left(\frac{\exp(\alpha_j)}{\sum_k \exp(\alpha_k)} \right)}{\partial \alpha_i} \tag{7}$$

$$\sum_{j=1}^{N} \frac{\partial L}{\partial g_j} \frac{\partial \left(\frac{\exp(\alpha_j)}{\sum_k \exp(\alpha_k)} \right)}{\partial \alpha_i} = \sum_{j=1}^{N} \frac{\partial L}{\partial g_j} p_j \left(\delta_{ij} - p_i \right). \tag{8}$$

According to (6), (7), and (8), parameters can be approximately estimated by using the gradient $w.r.t.$ and using $\partial L / \partial g_i$ in replace of $\partial L / \partial p_i$:

$$\frac{\partial L}{\partial \alpha_i} = \sum_{j=1}^{N} \frac{\partial L}{\partial p_j} \frac{\partial p_j}{\partial \alpha_i} = \sum_{j=1}^{N} \frac{\partial L}{\partial g_j} p_j \left(\delta_{ij} - p_i \right) \tag{9}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$.

Simultaneously, during each update step, one sampled path is enhanced (path weights increase), and another sampled path is

TABLE I
PERFORMANCE ON THE UWB-GESTURES PUBLIC DATASET

Model	FLOPs (M)	Params (M)	FPS (img/s)	Acc. (%)
SqueezeNet [36] †	2650	**73.60**	143.582	88.71
MobileNeXt [18] †	602.52	144.52	**160.806**	92.33
ShuffleNet v2 [17] †	583.48	92.85	155.316	92.61
MobileNet v2 [19] †	2650	**73.60**	143.582	93.43
GhostNet [20] †	846.54	268.02	98.614	95.59
EfficientNet [37] †	2900	2210.38	15.156	95.89
Ours	**595.06**	107.79	152.570	**96.52**

† means results referred from other papers.
The bold values highlight the best performance values obtained in the experiment presented in this article.

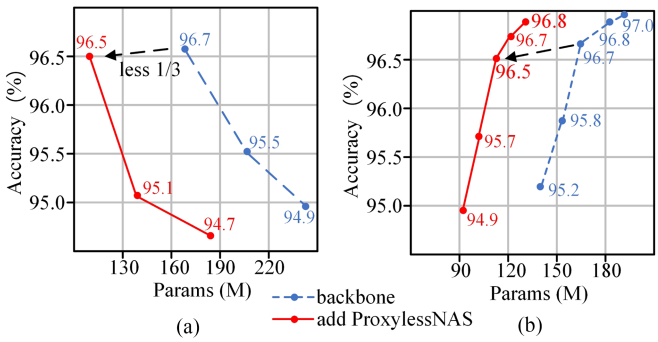

Fig. 5. Visualization of comparative experiment results performed at different experimental settings.

attenuated (path weights decrease), while all other paths remain unchanged. After each update, the algorithm prunes the paths, keeping the path with good performance and cutting out the ones with poor performance.

We optimize the network structure through the continuous update of the weight parameters and structure parameters. This approach does not increase computational complexity. Thereby, the memory requirement is reduced to the same level of training as a compact model.

V. EXPERIMENTS AND RESULTS

We mainly evaluate the proposed recognition method of hand gestures on the UWB-gestures public dataset [1]. We follow most of the standard training settings and hyperparameters.

A. Settings

The UWB-gestures public dataset consists of 12 gesture classes with 28 K training images [1]. We randomly sample 20% images from the training set as a validation set updated by using the Adam optimizer with an initial learning rate of 0.01. The overparameterized network is trained on the remaining training images with batch size 128. We train 300 epochs on the model in each part of the experiment. The configuration of the experimental equipment is $NVIDIA\ GeForce\ RTX\ 2080Ti$ with 1 GPU (12 GB) and 4 CPUs. The input image size is 224 × 224. The batch size is 32 for GPU. FLOPs row and FPS list the complexity at 224 × 224 input size.

B. Results

1) Experiment 1: The UWB-gestures public dataset [1] is taken into consideration for evaluation in this experiment. The quantitative comparison of the proposed method with cutting-edge UWB gesture classification methods is shown in Table I. With comparable numbers of flops, parameters, and FPS as MobileNeXt, our strategy increases the recognition accuracy by roughly 4%.

2) Experiment 2: For a comparable accuracy level (about 96%), GhostNet and EfficientNet must incur significant computing expenses. Specifically, GhostNet's FLOPs and parameter volume are 1.3 times higher than ours. EfficientNet's FLOPs and parameters are even more than four times ours. More importantly, our method achieves significantly higher resource efficiency, with our FPS is 1.5 times higher than GhostNet and 10 times higher than EfficientNet.

To evaluate the performance of the proposed method, the average recognition confusion matrix based on GPNAS is shown in Table II. The first column of each row represents the original class, while the first row represents the predicted class of gestures. The diagonal line values indicate the correct recognition rate of each of gesture, while the off-diagonal values indicate the false positive rate. For example, the value on (i, j) suggests the probability that the ith kind of gesture image is misclassified as the jth kind of gesture. From the confusion matrix, this model has a relatively balanced recognition accuracy rate for various gesture images (basically reaching 95%), and there is no problem with unbalanced class recognition. It turns out that GPNAS can not only achieve roughly good recognition performance but also exhibits strong robustness.

To further investigate the rationality of the network framework proposed in this article and the efficiency of the Gesture-ProxylessNAS Module, we used two series of experiments to verify. The specific experimental settings are as follows.

First, we only use the method's backbone network (combining the local binary encoder with multilayer ShuffleNet with depthwise separable convolution), without the Gesture-ProxylessNAS module. We conduct three experiments to replace the convolution kernel size in the Shuffle module, originally 3 × 3, with kernel sizes of 5 × 5 and 7 × 7. Subsequently, the Gesture-ProxylessNAS module is added to the initial three experiments to obtain six experimental results, as illustrated in Fig. 5(a).

Following the above experimental method, we carried several more experiments. The specific experimental settings are as follows. First, we use the backbone network of the method, without the Gesture-ProxylessNAS module. Three experiments are conducted to alter the number of deeply separable convolution layers from 2 to 6. Subsequently, the Gesture-ProxylessNAS module is added to the original three experiments to obtain six experimental results, as shown in Fig. 5(b).

Setting the convolution kernel sizes of ShuffleNet to 3 × 3, 5 × 5, and 7 × 7, respectively, yields the results as shown in Fig. 5(a). Among them, the experimental results obtained by setting the convolution kernel to 3 × 3 are the best (whether it

TABLE II
AVERAGE RECOGNITION CONFUSION MATRIX OF 12 GESTURES BASED ON GPNAS

	LR	RL	UD	DU	LU	LD	RU	RD	CW	AC	IP	EM
LR	**94.1**	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.4	0.1	0.0	0.0
RL	0.3	**97.2**	0.0	0.2	0.2	0.2	0.1	0.2	0.3	1.1	0.0	0.0
UD	0.4	0.0	**95.2**	0.3	0.3	0.3	0.0	0.1	0.1	0.3	0.0	0.0
DU	0.7	0.0	1.0	**94.5**	0.1	0.4	0.2	0.4	2.1	0.5	0.1	0.0
LU	1.5	0.2	0.2	1.3	**96.3**	0.2	0.3	0.2	0.5	0.0	0.0	0.0
LD	0.8	0.0	0.1	0.7	0.4	**96.5**	0.0	0.2	0.3	0.7	0.0	0.0
RU	0.2	0.2	0.0	0.5	0.0	0.0	**98.8**	0.4	0.8	0.0	0.0	0.0
RD	0.6	0.4	1.1	0.3	0.6	0.8	0.0	**97.5**	0.1	0.2	0.5	0.0
CW	0.5	0.4	0.3	0.8	0.6	0.5	0.0	0.0	**94.0**	0.6	0.0	0.2
AC	0.1	0.1	0.5	0.1	1.2	0.7	0.0	0.1	0.1	**96.3**	0.0	0.0
IP	0.2	1.3	1.4	0.5	0.0	0.3	0.0	0.0	0.0	0.2	**99.2**	0.4
EM	0.6	0.0	0.1	0.8	0.8	0.1	0.4	0.5	0.0	0.0	0.0	**98.4**
Overall recognition accuracy												**96.5**

The first column of each row represents the original class, whereas the first row represents the predicted class of gestures. The values on the diagonal line indicate the correct recognition rate of each class of gesture, while the values elsewhere indicate the false positive rate, for example: the value on (i, j) indicates the probability that the ith kind of gesture image is misjudged as the jth kind of gesture.
The bold values highlight the best performance values obtained in the experiment presented in this article.

is the backbone or the addition of the Gesture-ProxylessNAS module). At the same time, the accuracy of experimental results using 5×5 convolutional kernel size ranks second, and 7×7 experimental results ranks last. It demonstrates that too large a convolution core is not conducive to gesture recognition for gesture images. The horizontal comparison makes it clear from the graphic that the Gesture-ProxylessNAS module performs exceptionally well. Whether it is a 3×3, 5×5, or 7×7 convolution kernel, the model's accuracy is unchanged after adding the Gesture-ProxylessNAS module, and its parameter amount is significantly reduced.

Similar to this, setting the number of layers of depth-separable convolution layers to 2, 3, 4, 5, and 6 yields the result shown in Fig. 5(b). The chart shows that as the number of deep separable convolution layers rises, so does the number of parameters and the recognition accuracy. However, when the number of convolution layers is increased to 5 and 6, the model's price/performance ratio (the ratio of recognition accuracy and parameter quantity) is much smaller than that of 2, 3, and 4 layers. According to this perspective, the model can perform at its best when using four convolution layers. This also verifies the rationality and high performance of the method proposed in this article.

Similarly, in the horizontal comparison, it is clear from the figure that the performance of the Gesture-ProxylessNAS module is excellent. Regardless of the number of depth-separable convolution layers (ranging from 2 to 6), the module consistently maintains the model's precision while significantly reducing the parameter count.

C. Ablation Study

The proposed recognition method of hand gestures consists of three parts: LBP feature extraction, recognition networks, and Gesture-ProxylessNAS training methods. In this section, we evaluate them, respectively. We perform our ablation experiments on the UWB-gestures public dataset [1]. As shown in Table III, based on the backbone, the introduction of deep separable convolution layers can immensely improve the gesture recognition accuracy of the network (around 3%) but also introduces many parameters and FLOPs.

TABLE III
PERFORMANCE OF ABLATION STUDY ON UWB-GESTURES PUBLIC DATASET. MOST OF THE SETTINGS FOR THIS EXPERIMENT ARE THE SAME AS THOSE DESCRIBED ABOVE

Module	FLOPs (M)	Params (M)	FPS (img/s)	Acc. (%)
Shuffle Modules	583.48	92.85	155.316	92.61
+ DW Module	635.24	135.26	134.815	95.61
+ LBP encoder	690.17	165.32	124.781	96.69
+ Gesture-ProxylessNAS	595.06	107.79	152.570	96.52

VI. DISCUSSION

UWB gesture recognition is often used in scenarios with high real-time requirements and weak hardware device performance, such as in-vehicle interaction and smart homes. Despite this, the existing literature still primarily focuses on recognition accuracy as the sole metric to evaluate algorithm performance. To address this, we propose combining recognition accuracy with additional metrics such as parameter quantity, FLOPs, and FPS, to measure model lightness and real-time performance. And based on these metrics, we propose a lightweight network for detecting mid-air gesture based on UWB radar. Based on the lightweight network architecture multilayer ShuffleNet with depthwise separable convolution, it employs LBP encoders and adopts the model compression method named Gesture-ProxylessNAS, which improves the accuracy and efficiency of gesture recognition. Our

experiments demonstrate that our method outperforms previous studies on different gesture recognition algorithms [1] and lightweight neural networks [17], [18], [19], [20], [36], [37]. Although the performance of this method in the experiment is superior to other methods at present, there are still limitations. First, due to the limitations of dataset, this method only verifies the performance in an open dataset [1]. Second, after the introduction of Gesture-ProxylessNAS module, the recognition accuracy of this method still experiences a small reduction. Thus, future iterations of UWB gesture recognition data collection and model compression methods may have even greater potential.

VII. Conclusion

This article proposes a lightweight deep-learning network framework for UWB gesture recognition named GPNAS. By utilizing the advantages of multilayer ShuffleNet and Gesture-ProxylessNAS module, GPNAS learns spatial information through lightweight structure. Therefore, to improve the embedding of the textural features of the UWB picture, spatial features are encoded with the LBP module before multiscale features are retrieved with ShuffleNet. In order to completely use the feature information across different layers and enhance the recognition impact, multilayer ShuffleNet is used in the recognition stage to extract the solid complementary information between various hierarchical structures. Additionally, the Gesture-ProxylessNAS module reexamines the design spaces of the multilayer ShuffleNet with an optimization problem, which dramatically reduces the parameters and computational complexity of the network. Generally, the proposed GPNAS framework is an efficient, lightweight deep network architecture that can provide satisfactory recognition accuracy and performance with fewer parameters. The recognition results of real UWB-gestures datasets confirm the feature generalization ability of the proposed GPNAS. Future studies will focus on improving network recognition performance by creating a lighter, more effective network framework using an autonomous architectural search algorithm.

References

[1] S. Ahmed, D. Wang, J.-Y. Park, and S. H. Cho, "UWB-gestures, a public dataset of dynamic hand gestures acquired using impulse radar sensors," *Sci. Data*, vol. 8, 2021, Art. no. 102.

[2] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, pp. 1–54, 2015.

[3] G. Plouffe and A.-M. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 2, pp. 305–316, Feb. 2016.

[4] N. Ren, X. Quan, and S. H. Cho, "Algorithm for gesture recognition using an IR-UWB radar sensor," *J. Comput. Chem.*, vol. 4, no. 3, pp. 95–100, 2016.

[5] W.-Y. Kim and D.-H. Seo, "Radar-based human activity recognition combining range—time—doppler maps and range—distributed—convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 1002311.

[6] S. Franceschini et al., "Hand gesture recognition via radar sensors and convolutional neural networks," in *Proc. IEEE Radar Conf.*, 2020, pp. 1–5.

[7] F. Qi et al., "Generalization of channel micro-doppler capacity evaluation for improved finer-grained human activity classification using MIMO UWB radar," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 11, pp. 4748–4761, Nov. 2021.

[8] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 5, pp. 1328–1337, May 2009.

[9] O. H. Y. Lam, R. Kulke, M. Hagelen, and G. Mollenbeck, "Classification of moving targets using mirco-doppler radar," in *Proc. Int. Radar Symp.*, 2016, pp. 1–6.

[10] L. Zhang, J. Xiong, H. Zhao, H. Hong, X. Zhu, and C. Li, "Sleep stages classification by CW doppler radar using bagged trees algorithm," in *Proc. IEEE Radar Conf.*, 2017, pp. 0788–0791.

[11] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3278–3289, Apr. 2018.

[12] G. Li, S. Zhang, F. Fioranelli, and H. Griffiths, "Effect of sparsity-aware time–frequency analysis on dynamic hand gesture classification with radar micro-doppler signatures," *IET Radar Sonar Navigation*, vol. 12, no. 8, pp. 815–820, 2018.

[13] S. Skaria, A. Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna doppler radar with deep convolutional neural networks," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3041–3048, Apr. 2019.

[14] D. Miao, H. Zhao, H. Hong, X. Zhu, and C. Li, "Doppler radar-based human breathing patterns classification using support vector machine," in *Proc. IEEE Radar Conf.*, 2017, pp. 0456–0459.

[15] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-doppler signatures with convolutional neural network," *IEEE Access*, vol. 4, pp. 7125–7130, 2016.

[16] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.

[17] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[18] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[19] A. Souid, N. Sakli, and H. Sakli, "Classification and predictions of lung diseases from chest X-rays using mobilenet V2," *Appl. Sci.*, vol. 11, no. 6, 2021, Art. no. 2751.

[20] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1580–1589.

[21] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," 2018, *arXiv:1812.00332*.

[22] S. K. Leem, F. Khan, and S. H. Cho, "Detecting mid-air gestures for digit writing with radio sensors and a CNN," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1066–1081, Apr. 2020.

[23] N. Hendy, H. M. Fayek, and A. Al-Hourani, "Deep learning approaches for air-writing using single UWB radar," *IEEE Sensors J.*, vol. 22, no. 12, pp. 11989–12001, Jun. 2022.

[24] S. Skaria, A. Al-Hourani, and R. J. Evans, "Deep-learning methods for hand-gesture recognition using ultra-wideband radar," *IEEE Access*, vol. 8, pp. 203580–203590, 2020.

[25] Z. Xia, Y. Luomei, C. Zhou, and F. Xu, "Multidimensional feature representation and learning for robust hand-gesture recognition on commercial millimeter-wave radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4749–4764, Jun. 2021.

[26] B. Li, J. Yang, Y. Yang, C. Li, and Y. Zhang, "Sign language/gesture recognition based on cumulative distribution density features using UWB radar," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 2511113.

[27] S. Franceschini, M. Ambrosanio, V. Pascazio, and F. Baselice, "Hand gesture signatures acquisition and processing by means of a novel ultrasound system," *Bioengineering*, vol. 10, no. 1, 2023, Art. no. 36.

[28] M. Pietikinen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision Using Local Binary Patterns*, vol. 40. Berlin, Germany: Springer, 2011.

[29] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, "Local binary convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 19–28.

[30] B. Baker et al., "Designing neural network architectures using reinforcement learning," 2016, *arXiv:1611.02167*.

[31] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*.

[32] E. Real et al., "Large-scale evolution of image classifiers," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2902–2911.

[33] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," 2018, *arXiv:1806.09055*.

[34] G. Bender, P.-J. Kindermans, B. Zoph, V. K. Vasudevan, and Q. V. Le, "Understanding and simplifying one-shot architecture search," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 550–559.

[35] A. Brock et al., "Smash: One-shot model architecture search through hypernetworks," 2017, *arXiv:1708.05344*.
[36] F. N. Iandola et al., "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size," 2016, *arXiv:1602.07360*.
[37] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

Lihong Qiao received the B.Sc., M.Sc. and the Ph.D. degrees in applied mathematics from Hebei Normal University, China, in 2004, 2007, and 2010, respectively.

She is currently an Associate Professor with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China. Her research interests include medical image processing.

Zhixin Li is currently studying bachelor of engineering degree in communication engineering (bachelor's degree in progress) with Chongqing University of Posts and Telecommunications, Chongqing, China.

His research interest is medical image processing.

Bin Xiao received the B.S. and M.S. degrees in electrical engineering from Shanxi Normal University, Xi'an, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from Xidian University, Xi'an, in 2012, and conducted postdoctoral research at the University of Queensland in Australia in 2015.

He is currently a Professor with Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include image processing and pattern recognition.

Yucheng Shu received the M.S. and Ph.D. degrees in computer vision from the School of Software Engineering and School of Computer Science and Technology with Huazhong University of Science and Technology, in 2011 and 2015, respectively.

He is currently an Associate Professor with Chongqing University of Posts and Telecommunications and Chongqing Key Laboratory of Image Cognition. His research mainly focuses on computer vision, machine learning, and medical image processing.

Weisheng Li (Member, IEEE) received the B.S. degree in signal processing from the School of Electronics and Mechanical Engineering, Xidian University, Xi'an, China, in 1997, and the M.S. and Ph.D. degrees in signal processing from the School of Electronics and Mechanical Engineering and the School of Computer Science and Technology, Xidian University, in 2000 and 2004, respectively.

He is currently a Professor with Chongqing University of Posts and Telecommunications. His research focuses on intelligent information processing and pattern recognition.

Xinbo Gao (Senior Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees in signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively.

From 1997 to 1998, he was a Research Fellow with the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a Postdoctoral Research Fellow with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. He was with the School of Electronic Engineering, Xidian University, from 2001 to 2020. Since 2020, he has been the President of Chongqing University of Posts and Telecommunications, Chongqing, China. His current research interests include multimedia analysis, computer vision, pattern recognition, machine learning, and wireless communications. He has published five books and approximately 200 technical articles in refereed journals and proceedings, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, the *International Journal of Computer Vision*, and *Pattern Recognition* in the above areas. He is a Fellow of the Institution of Engineering and Technology. He has served as the General Chair/Co-Chair, the Program Committee Chair/Co-Chair, or a PC Member for approximately 30 major international conferences. He is on the Editorial Boards of several journals, including *Signal Processing* (Elsevier) and *Neurocomputing* (Elsevier).