This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

Variable Curvature Gabor Convolution and Multi-Branch Structures for Finger Vein Recognition

Jun Li , Huabin Wang , Shicheng Wei , Jian Zhou , Yuankang Shen , and Liang Tao

***Abstract*—Gabor filters are able to extract texture features from finger vein images from different directions and scales. However, manually crafted Gabor filters have problems such as relatively single direction and scale, and difficulties in parameter adjustment to adapt to specific datasets. To solve these problems, this paper proposes a neural network with a learnable variable curvature Gabor (VC-Gabor) convolutional layer. Firstly, the Gabor filter is improved by adding variable curvature to extract information about different curvature degrees in the vein curves. Secondly, the VC-Gabor filter is designed as a learnable convolutional filter, with parameters updated using neural network back-propagation. This facilitates the enrichment of learned VC-Gabor filter directions, scales, and curvatures, eliminating the need for intricate manual parameter tuning. Finally, we propose adaptive multi-branch structures for feature extraction, which are used to enhance the feature extraction capability of the model. Experimental results on publicly available datasets FV-USM and SDUMLA demonstrate that the proposed algorithm improves recognition accuracy and reduces the Equal Error Rate (EER), thereby substantiating the effectiveness of the approach.**

***Impact Statement*—Finger vein images are susceptible to light variations, resulting in poor image quality, which makes their recognition ineffective. Given that the line features of veins are obvious and stable, and the Gabor filter can effectively extract such line features, some researchers have used the Gabor filter for finger vein feature extraction. However, the traditional Gabor filter has a single directional scale and it is difficult to adjust the parameters to fit all the datasets. The VC-Gabor filter proposed in this paper improves the traditional Gabor filter in order to extract more obvious curvature features of vein curves, and uses a neural network to train the parameters of the filter, which avoids manual parameter tuning, and the method achieves good results on several finger vein datasets.**

***Index Terms*—VC-Gabor filter, multi-branch structure, finger vein recognition, convolutional neural network**

I. Introduction

IDENTITY authentication is to complete the confirmation of the user's identity through certain technical means. In recent years, traditional authentication methods, such as keys, documents and passwords, which are prone to cumbersome operation, easy loss, forgery, fraud and other defects, have been gradually replaced by biometrics. Commonly used biometrics, such as fingerprint [1], face [2], iris, [3] and voice [4] are widely used and researched due to their ease of use and rich identity information. However, as an emerging biometric identification technology, finger vein characteristics have the unique advantages of requiring in vivo detection and being distributed under the skin [5] [6]. As a result, it is not easy to be forged and has a wide range of application prospects.

Finger vein recognition generally includes the steps of vein image acquisition, vein image preprocessing, feature extraction, and feature matching [7]. Finger vein image acquisition is the process of illuminating the finger with near-infrared light of a specific wavelength to image the vein structure of the finger and capture the image [8]. Image preprocessing is done to remove the background information and obtain the Region of Interest (ROI) image. Feature extraction is to extract characteristic information from finger vein images that can describe and distinguish individual differences. Feature matching is to retrieve the identity with the highest match to the current feature in the feature database to complete the user identification. Since finger vein images are easily affected by finger pose, lighting, and other factors, resulting in poor image quality [9], how to extract effective features from low-quality finger vein images is the most critical step in finger vein recognition.

The use of Gabor filters for finger vein texture feature extraction is a time-honored method because they capture texture information and orientation information more efficiently than other filters. Yang *et al.* [10] proposed a finger vein extraction method based on the combination of a Gabor wavelet and a circular Gabor filter, which uses the Gabor wavelet to enhance the image of the vascular regions, which makes the finger vein network obviously prominent and eliminates the non-vascular regions. Xie *et al.* [11] proposed a bootstrap Gabor filter to extract finger vein features, which takes full advantage of the vein enhancement and denoising of the bootstrap filter and improves the recognition performance for low-quality images caused by factors such as low-contrast, illumination, or noise. Wang *et al.* [12] proposed a method to extract the finger vein features using a multiscale variable curvature Gabor filter to extract the features of the midline of finger veins, which extracts the features of the degree of curvature of the vein curve in the vein image by improving the Gabor filter and adding the curvature information. These methods

This work were supported by the National Natural Science Foundation of China (Grant No. 62106005) and the Natural Science Foundation for the Higher Education Institutions of Anhui Province (Grant No. 2022AH050091). (*Corresponding author: Huabin Wang*)

Jun Li, Huabin Wang, Jian Zhou, Yuankang Shen, and Liang Tao are with Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University, Hefei, Anhui, China 230601 (e-mail: e21301313@stu.ahu.edu.cn; wanghuabin@ahu.edu.cn; jzhou@ahu.edu.cn; e22301229@stu.ahu.edu.cn; taoliang@ahu.edu.cn)

Shicheng Wei is with School of Mathematics, Physics and Computing, University of Southern Queensland, Toowoomba, Australia(e-mail: Shicheng.Wei@usq.edu.au)

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

are effective in extracting texture and orientation information from finger vein images. However, the features of these filters are manually designed and have the disadvantage of being relatively homogeneous in terms of orientation, scale, and curvature, and the parameters of the manual modulation are difficult to optimize and are not robust to variations in the dataset.

The emergence of convolutional neural networks avoids complex manual parameterization and allows the extraction of deep features. In recent years, some researchers have applied convolutional neural networks to finger vein recognition with good results. For example, Radzi *et al.* [13] used a convolutional neural network for finger vein image feature extraction, they designed a four-layer convolutional CNN, thresholding the ROI of finger vein image to get a binary image as the input of the CNN, and feature extraction of the binary image with a four-layer convolutional neural network. Huang *et al.* [14] designed Deep-Vein, a deep convolutional neural network, the method stacks convolutional layers and pooling layers to deepen the layers of the network in the feature extraction stage, which improves the expressive power of the model by increasing the number of parameters of the model. Zhang *et al.* [15] designed an adaptive Gabor convolutional neural network for finger vein recognition, which replaces all the convolutional layers of the neural network in each layer with Gabor convolution in order to utilize the properties of Gabor filters to extract orientation and scale features of the veins. These depth-based methods train models suitable for finger vein feature extraction with a given dataset, but still have the following two problems: first, finger veins have rich line features, and the above methods are difficult to extract information about the degree of curvature of the vein curves; second, it is not conducive to the extraction of deeper features when the network hierarchy is shallow, and overfitting is prone to occur when stacking the number of convolutional layers in order to increase the depth of the network for a small-sized dataset such as finger veins.

In order to solve the above problems, this study proposes a neural network featuring learnable VC-Gabor convolution and multi-branch structures for finger vein recognition. Firstly, to capture curvature degree information from vein curves, we redesign the Gabor filter to incorporate variable curvature. Secondly, we integrate the VC-Gabor filter into a neural network to form the VC-Gabor convolution layer. This approach addresses the limitations of manually designed Gabor filters, such as uniformity in orientation, scale, and curvature, difficulty in parameter tuning, and vulnerability to dataset transformations. Finally, to address the insufficient feature representation due to the limited number of convolutional layers in the above model and the overfitting problem arising from model deepening, this study proposes a feature extraction module with a multi-branch structure. In this multi-branch structure, a shortcut branch is introduced to alleviate the gradient vanishing problem caused by deepening the model. The main contributions of this study are summarized as follows:

1) We propose a new VC-Gabor convolutional and multi-branch feature extraction structured neural network for finger vein recognition based on the characteristics of the finger vein dataset. Traditionally, Gabor-based feature extraction and CNN-based feature extraction are two separate steps. In contrast, our proposed neural network incorporates an improved Gabor filter into a convolutional neural network, which improves finger vein recognition and reduces the equal error rate.
2) We propose a variable curvature Gabor function based on the linear features of finger vein images and construct a learnable VC-Gabor convolutional layer. The proposed VC-Gabor filter has variable curvature and is incorporated into a neural network to extract rich vein line features in finger vein images. Conventional Gabor filters require manual parameterization, and Gabor filters have a single orientation and scale, making it difficult to extract rich features. In contrast, our proposed VC-Gabor filter with variable orientation, scale, and curvature is designed to be able to extract rich finger vein features.
3) We design adaptive multi-branch feature extraction structures based on the small size of the finger vein dataset. The proposed multi-branch structures use a two-branch structure for feature extraction at locations where downsampling is required, and a three-branch structure for feature extraction at locations where downsampling is not required. The traditional single-branch structure has relatively poor feature extraction capability and is prone to overfitting. In contrast, using these multi-branch structures can improve the feature extraction ability of the model and avoid overfitting.

II. Related Work

Over the past decades, researchers have investigated many effective finger vein feature extraction methods, which can be broadly classified into four categories: vein pattern-based methods, local feature-based methods, global feature-based methods, and convolutional neural network-based methods.

A. Vein Pattern-based Methods

Vein pattern-based methods utilize the shape and location information of the finger vein vessels as the basis for vein recognition. Miura *et al.* [16] proposed an algorithm based on maximum curvature vein pattern, which calculates the curvature values of the cross-sectional contours in different directions on the finger vein image, and distinguishes between the veins and the background information by the magnitude of the curvature values. Huang *et al.* [17] proposed the wide line detector algorithm (WLD) for finger vein feature extraction, which compares the pixel value of each pixel point with its circular neighboring points, and extracts the points on the vein line if the pixel value of the point is higher than a fixed threshold in the local region. Yang *et al.* [18] designed a set of Gabor filter banks dedicated to the analysis of finger veins for constructing the finger vein feature vectors to characterize the finger vein features at two filter scales, and then fused the two features to improve the reliability of recognition. These methods can improve image quality and recognition performance, however, it is difficult to effectively extract the

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

vein pattern in the blurred region of irregular shadows in finger vein images.

B. Local Feature-based Methods

The local feature-based approach extracts features from the local regions of the finger vein image chunks, and then these local features are fused together as the final features. Lee *et al.* [19] designed the weighted LBP method which extracts the LBP code from the 3×3 local region, and then the extracted LBP features are classified into three classes using an SVM classifier and different weights are assigned to each class. The method avoids the interference of irregular background information in the image. Rosdi *et al.* [20] proposed a local line binary pattern (LLBP) descriptor, which is different from the LBP descriptor in that the shape of the neighborhood of the method is not a square but a line, which is able to accurately describe the orientation information of the finger veins to obtain more stable features. Meng *et al.* [21] proposed the local direction coding (LDC) method, which extracts the rich orientation information hidden in the finger vein patterns, and firstly divides the finger vein region according to the luminance difference, and then obtains the orientation information of the finger veins. The method of local feature extraction needs to define the local neighborhood and radius size during the feature extraction process, and different neighborhood and radius choices may lead to different texture features being extracted, which needs to be adjusted and optimized according to the specific problem. This may require a certain amount of experience and expertise, increasing the subjectivity and complexity of the method.

C. Global Feature-based Methods

Global feature-based methods represent the entire data sample as a fixed-length vector, usually downscaling high-dimensional data and mapping the data into a low-dimensional space for processing. This method can effectively reduce the time and space complexity of data processing while improving the performance of the classifier. Typical methods are principal component analysis (PCA) [22], 2DPCA [23] and linear discriminant analysis (LDA) [24]. These algorithms transform the texture image to different subspaces and use the subspace coefficients (features) generated by subspace learning as discriminative features in the recognition matching stage. Wu *et al.* [25] proposed a hybrid feature extraction method integrating Principal Component Analysis (PCA) and LDA, which uses PCA and LDA to retain the main features and remove the noise. This method has the advantage of fast recognition speed due to the low feature dimension, but the global features are highly affected by pose, occlusion, deformation, and illumination, which are not suitable for extraction as finger vein image features.

D. Convolutional Neural Network-based Methods

Convolutional neural networks have been rapidly developed in the field of image classification in recent years, and some researchers have applied convolutional neural networks to finger vein recognition with good results. For example, Hu *et al.* [26] proposed the FV-Net network model, in which the first 7 layers of convolution use the parameters of the VGGFace-Net pre-trained model, and the latter 3 layers of convolutional layers are specially designed to extract semantic features for finger vein spatial information, and the method achieves excellent results on multiple datasets. Fang *et al.* [27] proposed a lightweight two-stream network to merge two images into a two-channel sample, extract features and pass the output to a similarity metric network, and finally input the extracted features into SVM for classification. Wang *et al.* [28] proposed a finger vein recognition method based on a multisensory field bilinear convolutional neural network, and devised a Dimensionally Interactive Attention Mechanism (DIAM) that enhances the correlation between channel and space, which improves the accuracy of finger vein recognition. Zhang *et al.* [12] combined deep learning with Gabor filters and proposed an adaptive Gabor convolutional neural network with receptive fields (AGCNN) for finger vein recognition, where each convolutional layer of the neural network is replaced with Gabor convolutional layer to extract vein features.

The above analysis shows that the above finger vein feature extraction methods still have many limitations. For example, the extracted features in the methods of [16]- [21] are hand-crafted, and although they have a high response speed in the recognition phase, the hand-crafted features can be considered superficial, sensitive to noise, difficult to modulate the parameters, and not robust to transformations of the dataset. The method proposed in [12] combines deep learning with Gabor filters to avoid complex manual parameter modulation and extracts multi-directional and multi-scale vein features, but the direction and degree of curvature are equally important to represent the structure of finger veins. Therefore, in this paper, we combine the curvature information and direction scale information to construct variable curvature Gabor filters, which are integrated into convolutional neural networks, and optimize the parameters of the VC-Gabor filters by calculating the loss function and back-propagation, without manually modulating the parameters of the filters, so that the VC-Gabor filters have multiple scales, directions, and curvature information, and have a better capability of feature extraction.

III. Methodology

In this chapter, the structure and advantages of the network model proposed in this paper are described in detail, including the VC-Gabor function, VC-Gabor convolution layer, and multi-branch network structures.

A. Network Framework

In this paper, a convolutional neural network based on variable curvature Gabor filter and multi-branch structures is proposed for finger vein image recognition. The overall structure of the network is shown in Fig. 1, and the input is a grayscale image of the finger vein to be trained. The network consists of four main parts: a VC-Gabor convolutional layer, multi-branch feature extraction structures, a fully connected layer, and a Softmax classification layer. First, the same as

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

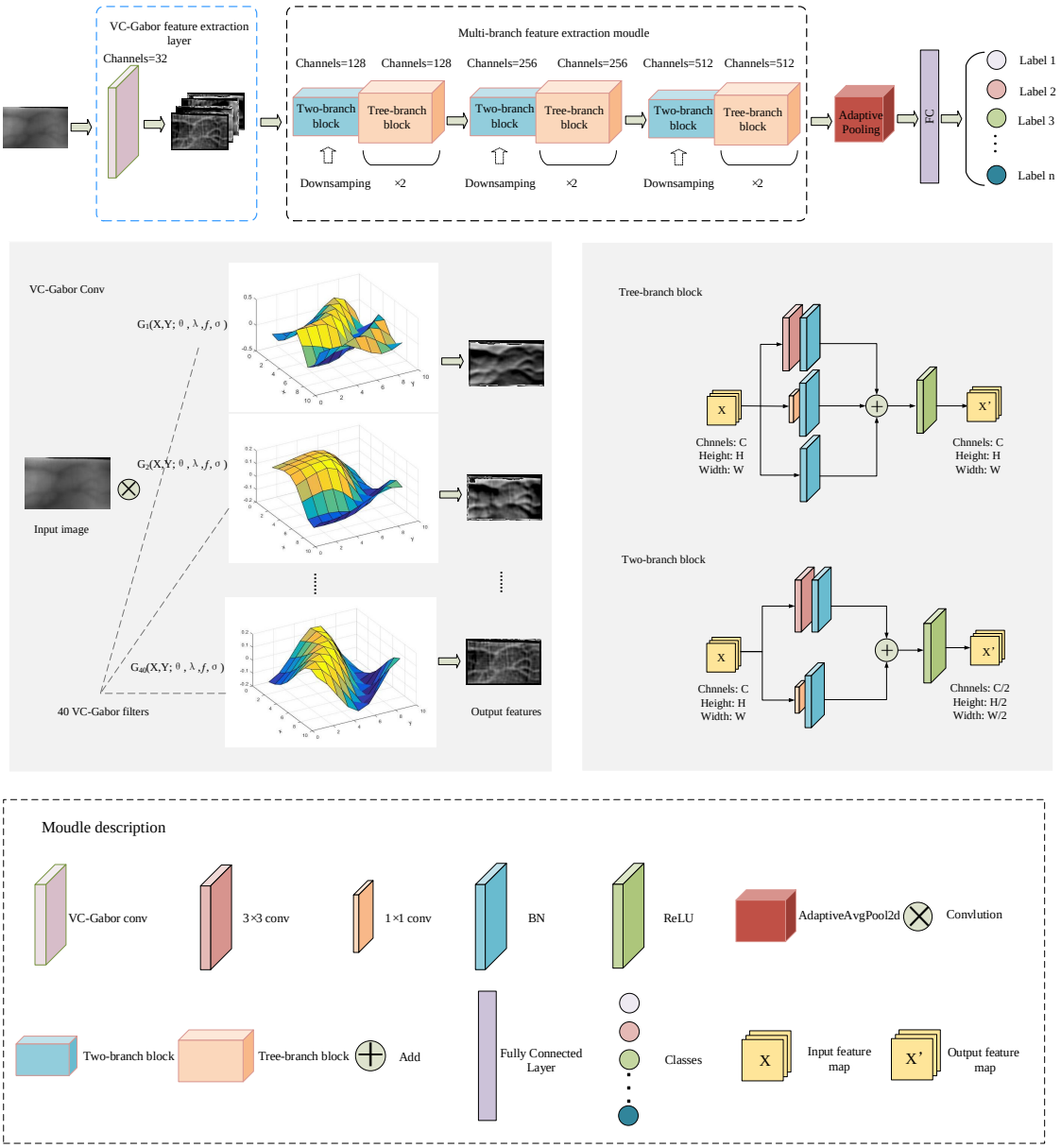


Fig. 1. The general architecture and detailed design of a network with VC-Gabor convolution and multi-branch feature extraction structures for finger vein recognition tasks.

the traditional CNN, all the weights of the convolutional layer and the fully connected layer can be learned by the neural network, but the difference is that in the first layer of the network, the traditional convolutional filters are replaced by the VC-Gabor convolutional filters that we designed, and the only part that needs to be learned is the parameters of the VC-Gabor filters. Secondly, after the VC-Gabor layer, multi-branch network structures are used for feature extraction. According to the existing theoretical basis, the ability of the multi-branch structure for feature extraction is better than that of the single-branch structure. In the part that needs downsampling, the two-branch structure as shown in the figure is used, and the width and height of the output feature map become half of the original. While in the part that does not need downsampling, the three-branch structure as shown in the figure is used, and at this time, the width and height of the feature map remain unchanged, and the specific implementation method is described in the following multi-branch structure feature extraction. Subsequently, the feature map is mapped to a feature vector of 512 dimensions by the adaptive maximum pooling downsampling layer, and this feature vector is connected to the fully connected layer. Finally, the raw output of the fully connected layer is converted into a probability distribution to perform multi-class classification of finger vein images.

B. VC-Gabor Filter

Gabor filter [29] is an efficient tool for spatially localized texture feature extraction. It is essentially a sinusoidal plane wave modulated by a Gaussian envelope, which is capable of localized and directional frequency analysis of two-

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

dimensional information to effectively extract texture information. With proper tuning, the Gabor filter can capture texture features in a specified direction and scale. The combination of multi-channel filtering allows Gabor filters with different parameters to extract localized texture features from finger vein images at different orientations and resolutions. In addition, the Gabor filters are insensitive to illumination changes, and therefore can effectively cope with the complex illumination changes that occur during the acquisition of finger vein images. Therefore, Gabor filters are widely used in traditional finger vein feature extraction methods.

By analyzing the finger vein image data, we can observe that the image contains rich vein texture information, which is very important to distinguish different types of images, so we should focus on the information of the vein region and ignore the background information when extracting the feature of the vein image. The traditional Gabor filter can not effectively extract the linear features of the vein with different curvatures, so this paper proposes a VC-Gabor filter to construct the finger vein image features at different scales, considering both the direction and the curvature of the vein curve. The VC-Gabor function is defined as follows:

$$G(x,y;\theta,\lambda,f,\sigma) = \frac{1}{2\pi\sigma^2}\cdot e^{-\frac{x^2+y^2}{2\sigma^2}}\cdot e^{2\pi i\left(\frac{1}{\lambda}(x\cos\theta+y\sin\theta)+f\sqrt{x^2+y^2}\right)} \quad (1)$$

In the formula, x and y are pixel coordinate positions; i denotes the unit of the imaginary number; θ is that angle between the sine function direction and the x-axis, namely the direction of the VC-Gabor kernel function, and the sine wave direction of the filter can be changed through adjustment θ , so that multi-direction detection is realized; λ are wavelengths of sine functions, and the wavelength of the plane wave of the VC-Gabor filter can be changed by adjustment λ, so that multi-scale detection can be realized; f is the curvature of the VC-Gabor filter, and the adjustment f can change the curvature of the filter, so as to realize the feature extraction of different bending degrees of the vein curve; σ is the standard deviation of the Gaussian function. The parameters of VC-Gabor filter banks have an important influence on their feature extraction ability. In order to obtain a more comprehensive feature representation of the image, a number of VC-Gabor filters with different parameters are usually combined into a filter bank to extract texture features of different scales, different directions, and different bending degrees.

The VC-Gabor function proposed in this paper combines a complex exponential function and a Gaussian function, which can be split into a real part and an imaginary part. The real part of the VC-Gabor function is the product of a Gaussian function and a cosine function, which responds significantly to edges and contours in an image and can be used for texture characterization. The formula is defined as follows:

$$G_{re}(x,y;\theta,\lambda,f,\sigma) = \frac{1}{2\pi\sigma^2}\cdot e^{-\frac{x^2+y^2}{2\sigma^2}}\cdot \cos\left(2\pi\left(\frac{1}{\lambda}(x\cos\theta+y\sin\theta)+f\sqrt{x^2+y^2}\right)\right) \quad (2)$$

The imaginary part of the VC-Gabor function is the product of a Gaussian function and a sine function, which excels in local feature extraction. The formula is defined as follows:

$$G_{im}(x,y;\theta,\lambda,f,\sigma) = \frac{1}{2\pi\sigma^2}\cdot e^{-\frac{x^2+y^2}{2\sigma^2}}\cdot \sin\left(2\pi\left(\frac{1}{\lambda}(x\cos\theta+y\sin\theta)+f\sqrt{x^2+y^2}\right)\right) \quad (3)$$

C. Learnable VC-Gabor Convolutional Layer

The traditional methods for VC-Gabor filter parameter setting are mostly based on experience, which can not guarantee the optimal parameters, nor can it adjust the parameters when the data set changes, resulting in poor robustness of the algorithm. The advantage of CNN is that it can obtain efficient feature extraction ability through a large amount of data training. It is found that among the filter parameters learned by CNN training, the first layer of convolutional layer parameters are similar to the Gabor filter kernel functions [29]. Inspired by this, we generate the convolutional filter from the VC-Gabor function and use CNN to learn the parameters of the VC-Gabor filter. Therefore, in order to make full use of the characteristics of the VC-Gabor filter and convolutional neural network, we use the real part of the VC-Gabor function to construct the VC-Gabor convolutional layer instead of the traditional convolutional layer, and use a convolutional neural network to learn the parameters of VC-Gabor filter and finally use the learned VC-Gabor convolution filter to extract the direction, scale and curvature features of the finger vein image texture.

Since VC-Gabor does not use learnable weights directly on the pixel space, the parameters learned in the VC-Gabor filter are different from the traditional convolution kernel parameters. Each pixel of conventional convolution kernels is the learnable parameter, for example, for a convolution kernel with kernel_size, the amount of parameters to be learned is K^2, as shown in Fig. 2(a). As for the VC-Gabor kernel, the parameters of the VC-Gabor convolutional kernel are invariant regardless of the kernel_size of the filter, and only θ,λ,f,σ are learnable parameters, as shown in Fig. 2(b).

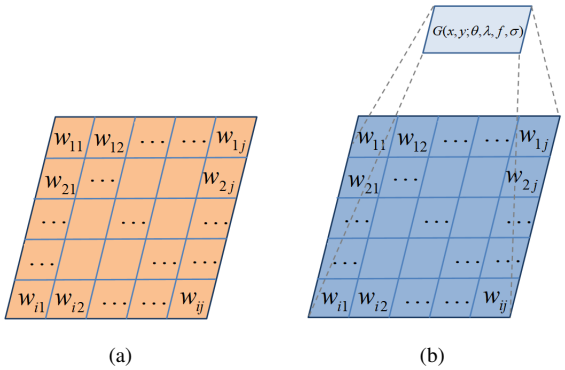


Fig. 2. Schematic comparison of traditional convolutional kernel parameters and VC-Gabor convolutional kernel parameters. (a) Schematic diagram of traditional convolutional kernel parameters. (b) Schematic diagram of VC-Gabor convolutional kernel parameters.

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

For these learnable VC-Gabor convolutional kernel parameters, we use backpropagation and chain derivation rules to update the parameters of the VC-Gabor function. The partial derivatives of G_{re} with respect to $\theta, \lambda, f, \sigma$ are first computed as shown in Eqs. (4)-(7).

$$\frac{\partial G_{re}}{\partial \theta} = -\frac{2\pi y'}{\lambda} \cdot G_{im} \tag{4}$$

$$\frac{\partial G_{re}}{\partial \lambda} = \frac{2\pi y'}{\lambda^2} \cdot G_{im} \tag{5}$$

$$\frac{\partial G_{re}}{\partial f} = -2\pi\sqrt{x^2+y^2} \cdot G_{im} \tag{6}$$

$$\frac{\partial G_{re}}{\partial \sigma} = (\frac{2}{\sigma} + \frac{x^2+y^2}{\sigma^3} \cdot G_{re} \tag{7}$$

Then the hidden parameters are updated using the back-propagation algorithm according to the chain rule, and the parameters of the VC-Gabor function are updated as shown in Eqs. (8)-(11).

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial L}{\partial G_{re}} \cdot \frac{\partial G_{re}}{\partial \theta} \tag{8}$$

$$\lambda_{t+1} = \lambda_t - \alpha \frac{\partial L}{\partial G_{re}} \cdot \frac{\partial G_{re}}{\partial \lambda} \tag{9}$$

$$f_{t+1} = f_t - \alpha \frac{\partial L}{\partial G_{re}} \cdot \frac{\partial G_{re}}{\partial f} \tag{10}$$

$$\sigma_{t+1} = \sigma_t - \alpha \frac{\partial L}{\partial G_{re}} \cdot \frac{\partial G_{re}}{\partial \sigma} \tag{11}$$

where, L is the objective function, α is the learning rate, and

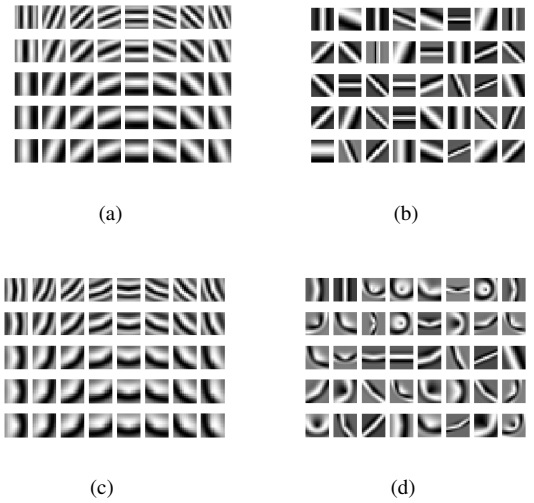
$$x^{'} = x\cos\theta + y\sin\theta \tag{12}$$

$$y^{'} = -x\sin\theta + y\cos\theta \tag{13}$$

In the model proposed in this paper, the first layer uses a 40-channel learnable VC-Gabor convolution filter for the extraction of line, orientation, and texture information while filtering out redundant information. Given an input image $X \in R^{80\times128\times1}$, the output feature map after the 40-channel VC-Gabor convolutional layer is $Y \in R^{80\times128\times40}$. Compared to the hand-crafted Gabor filter and the Gabor filter learned with a neural network, our learned VC-Gabor filter has variable direction, scale, and curvature, which is conducive to extracting richer features. As shown in Fig. 3, the visual comparison of the hand-crafted Gabor filter, the Gabor filter learned with neural network, and the VC-Gabor filter learned with neural network is demonstrated.

D. Multi-branch Structures

The traditional one-path network model is simple and easy to train, but its performance is not good, especially when the network layer is deep, and the gradient disappears easily. The current mainstream view is that the multi-branch structure has a better feature extraction effect than the single-path structure



(a) (b) (c) (d)

Fig. 3. Two-dimensional visual comparisons between hand-crafted Gabor filters, learned Gabor filters, hand-crafted VC-Gabor filters, and learned VC-Gabor filters. (a) Hand-crafted Gabor filters. (b) Learned Gabor filters. (c) Hand-crafted VC-Gabor filters. (d) Learned VC-Gabor filters.

[30]. Therefore, each convolution block after the VC-Gabor convolution layer uses a multi-branch structure.

There are two forms of convolutional block with multi-branch structure, one is for the part that needs to be down-sampled, we use the two-branch structure as shown in Fig. 4. Taking downsampling conv-block as an example, if the input feature map is given as $X \in R^{80\times128\times40}$ (width, height, and the number of channels are 80, 128, and 40 respectively). The first branch convolves the input feature map with 3×3, where stride is 2 and padding is 1, and then the feature map $X_1 \in R^{40\times64\times128}$ is obtained through the batch normalization layer; the second branch convolves the input feature map with 1×1, where stride is 2 and padding is 0, and then the feature map $X_2 \in R^{40\times64\times128}$ is obtained through the batch normalization layer. At the same time, through the network training out the multi-branch adaptive weights: $w = [w_1, w_2]$, the features extracted from the two branches are multiplied by the adaptive weights and summed respectively, and finally the output feature map of the downsampling conv-block is obtained by passing the summed feature data through the ReLU activation function layer as $Y \in R^{40\times64\times128}$. Given the input feature map X, the output feature map Y is computed under the formula:

$$Y = ReLU(w_1 \times BN(Conv(X)) + w_2 \times BN(Conv(X))) \tag{14}$$

Where Conv is the convolutional layer, BN is the batch normalization layer, ReLU is the nonlinear activation function, and w_1, w_2 are the adaptive weights on the two branches, respectively.

For the parts of each block that do not need to be down-sampled, a three-branch structure as in Fig. 5 is used. Take the Conv-block as an example, and take the feature map as input $X \in R^{80\times128\times40}$. The first branch convolves the input feature map x with 3×3, where stride is 1 and padding is 1,

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

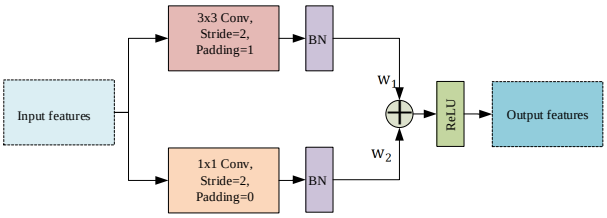


Fig. 4. Two-branch convolutional block

and then goes through the batch normalization layer to get the feature map $X_1 \in R^{40\times64\times128}$; the second branch convolves the input features with 1×1, where stride is 1 and padding is 0, and then goes through the batch normalization layer to get the feature map $X_2 \in R^{40\times64\times128}$; the third branch is a shortcut branch, which only batch normalizes the input features to get $X_2 \in R^{40\times64\times128}$. This shortcut branch structure not only obtains a more robust feature representation in the deeper layers of the network as the model deepens but also better handles the gradient vanishing problem in the deeper layers of the network. At the same time, the multi-branch adaptive weights: $w = [w_1, w_2, w_3]$ are trained through the network, the features extracted from the three branches are multiplied by the adaptive weights and summed up respectively, and finally the output feature map of Conv-block is obtained as $Y \in R^{40\times64\times128}$ by passing the summed feature data through the ReLU nonlinear activation function layer. The output feature map is calculated as follows:

$$\begin{aligned} Y = ReLU(&w_1 \times BN(Conv(X)) \\ &+w_2 \times BN(Conv(X))+w_3 \times BN(X)) \end{aligned} \tag{15}$$

These multi-branch structures have good model characterization ability and improve the accuracy of vein recognition significantly.

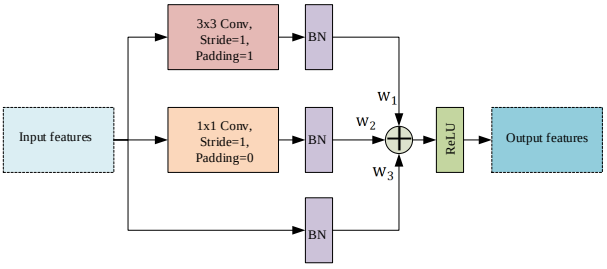


Fig. 5. Three-branch convolutional block

E. Loss Function

In the finger vein recognition task, the Softmax loss function is used to measure the difference between the model predictions and the true labels and to motivate the network to make appropriate parameter adjustments to improve the classification performance. Assume that the output vector of the last fully connected layer is obtained from the neural network as $z = [z_1, z_2, \ldots, z_C]$, where C is the number of categories. The output of the fully connected layer is then converted into category probabilities using the Softmax function, which is defined as follows:

$$p_i = \frac{e^{z_i}}{\sum_{k=1}^{C} e^{z_k}} \tag{16}$$

where p_j is the predicted probability of category j. The label of each sample is represented as a one-hot encoding. If the true category of a sample is i, the vector of true labels for that sample is $y = [0, 0, \ldots, 1, \ldots, 0]$, where only the ith position is 1 and the rest of the positions are 0. Subsequently, the one-hot encoding of the true labels is compared with the predicted probabilities to compute the Softmax loss, the expression for the Softmax loss function is as follows:

$$L = -\sum_{i=1}^{C} y_i \log(p_i) \tag{17}$$

where y_i is the ith element of the true label vector and p_j is the corresponding predicted probability. The parameters of the model proposed in this paper can be optimized by minimizing the above loss function. The specific steps of model optimization are shown in Algorithm 1.

IV. Experiments

In this section, we will first introduce the experimental environment, the dataset, and the evaluation metrics used in this paper. Subsequently, we will fully demonstrate the effectiveness of the proposed algorithm through ablation and comparison experiments.

A. Dataset

In this paper, we conduct experiments under two publicly available datasets, FV-USM and SDUMLA, which are described in detail below.

The FV-USM database was obtained from Universiti Sains Malaysia. It consists of the index and middle fingers of the left and right hands of 123 subjects, with six images for each finger. The images were divided into two sessions with 2952 images in each session, totaling 5904 images. The cropped ROI image resolution is 100×300 pixels as shown in Fig. 6.

The SDUMLA dataset was obtained from Shandong University and it contains finger vein images of 636 fingers from 106 subjects. Six grayscale images were obtained from each subject's index, middle, and ring fingers of the left and right hands, which in total consisted of 3816 images, and the cropped ROI images had a resolution of 150×96 pixels, as shown in Fig. 7.

In this experiment, we take the regions of interest extracted from the original finger vein images as input images [31]. The overall dataset is divided into training set, validation set, and test set according to the ratio of 4:1:1, where the validation set is used to detect whether the model converges and for model tuning, and the test set is used to test the performance of the model. In addition, in order to improve the generalization ability of the model and prevent overfitting, after dividing the dataset, we expanded the training set with samples by rotating, translating, flipping, and simulating light changes.

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

Algorithm 1 The optimization process of VC-Gabor Neural Networks

Input: Training set: $\{(x, y)\}_{i=1}^{N}$; Number of iterations: T.
// x represent an input sample; y represent a sample label;
// N represent the number of input samples.

Output: $G(w, b, \theta, \lambda, f, \sigma)$

Initialize w, b
Initialize $\theta, \lambda, f, \sigma$
for $i = 0$ to T **do**
  Calculate forecast categories: $p = [p_0, p_1, \ldots, p_{C-1}]$
  Set the one-hot representation of the sample label: $y = [y_0, y_1, \ldots, y_{C-1}]$
  Calculate loss: $L = -\sum_{i=1}^{C} y_i \log(p_i)$
  $w \leftarrow w - \alpha \frac{\partial L}{\partial w}$
  $b \leftarrow b - \alpha \frac{\partial L}{\partial b}$
  $\theta \leftarrow \theta - \alpha \frac{\partial L}{\partial G_{re}} \cdot \frac{\partial G_{re}}{\partial \theta}$
  $\lambda \leftarrow \lambda - \alpha \frac{\partial L}{\partial G_{re}} \cdot \frac{\partial G_{re}}{\partial \lambda}$
  $f \leftarrow f - \alpha \frac{\partial L}{\partial G_{re}} \cdot \frac{\partial G_{re}}{\partial f}$
  $\sigma \leftarrow \sigma - \alpha \frac{\partial L}{\partial G_{re}} \cdot \frac{\partial G_{re}}{\partial \sigma}$
end for
return $G(w, b, \theta, \lambda, f, \sigma)$

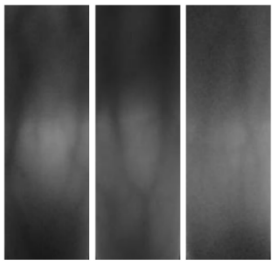


Fig. 6. FV-USM dateset

B. Experimental Environment and Parameter Settings

In this paper, the proposed method is implemented using the Pytorch framework, and experiments are conducted on a Linux server. The main experimental environment configurations are shown in Table I. In this model, the batch size is set to 32, and the Adam optimizer is used for training with an initial learning rate of 0.001, and the learning rate is adjusted according to the learning rate adjustment strategy of cosine annealing with a period of 20. We ensure the fairness of the experiments through the following methods:

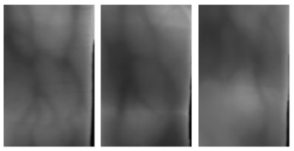


Fig. 7. SDUMLA dateset

In the ablation experiments, we established five experimental groups to validate the effectiveness of each module. Each experimental group only modified one specific module in the model, while keeping other modules unchanged. Additionally, the hyperparameters and test set remained unchanged. In the comparative experiments, we reproduced the experiments from the compared paper and obtained experimental results using the same test set as ours.

TABLE I
EXPERIMENTAL ENVIRONMENT CONFIGURATIONS

Name	disposition
SYSTEM	18.04.1-Ubuntu
CPU	Intel(R) Xeon(R) CPU E5-2678 v3 @ 2.50GHz
GPU	NVIDIA GeForce RTX 3090 24GB
MEMORY	64GB
HARD DISK	1TB

C. Evaluation Metrics

In order to fairly evaluate the performance of different methods, we use two metrics, Accuracy Rate (ACC) and Equal Error Rate (EER), to evaluate the performance of the models, and compare the performance of the models more clearly through Detection Error Tradeoff (DET) curves. The ACC denotes the proportion of samples correctly classified by the classifier to the total number of samples, which is calculated by the following formula:

$$ACC = \frac{Number\ of\ Correctly\ Classified\ Samples}{Total\ Number\ of\ Samples} \times 100\% \tag{18}$$

The EER measures and balances the performance of False Rejection Rate (FRR) and False Acceptance Rate (FAR) in an integrated manner. By adjusting the thresholds, when FAR and FRR are equal, the value of FAR and FRR at that point is the EER. The definitions of FAR and FRR above are as follows:

$$FAR = \frac{FP}{FP + TN} \tag{19}$$

$$FRR = \frac{FN}{TP + FN} \tag{20}$$

where FP refers to the number of samples that are originally negative samples but are determined to be positive samples; TN refers to the number of samples that are originally negative

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

samples and are judged to be negative samples; TP refers to the number of samples that are originally positive samples and are determined to be positive samples; FN is the number of samples that were originally positive and were judged to be negative.

DET curve is a commonly used classifier evaluation index, which has a very objective role in evaluating the performance of the model. It is a curve with False Positive Rate (FPR) and False Negative Rate (FNR) as axes to help us weigh the trade-off between the false positive rate and the false negative rate of the classifier.

Usually, the DET curve is a curve in the lower left direction of the coordinate axis. We can judge the performance of the classifier by comparing the area of the DET curve and the shape of the curve. The larger the area enclosed by the curve and the upper right coordinate axis or the closer to the origin of the coordinate axis $(0, 0)$, the better the performance of the classifier.

D. Selection of Kernal Size for VC-Gabor Filters

The size of the VC-Gabor convolution kernel is an important parameter in convolution operation, and the response-ability of the image is different with different sizes of convolution kernel. In order to select the optimal sizes of the VC-Gabor convolution kernel for the model performance, we make other conditions the same, only the size of the VC-Gabor convolution kernel is different (K=3, K=9, K=15 in turn) and carry out experiments on FV-USM dataset and SDUMLA dataset. The experimental results are shown in Table II:

TABLE II
PARAMETERS EXPERIMENT

Kernel size	FV-USM		SDUMLA	
	ACC(%)	EER(%)	ACC(%)	EER(%)
(3,3)	99.81	0.16	98.94	0.67
(9,9)	99.88	0.13	99.30	0.51
(15,15)	99.83	0.17	98.99	0.57

From Table II, it can be seen that the feature extraction capability is not optimal when the convolution kernel is too small or too large. This is because a larger convolution kernel in the shallow layer of the network can obtain a larger sensory field, while a convolution kernel that is too large tends to lose some local detail information. In this experiment, when the convolution kernel size K=9, the accuracy and EER of the model are relatively optimal. Therefore, in the next experiments, the size of the VC-Gabor convolution kernel is set to K=9.

E. Model Visualization

In order to demonstrate that the model proposed can better learn the features of the vein region, model visualization experiments are designed in this paper, the features obtained from each convolutional layer are displayed by the reverse convolution operation on the feature map, and these features are visualized to obtain the heat map [32]. In this paper, the parts in red represent the parts that the model focuses more on.

The heat map of the traditional convolutional neural network with multi-branch structures and the VC-Gabor convolutional and multi-branch structure neural network proposed in this paper is shown in Fig. 8. From the figure, it can be seen that both neural network models focus on the vast majority of the vein curve part of the finger vein image, which indicates that this part of the finger vein region labeled in red plays a crucial role in the recognition results, and also indicates that our model effectively extracts this part of the vein features. However, for the traditional convolutional neural network with multi-branch structures, there are still some vein curves in the more ambiguous regions that are not attended to. The VC-Gabor and multi-branch structure network proposed in this paper, on the other hand, focuses on this part of the blurred finger vein region (the red dashed box part), which indicates that the VC-Gabor filter with variable direction, scale, and curvature learned by the model in this paper can extract the vein features more efficiently in the shallow part of the neural network.

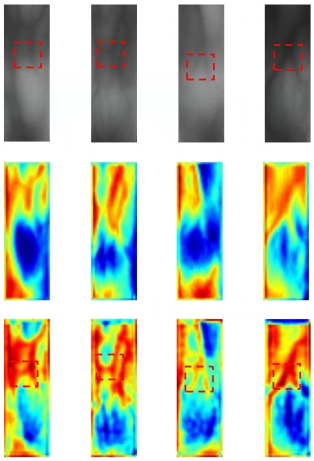


Fig. 8. Comparison of the heat maps of the traditional convolutional neural network with multi-branch structures and the neural network with VC-Gabor convolution and multi-branch structure proposed in this paper. In the figure, the first layer is the original finger vein images, the middle is the heat maps of the traditional convolutional neural network with multi-branch structures, and the bottom layer is the heat maps of the VC-Gabor and multi-branch structures neural network.

F. Ablation Experiment

In order to evaluate the contribution of the VC-Gabor convolutional layer and the multi-branch feature extraction structure in the model proposed in this paper to the performance of the algorithm, we designed several control groups for ablation experiments, which were carried out on the FV-USM and SDUMLA datasets, respectively. The experimental results are given in Table III and the corresponding DET curves are given in Fig 9, 10, 11 and 12. The experimental control groups were set up as follows:

(1) CNN+SBS (single-branch Structure): The first convolutional layer of the network uses a traditional convolutional

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

filter, and all subsequent feature extraction blocks use single-branch structures.

(2) CNN+MBS(multi-branch structure): The first convolutional layer of the network is an traditional convolutional filter, and all subsequent feature extraction blocks use multi-branch structures.

(3) VC-Gabor+SBS: The first convolutional layer of the network uses VC-Gabor convolutional filters, and all subsequent feature extraction blocks use single-branch structures.

(4) Gabor+MBS: The first convolutional layer of the network uses Gabor convolutional filters, and all subsequent feature extraction blocks use multi-branch structures.

(5) VC-Gabor+MBS: The first convolutional layer of the network uses VC-Gabor convolutional filters, and all subsequent feature extraction blocks use multi-branch structures.

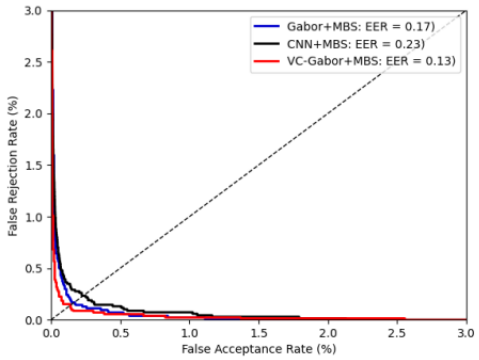


Fig. 9. DET curves for experiments (2), (4) and (5) on the FV-USM dataset

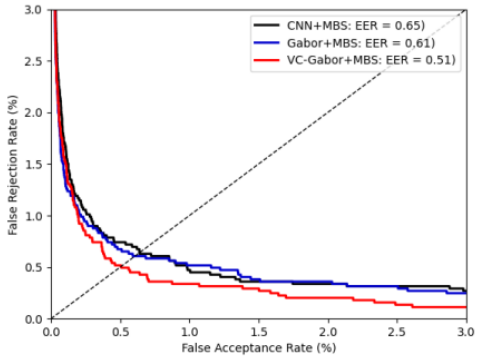


Fig. 10. DET curves for experiments (2), (4) and (5) on the SDUMLA dataset

Observing Table III, Fig. 9 and Fig. 10, by comparing Experiments (2), (4), and (5), it can be found that under the same condition of using multi-branch structural feature extraction, when replacing the first layer of the network from traditional convolution to Gabor convolution, the recognition accuracy is improved by 0.06% and 0.09% on the FV-USM dataset and SDUMLA dataset, respectively, and the equal-error rate is reduced by 0.06% and 0.04%. The experimental results

TABLE III
ABLATION EXPERIMENT

Type	Method	FV-USM		SDUMLA	
		ACC(%)	EER(%)	ACC(%)	EER(%)
(1)	CNN+SBS	99.54	0.36	98.61	0.74
(2)	CNN+MBS	99.75	0.23	98.97	0.65
(3)	VC-Gabor+SBS	99.69	0.29	99.10	0.61
(4)	Gabor+MBS	99.81	0.17	99.06	0.61
(5)	**VC-Gabor+MBS**	**99.88**	**0.13**	**99.30**	**0.51**

show that adding Gabor convolution to the neural network can effectively extract the direction and scale information of vein lines, thus improving the recognition rate and reducing the equal error rate. By replacing the first layer of the network with VC-Gabor convolution, the recognition accuracy is further improved and the equal error rate is further decreased, compared with the traditional convolution on the two datasets, the accuracy is improved by 0.13% and 0.33% respectively, and the equal error rate is decreased by 0.10% and 0.14% respectively. This is because we added variable curvature information to the Gabor filter, which not only extracts the vein feature information in different directions and scales but also extracts the curvature feature information of the vein line features.

Observing Table III, Fig. 11 and Fig. 12, by comparing the results of Experiments (1), (2), (3) and (5), we can find that using multi-branch structure feature extraction achieves better results than single-branch structure while ensuring all other things are equal. The accuracy is improved by 0.21% and 0.36% on both datasets, and the equal error rate is reduced by 0.13% and 0.09%, respectively. Similarly, comparing the results of Experiment 2 and Experiment 8, we can find that the experimental results achieved by the multi-branch structure are still better than those of the single-branch structure, and the accuracy is improved by 0.19% and 0.20% on both datasets, and the equal error rate is reduced by 0.16% and 0.10%, respectively. Therefore, the adaptive multi-branch structure feature proposed in this paper is conducive to improving the feature extraction ability of finger vein images, thus enhancing the recognition performance.

In summary, no matter in the multi-branch condition or the single-branch condition, through experiments, we conclude that the experimental effect of the variable curvature Gabor convolutional layer is improved on the FV-USM dataset and the SDUMLA dataset.

G. Contrast Experiment

In order to verify the performance of the algorithm in this paper, we compare the algorithm proposed in this paper with the classical finger vein recognition algorithms based on traditional methods and deep learning, and the results of the comparison experiments are shown in Table IV.

Compared with traditional methods for finger vein recognition [33], [34], [17], our algorithm achieves a maximum improvement of 4.74% and 3.51% in recognition accuracy on

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

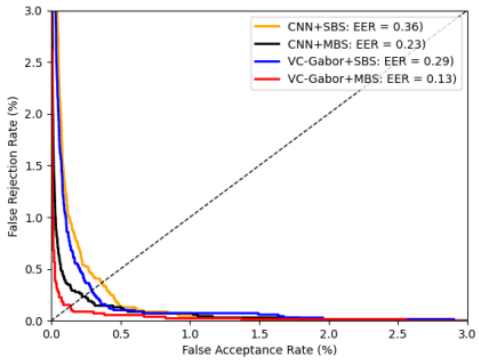


Fig. 11. DET curves for experiments (1), (2), (3), and (5) on the FV-USM dataset

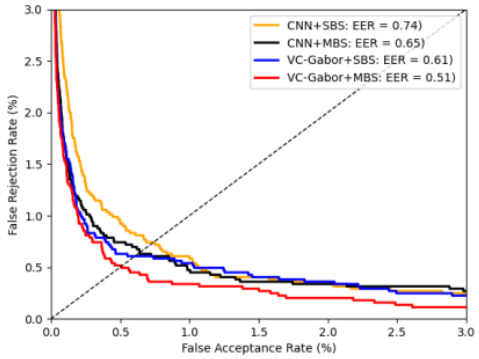


Fig. 12. DET curves for experiments (1), (2), (3), and (5) on the SDUMLA dataset

the FV-USM and SDUMLA datasets, respectively. Moreover, the EER is reduced by up to 2.56% and 2.26%, respectively. These results demonstrate that our deep learning-based finger vein recognition algorithm exhibits stronger feature extraction capabilities compared to traditional methods. The learned features are more discriminative, leading to superior performance in terms of recognition accuracy and EER evaluation metrics.

Compared with the pre-trained VGG-based model proposed by Hong *et al.* [35], the recognition rate of the algorithm proposed in this paper is improved by 2.87% and 2.62% on the FV-USM and SDUMLA datasets, respectively, and the equal error rate is reduced by 1.15% and 1.64%, respectively. This is due to the fact that the algorithm in this paper does not use pre-trained models, an approach that not only trains models with better generalization ability based on the characteristics of the finger vein dataset. Moreover, this way of training from scratch allows for easier tuning of hyperparameters and monitoring of performance metrics during the training process for finer-grained tuning and optimization of the model. Compared with the adaptive AGCNN method proposed by Zhang *et al.* [15], our network achieves a significant improvement in accuracy on the FV-USM and SDUMLA datasets, with increases of 6.08% and 8.07%, respectively, and a reduction in the EER of 3.76% and 3.63%, respectively. Zhang *et al.* replaced all traditional convolutional layers with Gabor convolutional layers, significantly reducing the parameter count but resulting in a decrease in recognition accuracy. In contrast, our algorithm only replaces the first layer's traditional convolutional layers with VC-Gabor convolutional layers to extract directional, scale, and curvature features of finger vein curves in images. Additionally, we utilize a multi-branch structure in the subsequent network, which enhances the feature extraction capabilities compared to Zhang *et al.*'s single-branch structure. Compared with the lightweight CNN finger vein recognition model proposed by Zhao *et al.* [36], the recognition accuracies of this paper are improved by 1.93% and 1.72% on FV-USM and SDUMLA, respectively, and the equal error rates are reduced by 0.94% and 1.25%, respectively. This is due to the deeper network model proposed in this paper, which tends to have better feature extraction capability, and the residual structure of the method in this paper can effectively avoid overfitting. Compared with the attention-based models proposed by Huang *et al.* [7] and Zhang *et al.* [38], as well as the capsule network model method proposed by Li *et al.* [37], our network model still exhibits some performance improvements. Specifically, on the FV-USM dataset and SDUMLA dataset, our proposed model achieves at least 0.45% and 0.39% improvement in recognition accuracy, respectively, with a reduction in the EER of at least 0.06% and 0.17%. These results indicate that the VC-Gabor convolution and multi-branch feature extraction structure proposed in our study possess stronger discriminative capabilities for distinguishing different classes of finger vein images.

TABLE IV
COMPARISON WITH EXISTING CONVENTIONAL FINGER VEIN RECOGNITION METHODS AND DEEP LEARNING BASED FINGER VEIN RECOGNITION METHODS ON FV-USM AND SDUMLA DATASETS

Setting	Method	FV-USM		SDUMLA	
		ACC(%)	EER(%)	ACC(%)	EER(%)
Yang *et al.*, 2013 [33]	SWG	95.14	2.69	96.95	2.77
Qiu *et al.*, 2016 [34]	DWP	97.02	2.32	97.61	1.59
Huang *et al.*, 2010 [17]	WLD	96.51	2.37	95.79	2.24
Hong *et al.*, 2017 [35]	CNN(pre-trained VGG16)	97.01	1.28	96.68	2.15
Zhang *et al.*, 2019 [15]	AGCNN	93.80	3.89	91.23	4.14
Zhao *et al.*, 2020 [36]	lightweight CNN	97.95	1.07	97.58	1.76
Huang *et al.*, 2021 [7]	JAFVNet	99.36	0.42	98.74	0.68
Li *et al.*, 2022 [37]	ViT-Cap	98.68	0.28	93.24	1.30
Zhang *et al.*, 2023 [38]	LCAModel	99.43	0.19	98.91	0.74
Ours	**VC-Gabor+MBS**	**99.88**	**0.13**	**99.30**	**0.51**

V. CONCLUSION

Because the direction and scale of the traditional Gabor filter are relatively single, the curvature information in the vein curve can not be effectively extracted, and the parameter tuning is difficult, so the effect of finger vein recognition is not good. To address this issue, this paper proposes a neural network based on the learnable VC-Gabor convolution and multi-branch structure. Firstly, the Gabor filter is improved and curvature information is added to extract different curvature information from the vein curve. Secondly, the VC-Gabor function is constructed as a learnable convolution layer, and the parameters of the VC-Gabor convolution layer are updated

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3397293

using backpropagation in the neural network. This allows the learned VC-Gabor filters to be rich in orientation, scale, and curvature while eliminating the need for complex manual parameter tuning. Finally, we propose an adaptive multi-branch structure for feature extraction, which can enhance the feature extraction ability of the model and avoid overfitting. Experimental result shows that the proposed algorithm is superior to the current mainstream algorithms in classification accuracy and EER.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their comments to improve the quality of this article.

REFERENCES

[1] P. Gupta, K. Tiwari, and G. Arora, "Fingerprint indexing schemes - A survey," *Neurocomputing*, vol. 335, pp. 352–365, Mar. 2019.

[2] J. Galbally, S. Marcel, and J. Fierrez, "Biometric Antispoofing Methods: A Survey in face recognition," *IEEE Access*, vol. 2, pp. 1530-1552, 2014.

[3] X. Liu, Y. Bai, Y. Luo, Z. Yang, and Y. Liu, "Iris recognition in visible spectrum based on multi-layer analogous convolution and collaborative representation," *Pattern Recognit. Lett.*, vol. 117, pp. 66-73, Jan. 2019.

[4] Z. Meng, N. Kanda, Y. Gaur, S. Parthasarathy, E. Sun, L. Lu, X. Chen, J. Li, and Y. Gong, "Internal language model training for domain-adaptive end-to-end speech recognition," *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 7338-7342, 2021.

[5] Z. Zhang, F. Zhong, and W. Kang, "Study on reflection-based imaging finger vein recognition," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2298–2310, 2022.

[6] H. Ma, N. Hu, and C. Fang, "The biometric recognition system based on near-infrared finger vein image," *Infrared Phys. Technol.*, vol. 116, Aug. 2021.

[7] J. Huang, M. Tu, W. Yang, and W. Kang, "Joint attention network for finger vein authentication," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1-11, 2021.

[8] J. Yang, Y. Shi, and G. Jia, "Finger-vein image matching based on adaptive curve transformation," *Pattern Recognit.*, vol. 66, pp. 34-43, Jun. 2017.

[9] P. Zhao, S . Zhao, J. Xue, W. Yang, and Q. Liao, "The neglected background cues can facilitate finger vein recognition," *Pattern Recognit.*, vol. 136, pp. 109199, 2023.

[10] J. Yang, J. Yang, and Y. Shi, "Combination of gabor wavelets and circular gabor filter for finger-Vein extraction," *Emg. Intell. Comput. Tech. Apps. (ICIC)*, pp. 346-354, 2009.

[11] S. Xie, J. Yang, S. Yoon, L. Yu, and D. S. Park, "Guided gabor filter for finger vein pattern extraction," *8th Int. Conf. Signal Image Techn. Internet Based Syst. (SITIS)*, pp. 118-123, 2012.

[12] H. Wang, M. Du, J. Zhou, and L. Tao, "Weber local descriptors with variable curvature gabor filter for finger vein recognition," *IEEE Access*, vol. 7, pp. 108261-108277, 2019.

[13] S. A. Radzi, M. K. Hani, and R. Bakhteri, "Finger-vein biometric identification using convolutional neural network," *Turk. J. Elec. Eng. Comp. Sci.*, vol. 24, no. 3, pp. 1863-1878, Jan. 2016.

[14] H. Huang, S. Liu, and H. Zheng, "DeepVein: Novel finger vein verification methods based on deep convolutional neural networks," *IEEE Int. Conf. Identity Secur. Behav. Anal. (ISBA)*, pp. 1-8, 2017.

[15] Y. Zhang, W. Li, L. Zhang, X. Ning, L. Sun, and Y. Lu, "Adaptive learning gabor filter for finger-vein recognition," *IEEE Access* vol. 7, pp. 159821-159830, 2019.

[16] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE Trans. Info. Syst.*, vol. 90, no.8, pp. 1185-1194, Aug. 2007.

[17] B. Huang, Y. Dai, R. Li, D. Tang, and W. Li, "Finger-vein authentication based on wide line detector and pattern normalization," *20th Int. Conf. Pattern Recognit. (ICPR)*, pp. 1269–1272, 2010.

[18] J. Yang, Y. Shi, J. Yang, "Finger-vein recognition based on a bank of gabor filters," *Asian Conf. Comput. Vis. (ACCV)*, vol. 1, pp. 374-383, 2009.

[19] H. Lee, B. Kang, E. Lee, and K. R. Park, "Finger vein recognition using weighted local binary pattern code based on a support vector machine," *J. Zhejiang Univ. Sci. C*, vol. 11, no. 7, pp. 514–524, Jul. 2010.

[20] B. A. Rosdi, C. W. Shing, and S. A. Suandi, "Finger vein recognition using local line binary pattern," *Sensors*, vol. 11, no. 12, pp. 11357–11371, Nov. 2011.

[21] X. Meng, G. Yang, Y. Yin, and R. Xiao, "Finger vein recognition based on local directional code," *Sensors*, vol. 12, no. 11, pp. 14937–14952, Nov. 2012.

[22] J. Wu, and C. Liu, "Finger-vein pattern identification using principal component analysis and the neural network technique," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5423-5427, May 2011.

[23] G. Yang, X. Xi, and Y. Yin, "Finger vein recognition based on (2D) PCA and metric learning," *BioMed Res. Int.*, 2012.

[24] X. Xi, L. Yang, and Y. Yin, "Learning discriminative binary codes for finger vein recognition," *Pattern Recognit.*, vol. 66, pp. 26–33, Jun. 2017.

[25] J. Wu, and C. Liu, "Finger-vein pattern identification using SVM and neural network technique," *Expert Syst. Appl.*, vol. 38, no. 11, pp. 14284-14289, Oct. 2011.

[26] H. Hu, "FV-Net: learning a finger-vein feature representation based on a CNN," *24th Int. Conf. Pattern Recognit. (ICPR)*, pp. 3489-3494, 2018.

[27] Y. Fang, Q. Wu, and W. Kang, "A novel finger vein verification system based on two-stream convolutional network learning," *Neurocomputing*, vol. 290, pp. 100-107, 2018.

[28] K. Wang, G. Chen, and H. Chu, "Finger vein recognition based on multi-receptive field bilinear convolutional neural network," *IEEE Signal Process. Lett.*, vol. 28, pp. 1590-1594, 2021.

[29] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4357-4366, 2018.

[30] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making -style convnets great again," *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 13728-13737, 2021.

[31] W. Yang, W. Luo, W. Kang, Z. Huang and Q. Wu, "FVRAS-Net: An embedded finger-vein recognition and anti-spoofing system using a unified CNN," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 11, pp. 8690-8701, 2020.

[32] M. D. Zeiler, and R. Fergus, "Visualizing and understanding convolutional networks," *Euro. Conf. Comput. Vision*, pp. 818-833, Nov. 2014.

[33] L. Yang, G. Yang, and Y. Yin, "Sliding window-based region of interest extraction for finger vein images," *Sensors*, vol. 13, no. 3, pp. 3799-3815, 2013.

[34] S. Qiu, Y. Liu, and Y. Zhou, "Finger-vein recognition based on dual-sliding window localization and pseudo-elliptical transformer," *Expert Syst. Appl.*, vol. 64, pp. 618-632, Dec. 2016.

[35] H. Hong, M. Lee, and K. Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors," *Sensors*, vol. 17, no. 6, Jun. 2017.

[36] D. Zhao, H. Ma, and Z. Yang, "Finger vein recognition based on lightweight CNN combining center loss and dynamic regularization," *Infrared Phys. Technol.*, vol. 105, Mar. 2020.

[37] Y. Li, H. Lu, and Y. Wang, "ViT-Cap: A novel vision transformer-based capsule network model for finger vein recognition," *Appl. Sci.*, vol. 12, 2022.

[38] Z. Zhang, and M. Wang, "Finger vein recognition based on lightweight convolutional attention model," *IET Image Process.*, vol. 17, no. 6, pp. 1864-1873, Feb. 2023.

Authorized licensed use limited to: University of Southern Queensland. Downloaded on August 22,2024 at 11:57:11 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.