IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 November 2023, accepted 20 December 2023, date of publication 28 December 2023,
date of current version 5 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347922

RESEARCH ARTICLE

Finger Vein Recognition Based on ResNet With Self-Attention

ZHIBO ZHANG[1], **GUANGHUA CHEN**[1,2], **WEIFENG ZHANG**[3], **AND HUIYANG WANG**[1]

[1]School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China
[2]Microelectronic Research and Development Center, Shanghai University, Shanghai 200444, China
[3]College of Electromechanical Engineering, Qingdao University of Science and Technology, Qingdao 266061, China

Corresponding author: Guanghua Chen (chghua@shu.edu.cn)

ABSTRACT To solve the problem of low accuracy and high computational resource consumption in finger vein recognition, a finger vein recognition model based on ResNet with self-attention (FV-RSA) is proposed. This model combines global focusing ability of self-attention mechanism and local feature extraction ability of CNN, which improves recognition accuracy. To reduce the number of parameters and floating-point operations, self-attention and convolution share linear projections by pointwise convolution. Self-attention and CNN are fused in the convolution and self-attention (CASA) block connected by skip connection to avoid gradient vanishing or gradient exploding. During the training phase, we use a variable learning rate with cosine annealing to avoid falling into local optimum. Experiments show that the method works well on the public database, which can not only improve the accuracy, but also reduce the number of parameters and computational complexity.

INDEX TERMS Finger vein recognition, deep learning, ResNet, self-attention, variable learning rate.

I. INTRODUCTION

In today's digital and information age, people's awareness of the protection of personal information is constantly strengthening. Traditional identity authentication technologies such as magnetic card, certificate, password and other identity identifiers, are easy to be lost and stolen, which can not meet the needs of high security and confidentiality of personal identity information. The convenience and reliability of biometrics make them a prominent solution to tackle this issue. The first-generation biometrics such as fingerprint recognition [1], face recognition [2], and iris recognition [3] have been successfully applied. However, they still have some shortcomings. Surface features such as fingerprints are easily damaged. Face recognition conditions are easily limited, and features are not stable enough. Iris recognition is not convenient enough. Voice features are not stable enough. All of the above recognition methods have different degrees of influence on the recognition results. Finger vein recognition has the advantages of liveness recognition and internal characteristics. Contactless identification is more hygienic. Compared with the first-generation biometric identification technology, Finger vein recognition has higher security level, which is more stable and difficult to forge characteristics.

Traditional finger vein recognition methods use a hand-designed approach to extract features. Image processing and statistic are mainly used, which generally extracting shallow features. Kono et al. proposed firstly to use finger vein features for identity authentication, which enhanced the vein features by a background noise reduction filter. After the corrected image was rotated, the normalized maximum value of the cross-correlation was used to quantify the similarity between the two vein features [4]. Miura et al. proposed a repetitive linear tracking (RLT) algorithm based on local grayscale differences in finger vein images to track and extract vein curves pixel by pixel [5]. Later, they proposed the maximum curvature (MC) algorithm to extract finger vein features by searching the maximum value of the local cross-section curvature of the image [6]. These two algorithms had become the two classic algorithms in the field of finger vein feature extraction. The negative average curvature points of the valley-shaped structure based on the geometric shape of the grayscale distribution of finger vein images was used to extract finger vein features [7]. Asaari et al. used a modified

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

Gaussian filter to extract vein texture features from the acquired finger vein ROI images, and used the band-limited phase correlation method to identify feature similarity. In the recognition stage, they proposed the geometric feature of the width centroid contour distance, which is related to the finger vein texture feature fusion for matching recognition [8]. Traditional feature extraction methods require preprocessing, which requires a lot of time. Traditional machine learning methods such as SVM [9] and Multi-SVM [10] are used to recognize finger vein, but the accuracy of these methods in recognition are lower than other methods.

Deep learning method can be directly used to achieve end-to-end recognition without too much manual intervention, which can obtain deep features with better stability. Many researchers have already done related work, such as FV-GAN [11], FVRAS-Net [12]. In these works, several convolutional neural network architectures were used. In recent years, the transformer [13]architecture has performed well in the field of computer vision. Some researchers used the transformer architecture for finger vein recognition, such as FV-ViT [14]. All convolution blocks are replaced by CNN mixed transformer block in some methods, which is inefficient [15]. MobileVit [16] has proven by experiments that there is a lot of redundancy in shallow layer. Transformer took up a lot of computing resources, and it was proven to be even less effective than convolution when the dataset is small [17]. These methods did not effectively use the core of transformer which is self-attention mechanism. Convolutional neural networks and attention blocks were used in some works, which used channel attention or spatial attention, such as ResNet+SE [18] and ECA-Resnet [19].These methods can effectively improve the accuracy of recognition, but some information is lost. CNN combined with self-attention mechanism is necessary.

In order to increase the accuracy of finger vein recognition and decrease the computational resource consumption of the model, we propose ResNet with self-attention (FV-RSA) model which is composed of convolution and self-attention (CASA) blocks. The CASA block is shown in Fig.1, self-attention is projected to queries, keys and values by pointwise convolution [20], followed by calculating attention weights and weighted sum of values. Convolution with $k_2 \times k_2$ kernal is decomposed into k_2^2 pointwise convolutions, followed by shift and summing values of feature maps for different kernels. The CASA block combines global focusing ability of self-attention and local feature extraction ability of CNN, and achieves higher accuracy for finger vein recognition than both the traditional convolutions and the self-attention. In terms of the number of parameters and Flops, the CASA block is more lightweight than traditional convolution and self-attention. The CASA block simultaneously solves the two problems of low accuracy and high computational resource consumption. The learning rate of cosine decay is used, which effectively speeds up the convergence of the fitting curve and avoids falling into local optimum. The FV-RSA model has the following three main contributions. Firstly, CASA block is used to improve feature extraction ability. Then pointwise convolutions are shared to reduce the number of parameters and computational complexity. Finally, The cosine annealing algorithm is used to speeds up the convergence and avoid falling into local optimum.

II. DECOMPOSITION OF CNN AND SELF-ATTENTION

CNN and self-attention are decomposed into two steps. The first step is pointwise convolution operation. Then the feature maps are unfolded as queries, keys, and values for self-attention. On the other hand, the feature maps are shifted and summed for convolution.

A. DECOMPOSITION OF SELF-ATTENTION

Three pointwise convolutions are used to project. After the pointwise convolution operations, the obtained feature maps are unfolded respectively. which are converted to key, query and value. The key and query are used to calculate attention weights of the value. The illustration is shown in Fig.2.

To gather more information, we employ multi-head self-attention. The multi-head self-attention have N heads. H, W is the length and width of feature map. C is the input and output channel. The channels are equally divided by N, so the number of channels becomes $\frac{C}{N}$. The feature map is unfolded into single-pixel blocks. k_1^2 is the size of the sliding window which is used in unfold operation. Let $X \in \mathbb{R}^{C_{in} \times H \times W}$ denote input feature map. Let $Y \in \mathbb{R}^{C_{out} \times H \times W}$ denote output feature map. $x_{(i,j)} \in \mathbb{R}^{C_{in}}$, $x_{(f,g)} \in \mathbb{R}^{C_{in}}$, $y_{(i,j)} \in \mathbb{R}^{C_{out}}$are corresponding tensor of pixel(i, j). Three pointwise convolutions are used on the feature map to project the feature map into queries, keys and values, we can get:

$$q_{(i,j)}^{(h)} = W_q^{(h)} x_{(i,j)}, k_{(i,j)}^{(h)} = W_k^{(h)} x_{(i,j)}, v_{(i,j)}^{(h)} = W_v^{(h)} x_{(i,j)} \quad (1)$$

$$k_{(f,g)}^{(h)} = W_k^{(h)} x_{(f,g)}, v_{(f,g)}^{(h)} = W_v^{(h)} x_{(f,g)} \quad (2)$$

where we donate $W_q^{(h)}$, $W_k^{(h)}$, $W_v^{(h)}$are the projection matrices for queries, keys and values in Head h.

After the unfold operation, the feature map is expanded into a sequence $Z_{(i,j)}$. $x_{(i,j)}$ denote a pixel in sequence $Z_{(i,j)}$. $x_{(f,g)}$ denote any other pixel on sequence $Z_{(i,j)}$ except for $x_{(i,j)}$. $x_{(i,j)}$not only performs self-attention operation on itself, but also performs attention on $x_{(f,g)}$.

For convenience, all pixels in sequence $Z_{(i,j)}$, including $x_{(i,j)}$ and $x_{(f,g)}$, are referred to as $x_{(m,n)}$. The attention weights are:

$$\mathrm{A}(q_{(i,j)}^{(h)}, k_{(m,n)}^{(h)}) = \underset{Z_{(i,j)}}{\mathrm{softmax}} \left(\frac{(q_{(i,j)}^{(h)})(k_{(m,n)}^{(h)})^T}{\sqrt{d}} \right) \quad (3)$$

where d is the feature dimension of $q_{(i,j)}^{(h)}$.

The output $y_{(i,j)}^{(s)}$ of a single head is:

$$y_{(i,j)}^{(s)} = \sum_{m,n \in Z_{(i,j)}} \mathrm{A}(q_{(i,j)}^{(h)}, k_{(m,n)}^{(h)}) v_{(m,n)}^{(h)} \quad (4)$$

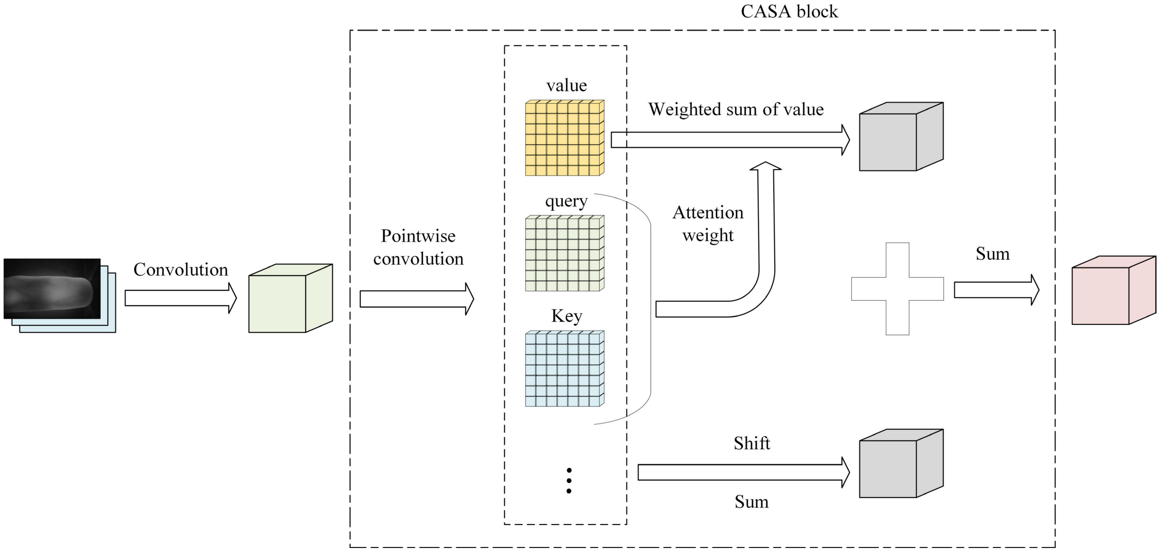


FIGURE 1. The basic structure of convolution and self-attention.

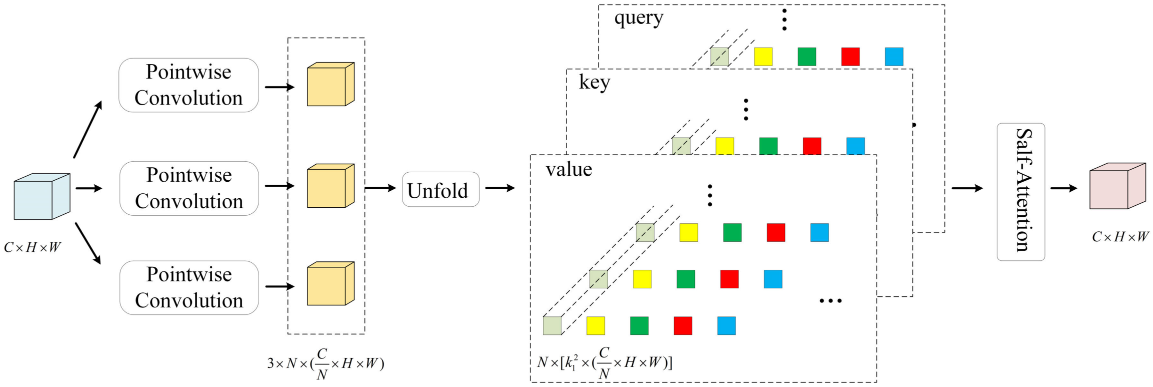


FIGURE 2. The illustration of self-attention.

Different heads need to be connected to obtain the final output $y_{(i,j)}$.

$$y_{(i,j)} = \mathop{C}\limits_{h=1}^{N} y^{(s)}_{(i,j)} \tag{5}$$

The self-attention mechanism for images is decomposed into two steps. The first step is performing pointwise convolutions which projects input features as queries, keys, and values. The second step is calculating the attention weights and weighted sum of values.

B. DECOMPOSITION OF CONVOLUTION

Traditional convolution with kernel size $k_2 \times k_2$ is decomposed into k_2^2 pointwise convolutions, which are used to project. Parallel shift is achieved by depthwise convolutions with specific parameters. Group convolutions [21] are used to ensure that the final dimension is same as self-attention part. The illustration is shown in Fig.3.

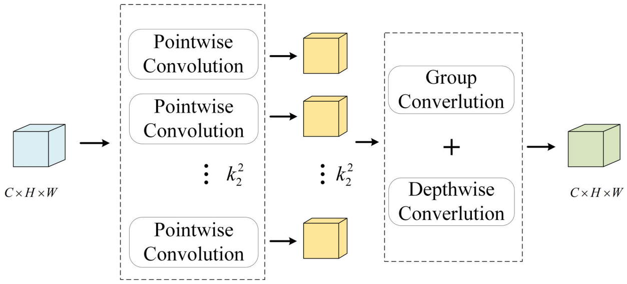


FIGURE 3. The illustration of convolution.

For traditional convolutional neural networks, the computing paradigm is:

$$y_{(i,j)} = \sum_{a,b=1}^{k_2} K_{(a,b)} x_{(i+a-\lfloor k_2/2\rfloor, j+b-\lfloor k_2/2\rfloor)} \tag{6}$$

where $K \in \mathbb{R}^{C_{in}\times C_{out}\times k_2\times k_2}$, $K_{(a,b)}$ is the weight of kernel K, a,b is the kernel position.$a, b \in (1, 2\cdots k_2)$.

Equation (6) shows that the traditional convolution operation is decomposed. For a single position (a, b) of the convolution kernel, the single pixel output is calculated firstly. The output is $y_{(i,j)}^{(a,b)} \in \mathbb{R}^{C_{out}}$.

$$y_{(i,j)}^{(a,b)} = K_{(a,b)} x_{(i+a-\lfloor k_2/2 \rfloor, j+b-\lfloor k_2/2 \rfloor)} \tag{7}$$

The sum of k_2^2 outputs is obtained as follows.

$$y_{(i,j)} = \sum_{a,b=1}^{k_2} y_{(i,j)}^{(a,b)} \tag{8}$$

$y'^{(a,b)}_{(i,j)}$ is the feature map obtained by pointwise convolution, and $y_{(i,j)}^{(a,b)}$ is the feature map obtained by summing a part of each feature map.

$$y'^{(a,b)}_{(i,j)} = K_{(a,b)} x_{(i,j)} \tag{9}$$

$$y_{(i,j)}^{(a,b)} = shift(y'^{(a,b)}_{(i,j)}, a - \lfloor k_2/2 \rfloor, b - \lfloor k_2/2 \rfloor) \tag{10}$$

As shown in Fig.4, convolution operation is achieved by pointwise convolution, shift and summation. Where the convolution kernel size is 2 × 2, stride is 1, the weight of convolution kernel is 1, 2, 3, 4, the weights of pointwise convolution kernels are 1, 2, 3 and 4. Pointwise convolution operate in positions (1, 1), (1, 2), (2, 1) and (2, 2) of the traditional convolution. the single pixel outputs are calculated by pointwise convolution. The output is same as traditional convolution which is shifted and summed.

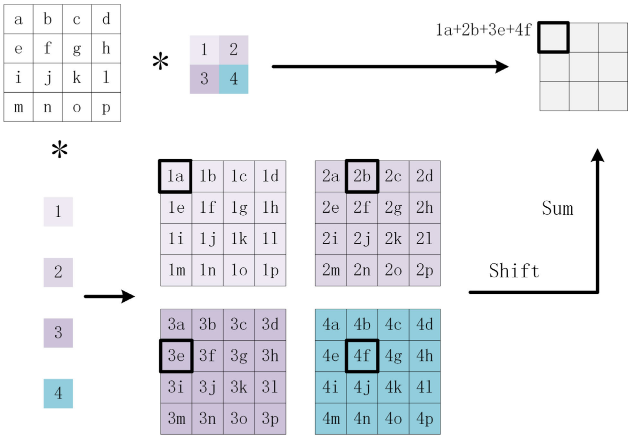


FIGURE 4. Convolution operation using pointwise convolution.

Sliding windows for shift is a serial operation, which is very inefficient. Shift is implemented by depthwise convolution with specified parameters, which greatly improves the degree of parallelism, as show in Fig.5. The number of pointwise k_2^2 is 9. The convolution kernel size of the depthwise convolution used is 3×3. The input feature maps are spliced in the channel dimension, and the convolution kernel is also spliced in the channel dimension. In order that the final feature map is spliced with self-attention, group convolution is used to change the channel.

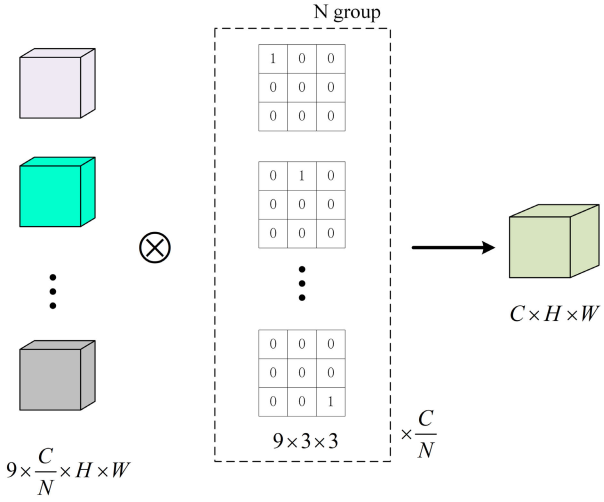


FIGURE 5. Depthwise convolution instead of shift.

In particular, in order to further improve the learning ability of the network, the convolution kernel parameters of depthwise convolution are set to be learnable based on the original initial values as fixed values, and the weights are adjusted during training. In this way, the required features are flexibly adjusted when shifting.

Same as self-attention, the first step is performing pointwise convolutions. The second step is performing shift and summation. After the first step, nine feature maps are selected to implement the convolution operation, and three feature maps are selected to implement the self-attention mechanism. After the second step, convolution extracts local features, and self-attention focuses on key areas.

III. MODULE REPLACEMENT IN RESNET

ResNet's skip connection can effectively prevent gradient disappearance or gradient explosion [22]. The CASA block is connected through the Residual network. CASA blocks are used to replace the traditional convolutional blocks in the deep layers of ResNet. A cosine decaying learning rate is used in training time.

A. INTEGRATION OF SELF-ATTENTION AND CNN

The implementation of self-attention is decomposed into two steps. In first step, the feature map is projected to key, query and value by pointwise convolutions, as in (11). The Flops for this step is $3C^2$, The number of parameters is $3C^2$.

$$\begin{aligned} q_{(i,j)}^{(h)} &= W_q^{(h)} x_{(i,j)}, \\ k_{(m,n)}^{(h)} &= W_k^{(h)} x_{(m,n)}, \\ v_{(m,n)}^{(h)} &= W_v^{(h)} x_{(m,n)} \end{aligned} \tag{11}$$

The second step is shown in (12). The Flops of second step is $2k_1^2 C$, and the number of parameters is 0.

$$y_{(i,j)} = \mathop{C}_{h=1}^{N} \Big(\sum_{m,n \in Z_{(i,j)}} \underset{Z_{(i,j)}}{\text{softmax}} \Big(\frac{(q_{(i,j)}^{(h)})(k_{(m,n)}^{(h)})^T}{\sqrt{d}} \Big) v_{(m,n)}^{(h)} \Big) \tag{12}$$

The implementation of CNN can also be decomposed into two steps. In first step, the feature map is projected by pointwise convolutions as in (13). This step is shared with the first step of implementing self-attention. Flops is $k_2^2C^2$. The number of parameters is $k_2^2C^2$.

$$y'^{(a,b)}_{(i,j)} = K_{(a,b)}x_{(i,j)} \tag{13}$$

Then the second step is shift and summation. The Flops of this step is k_2^2C, and the number of parameters is 0.

Then the second step is shift and sum, as in (14) and (15). The Flops of this step is k_2^2C, and the number of parameters is 0.

$$y^{(a,b)}_{(i,j)} = shift(y'^{(a,b)}_{(i,j)}, a - \lfloor k_2/2 \rfloor, b - \lfloor k_2/2 \rfloor) \tag{14}$$

$$y_{(i,j)} = \sum_{a,b=1}^{k} y^{(a,b)}_{(i,j)} \tag{15}$$

The CASA block is shown in Fig.6. k_2^2 pointwise convolutions are used to project. The projected feature maps are used to do self-attention and convolution respectively. The output of self-attention is Y_{sa}, and the output of convolution is Y_{conv}. The two feature maps are summed according to the proportion. Two ratios are set, which are learnable.

$$Y = r_1 Y_{sa} + r_2 Y_{conv} \tag{16}$$

where r_1 is the ratio of Y_{sa}, and r_2 is the ratio of Y_{conv}.

In the first step, pointwise convolutions are shared, the total Flops of the first step is $3C^2$ and the total number of parameters is $3C^2$. The total Flops of the second step are $(2k_1^2 + 4k_2^2 + k_2^4)C$, and the total number of parameters are $3k_2^2N + k_2^4C$. k_1 and k_2 are very small numbers. C is much greater than k_1 and k_2. The first step takes up most of the computing resources. Therefore, pointwise convolution operation is shared. Compared with CNN, only few parameters are introduced in our method.

Because image has a lot of data redundancy, especially in the shallow layer feature maps, many adjacent pixels are almost no difference. In the shallow layer, traditional convolution operation is used. Convolutional and Self-Attention block replace the traditional convolution operation in the deep layer.

Firstly, Compared with ResNet, the focusing ability of FV-RSA model is significantly improved, which has good feature extraction without the excessively deep network. And the model overcomes also the shortcomings of using the transformer module directly on small datasets, which is not effective on small datasets such as finger vein images. Finally, convolution is used for local representation, and self-attention mechanism is used for global representation, which can solve the problem of the lack of spatial induction biases of traditional self-attention.

B. ADDING COSINE ANNEALING ALGORITHM

Due to the particularity of finger vein images, the differences between different images are very small. The model is easy to fall into local optimum. To solve this problem, cosine annealing is used to adjust the learning rate. If model falls into a local optimum, the learning rate is increased to jump out of the local minimum of the loss function.

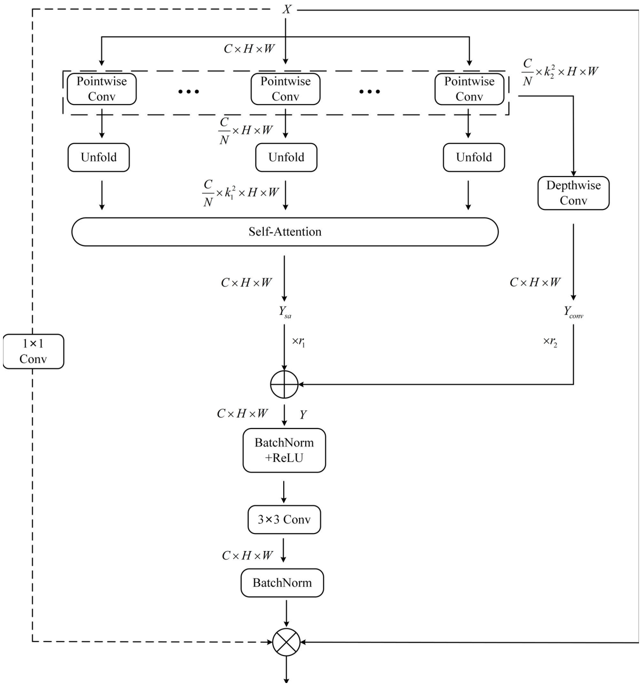


FIGURE 6. Convolution with Self-attention block.

$$\eta_t = \eta^i_{\min} + \frac{1}{2}(\eta^i_{\max} - \eta^i_{\min})(1 + \cos(\frac{T_{cur}}{T_i}\pi)) \tag{17}$$

where i is the number of runs. Let $\eta^i_{\max}$ and $\eta^i_{\min}$ denote the maximum and minimum learning rate, respectively. T_{cur} denote the number of epochs that have been executed. For example, if the total number of samples is 80 and the size of each batch is 16, then the batch will be read in five times in an epoch, and after the first batch is executed in the first epoch, T_{cur} will be updated to 1/5=0.2, and so on. T_i denotes the total number of epochs in the ith run. This paper does not involve restarting which does not need to consider. We fix T_i to be the number of epochs in the model.

IV. EXPERIMENTS AND RESULT ANALYSIS

A. DATASETS AND EXPERIMENT ENVIRONMENT

In order to verify the effectiveness of the model, we verify it on public databases. Datasets from different regions are used. We conduct experiments on three databases: FV-USM, SDUMLA-HMT [23] and THU-FVFDT3 [24].The FV-USM has finger vein images from 123 volunteers. Finger vein images of four fingers are taken 6 times. The SDUMLA-HMT has images from 106 volunteers. Finger vein images of six fingers are taken 6 times. THU-FVFDT3 has finger vein images from 610 volunteers. Finger vein images are taken 8 times. The factors such as different lighting conditions, angles and occlusions are taken into account in the public databases. To ensure that the training datasets and test datasets do not overlap, the datasets are randomly divided into training and test datasets according to the ratio of 9:1.

The number of training epochs is set to 200. The experiments are conducted by the PyTorch framework with version 1.12.1 and Python with version 3.9.12. The training and test run on the NVIDIA RTX 3070Ti GPU.

B. OPTIMIZER SELECTION AND SETTING

The optimizer uses SGD. The learning rate of the optimizer is attenuated using cosine annealing [25]. The initial learning rate is set to 0.001. As the number of epochs increases, the learning rate starts to cosine decay and eventually decreases to one-tenth of the original, which is 0.0001. As is shown in Fig.7.

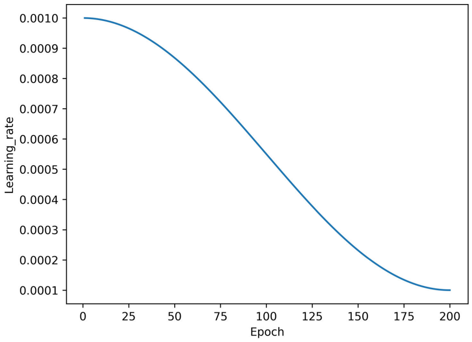


FIGURE 7. Cosine Annealing learning rate.

Big learning rate is used at the beginning, which not only speeds up convergence, but also avoids getting stuck in a local optimum. Experiments show that the model falls into local optimum and seriously affect the convergence without variable learning rate. FV-USM database is used as an example. With a fixed learning rate, the training of first epoch takes 98 seconds, but with a variable learning rate, the training of first epoch takes only 66 seconds. The choice of different optimizer strategies has a significant impact on test accuracy and loss in multiple experiments. Test accuracy and loss for different optimizer strategies are shown in Fig.8. Using the Adam [26] optimizer alone, the model failed to converge. Using the Adam optimizer and cosine decay learning rate, the model is converged, but the accuracy curve has large oscillation amplitude. Using the SGD optimizer and cosine decay learning rate, good convergence result is achieved. The model loss curve using Adam drops too quickly at the beginning which falls into local optimum. A big learning rate is necessary, which helps model to avoid falling into local optimum.

C. EXPERIMENTAL RESULTS AND RESOURCE OCCUPANCY ANALYSIS

This model takes into account the local feature extraction ability of convolution and the global focusing ability of self-attention. We test the model on FV-USM, SDUMLA-HMT and THU-FVFDT3 databases, respectively. The accuracy curve of the public database is shown in Fig.9. The accuracies of both SDUMLA-HMT and THU-FVFDT3 databases are 100%. The accuracy of FV-USM databases is 99.90%. Experiments show that the model has a good result on finger vein recognition.

The proposed network model is compared with the mainstream image feature extraction models in terms of parameters and Flops. Some networks like ResNet18 completely use traditional convolutions. The other networks use self-attention mechanisms. These models adopt the lightweight version. ViT-base uses transformer modules directly. Swin-small [27] is shifted windowself-attention. PVT-small [28] and PVTv2 [29] is sparse global self-attention. 224×224×3 tensor data is used for test. The network parameters are set to the same, and the experimental results are given in Table 1. The number of parameters of our model is about 74% of ResNet18 and about 10% of ViT-base. The Flops of our model is about 89% of ResNet18 and about 9% of ViT-base. From the result, the number of parameters and computational complexity of our model is is reduced.

TABLE 1. parameters and Flops comparison with other methods.

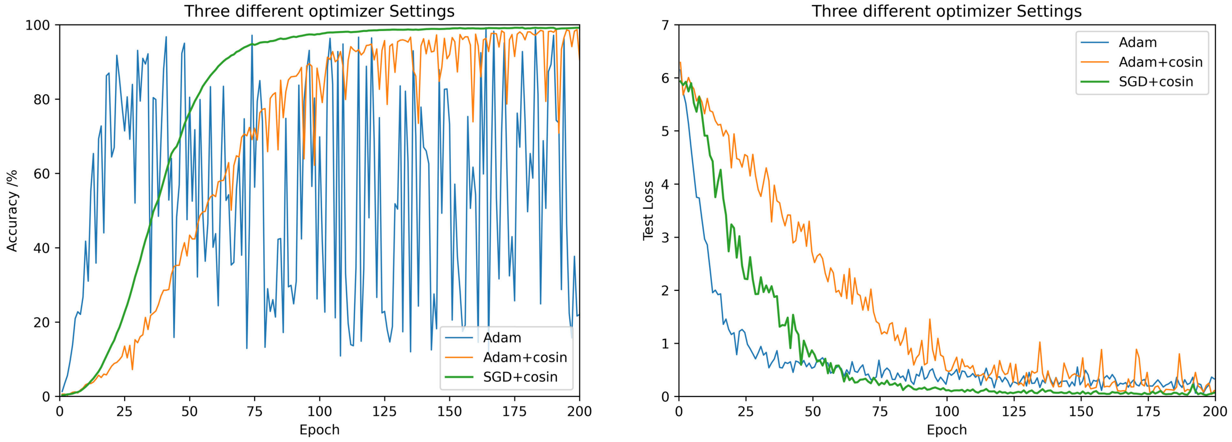
Method	Params	Flops
ResNet18	11.7M	1.8G
ViT-base	86.4M	16.9G
Swin-small	49.6M	8.5G
PVT-small	24.1M	3.7G
PVTv2	14.0M	2.0G
Our method	8.7M	1.6G

TABLE 2. Accuracy (%) comparison with traditional techniques.

Method	Accuracy/%
LLBP	98.56
PLLBP	99.21
Multi-SVM	94.00
SVM	98.00
Our method	100

D. COMPARISON AND ANALYSIS WITH OTHER WORKS

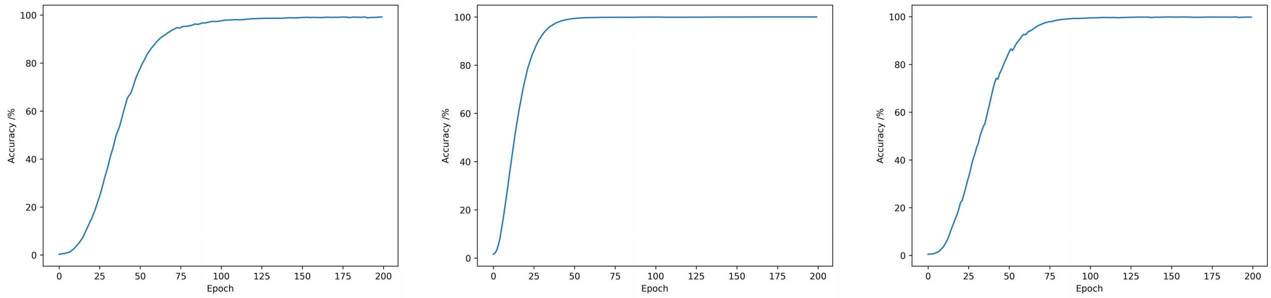
SDUMLA-HMT is used to compare the performance of our model with traditional finger vein recognition methods. In Table 2, our method is contrasted with traditional finger vein recognition technique. Traditional techniques include traditional feature extraction methods and machine learning methods. Traditional feature extraction methods of finger vein images have LBP [30], LLBP [31] and PLLBP [32]. These methods work well, but they require a tedious data



(a)

(b)

FIGURE 8. **Test accuracy and loss for different optimizer strategies. (a) Test accuracy; (b) Test accuracy loss.**



(a)

(b)

(c)

FIGURE 9. **Test accuracy on different datasets. (a) FV-USM; (b) SDUML-HMT; (c) THU-FVFDT3.**

TABLE 3. **Accuracy (%) comparison with other SOTA works.**

Method	Accuracy/%		
	FV-USM	SDUMLA-HMT	THU-FVFDT3
ResNet+SE	99.59	96.70	-
DS-CNN	-	98.50	90.00
Merge CNN	96.75	99.48	99.56
Triplet-Classifier GAN	99.29	94.10	-
ViT-cap	98.68	90.25	-
FV-ViT	99.73	92.77	-
Our method	99.90	100	100

preprocessing. The accuracy of traditional machine learning methods such as Multi-SVM and SVM is lower than other methods. It is very necessary to adopt the method of deep learning.

Further experiments compare the performance of models which use deep learning methods over the past three years. In Table 3, the comparison between our model and other related works is listed. As can be seen from the table, our model has a higher recognition accuracy and performs better than the models for finger vein recognition.

The ResNet+SE model adds channel attention on the basis of ResNet, which ignore spatial information. SDUMLA-HMT images have illumination difference between the edge region and the middle region, which requires the model that has a high ability for extracting spatial information. DS-CNN [33] model uses depthwise convolution, which just reduces the number of parameters. Merge CNN [34] model uses deeply-fused convolution, which is still traditional convolution in essence. Their feature extraction ability is relatively weak. Triplet-Classifier GAN [35] model uses GAN to augment the training data to improve the discrimination ability of CNN. This method only performs data augmentation and does not improve the feature extraction ability. ViT-cap [36] and FV-ViT use the ViT model directly, that is the transformer module. The above two, they just use global attention, but they do not even perform as well as a convolutional neural network on the small datasets of finger veins.

V. CONCLUSION

According to the characteristics of finger veins, the FV-RNSA model is proposed in this paper. In order to have focus globally ability, the self-attention mechanism is used, which shares the linear projection process with convolution. This model combines global focusing ability of self-attention and local feature extraction ability of convolution, which is lightweight and has high feature extraction ability. The model adds cosine decay learning rate to avoid falling into local optimum. Experiments show that the proposed model works well on the three databases. The model can reach 100% accuracy on SDUMLA-HMT and THU-FVFDT3. The model accuracy on FV-USM is also improved compared with other works. The number of parameters and computational complexity of our model are lower than other models using CNN or self-attention. We will conduct further lightweight research without reducing the accuracy in the future.

REFERENCES

[1] B. Moayer and K.-S. Fu, "A tree system approach for fingerprint pattern recognition," *IEEE Trans. Comput.*, vol. C-25, no. 3, pp. 262–274, Mar. 1976.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, Jan. 1991.

[3] R. P. Wildes, J. C. Asmuth, G. L. Green, S. C. Hsu, R. J. Kolczynski, J. R. Matey, and S. E. McBride, "A system for automated iris recognition," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Feb. 1994, pp. 121–128.

[4] M. Kono, "A new method for the identification of individuals by using of vein pattern matching of a finger," in *Proc. 5th Symp. Pattern Meas.*, Yamaguchi, Japan, 2000, pp. 9–12.

[5] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Mach. Vis. Appl.*, vol. 15, no. 4, pp. 194–203, Oct. 2004.

[6] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE Trans. Inf. Syst.*, vols. E90–D, no. 8, pp. 1185–1194, Aug. 2007.

[7] W. Song, T. Kim, H. C. Kim, J. H. Choi, H.-J. Kong, and S.-R. Lee, "A finger-vein verification system using mean curvature," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1541–1547, Aug. 2011.

[8] M. S. Mohd Asaari, S. A. Suandi, and B. A. Rosdi, "Fusion of band limited phase only correlation and width centroid contour distance for finger based biometrics," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3367–3382, Jun. 2014.

[9] R. P. Kumar, R. Agrawal, S. Sharma, M. K. Dutta, C. M. Travieso, and J. B. Alonso-Hernández, "Finger vein recognition using integrated responses of texture features," in *Proc. 4th Int. Work Conf. Bioinspired Intell. (IWOBI)*, Jun. 2015, pp. 209–214.

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

[11] W. Yang, C. Hui, Z. Chen, J.-H. Xue, and Q. Liao, "FV-GAN: Finger vein representation using generative adversarial networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2512–2524, Sep. 2019.

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

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[14] X. Li and B.-B. Zhang, "FV-ViT: Vision transformer for finger vein recognition," *IEEE Access*, vol. 11, pp. 75451–75461, 2023.

[15] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, and G. Huang, "On the integration of self-attention and convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 805–815.

[16] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. Gelly, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[18] H. Ren, L. Sun, J. Guo, C. Han, and F. Wu, "Finger vein recognition system with template protection based on convolutional neural network," *Knowl.-Based Syst.*, vol. 227, Sep. 2021, Art. no. 107159.

[19] B. Hou and R. Yan, "ArcVein-Arccosine center loss for finger vein verification," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

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

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[23] Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A multimodal biometric database," in *Proc. 6th Chin. Conf. Biometric Recognit.* Cham, Switzerland: Springer, 2011, pp. 260–268.

[24] W. Yang, G. Ma, F. Zhou, and Q. Liao, "Feature-level fusion of finger veins and finger dorsal texture for personal authentication based on orientation selection," *IEICE Trans. Inf. Syst.*, vol. 97, no. 5, pp. 1371–1373, 2014.

[25] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[28] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 548–558.

[29] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "PVT V2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, Sep. 2022.

[30] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[31] 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.

[32] Y. Lu, S. J. Xie, S. Yoon, and D. S. Park, "Finger vein identification using polydirectional local line binary pattern," in *Proc. Int. Conf. ICT Converg. (ICTC)*, Oct. 2013, pp. 61–65.

[33] K. Shaheed, A. Mao, I. Qureshi, M. Kumar, S. Hussain, I. Ullah, and X. Zhang, "DS-CNN: A pre-trained exception model based on depth-wise separable convolutional neural network for finger vein recognition," *Expert Syst. Appl.*, vol. 191, Apr. 2022, Art. no. 116288.

[34] I. Boucherit, M. O. Zmirli, H. Hentabli, and B. A. Rosdi, "Finger vein identification using deeply-fused convolutional neural network," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 3, pp. 646–656, Mar. 2022.

[35] B. Hou and R. Yan, "Triplet-classifier GAN for finger-vein verification," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[36] Y. Li, H. Lu, Y. Wang, R. Gao, and C. Zhao, "ViT-Cap: A novel vision transformer-based capsule network model for finger vein recognition," *Appl. Sci.*, vol. 12, no. 20, p. 10364, Oct. 2022.

ZHIBO ZHANG received the B.E. degree in electrical engineering and automation from Lanzhou Jiaotong University, Lanzhou, China, in 2021. He is currently pursuing the M.S. degree in measuring and testing technologies and instruments with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China.

His current research interests include computer vision and deep learning.

GUANGHUA CHEN received the B.S. degree in applied geophysics from the Central South University of Technology, in 1996, and the Ph.D. degree in control theory and control engineering from Shanghai University, in 2002.

He joined the Microelectronic Research and Development Center, Shanghai University, in 2002, where he is currently an Associate Research Professor. He has published more than 80 papers in various journals and conference proceedings. His research interests include deep learning, VLSI design for video signal processing, and embedded hardware systems design and application.

WEIFENG ZHANG received the B.S. degree in mechanical engineering from the Qingdao University of Science and Technology, Qingdao, China, in 2000, the M.S. degree in mechanical engineering from Shandong University, Jinan, China, in 2005, and the Ph.D. degree in mechanical engineering from the Qingdao University of Science and Technology, in 2017.

From 2006 to 2020, he was a Lecturer in mechanical engineering automation. Since 2020, he has been an Adjunct Professor with the Mechanical and Electrical Engineering College, Qingdao University of science and Technology. He is the author of four books more than 30 articles, and more than 20 inventions. His research interests include the micro-nano and ultrasonic special processing technology and equipment, industrial visual detection and image processing, intelligent manufacturing and automation equipment, and robot technology.

Dr. Zhang was a Trustee of the Group and Intelligent Integration Technology Branch of China Mechanical Engineering Society, and also the Vice Dean of the School of Mechanical and Electrical Engineering, Qingdao University of Science and Technology.

HUIYANG WANG received the B.E. degree in measurement and control technology and instruments with a primary focus on embedded development from Yangtze University, Jingzhou, China, in 2022. He is currently pursuing the M.S. degree in measuring and testing technologies and instruments with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China.

His current research interests include image processing and deep learning.