Double Branch Lightweight Finger Vein Recognition based on Diffusion Model

Zhiyong Tao[1], Yajing Gao[2], Sen Lin[3]
School of Electronic and Information Engineering, Liaoning Technical University, Huludao, China 125105[1, 2]
School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang, China 110159[3]

***Abstract*—Aiming at the problems of high complexity, insufficient global information extraction and easy overfitting in finger vein recognition, a finger vein recognition method based on diffusion model is proposed. Firstly, finger vein images are generated according to the dataset by diffusion model, which is used to prevent overfitting; secondly, a streamlined convolutional neural network is used to form a two-branch lightweight backbone network with an improved multi-head self-attention mechanism, which can effectively reduce the complexity of the model; and finally, in order to maximally extract the image's overall information, the convolution is used to merge the extracted local and global features, and the recognition results are output. The algorithm can reach a maximum recognition rate of 99.78% on multiple datasets, while the number of references is only 2.15M, which further reduces the complexity of the algorithm while maintaining a high accuracy compared to other novel finger vein recognition algorithms as well as lightweight convolutional neural network models. As the first attempt in this field, it will provide new ideas for future research work.**

Keywords*—Finger vein recognition; convolution neural network; diffusion model; multi-head self-attention mechanism; lightweight network

I. Introduction

Lately, the focus of researchers on finger vein recognition technology has intensified, attributed to its exceptional security and precision. Since the vein network of each individual is hidden under the skin, finger vein-based biometrics has a massive advantage in live identification. As a developing technology, finger vein recognition-based biometrics is far from flawless. Various internal and external factors can impact the performance of finger vein verification. These factors include: Lighting Conditions, Finger Placement Angle, and Uniform illumination. Therefore, high-precision and high-robustness algorithms are essential for feature extraction, recognition, or verification of finger vein images. A typical finger vein recognition process includes image acquisition, preprocessing, feature extraction, and matching. In the finger vein image acquisition process, a finger vein image acquisition device consisting of an image sensor and an infrared light source is used. Preprocessing of the acquired finger vein images is carried out to facilitate the subsequent feature extraction process. Suppressing noise, improving image contrast, and performing data augmentation are common image preprocessing methods.

Feature extraction in finger vein recognition involves two main categories: traditional recognition methods and deep learning methods. Traditional recognition methods for feature extraction can be further classified into three distinct categories: template-based methods [1], representation-based methods [2], and feature-based [3][4][5] learning methods. These methods require manual labeling of parameters, depend on image quality, and have cumbersome recognition steps. Compared with machine learning methods, deep learning [6][7][8][9] based methods can achieve more stable recognition results by acquiring more profound image features through Convolutional Neural Networks (CNN). Therefore, some researchers proposed deep learning-based finger vein recognition methods. For example, Radzi et al. [10] proposed a CNN-based finger vein recognition method, Fang et al. [11] proposed a lightweight two-channel network to improve the verification of finger veins by extracting the mini-region of interest (ROI), Zhang et al. [12] proposed Domain Adaptation Finger Vein Network (DAFVN) improve the final recognition result by extracting illumination invariant features in the image and reducing the effect of light on the recognition result. Recently, researchers proposed the Vision Transformer (ViT) [13] method, which has attracted widespread attention in deep learning. Compared with CNN, ViT focuses more on global features and has shown excellent performance in several domains. In addition, researchers have proposed some improved methods, such as Liu et al. [14] proposed Swin Transformer, which obtains global and local features by constructing hierarchical feature maps and sliding windows, with better experimental results but high model complexity, and Peng et al. [15] proposed Parallel Network Architecture, which makes use of convolution and Multi-Head Self-Attention (MHS) mechanism. Head Self-Attention (MHSA) to extract local and global features in parallel, which improves the network performance but is ineffective for small datasets. Based on the advantages of Transformer, researchers started applying it to finger vein recognition. Huang et al. [16] proposed Finger Vein Transformer (FVT) model for recognition, which achieves multi-scale feature extraction by reducing the number of tokens layer by layer, but exploiting the Transformer increases the complexity and computation at the same time.

To enhance the efficiency of recognizing finger veins, some researchers have introduced data enhancement techniques to make the model better adapt to finger vein images in various scenarios. Yang et al. [17] proposed Finger Vein Representation Using the Generative Adversarial Networks (FV-GAN) model, which was the first time GAN was in the field of finger vein recognition. Choi et al. [18] proposed a Conditional Generative Adversarial Network (CGAN) to recover blurred images. They used a deep convolutional neural

network for the finger vein images—a convolutional neural network for finger vein image recognition. Hou et al. [19] proposed a ternary classifier, GAN, for generating training data to improve the learning ability of the CNN classifier. Although high-quality images can be generated using GAN networks, they require more extensive databases and may be unstable during training.

Diffusion Model (DM) [20] is a recently emerged deep generative model for high-quality image generation, which is rapidly evolving and is widely used in tasks such as text-to-image generation, image-to-image generation, and video image generation. Also, one of the data enhancement methods, the diffusion model has a more straightforward training process and generates higher-quality images compared to GAN networks. Jonathan et al. [20] proposed the Denoising Diffusion Probabilistic Model (DDPM), which is used for image generation tasks, generating the image quality is higher than other generation models such as GAN. Robin [21] et al. proposed Latent Diffusion Models (LDM), which achieve image generation by introducing a cross-attention conditioning mechanism, which significantly improves the training and sampling efficiency without degrading its quality.

Summarizing the above research methods, the existing algorithms for finger vein recognition generally have high complexity, recognition accuracy needs to be improved, and the training process is unstable, so a two-branch lightweight finger vein recognition model (FV-DM) based on the diffusion model is designed to achieve finger vein image generation using the diffusion model to solve the problem of overfitting due to the small finger vein dataset. The CNN and the improved E-MSHA module are used to extract image features in parallel with the dual-branching in the feature extraction process to avoid the problem of low accuracy caused by insufficient feature extraction, while the diffusion model is used in the finger vein recognition process in order to explore a new way of finger vein recognition. The comprehensive experiments on the self-constructed dataset and three public datasets show that FV-DM all achieve better recognition results, as well as lower model parameters and computational complexity, shorter recognition time, and lower Equal Error Rate (EER).

The remainder of this paper is organized as follows. Section 2 introduces our modelling approach and explains how it works. Section 3 describes the experiments conducted to validate the performance of the model. In Section 4, we summarize the paper and make suggestions for future work.

II. METHODOLOGY

A. Diffusion Model

Recently, the diffusion model as a generative model has received more and more attention from researchers due to its powerful image generation ability. As shown in Fig. 1, the diffusion model is mainly divided into the forward noise addition process and the reverse denoising process. The solid line indicates the forward noise addition process and the dashed line indicates the reverse denoising process.

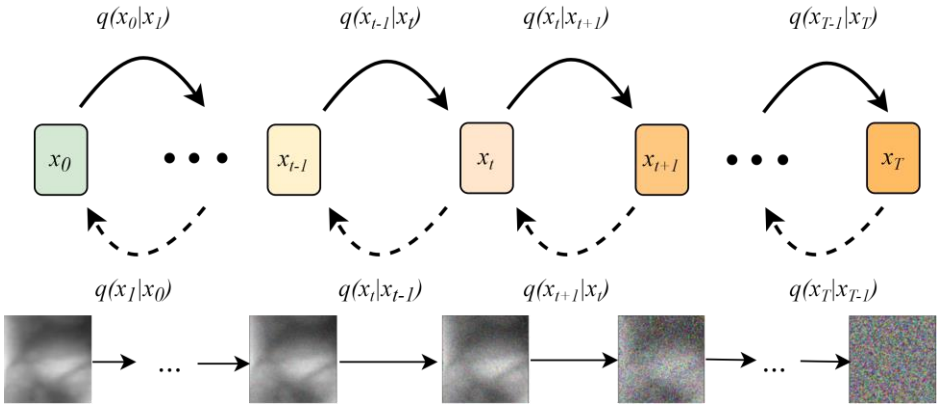


Fig. 1. Network structure diagram of diffusion model.

1) Forward noise addition process: The forward noise addition process uses Gaussian noise to gradually add noise to the input image, generating a series of noise samples $x_0, x_1, ..., x_T$ until the image becomes a pure noise image. Assuming that $q(x_0)$ is the probability distribution of the real image, $q(x_t \mid x_{t-1})$ represents the probability distribution of the current image x_t obtained by adding noise to the previous step image x_{t-1} in the forward noise addition process, and the mathematical expressions for each step of the process of adding Gaussian noise are shown in (1):

$$q(x_t \mid x_{t-1}) = N(x_t \mid \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I) \tag{1}$$

Where β_t is the diffusivity and t varies with time. The formula is expressed as a mean $\mu_t = \sqrt{1-\beta_t}\, x_{t-1}$ with a variance Gaussian $\sigma_t^2 = \beta_t$ distribution. If the final image x_T is obtained through x_0, the whole process can be regarded as a Markov chain from $t=1$ to the moment $t=T$, as shown in Eq. (2):

$$q(x_{0:T}) = q(x_0)\prod_{t=1}^{T} q(x_t \mid x_{t-1}) \tag{2}$$

In the forward noise addition process, Eq. (1) can be expressed as by the simplified way in literature [23]:

$$x_t = \sqrt{1-\beta_t}\,x_{t-1} + \sqrt{\beta_t}\,z_{t-1} \tag{3}$$

Where z_{t-1} denotes the noise at moment $t-1$. The x_t at any moment is obtained from the original image x_0 with the formula shown in the following equation:

$$\alpha_t = 1-\beta_t \tag{4}$$

$$\overline{\alpha_t} = \prod_{i=1}^{t}\alpha_i \tag{5}$$

$$\begin{aligned} x_t &= \sqrt{\alpha_t}\,x_{t-1} + \sqrt{1-\alpha_t}\,z_{t-1} \\ &= \sqrt{\overline{\alpha_t}}\,x_0 + \sqrt{1-\overline{\alpha_t}}\,z_t \end{aligned} \tag{6}$$

$$q(x_t \mid x_0) = \mathrm{N}\left(x_t \mid \sqrt{\overline{\alpha_t}}\,x_0, (1-\overline{\alpha_t})I\right) \tag{7}$$

z_t is denoted as a Gaussian distribution satisfying $\mathrm{N}(0, I)$

2) Reverse denoising process: The reverse denoising process is also known as the inverse diffusion process. The main purpose is to gradually predict the target image x_0 from the purely noisy image x_T, i.e., to derive the x_{t-1} distribution from x_t, which can be transformed into what is shown in Eq. (8) by using Bayes' formula:

$$q(x_{t-1} \mid x_t) = q(x_t \mid x_{t-1})\frac{q(x_{t-1})}{q(x_t)} \tag{8}$$

According to the forward noise addition process, $q(x_t \mid x_{t-1})$ is known, and for $q(x_{t-1})$ and $q(x_t)$, it can be solved by adding the known condition x_0, as shown in the following equation:

$$q(x_{t-1} \mid x_t, x_0) = q(x_t \mid x_{t-1}, x_0)\frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} \tag{9}$$

In the process of reverse denoising, the features in the input noisy image are predicted by the neural network, and this paper chooses U-Net as the model for noise prediction. U-Net is a U-shaped network structure, which consists of downsampling on the left side, upsampling on the right side, and cross-layer connections. The downsampling reduces the size of the feature map through the convolution operation and reduces the computational cost. The upsampling gradually restores the feature map to its original size through the inverse convolution operation, and the cross-layer connection is used to splice the features between the downsampling and the upsampling, which can effectively integrate the features of different levels of the image. For normalisation, Group Normalization (GN) is chosen. Finally, for the downsampling and upsampling operations in U-Net, the convolution with a step size of 2 and the inverse convolution are chosen, respectively. The specific structure is shown in Fig. 2.

B. Design of the Network Model

Inspired by DDPM [20], a finger vein recognition network based on a diffusion model is designed. It is shown in Fig. 3

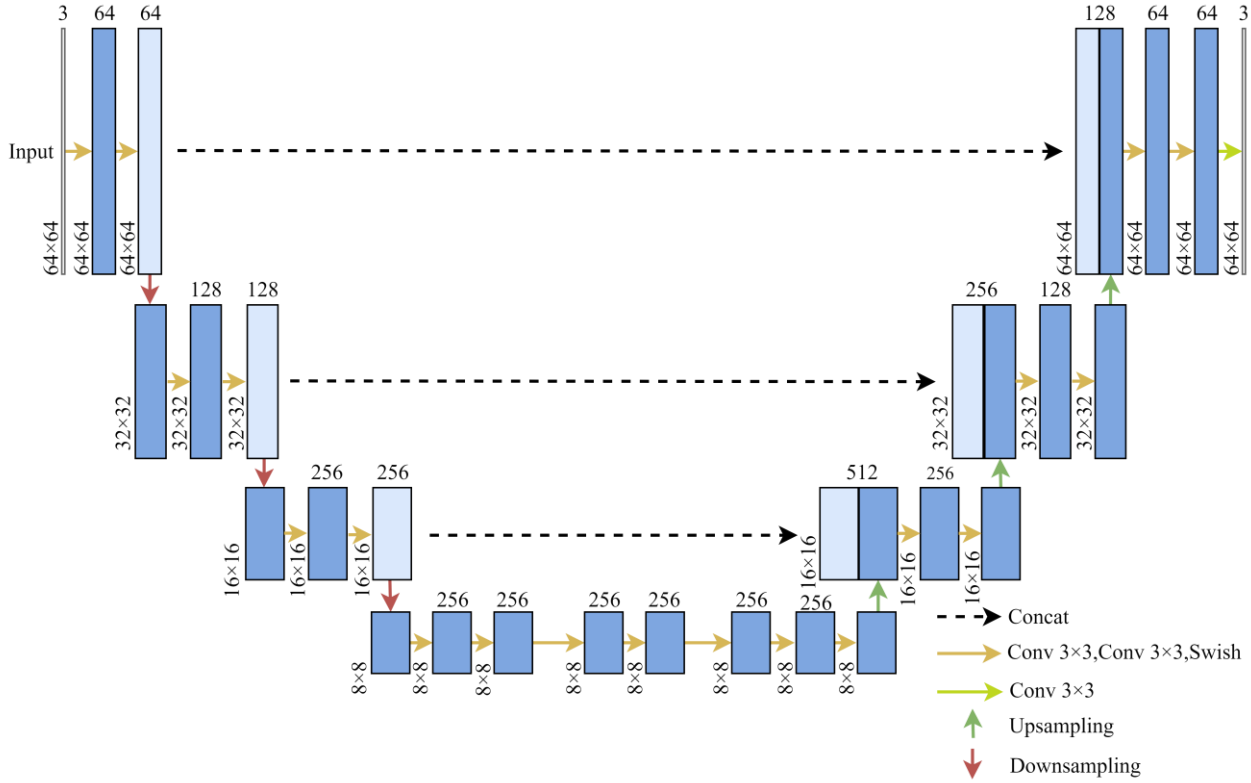


Fig. 2. U-Net Structure diagram.

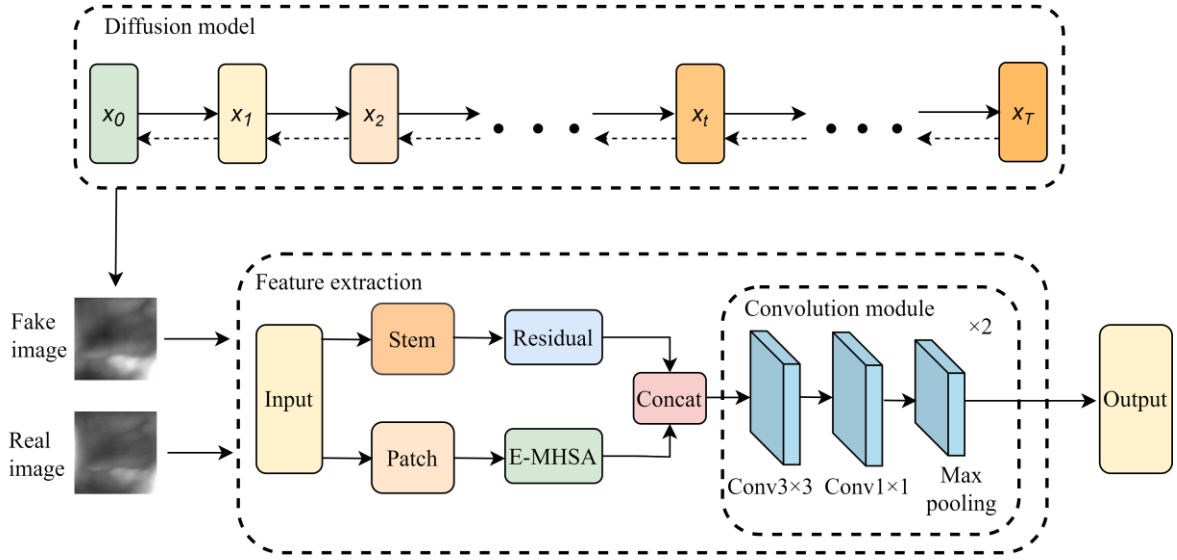


Fig. 3. Structure diagram of diffusion model.

The network structure contains two main parts: the image generation part and the image feature extraction part. Firstly, the diffusion model is used to achieve image generation by forward noise addition process and reverse denoising process. Then, the generated image is passed into the feature extraction network along with the actual image, and the Residual [22] module and the E-MHSA module extract the global and local features of the image, respectively, and stitch the features after extraction, which allows for better fusion of the features and further improves their expressive ability. The fused features are passed through the convolution module to achieve the extraction of deeper features. As shown in the figure, the convolution module consists of an ordinary convolution of size, an ordinary convolution of size, and a maximum pooling layer and is stacked twice in the feature extraction process to extract image features more comprehensively.

C. Residual Structure

The model uses a Residual module and an improved E-MHSA module for local and global feature extraction in the early stage of feature extraction. The Residual module consists of an inverted residual structure, which can effectively reduce the computational cost while extracting the local features of the image. The specific structure is shown in Fig. 4.

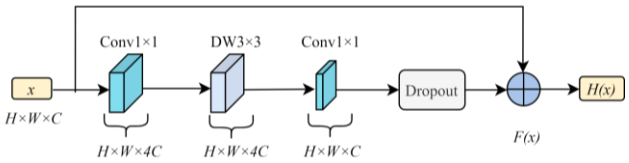


Fig. 4. Residual structure diagram.

The inverted residual structure contains two ordinary convolutions, a DW convolution, a Dropout layer and a jump connection. The input information is first increased by 1×1 size ordinary convolution, then the image size is transformed by DW convolution with a convolution kernel size of 3×3, and finally the number of channels is decreased by 1×1 size ordinary convolution, and the Dropout is used to randomly discard the features to prevent the parameter from relying too much on the training data and the phenomenon of overfitting. Finally, the output of the Dropout layer is added with the result of the jump join to complete the output of the information. The jump connection is mathematically defined as:

$$H(x) = F(x) + x \tag{10}$$

Where $F(\cdot)$ is a function containing convolution, pooling, and modified linear unit operations, x inputs the feature map, and $H(x)$ is the output of the inverted residual structure. The inclusion of jump connections in the inverted residual accelerates the convergence of the network and improves the generalization of the model.

D. E-MHSA Structure

Since MHSA has a strong ability to capture low-frequency signals, which are used to provide global information, the enhanced E-MHSA module is used in this paper for global feature extraction. Compared to the traditional MHSA, E-MHSA incorporates average pooling operation and down-sampling before the computation of the attention mechanism in order to reduce the computational cost and achieve a more efficient and lightweight deployment. As shown in Fig. 5, the E-MHSA module is similar to the Transformer Block in ViT, which first captures the low-frequency signals through E-MHSA with the following formula:

$$\text{E-MHSA}(x) = concat(SA(x_1), SA(x_2), ..., SA(x_h))W^0 \tag{11}$$

Where $x = [x_1, x_2, ..., x_h]$ denotes the division of input feature x into multiple heads in the channel dimension and h is the number of heads divided. In this paper, we take 8 as the number of heads for the attention mechanism. $SA(\cdot)$ is the computational formula for the attention mechanism, and the formula is as follows:

$$SA(x) = Attention(X \cdot W^{Q}, P_s(X \cdot W^{K}), P_s(X \cdot W^{V})) \quad (12)$$

where P_s represents the average pooling operation with step size s.

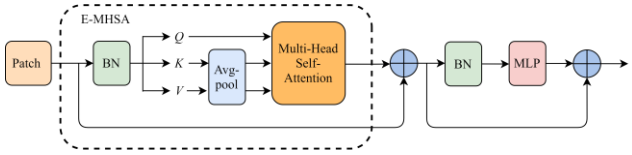


Fig. 5. Structure diagram of E-MHSA.

III. Experiments and Analyses

A. Presentation of Datasets

The experiments were conducted on three public datasets, FV-USM [24], SDUMLA-HMT [25], THU-FVFDT2 [26], and with a self-constructed dataset, FV-SIPL, which were divided in a 2:1 ratio, except for the THU-FVFDT2 dataset, in which the training and test sets were equally divided. The data information is shown in Table I.

TABLE I. Data Information From Four Datasets

Dataset	Total number of categories	Total image count	Total training sets	Total test sets
FV-USM	492	5904	3936	1968
SDUMLA-HMT	636	3816	2544	1272
THU-FVFDT2	610	1220	610	610
FV-SIPL	108	1296	864	432

1) FV-USM: The dataset was provided by Universiti Teknologi Malaysia and contained finger vein images from 123 volunteers, with 12 images captured from each of the four fingers of each volunteer. Therefore, the whole dataset covers a total of 492 finger categories and 5904 images. The size of each of these images is 640×480pixels.

2) SDUMLA-HMT: The dataset was provided by Shandong University, which contains finger vein images of 106 volunteers, and 6 images were collected for each index, middle, and ring finger of each volunteer's hands the whole dataset covers a total of 636 finger categories and 3816 images, where each image size is 320×240pixels.

3) THU-FVFDT2: The dataset was provided by Tsinghua University and contained finger vein images of 610 volunteers. Finger vein images were collected twice for each volunteer, with a total of 1220 images, each with a size of 200×100pixels.

4) FV-SIPL: This dataset was made by the Signal and Information Processing Laboratory of Liaoning University of Engineering and Technology by using infrared finger vein acquisition sensors to collect finger vein images from 27 volunteers. Among them, 12 images were acquired for each of the four fingers of each volunteer, and the whole dataset covered 108 finger categories and 1296 images in total. The size of each image is 176×415 pixels.

B. Image Preprocessing

In order to facilitate the subsequent process of image feature extraction, preprocessing operations are performed on the image. Taking the FV-USM dataset as an example, the main processes are shown in Fig. 6(a) to (d) below.

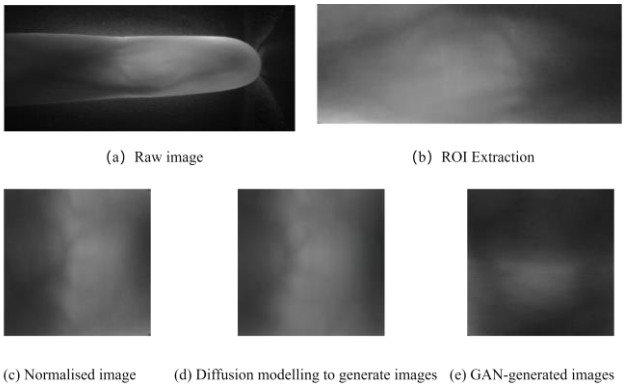


Fig. 6. Image preprocessing process.

For the original images in the dataset, first of all, through the ROI extraction operation, to reduce the interference of irrelevant information on the recognition results, and then carry out image normalisation, pass the normalised images into the diffusion model, and set the number of iterations in the training process to be 10000, and the time T to be 1000. the same parameter settings are carried out on the commonly used generative model GAN, and it can be seen through Fig. 6(d) and Fig. 6(e) that the images generated by the diffusion model are clearer and show more similar image features to the original image, so the use of diffusion model is chosen as the data enhancement method in FV-DM.

C. Experimental Environment and Parameter Settings

The experiments were conducted under the Linux operating system using PyTorch1.7 framework, and the graphics card used for training and testing was GeForce RTX 3090. The learning rate was set to 0.001, the batch size was set to 16, and Stochastic Gradient Descent (SGD) was chosen as the optimiser, where the momentum was set to 0.9. The input size of finger veins was uniformly adjusted to 224×224pixels, and the final experimental results were obtained by training iterations 100 times.

D. Evaluation Indicators

In order to evaluate the performance and advantages of the model, metrics such as Accuracy, Equal Error Rate, Average Processing Time for a Single Image, Number of Parameters, and Floating Point Operations (FLOPs) are selected for evaluation. Accuracy rate, as one of the commonly used metrics in finger vein recognition, can reflect the ability of the model to correctly identify different categories of samples in the entire dataset. The formula for accuracy rate is shown in (13):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

Where *TP* denotes the number of correct positive sample predictions, *TN* denotes the number of correct negative sample predictions, *FN* denotes the number of incorrect negative sample predictions, and *FP* denotes the number of incorrect positive sample predictions. In image recognition tasks, the EER value is usually used as an indicator to evaluate the good or bad performance of the model, which is determined by the False Acceptance Rate (FAR) and the False Rejection Rate (FRR). The formulas for FAR and FRR are shown below:

$$FAR = \frac{FP}{FP + TN} \tag{14}$$

$$FRR = \frac{FN}{TP + FN} \tag{15}$$

Wherein the number of samples for incorrect acceptance and incorrect rejection is defined by a predetermined threshold. When the threshold of matching is greater than the preset threshold, it is determined to be incorrectly accepted, and vice versa is determined to be incorrectly rejected. The value when *FRR* and *FAR* are equal is the equal error rate. The equal error rate reflects the overall performance of the recognition method. The smaller the value of equal error rate, the better the performance of the recognition method.

E. Comparison Experiment

In order to verify the effectiveness of the FV-DM method, it is compared with the classical Transformer network models: the VIT-B, Swin-T, Conformer-B, Next-ViT and the lightweight CNN network model EfficientNetV2. The recognition accuracy results of the different methods on the datasets are shown in Table II. The results in the table show that the methods proposed in this paper achieve the best recognition results on all four datasets. Bolding indicates the best results and underlining indicates the second best results. In addition to the accuracy comparison, the average processing time, number of parameters and FLOPs of individual images for the different methods were also compared, as shown in Table III. In terms of the average processing time for a single image, MobileNetV2 is 2.27ms, which is 0.53ms faster than FV-DM, which is due to the MSHA contained in FV-DM. Other than that, FV-DM outperforms the other methods.

TABLE II. RECOGNITION ACCURACY OF DIFFERENT METHODS ON FOUR DATA SETS (UNIT: %)

Method	FV-USM	SDUMLA-HMT	THU-FVFDT2	FV-SIPL
VIT-B[13]	58.67	63.66	59.55	76.28
Swin-T[14]	95.0	93.33	76.01	95.12
Conformer-B[15]	99.0	98.67	96.64	99.33
Next-ViT[27]	98.56	99.0	98.87	99.53
EfficientNetV2[28]	98.10	98.07	97.78	98.20
MobileNetV2[22]	98.12	99.0	98.32	99.0
ResNet101[29]	98.33	98.34	98.21	99.0
FV-DM(Our)	**99.67**	**99.66**	**99.10**	**99.78**

TABLE III. COMPARISON OF EVALUATION INDEX RESULTS OF DIFFERENT METHODS

Method	Time/ms	Parameters/M	FLOPs/G
VIT-B	11.30	103.03	16.88
Swin-T	7.21	28.27	4.37
Conformer-B	7.15	96.63	21.01
Next-ViT	3.52	31.76	5.79
EfficientNetV2	3.49	21.46	2.90
MobileNetV2	**2.27**	3.50	0.33
ResNet101	7.61	44.55	7.84
FV-DM(Our)	2.80	**2.15**	**0.19**

In this paper, four datasets are used to compare different recognition methods, including VIT-B, Swin-T and Conformer-B. The results are shown in Fig. 7.

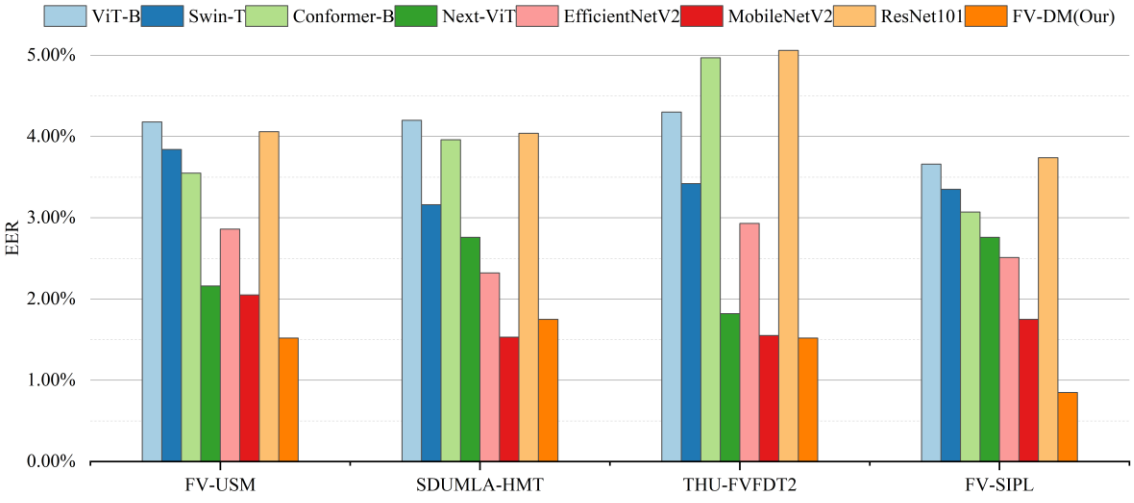


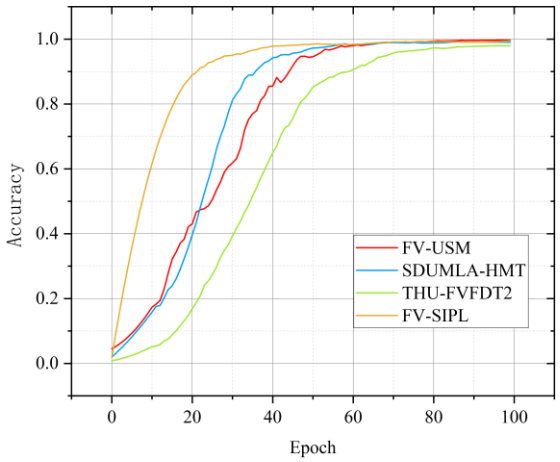
Fig. 7. Comparing the equal error rates of different methods.

On the SDUMLA-HMT dataset, the equal error rate of FV-DM is slightly higher than that of MobileNetV2, but except for that, FV-DM maintains the lowest equal error rate, which indicates that the FV-DM method has excellent performance in finger vein recognition, and it can be used as an effective recognition method. Compared with other methods, the FV-DM method has higher accuracy and better robustness, so it has a wide range of application prospects in practical applications.

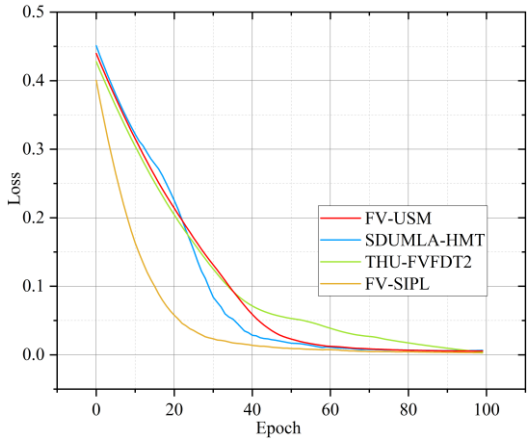
TABLE IV. RECOGNITION ACCURACY OF DIFFERENT METHODS ON PUBLIC DATASETS (UNIT: %)

Method	FV-USM	SDUMLA-HMT	THU-FVFDT2
Merge CNN[30]	96.15	89.99	—
DS-CNN[31]	—	98.00	89.00
Semi-PFVN[32]	94.67	96.61	—
LFVRN_CE[33]	98.58	97.75	—
DGLFV[34]	—	99.25	—
CMrFD[35]	98.33	98.92	—
FVT	**99.73**	97.90	90.66
TFHFT-DPFNN[36]	—	98.00	—
CNNs[37]	97.95	—	—
Coding SchemeA[38]	99.59	95.91	—
FV-GAN	—	—	98.52
Triplet-classifier GAN	99.66	99.53	—
FV-DM(Our)	99.67	**99.66**	**99.10**

FV-DM is compared with novel finger vein models in recent years, and the results are shown in Table IV. Among them, FV-DM obtained the highest recognition accuracy on both public datasets, SDUMLA-HMT and THU-FVFDT2. The recognition accuracy on the FV-USM dataset is lower than that of the FVT method by 0.06%, but it is higher than that of FVT on the SDUMLA-HMT and THU-FVFDT2 datasets by 1.77% and 9.03%, respectively. Therefore, from the overall results, FV-DM recognition results are better.



(a) Recognition accuracy curves for the four datasets.



(b) Loss curves for the four datasets.

Fig. 8. Recognition accuracy and loss curve of FV-DM on four datasets.

By comparing the novel finger vein recognition algorithms in recent years, FV-DM has better performance in terms of recognition accuracy, recognition time, complexity, etc. the recognition accuracy versus test loss curves of FV-DM on the four datasets are shown in Fig. 8.

F. Ablation Experiment

Ablation experiments were conducted in order to better validate the effectiveness of the modules in each part of the network. Under the premise that the rest of the conditions remain unchanged, modules such as Residual, E-MHSA, convolutional module and diffusion model are added to the network sequentially, and the accuracy rate is tested with the FV-SIPL dataset as an example, and the experimental results are shown in Table V. From the table, it can be seen that the accuracy rate increases step by step after the modules are added. Residual and E-MHSA need to be further fused after extracting the local and global features, respectively, to achieve a more comprehensive feature extraction, so the accuracy rate is increased by 42.92% after adding the convolution module compared with the previous one. The introduction of the diffusion model can achieve intra-class enhancement of the data and avoid the overfitting problem, so the accuracy is further improved after adding the diffusion model, which verifies the correctness of the conjecture.

TABLE V. ACCURACY COMPARISON ON THE FV-SIPL DATASET (UNIT: %)

Residual	E-MHSA	Convolution module	Diffusion model	Accuracy
√				41.40
√	√			55.58
√	√	√		98.50
√	√	√	√	**99.78**

IV. CONCLUSION

Aiming at the finger vein recognition process that does not fully consider the global features of the image, is easy to overfit, and has other problems, this paper proposes a two-

branch lightweight finger vein recognition method based on the diffusion model. Firstly, the diffusion model is used to generate finger vein images to expand the finger vein dataset. Secondly, a two-branch network composed of a convolutional neural network and improved E-MHSA is used to extract global and local features from the expanded dataset. Then, the extracted global and local features are fused by the convolutional module, and the image features are further extracted. Finally, the recognition results are output, and the effectiveness of the method is verified on multiple datasets at the same time. Experiments show that the method in this paper can improve the recognition performance while keeping the computational cost small. In future work, the application of the diffusion model in finger vein recognition will be explored deeply to seek more possibilities.

ACKNOWLEDGMENT

This work has received support from the Liaoning Provincial Department of Education Fund in China (Approval No. LJKZ0349, LJKMZ20220679).

REFERENCES

[1] Yang L, Yang G P, Yin Y L, et al. Finger Vein Recognition With Anatomy Structure Analysis[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 28: 1892-1905.

[2] Lu Y, Yoon S, Wu S Q, et al. Pyramid Histogram of Double Competitive Pattern for Finger Vein Recognition[J]. IEEE Access, 2018, 6: 56445-56456.

[3] Kang W X, Lu Y T, Li D J, et al. From Noise to Feature: Exploiting Intensity Distribution as a Novel Soft Biometric Trait for Finger Vein Recognition[J]. IEEE Transactions on Information Forensics and Security, 2019, 14: 858-869.

[4] Yang L, Yang G P, Xi X M, et al. Finger Vein Code: From Indexing to Matching[J]. IEEE Transactions on Information Forensics and Security, 2019, 14: 1210-1223.

[5] Liu H Y, Yang G P, Yang L, et al. Anchor-Based Manifold Binary Pattern for Finger Vein Recognition[J]. Science China Information Sciences, 2019, 62: 1-16.

[6] Chen Y, Dai X, Chen D, et al. Mobile-former: Bridging mobilenet and transformer[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 5270-5279.

[7] Zhang Q, Yang Y B. Rest: An efficient transformer for visual recognition[J]. Advances in neural information processing systems, 2021, 34: 15475-15485.

[8] Lee S H, Lee S, Song B C. Vision transformer for small-size datasets[J]. arXiv preprint arXiv:2112.13492, 2021.

[9] Tan M, Le Q V. Mixconv: Mixed depthwise convolutional kernels[J]. arXiv preprint arXiv:1907.09595, 2019.

[10] Radzi S A, Khalih-hani M, Bakhteri R. Finger-Vein Biometric Identification Using Convolutional Neural Network[J]. Journal of Signal Processing, 2016, 24:1863-1878.

[11] Fang Y X, Wu Q X, Kang W X. A Novel Finger Vein Verification System Based on Two-Stream Convolutional Network Learning[J]. Neurocomputing, 2018, 29: 100-107.

[12] Zhang Z J, Zhong F, Kang W X. Study on Reflection-Based Imaging Finger Vein Recognition[J]. IEEE Transactions on Information Forensics and Security, 2021, 17: 2298-2310.

[13] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[14] Liu Z, Lin Y T, Cao Y, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.

[15] Peng Z L, Huang W, Gu S Z, et al. Conformer: Local Features Coupling Global Representations for Visual Recognition [C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 367-376.

[16] Huan J D, Luo W J, Yang W L, et al. FVT: Finger vein transformer for authentication[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 1-13.

[17] Yang W M, Hui C Q, Chen Z Q, et al. FV-GAN: Finger vein representation using generative adversarial networks[J]. IEEE Transactions on Information Forensics and Security, 2019, 14: 2512-2524.

[18] Choi J, Noh K J, Cho S W, et al. Modified conditional generative adversarial network-based optical blur restoration for finger-vein recognition[J]. IEEE Access, 2020, 8: 16281-16301.

[19] Hou B, Yan R. Triplet-classifier GAN for finger-vein verification[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 1-12.

[20] Ho J , Jain A , Abbeel P .Denoising diffusion probabilistic models[J]. arXiv preprint arXiv:2006.11239, 2020.

[21] Rombach R, Blattmann A., Lorenz D, et al. High-resolution image synthesis with latent diffusion models [C]//Conference on Computer Vision and Pattern Recognition (CVPR), 2021: 10674-10685.

[22] Luo C. Understanding Diffusion Models: A Unified Perspective[J]. arXiv preprint arXiv: 2208.11970, 2022.

[23] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. New York, 2018: 4510-4520.

[24] Asaari M S M, Suandi S A, Rosdi B A. Fusion of band limited phase only correlation and width centroid contour distance for finger based biometrics[J]. Expert Systems with Applications, 2014, 41(7): 3367-3382.

[25] Yin Y, Liu L, Sun X. SDUMLA-HMT: A multimodal biometric database [C]//Biometric Recognition: 6th Chinese Conference. Beijing, 2011: 260-268.

[26] Yang W, Qin C, Liao Q. A database with ROI extraction for studying fusion of finger vein and finger dorsal texture [C]//Biometric Recognition: 9th Chinese Conference. Shenyang, 2014: 266-270.

[27] Li J, Xia X, Li W, et al. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios[J]. arXiv preprint arXiv:2207.05501, 2022.

[28] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks [C]//International conference on machine learning. New York, 2019: 6105-6114.

[29] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. New York, 2016: 770-778.

[30] Zhao D, Ma H, Yang Z, et al. Finger vein recognition based on lightweight CNN combining center loss and dynamic regularization[J]. Infrared Physics & Technology, 2020, 105: 103221.

[31] Shaheed K, Mao A, Qureshi I, et al. DS-CNN: A pre-trained Xception model based on depth-wise separable convolutional neural network for finger vein recognition[J]. Expert Systems with Applications, 2022, 191: 116288.

[32] Chai T, Li J, Prasad S, et al. Shape-driven lightweight CNN for finger-vein biometrics[J]. Journal of Information Security and Applications, 2022, 67: 103211.

[33] Zhong Y, Li J, Chai T, et al. Different Dimension Issues in Deep Feature Space for Finger-Vein Recognition [C]//Chinese Conference on Biometric Recognition. Cham, 2021: 295-303.

[34] Tao Z, Wang H, Hu Y, et al. DGLFV: Deep Generalized Label Algorithm for Finger-Vein Recognition[J]. IEEE Access, 2021, PP(99):1-1.

[35] Shen J, Liu N, Xu C, et al. Finger vein recognition algorithm based on lightweight deep convolutional neural network[J]. IEEE Transactions on Instrumentation and Measurement, 2021, 71: 1-13.

[36] Muthusamy D, Rakkimuthu P. Trilateral Filterative Hermitian feature transformed deep perceptive fuzzy neural network for finger vein verification[J]. Expert Syst. Appl. 2022, 196: 116678.

[37] Zhao D , Ma H , Yang Z , et al. Finger vein recognition based on lightweight CNN combining center loss and dynamic regularization[J]. Infrared Physics & Technology, 2020, 105(8):103221.

[38] Ren H, Sun L, Guo J, et al. Finger vein recognition system with template protection based on convolutional neural network[J]. Knowl. Based Syst. 2021, 227, 107159.