applied sciences

MDPI

Article

ViT-Cap: A Novel Vision Transformer-Based Capsule Network Model for Finger Vein Recognition

Yupeng Li, Huimin Lu *, Yifan Wang, Ruoran Gao and Chengcheng Zhao

School of Computer Science and Engineering, Changchun University of Technology, Changchun 130102, China
* Correspondence: luhuimin@ccut.edu.cn

Abstract: Finger vein recognition has been widely studied due to its advantages, such as high security, convenience, and living body recognition. At present, the performance of the most advanced finger vein recognition methods largely depends on the quality of finger vein images. However, when collecting finger vein images, due to the possible deviation of finger position, ambient lighting and other factors, the quality of the captured images is often relatively low, which directly affects the performance of finger vein recognition. In this study, we proposed a new model for finger vein recognition that combined the vision transformer architecture with the capsule network (ViT-Cap). The model can explore finger vein image information based on global and local attention and selectively focus on the important finger vein feature information. First, we split-finger vein images into patches and then linearly embedded each of the patches. Second, the resulting vector sequence was fed into a transformer encoder to extract the finger vein features. Third, the feature vectors generated by the vision transformer module were fed into the capsule module for further training. We tested the proposed method on four publicly available finger vein databases. Experimental results showed that the average recognition accuracy of the algorithm based on the proposed model was above 96%, which was better than the original vision transformer, capsule network, and other advanced finger vein recognition algorithms. Moreover, the equal error rate (EER) of our model achieved state-of-the-art performance, especially reaching less than 0.3% under the test of FV-USM datasets which proved the effectiveness and reliability of the proposed model in finger vein recognition.

Keywords: finger vein; biometrics; computer vision; deep learning

Citation: Li, Y.; Lu, H.; Wang, Y.; Gao, R.; Zhao, C. ViT-Cap: A Novel Vision Transformer-Based Capsule Network Model for Finger Vein Recognition. *Appl. Sci.* **2022**, *12*, 10364. https://doi.org/10.3390/app122010364

Academic Editor: Yu-Dong Zhang

Received: 8 September 2022
Accepted: 11 October 2022
Published: 14 October 2022

1. Introduction

The Internet era makes a face [1], iris [2], fingerprint [3], and other biometric features of people's digital identification. Biometrics can automatically detect, capture, process, analyze, and identify these digital physiological or behavioral signals, which is a typical and complex pattern recognition problem and has been at the forefront of the development of artificial intelligence technology. Finger vein recognition is a biometric technique that uses human finger vein images to identify individuals. Finger vein recognition refers to the use of a charge-coupled device (CCD) camera to obtain an individual's finger vein distribution map by irradiating fingers with near-infrared light and then using advanced filtering image binarization and other means to extract digital image features by comparing them with the finger vein feature values stored in the host and using a complex matching algorithm to match the characteristics of the finger veins, so as to realize personal identification. Compared with other biometrics, finger vein recognition has become the second generation of biometrics because of its advantages in high security, precision, stability, and ease of use.

Early finger vein recognition is mainly based on feature engineering methods, which extract distinguishable features from pre-processed finger vein images, such as local texture features [4], vein pattern features [5], minutiae features [6], etc., by measuring the similarity between the image features to be compared and the extracted features to achieve recognition. However, feature extraction of finger veins is greatly affected by ambient temperature [7].

When the finger is cold, the veins of the finger shrink and thin, making vein information scarce. Secondly, the surrounding strong ambient light (such as sunlight) will cause different degrees of interference with the near-infrared image, affecting the authentication rate of vein recognition. At the same time, finger vein acquisition devices have certain requirements for the acquisition posture, so the acquisition posture will have a certain impact on the image quality. These defects affect the effect of traditional feature extraction algorithms and have a negative impact on the performance of finger vein recognition. With the rapid development of deep learning, self-learning features based on the deep framework have made great progress in image recognition in recent years. Compared with traditional algorithms, the goal of deep learning is to learn features. The problem of manually extracting feature points was solved by obtaining the feature information of each layer through the network. Based on the learning of massive data and under the constraints of deep framework theory, we adjusted the parameters of the multi-layer network, established the optimal nonlinear fitting network between input and output nodes, and then compared the target samples with the samples mapped by the deep network, marking the correspondence between them to get as close as possible to the real distribution. By training the deep network model, the maximum probability distribution of the target classification was obtained. Finger vein recognition [8] is a recognition technology for complete image classification. Many methods of finger vein recognition based on deep learning have been proposed in recent years and achieved satisfactory results. Das et al. [9] proposed a CNN-based finger vein recognition model and tested its effectiveness on four public finger vein image datasets. Wang et al. [10] proposed an HGAN-based data expansion strategy for the CNN finger vein recognition model and compressed the model using filtered pruning and low-rank decomposition. Lu [11] et al. proposed a CNN-based local descriptor named CNN-Competitive Order (CNN-CO) for the finger vein recognition model of the Deep Convolutional Neural Network (DCNN). However, CNN did not consider the spatial relationship between potential target features and performed poorly in exploring the spatial relationship between features. Moreover, the pooling layer of CNN loses a lot of valuable information, which makes the result of finger vein recognition unable to achieve a great improvement. Hinton et al. [12] presented the capsule network, which defined features in a more reasonable way than CNN by ensuring the invariance of translation and rotation. Dilara Gumusbas et al. [13] used capsule networks for recognition using a limited number of samples in four finger vein datasets. Although the capsule network overcomes some of the drawbacks of CNNs, the model cannot selectively focus on the important information in the image, resulting in a much smaller receptive field during actual processing than the theoretical receptive field. In fact, after we detected the key points, object boundaries, and other basic units that made up visual elements, high-level visual semantic information tended to focus more on how these elements related to each other to form a target object and how the spatial position relationships between these target objects constituted a scene. However, models such as the capsule network do not achieve the desired effect when dealing with the relationships between these elements. Dosovitskiy [14] put forward a transformer model applied to computer vision, which achieved good results on multiple image recognition benchmarks. Unlike CNN models, vision transformer uses a self-attention mechanism to integrate information across the entire image. Even at the lowest level, the vision transformer captures global contextual information by using self-attention to establish remote dependencies on targets and extract more powerful features.

To solve the problem that the capsule network lacks the ability to encode the long-range dependencies in the image and cannot selectively pay attention to important image feature information when the capsule network is used for image classification, we integrated the advantages of the capsule network in processing the underlying vision information, as well as the advantages of the transformer in processing the relationship between the visual elements and the target objects, and propose a new vision transformer-based capsule network model (ViT-Cap) for finger vein recognition. The model can encode the dependen-

cies between image features so as to improve the effect of image classification, especially multi-label classification. Experimental results showed that the proposed model has better recognition performance than the existing recognition methods.

The main contributions of this article are as follows. A new vision transformer model based on a capsule network is proposed for finger vein recognition.

(1) The model can encode long-range dependencies in images and can better understand the relationship between advanced semantic visual information and basic visual elements in finger vein images.
(2) Based on the vision transformer, we introduced the local information shared by the capsule network module and constructed a finger vein recognition model based on global attention and local attention, which better obtained the features of finger veins and maximized the performance of the model.
(3) Compared with CNNs, the model we proposed has higher logical interpretability, and we visualized part of the training process of the model. At the same time, due to the dynamic routing mechanism in the model, better performance was obtained when processing small-scale image data, which solved the limitation of a small amount of finger vein data.

The remainder of this article is organized as follows: Section 2 briefly introduces the relevant work of finger vein recognition methods; Section 3 describes the motivation behind our proposed new model and the main methods based on our proposed model; Section 4 gives the detailed experimental results and analysis; and finally, we summarize the work and look forward to the future work in Section 5.

2. Related Work

Digital vein recognition technology was first invented by Joseph Rice [15] in 1983 and named Veincheck, which is the prototype of modern vein recognition technology. Japanese researchers Kono et al. [16] first proposed the use of finger vein features for identity authentication in 2000. Like other biometrics, finger vein recognition processes mainly include four stages: image acquisition, image pre-processing, feature extraction, and feature matching. So far, many researchers have proposed many interesting methods of finger vein feature extraction.

Generally speaking, the recognition methods of finger veins are divided into the following two categories: The first type of finger vein recognition method is based on the traditional manual feature extraction. Miura et al. [17] proposed a classical linear tracking algorithm to extract vein pattern features based on a local grayscale difference of finger vein images. Qin et al. [18] presented a method of finger vein pattern extraction based on region growth, and the extraction effect was better. To further improve the effectiveness of feature extraction, Miura et al. [19] also proposed the maximum curvature algorithm for finger vein feature extraction by finding the maximum curvature of a local cross-section of the image, which has become a presentative algorithm in the field of finger vein feature extraction. However, low-quality finger vein images may reduce recognition performance, and Gupta et al. [20] proposed the use of local multi-scale matching filters to alleviate the noise generated by uneven illumination in digital vein images. At the same time, global features are used to enhance finger vein images to obtain better recognition performance. Rosdi et al. [21] proposed a local linear binary pattern feature based on the local feature of finger veins to extract the coding feature of the linear local region and obtained good experimental results. Van et al. [22] presented a new extraction method for digital vein features, which focused on local invariant directional feature extraction. The extracted features are further processed by GridPCA to remove redundant data, which can obtain a more accurate recognition effect of finger veins.

The second type of finger vein recognition method is based on the feature extraction methods of deep learning. The goal of deep learning is to learn features and obtain the feature information of each layer through the network, thus solving the problem of manually extracting feature points. Das et al. [9] proposed a CNN-based finger vein

recognition model and tested its effectiveness on four public finger vein image datasets. Hong et al. [23] proposed a CNN-based finger vein recognition model that is robust to image shading and misalignment. This model was tested on three public finger vein databases and achieved good recognition results. Zeng et al. [24] presented a finger vein verification algorithm based on fully CNN and conditional random field and tested the proposed model on three public finger vein datasets, and the experimental results verified its superior performance in the finger vein verification task. Wang et al. [25] proposed a Multi-Receptive Field Bilinear Convolutional Neural Network (MRFBCNN) to obtain second-order features of finger veins and better distinguish finger veins, with little difference between classifications.

3. Materials and Methods

In this section, we will first introduce the motivation of the finger vein recognition model we proposed, and then we will detailly describe the vision transformer model and capsule network module, and then focus on the ViT-Cap model we designed and developed to better achieve the finger vein recognition task.

3.1. Motivation

At present, CNNs are the dominant deep neural network architectures in computer vision. From image classification [26], object detection [27], and image segmentation [28] to action recognition [29], various types of CNNs have proved their effectiveness in these computer vision tasks and have also shown good performance in the field of finger vein recognition. Although CNN offers translation invariance, the translation invariance it provides through pooling technology is limited, which is also the reason why CNN needs much data rotated from different viewpoints.

3.1.1. Overview of Vision Transformer

Recently, the amazing effects of transformer models in natural language tasks have attracted the attention of computer vision. Researchers have proposed many transformer-based vision models, such as ViT [14], DETR [30], DeiT [31], GLiT [32], etc. Vision transformer is a modified NLP transformer for image classification without any convolutional layers. The image is split into patches, which are then flattened and put into a lower-dimensional embedding space. An extra token is added to each vector to denote its relative location in the image, and another learnable token is added to the entire sequence of vectors to denote the class. The sequence of vectors is fed to a standard transformer encoder, which has been modified with an extra fully connected layer at the end for classification.

For vision transformers, the attention mechanism provides key advantages that CNNs do not have. The transformer can capture the long-range relationship and has dynamic adaptive modelling capability. Moreover, the transformer's self-attention mechanism has a built-in attention map, which provides a new way for models to make decisions. CNNs are not highly interpretable and can only provide a rough visualization. Transformer tokens provide more detailed attention images than CNN's limited receptive field, and the self-attention mechanism clearly simulates the interactions between each area in the image. However, vision transformers need a very large dataset to surpass CNNs. The performance of ViT can only achieve the performance of SOTA when it is trained on Google's private image dataset JFT-300M. This problem is particularly serious in the field of finger vein recognition due to the small data set of finger veins. As with the ViT model, CNNs perform worse when data is scarce.

3.1.2. Overview of Capsule Network

Hinton et al. [12] proposed a capsule network in view of the shortcomings of CNN models. A capsule is a set of neurons whose activity vectors represent the instantiation parameters of a specific type of entity, such as an object or object part. It uses the length of the activity vector to represent the probability of entity existence and its orientation to

represent the instantiation parameters. Multiple capsules form a hidden layer, and the relationship between the two hidden layers is determined by a dynamic routing algorithm.

There are two main advantages of a capsule network. First, the capsule network model requires less data for training, and the capsule network can promote what it has learned to new scenarios. Second, the capsule network is not interfered with by the overlapping of multi-categories, and it can handle complex scenes of overlapping targets. Through the dynamic routing mechanism, the capsule network can realize the identification and prediction of overlapping targets. However, the capsule network has its own limitations. It lacks the ability to encode long-range dependencies in images and cannot selectively focus on important image feature information. When dealing with the relationship between high-level visual semantic information and basic visual elements in the image, it does not achieve the desired effect. Although the capsule network can almost reach the best performance when processing small-scale image data, it still needs to be improved in large-scale image processing.

3.1.3. Overview of ViT-Cap Module

Motivated by these factors, this study aimed to propose a vision transformer-based capsule network model for finger vein recognition. A capsule network can be regarded as modelling shared local information. Compared with CNN and traditional neural networks, the capsule network is more adaptable to bad data and can adapt to affine transformation for input data.

We introduced transformer-based modelling of shared local information and then explored the network structure of the vision transformer based on global and local attention. Our model combined the advantages of capsule networks in processing the underlying vision with the advantages of transformers in processing the relationships between visual elements and objects. Our proposed model solved the problem of the lack of long-range dependence on the encoding image and its inability to selectively focus on important image feature information when the capsule network is used for image classification, thereby achieving better finger vein classification results. In addition, the capsule network can better explore the relationship between features. In the case of a small amount of data, a capsule network has better generalization abilities than CNNs and can resist overfitting better. The model we proposed solves the limitation of a small amount of finger vein data and maximizes the performance of the model.

The complete architecture of our proposed model is shown in Figure 1. We used the encoder part of the transformer architecture combined with our capsule networks and their dynamic routing to implement finger vein recognition. The model we proposed mainly consisted of three parts. The first part was to process the input finger vein image and linearly embed it into the transformer encoder, the second part was to build the transformer module to process the embedded finger vein image, and the third part was to deploy the capsule network module. The data processed by the transformer is inputted into the capsule network module for further training, and finally, the model results are outputted. A detailed description of the model architecture is expanded in Sections 3.2–3.4.

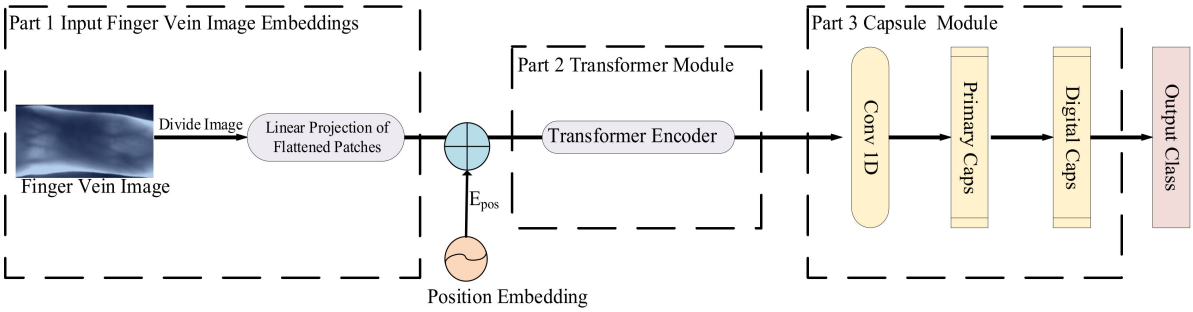

Figure 1. The overall architecture of the ViT-Cap model.

3.2. Linear Embedding of Finger Vein Images

Transformers were originally applied in the field of natural language processing, where the standard transformer receives a one-dimensional sequence of token embeddings as input [20]. Therefore, when we used a transformer to process 2D finger vein images, to fit the transformer architecture, we split the finger vein images into patches, linearly embedded each patch, and fed the resulting sequence of vectors into the transformer encoder [14,30,31].

As shown in Figure 2, first, we needed to chunk the input finger vein image, assuming that the image information parameter of the input image x is (H, W, C), where (H, W) is the resolution of the finger vein image and C is the number of channels. Assuming the patch size was (P, P), then $N = HW/P^2$ image blocks were divided for the vein image with input parameters (H, W), and we chose the patch size P, which was 16×16 or 28×28. If the patch size was too large, the parameter scale of the model exponentially increased, and conversely, if the size of the patch was too small, the final performance of the model was affected. Therefore, in the process of finger vein images in our proposed model, the patch size was selected as 28×28. Then, each block was regarded as a vector, and all the vectors were combined into a sequence, resulting in a dimension of $\left(N, P^2C\right)$ for the data.

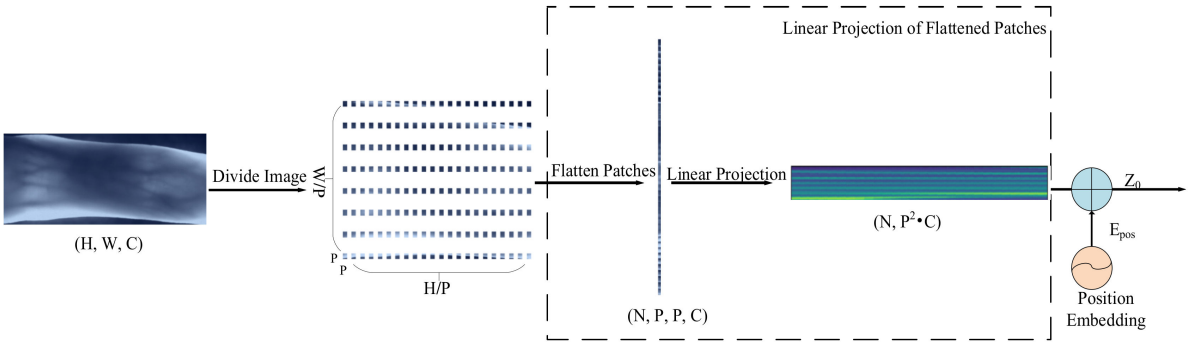

Figure 2. The first part of ViT-Cap model architecture: linear embedding of finger vein images.

Transformers use a constant vector size D in all layers, so we flattened all the patches and mapped them to D dimensions through a trainable linear projection. The final embedded sequence of the patch, with the token Z_0, is given in Equation (1), where x_{class} is a learnable classification token that is needed to perform the classification task. The patch embeddings are the output of this projection.

$$Z_0 = \left[x_{class}; x_p^1 E; x_p^2 E; \cdots; x_p^N E\right] + E_{pos}, \qquad E \in R^{(P^2 \cdot C) \times D}, E_{pos} \in R^{(N+1) \times D} \tag{1}$$

3.3. Transformer-Based Module

The resulting sequence of embedded patches Z_0 was passed to the transformer encoder. Figure 3 shows the second part of our model. The encoder consisted of L identical layers, which were composed of alternating layers of the multi-head self-attention block (MHSA, Equation (2)) and a fully connected, feed-forward dense block. Each layer was a feed-forward network after the attention, and the function of the feed-forward dense block was spatial conversion. Feed-forward networks introduce a nonlinear ReLU activation function, which transforms the space of attention output, thereby improving the performance of the model. Layer norms were applied before every block, and the residual connections were applied after every block.

$$Z_l = MHSA(LN(Z_{L-1})) + Z_{L-1}, \quad l = 1, \ldots, L \tag{2}$$

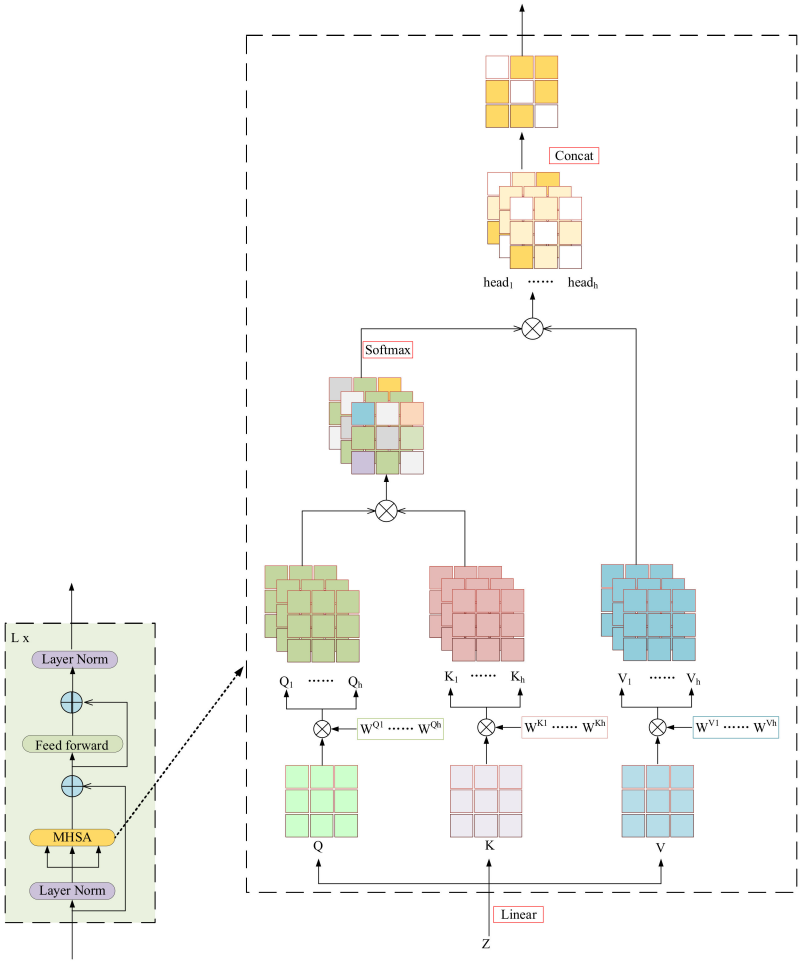

Figure 3. The second part of the ViT-Cap model architecture: transformer encoder module.

The MHSA was the core component of the transformer encoder, and its role was to determine the relative importance of a single patch embedding relative to other embeddings in the sequence. MHSA consists of four layers: a linear layer, a self-attention layer, a concatenation layer, and a final linear layer. The operation of the MHSA is as follows.

Firstly, we inputted the linear transformation of image matrix Z and divided it into three matrices Q, K, and V, which were obtained by three different weight transformation matrices, W^Q, W^K, and W^V (Equation (3)).

$$[Q, K, V] = \left[ZW^Q, ZW^K, ZW^V\right] \tag{3}$$

Then the matrices Q, K, and V, obtained above, were linearly projected into h different subspaces, h equaled 12, and the self-attention values in each subspace were calculated (Equations (4)–(7)).

$$[Q_1, \cdots, Q_h] = \left[QW^{Q_1}, \cdots, QW^{Q_h}\right] \tag{4}$$

$$[K_1, \cdots, K_h] = \left[KW^{K_1}, \cdots, KW^{K_h}\right] \tag{5}$$

$$[V_1, \cdots, V_h] = \left[VW^{V_1}, \cdots, VW^{V_h}\right] \tag{6}$$

$$Head_i = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d}}\right) V_i \tag{7}$$

Finally, we concatenated all the attention heads and then projected them through a feed-forward layer with a learnable weight W (Equation (8)).

$$MultiHead(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = Concat(Head_1, \ldots, Head_h)\mathrm{W}^O \quad (8)$$

In the process of finger vein recognition, attention is an effective non-local information fusion technology. Attention essentially means the average weighting according to the matrix of a relation, which means different information can be fused. CNN itself has a defect, that is that each of its operations can only focus on the information near the convolution kernel and cannot fuse distant information. However, attention can realize a weighted fusion of distant information, which plays an auxiliary role. On the other hand, attention has a higher logical interpretability than CNN, and the weighted analysis of attention naturally has the property of visualization.

As shown in Figure 4, we trained the model on four public datasets of finger veins and visualized the attention maps obtained from the four datasets. The more important the area in Figure 4, the brighter the pixels. Figure 4 shows the process of obtaining finger vein information by the multi-head attention mechanism. Through this module, the important feature information of finger vein images was gradually obtained. For example, in FV-USM, the attention map head-1 only showed partial attention to the vein feature information, but the corresponding attention was also given to irrelevant information outside the veins. This problem was gradually improved in subsequent attention map heads. By the time of attention head-12, we saw that less attention was paid to irrelevant regions, and more attention was paid to the vein feature information in the image. For MMCBNU, attention head-1 mainly focused on irrelevant information, and then the model attempted to capture the discriminative areas that corresponded to the finger vein. For SDUMLA and HKPU, as attention heads increased, the model tended to pay more attention to the region of finger veins. After attention head-6, we saw that less attention was paid to irrelevant regions. By analyzing the visualized finger vein results, the attention mechanism learned very strong structural features.

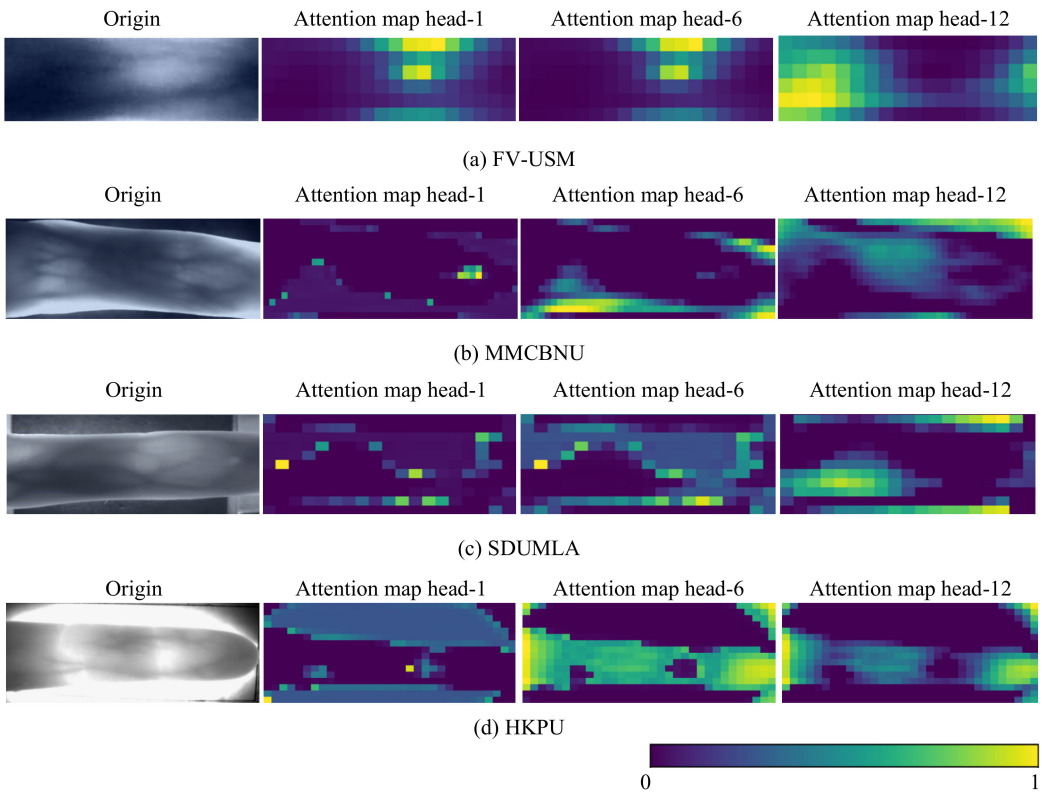

Figure 4. Attention maps obtained on four public finger vein datasets.

3.4. Capsule Network-Based Module

We fed the vector X generated by the transformer module into the capsule module, and the architecture of the third part of the proposed model is shown in Figure 5. First, we used a convolutional layer to process the feeding vector X. Conv1 had a 9×9 convolutional kernel with a stride of 1 and an activation function of ReLU. This layer converted pixel intensity into activities of the local feature detector, which were then used as input for PrimaryCaps.

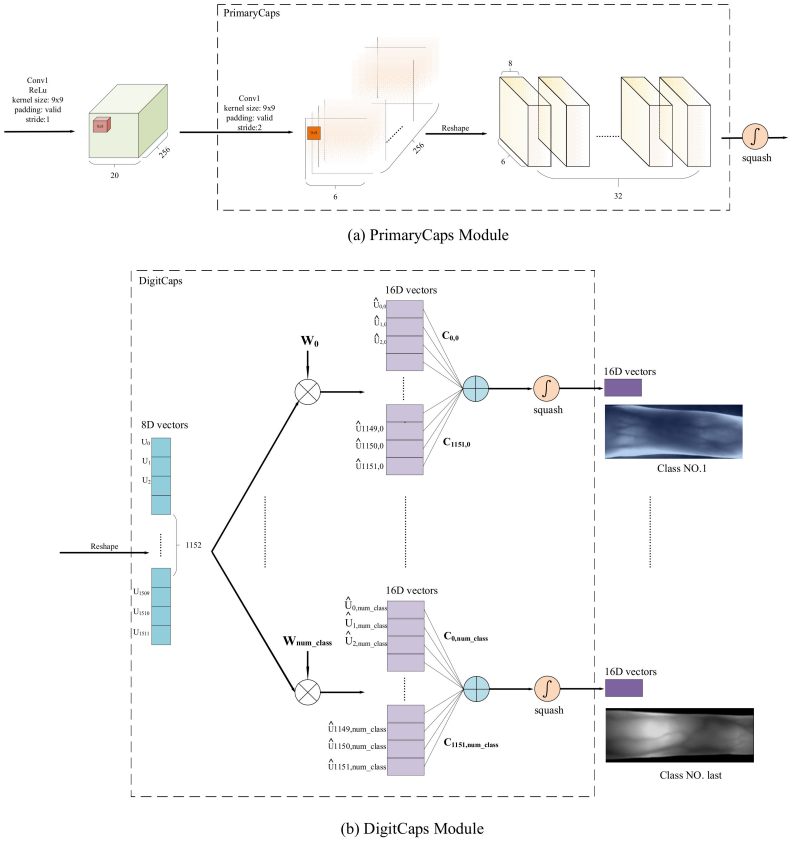

Figure 5. The third part of ViT-Cap model architecture.

The PrimaryCaps module was a convolutional capsule layer containing 32 channels of convolutional 8D capsules. Each primary capsule contained 8 convolutional units, a 9×9 kernel, and a stride of 2. PrimaryCaps had $[32 \times 6 \times 6]$ capsule output. Each layer of the DigitCaps module had a 16D capsule, and each capsule received input from all capsules in the layer below.

We used dynamic routing between PrimaryCaps and DigitCaps, as shown in Figure 5b. For the capsules, the input u_i and output v_j of the capsule were vectors. We applied a transformation matrix W_{ij} to the capsule output u_i of the previous layer, and then calculate the weighted sum s_j of the weight c_{ij}. c_{ij} was the coupling coefficient calculated by the iterative dynamic routing process.

$$\hat{u}_{j|i} = W_{ij} u_i \tag{9}$$

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{10}$$

We applied a squashing function (Equation (11)) to ensure that short vectors shrank to almost zero in length and long vectors shrank to almost one length.

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||} \tag{11}$$

Prediction vector $\hat{u}_{j|i}$ was the prediction of the output of capsule j from capsule i. If the activity vector was very similar to the prediction vector, we concluded that both capsules were highly correlated. This similarity was measured using the predicted scalar product and the activity vector.

$$b_{ij} \leftarrow \hat{u}_{j|i} \cdot v_j \tag{12}$$

The coupling coefficient c_{ij} was calculated as the Softmax of b_{ij}.

$$c_{ij} = \frac{expb_{ij}}{\sum_k expb_{ik}} \tag{13}$$

To make b_{ij} more accurate, it was iteratively updated in 3 or 4 iterations in finger vein recognition experiments.

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i \cdot v_j} \tag{14}$$

Finally, the model-encoded finger vein image was fed into a dense layer with a Softmax activation function that mapped the embedded vector to the desired class in one-hot format.

4. Results

4.1. Finger Vein Datasets Description

In this section, we first discuss the details of the datasets used, followed by the setup of the experiments. The effectiveness of our proposed ViT-Cap model was evaluated on four publicly available finger vein datasets: MMCBNU from Jeonbuk National University in South Korea [33], SDUMLA from Shandong University in China [34], FV-USM from University Sains Malaysia [35], HKPU from the Hong Kong Polytechnic University [7], and MIXFV. The primary reason for using these specific datasets was that most existing finger vein recognition methods are evaluated by tests performed on one or more of these datasets. The details of these four public finger vein databases are shown in Table 1.

Table 1. Details of public finger vein datasets.

Database	Sampling Object	Image Resolution	Number of Classes	Number of Fingers	Total Images
MMCBNU	100	320 × 240	600	6	6000
SDUMLA	106	320 × 240	636	6	3816
FV-USM	123	640 × 480	492	4	5904
HKPU	156	513 × 256	624	4	3132

Information from the MMCBNU database was collected from 100 volunteers, including 83 men and 17 women. Each volunteer provided the index, ring, and middle fingers of both hands, and the collection was repeated 10 times per 6 fingers, for a total of 60 finger vein images from each volunteer.

The SDUMLA database was created by Shandong University in China, with a dataset of finger veins from 106 volunteers, each of whom provided the index, middle, and ring fingers of both hands, so a total of 3816 images were collected with a resolution of 320 × 240 pixels.

The FV-USM database was collected from 123 volunteers, including 83 men and 40 women, ranging in age from 20 to 52. Each volunteer provided four fingers: left index finger, left middle finger, right index finger, and right middle finger, resulting in a total of 492 finger classes, and the finger image resolution captured was 640 × 480.

The Hong Kong Polytechnic University dataset consisted of finger vein images collected from men and women volunteers. About 93% of the participants were under 30 years old. Finger images were obtained over two separate time periods, with a minimum interval of one month and a maximum interval of more than six months. In each session, each volunteer provided six image samples from the index and middle fingers, and each sample consisted of a finger vein and finger texture image from the left hand.

The MIXFV contains 100 categories selected from the top 100 categories of SDUMLA, FV-USM, MMCBNU and HKPU. The MXIFV database is divided as follows: the training set has 27 training samples for each category from four different public finger vein databases, and the test set has 13 test samples for each category, also from four different public finger vein databases.

4.2. Finger Vein Image Pre-Processing

In the process of finger vein image acquisition, accurate identification of finger veins is a very difficult task due to the influence of uneven illumination, equipment noise, and other factors. Therefore, before finger vein recognition, it is necessary to pre-process the finger vein datasets to improve the recognition performance. To test the performance of our proposed model to the maximum extent, we only performed region of interest (ROI) localization processing on the finger vein image to maximize the retention of the vein feature information in the original image.

For the MMCBNU and FV-USM datasets, we used the ROI images included in the datasets for model training, which ensured the reliability of the vein image information. For HKPU and SDUMLA datasets, we adopted the ROI extraction method proposed by Lu et al. [36]. We retained the finger vein area in the image to the greatest extent and conducted ROI localization along the outer tangent direction of the finger. We only removed irrelevant background information in the image, but not the part with unclear vein lines in the original image. Although there were still some problems, such as unclear partial vein texture information and missing vein texture information caused by severe exposure in some images after ROI extraction, the vein image after ROI localization retained the complete vein information of the original image and such data could better show the robustness of the model we proposed. Figure 6 shows the comparison of ROI-localized finger vein images with the original finger vein images from four public datasets.

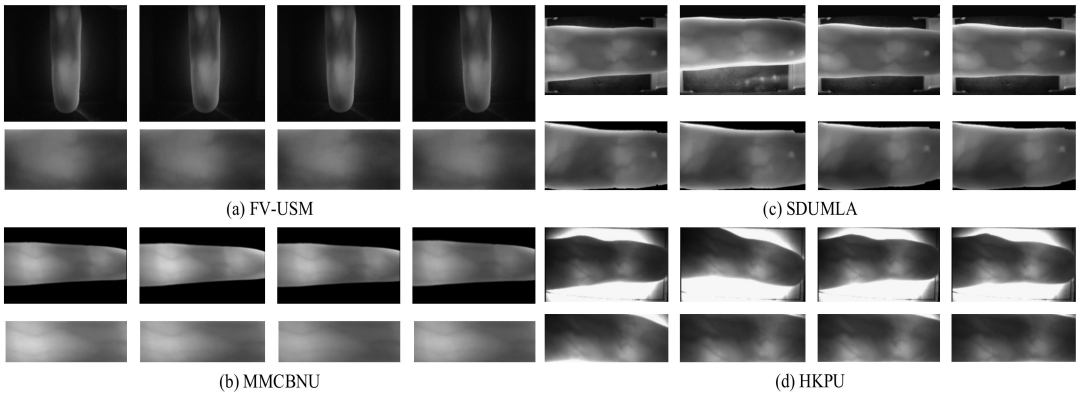

Figure 6. Comparison of finger vein images before and after ROI in four public datasets.

4.3. Experimental Setup for Finger-Vein Recognition

We conducted three sets of experiments on four public datasets. In the first set of experiments, we changed the number of multi-heads and studied the relationship between the attention mechanism and model performance. Then we changed the number of encoder layers and the number of routing iterations and studied the relationship between network depth and model performance. In addition, equal error rate (EER) and recognition accuracy (ACC) were used to evaluate the performance of the finger vein recognition model. The EER is the error rate when the false acceptance rate (FAR) equals the false rejection rate (FRR). The FRR is the probability that the correct sample is wrongly rejected by the system, while FAR is the probability that the wrong sample is considered correct by the system.

The recognition accuracy was selected to evaluate the recognition performance of the model. By feeding the test set into the training model and outputting it through Softmax, the maximum probability was selected as the category of the current image. By comparing

with the labels, the proportion of the number of samples with the same prediction category and categories in the total number of samples was calculated, as shown in Equation (15) (where N_T represents the number of correctly classified samples and N represents the total number of samples).

$$Accuracy = \frac{N_T}{N} \tag{15}$$

We clearly described the calculation method of the EER. For multi-classification methods, such as ours, we regarded the prediction of each class as a binary classification task. For instance, we fed an image of a finger vein, which belonged to class three, into the model, which outputted a list of predictions. We assumed that there were 100 predictions: [0.0031, 0.5362, 0.9966, 0.0042, . . . , 0.0033]. The threshold (from 0 to 1) was set to 0.5. If the prediction probability was greater than 0.5, the corresponding sample was marked as positive (P); otherwise, it was marked as negative (N). If the predicted label of the corresponding sample was the same as its true label, we marked it as a true (T) sample; otherwise, it was a false (F) sample. For the value 0.9966, the true label of its corresponding sample was three, which was equal to the predicted result, so it was a TP sample. Similarly, 0.0031 was a TN sample, and 0.5362 was an FP sample. Then, the FAR and FRR were calculated (as shown in Equations (16) and (17)). Multiple sets of FAR and FRR were obtained by setting a series of thresholds. We connected these points to form a receiver operating characteristic curve (ROC), and its intersection with the diagonal was EER, while AUC was the area under the ROC curve.

$$FAR = \frac{FP}{FP + TN} \tag{16}$$

$$FRR = \frac{FN}{FN + TP} \tag{17}$$

In the second set of experiments, we compared the results of our model with several state-of-the-art methods, including ACC and EER values. Finally, we tested the model's performance on datasets of small categories, selecting 50, 100, 150, and 200 categories on four public finger vein datasets for experimentation. We also used fewer images for training and compared the results. The main purpose of the third set of experiments was to test the performance of our proposed model in processing small categories and small-scale image data.

In all experiments, the basic framework of our proposed ViT-Cap model did not change. What we changed during the experiment were the layers of the encoder, the number of multi-heads, and the number of routing iterations. The embedding dimension of the ViT-Cap model was 784, and the number of routing iterations used in the experiment was three or four. To improve the experimental effect, we adjusted the ROI pre-processed image size to 224 $\times$ 224 and the patch size to 28 $\times$ 28. During model training, we used PyTorch (a Python-based deep learning framework) to resize the input vein images to 224 $\times$ 224.

For comparison, we evaluated the model's performance based on standard population accuracy, which was the number of correctly classified finger vein images divided by the total number of finger vein images. We also used the EER value and AUC value obtained by the experiment as important indicators to evaluate the performance of the model. To analyze the performance of the proposed model, we compared it with state-of-the-art finger vein recognition models. The results of the ViT-Cap model on the four datasets mentioned above showed competitive or even better performance.

All the algorithms in this article were implemented and run using Python 3.8 on a server with an Intel(R) Silver 4214 CPU, 256 GB of RAM, and a Tesla T4 graphics card, and we used PyTorch 1.7.1, an open-source deep neural network library written in Python.

4.4. Experiment 1: Preliminary Analysis

We tested the models on four publicly available finger vein datasets, each of which was first divided into training and test sets. For the FV-USM dataset, we set the percentage of

training data to test data at 8-4; for the SDUMLA dataset, we set the percentage of training data to test data at 5-1; for the MMCBNU dataset, the percentage of training data to test data was 6-4; for the HKPU dataset, the percentage of training data to test data was 8-4.

Under the condition that the overall framework of the model remains unchanged, we used the method of controlling variables to set the number of encoder layers in the transformer module to one and two, the number of heads to 12 and 24, and the number of routing iterations in the capsule module to three and four, respectively. The number of training rounds of the model was unified to 300 epochs, and the accuracy, EER, and AUC values of the model under different parameters were obtained through experiments to evaluate the performance of the model. Table 2 shows the experimental results obtained under different parameters.

Table 2. Performance comparison of ViT-Cap on four datasets and under different parameter settings.

Database	Number of Layers	Heads	Routing Iterations	ACC (%)	EER (%)	AUC
MMCBNU	1	12	3	96.54	0.66	1.0
		24		97.52	0.63	1.0
	2	12	4	95.29	1.12	1.0
		24		97.25	0.65	1
SDUMLA	1	12	3	91.98	2.15	0.99
		24		94.97	1.13	1
	2	12	4	92.45	2.38	0.99
		24		93.24	1.74	0.99
FV-USM	1	12	3	97.76	1.01	1.0
		24		98.68	0.29	1.0
	2	12	4	97.82	0.28	1.0
		24		97.92	0.60	1.0
HKPU	1	12	3	92.62	3.53	0.99
		24		93.45	2.93	0.99
	2	12	4	95.36	2.56	0.99
		24		95.61	1.66	0.99

The results in Table 2 clearly show that the performance of our proposed model was very good. On the four publicly available finger vein datasets, the average recognition accuracy of ViT-Cap reached 95.53%. On the FV-USM dataset, the ViT-Cap model achieved a minimum EER value of 0.28% and an AUC value of almost one. Through experimental analysis, when only the number of heads was changed, the results of the experiment using 24 heads were significantly better than that using only 12 heads.

After completing the experiments of Table 2, we conducted an extreme experiment at the same time. We set the encoder layers, heads, and routing iterations to 8-24-6, respectively. The experimental results in Table 3 further show that when the performance of the model reached a certain level, changing the depth of the encoder layer and the number of routing iterations did not improve the recognition performance.

For most deep learning models, increasing the depth of the model greatly improved the performance of the model. However, the framework of our ViT-Cap model determined that increasing the depth of the model did not greatly improve the overall performance of the model. We extracted the local feature information of the finger vein image through the encoder layer (layers of 1–2), which obtained the basic vein feature information, and then use the capsule network module to perform a deeper global vein feature information extraction of the finger vein features obtained by the encoder layer. The encoder layer and capsule network module complemented each other, enabling our model to encode dependencies between vein image features, thereby improving finger vein recognition performance. The setting of the number of routing iterations was essentially a hyperparameter. It was most reasonable to set routing iteration to three and four. If the parameters of routing iteration of the model were set too large, it greatly increased the training cost of the model.

Table 3. Performance comparison of ViT-Cap with deep layers and routing iterations on four datasets.

Database	Number of Layers	Heads	Routing Iterations	ACC (%)	EER (%)	AUC
MMCBNU	12	24	6	96.08	0.72	1
SDUMLA	12	24	6	92.14	1.63	1
FV-USM	12	24	6	98.35	0.55	1
HKPU	12	24	6	95.48	1.71	0.99

Figure 7 shows the ROC curve obtained by the experiment. For the MMCBNU dataset, continuing to increase the depth of the network model degraded the performance of the model as the depth of the network model increased to a certain level. As can be seen from Figure 7a, increasing the number of layers of the encoder significantly reduced the performance of the network model without changing the number of heads. On the contrary, if we increased the number of heads, the performance of the model was further improved. Since the multi-head attention mechanism acquired long-range contextual information in the image without considering distance, it was possible to obtain the relationship between complex high-level semantic information in finger vein images thereby improving the accuracy of recognition. The ROC curve obtained after training the model with FV-USM dataset is shown in Figure 7b. The optimal model configuration was an encoder layer with 24 heads, and the number of routing iterations is set to 3. Furthermore, when the experimental results reached the upper limit of the model performance, continuing to change the configuration of the model did not improve the experimental results.

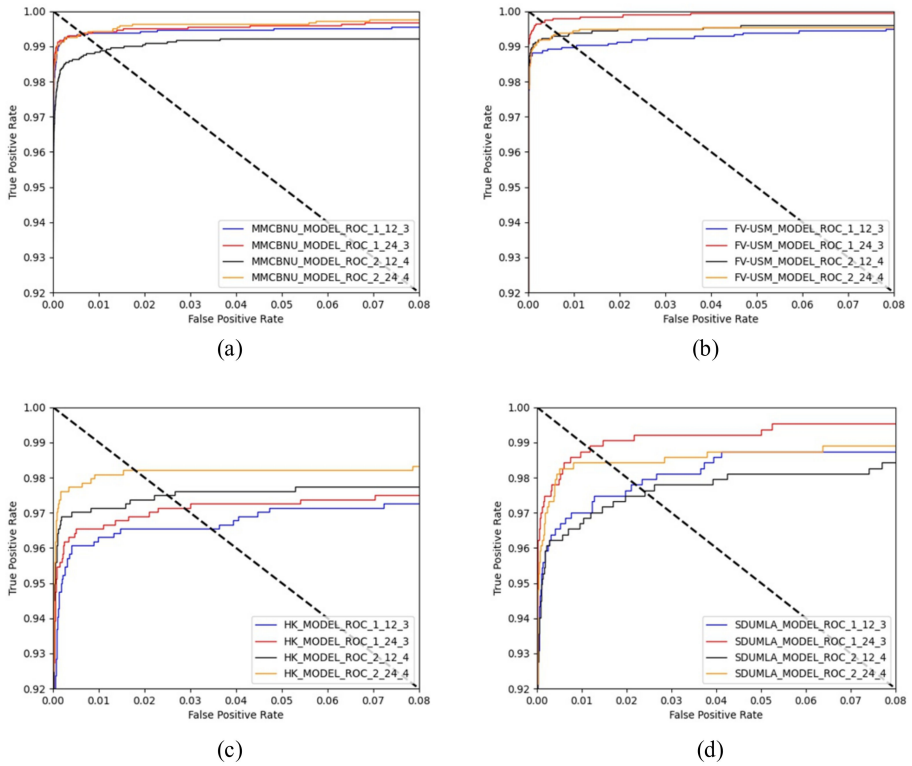

Figure 7. ROC curves of the ViT-Cap model on four public finger vein datasets:(**a**) MMCBNU (**b**) FV-USM (**c**) HKPU (**d**) SDUMLA.

We designed two sets of ablation experiments. We used Resnet50 and MobilenetV3 as the backbone to test performance. The experimental results are shown in Table 4. The experimental results showed that the two sets of ablation experiments achieved 93.13% and 89.07% of the accuracy, respectively, and the accuracy of our proposed model is 95.23%.

Table 4. Ablation study on MIXFV.

Model	ACC (%)
Resnet50 + CapsNet	93.13
MobilenetV3 + CapsNet	89.07
Proposed	95.23

4.5. Experiment 2: Comparison with State-of-the-Art Finger Vein Recognition Algorithms

In the second set of experiments, we selected four state-of-the-art methods to conduct experiments on four publicly available datasets. The first algorithm we chose was the vision transformer model, and we applied it to finger vein recognition. The second algorithm we chose was the capsule network model, which used the capsule network to train finger vein images, and the third algorithm selected was the finger vein recognition algorithm based on convolutional neural network proposed by Das et al. [14], which obtained stable and high-precision performance when processing finger vein images of different qualities. The last algorithm we chose was the method based on maximum curvature proposed by Miura et al. [23]. The algorithm extracted the features used for finger vein recognition by calculating the local maximum curvature point of the finger vein cross-section, and it becme a presentative algorithm in the field of finger vein recognition. Table 5 shows the recognition accuracy achieved by four comparative finger vein recognition algorithms and ViT-Cap algorithm. Experimental results indicated that the recognition performance of the proposed method was better than that of other methods.

Table 5. Comparison with state-of-the-art methods of finger vein recognition.

Datasets	Training	Testing	Accuracy				
			State-of-the-Art Methods				
			ViT	Capsule Net	CNN [9]	MC [19]	ViT-Cap
MMCBNU	6 images	remaining 4 images	95.13	96.29	-	-	**97.52**
SDUMLA	5 images	remaining 1 image	90.88	95.66	97.48	**97.95**	93.24
FV-USM	8 images	remaining 4 images	95.99	96.47	97.53	90.34	**98.68**
HKPU	8 images	remaining 4 images	87.62	95.07	95.13	85.24	**95.61**

As can be seen from the obtained recognition accuracy, the average recognition accuracy of our proposed model on four public datasets was 96.26%, which was significantly better than other advanced algorithms. The recognition accuracy of ViT-Cap model on FV-USM, SDUMLA, MMCBNU, and HKPU datasets was 98.68%, 93.24%, 97.52%, and 95.61%, respectively, while the average accuracy of the vision transformer and capsule network model was only 92.41% and 95.87%, respectively. This indicated that our proposed model has excellent extraction ability of finger vein features. Compared with the existing methods, the ViT-Cap provided a new reference model for finger vein recognition.

EER plays an important role in evaluating the recognition effect of finger vein models, so we selected the recognition algorithms proposed in six cutting-edge papers on finger vein recognition and experimented with our algorithm and compared the obtained EER results. Table 6 shows the EER values obtained by our model and the EER results of other related algorithms. Repeated line tracking algorithm and maximum curvature algorithm were representative finger vein recognition methods, but all had high EER values in three public datasets. We also compared our model with the deep learning-based recognition algorithms proposed in recent years [37–40] and found that our model still achieved the best results. The results of Table 6 are the results of a direct citation of the original paper. We ensured that we compared each method based on the same datasets and metrics and used the same number of training and testing datasets. For each, the best results were selected for comparison. Therefore, we believe that the comparison was fair, and our model can withstand the test of different methods.

Table 6. Performance comparison in terms of EER (%).

Algorithm	Year	MMCBNU	SDUMLA	FV-USM	HKPU
Miura et al. [17]	2003	5.74	5.85	-	3.32
Miura et al. [19]	2005	2.69	3.65	-	2.41
Qin et al. [37]	2018	-	-	0.80	2.33
Kang et al. [38]	2019		1.69	0.94	2.40
Yao et al. [39]	2021	1.68	-	2.12	4.23
Tao et al. [40]	2021	-	2.23	-	-
Proposed	2021	**0.63**	**1.3**	**0.28**	**1.66**

4.6. Experiment 3: Performance Analysis of algorithms on Small Sample and Small Categories of Finger Vein Datasets

Artificial intelligence technology, represented by deep learning, is booming, and there are more applications in the field of finger vein recognition. However, deep learning is more suitable for large datasets in applications. Deep learning needs to automatically learn features from the data, which is usually only possible with a large amount of training data. However, the model we proposed can perform better on the small amount of finger vein datasets. Our model could better explore the relationship between data features and maximize the performance of the model.

In most cases, to improve the performance of deep learning-based finger vein recognition models, the finger vein datasets are augmented to obtain more experimental samples. We redesigned the experiment to test the model's performance on a finger vein dataset with fewer samples. We used a lesser number of images for training and compared the results, and we compared the proposed method with capsule networks. We redivided the four public datasets, and when the test set was determined to be one sample, the data of the training set was reduced in turn, and the division methods of 4-1 and 3-1 were selected, respectively, to test the effect of the model in the case of a small sample of data. Table 7 shows the comparison results of the proposed model ViT-Cap and capsule network in accuracy, EER and AUC values under the condition of reducing training samples. The experimental results showed that the performance of our model decreased to a certain extent when the sample size was reduced. However, compared with deep learning models, such as capsule networks, that process small sample data better, our model could still achieve better performance than the capsule network model with fewer samples, and the average recognition accuracy of our model was 91.7%.

Table 7. Finger vein recognition performance of ViT-Cap on small sample datasets.

Datasets	Training	Testing	Model	ACC (%)	EER (%)	AUC
MMCBNU	4 images	1 image	Capsule	90.83	1.87	0.99
			ViT-Cap	91.66	1.17	0.99
	3 images	1 image	Capsule	86.66	2.53	0.99
			ViT-Cap	87.16	2.49	0.99
SDUMLA	4 images	1 image	Capsule	88.50	5.06	0.98
			ViT-Cap	90.25	3.11	0.99
	3 images	1 image	Capsule	81.13	8.02	0.97
			ViT-Cap	82.71	4.13	0.99
FV-USM	4 images	1 image	Capsule	93.08	1.46	0.99
			ViT-Cap	94.31	2.21	0.99
	3 images	1 image	Capsule	88.33	4.52	0.99
			ViT-Cap	89.02	3.61	0.99
HKPU	4 images	1 image	Capsule	89.04	4.24	0.98
			ViT-Cap	93.09	2.35	0.99
	3 images	1 image	Capsule	82.38	8.94	0.96
			ViT-Cap	92.14	2.31	0.99

5. Conclusions

In this study, a new finger vein recognition model based on a capsule network of vision transformers was proposed. We combined the advantages of a capsule network in processing the underlying vision with the advantages of a transformer in processing the relationship between visual elements and objects, thus solving the problem that capsule networks lack the ability to encode long-range dependencies in images and cannot selectively focus on important image feature information when it is used for image classification. Thus, a better classification effect of the finger vein was achieved. At the same time, we used dynamic routing mechanisms to solve the problem of poor model training effect caused by the small amount of finger vein data. It is worth noting that the ViT-Cap model we first proposed was used to solve the task of finger vein recognition, which provided a completely new reference model for finger vein recognition. ViT-Cap showed a significant advantage over other CNNs in that it was more effective in processing smaller categories of data. Different from the traditional CNN, ViT-Cap endowed the model with a new expression of the relationship between local features and overall features, which provided a new idea for the direction of computer vision.

In the future, we will study how to clarify the correlation between intra-patch and inter-patch and conduct experiments in more open-finger vein datasets. In view of the problem that direct resizing in the process of model training may affect the performance of the model, we will seek a specific model or method to achieve lossless resizing of images.

Author Contributions: Conceptualization, Y.L. and H.L.; methodology, Y.L.; software, Y.L., Y.W. and C.Z.; validation, H.L. and R.G.; formal analysis, Y.L.; investigation, Y.W., H.L. and C.Z.; resources, Y.L.; data curation, R.G.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L. and H.L.; visualization, Y.W.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is sponsored by the Key R&D Project of Jilin Provincial Science and Technology Development Plan in 2020 (No. 20200401103GX).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhong, Y.; Deng, W.; Hu, J.; Zhao, D.; Li, X.; Wen, D. Sface: Sigmoid-constrained hypersphere loss for robust face recognition. *IEEE Trans. Image Process.* **2021**, *30*, 2587–2598. [CrossRef] [PubMed]
2. Wang, K.; Kumar, A. Periocular-assisted multi-feature collaboration for dynamic iris recognition. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 866–879. [CrossRef]
3. Liu, F.; Liu, G.J.; Zhao, Q.J.; Shen, L.L. Robust and high-security fingerprint recognition system using optical coherence tomography. *Neurocomputing* **2020**, *402*, 14–28. [CrossRef]
4. Ma, H.; Hu, N.; Fang, C.X. The biometric recognition system based on near-infrared finger vein image. *Infrared Phys. Technol.* **2021**, *116*, 103734. [CrossRef]
5. Jin, J.; Di, S.; Li, W.; Sun, X.; Wang, X. Finger vein recognition algorithm under reduced field of view. *IET Image Process.* **2020**, *15*, 947–955. [CrossRef]
6. Yang, L.; Yang, G.; Xi, X.; Su, K.; Chen, Q.; Yin, Y. Finger vein code: From indexing to matching. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1210–1223. [CrossRef]
7. Kumar, A.; Zhou, Y.B. Human identification using finger images. *IEEE Trans. Image Process.* **2012**, *21*, 2228–2244. [CrossRef]
8. Shaheed, K.; Liu, H.; Yang, G.; Qureshi, I.; Gou, J.; Yin, Y. A systematic review of finger vein recognition techniques. *Information* **2018**, *9*, 213. [CrossRef]
9. Das, R.; Piciucco, E.; Maiorana, E.; Campisi, P. Convolutional neural network for finger-vein-based biometric identification. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 360–373. [CrossRef]
10. Wang, G.Q.; Sun, C.M.; Sowmya, A. Learning a compact vein discrimination model with ganerated samples. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 635–650. [CrossRef]

11. Lu, Y.; Xie, S.J.; Wu, S.Q. Exploring competitive features using deep convolutional neural network for finger vein recognition. *IEEE Access* **2019**, *7*, 35113–35123. [CrossRef]
12. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. In Proceedings of the 31th Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 3856–3866.
13. Gumusbas, D.; Yildirim, T.; Kocakulak, M.; Acir, N. Capsule network for finger-vein-based biometric identification. In Proceedings of the IEEE Symposium Series on Computational Intelligence, Xiamen, China, 6–9 December 2019; pp. 437–441.
14. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An image is worth 16 × 16 words: Transformers for image recognition at scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 3–7 May 2021.
15. Rice, J. Apparatus for the Identification of Individuals. US Patent No. 4,699,149, 19 March 1985.
16. Kono, M.; Ueki, H.; Umemura, S. A new method for the identification of individuals by using of vein pattern matching of a finger. In Proceedings of the Fifth Symposium on Pattern Measurement, Yamaguchi, Japan, 20–22 January 2000; pp. 9–12.
17. Miura, N.; Nagasaka, A.; Miyatake, T. Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. *Mach. Vis. Appl.* **2004**, *15*, 194–203. [CrossRef]
18. Qin, H.F.; Qin, L.; Yu, C.B. Region growth-based feature extraction method for finger-vein recognition. *Opt. Eng.* **2011**, *50*, 214–229. [CrossRef]
19. Miura, N.; Nagasaka, A.; Miyatake, T. Extraction of finger-vein patterns using maximum curvature points in image profiles. *IEICE Trans. Inf. Syst.* **2007**, *90*, 1185–1194. [CrossRef]
20. Gupta, P.; Gupta, P. An accurate finger vein based verification system. *Digit. Signal Process.* **2015**, *38*, 43–52. [CrossRef]
21. Rosdi, B.A.; Chai, W.S.; Suandi, S.A. Finger vein recognition using local line binary pattern. *Sensors* **2011**, *11*, 11357–11371. [CrossRef]
22. Van, H.T.; Thai, T.T.; Le, T.H. Robust finger vein identification base on discriminant orientation feature. In Proceedings of the Seventh International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City, Vietnam, 8–10 October 2015; pp. 348–353.
23. Hong, H.G.; Lee, M.B.; Park, K.R. Convolutional neural network-based finger-vein recognition using NIR image sensors. *Sensors* **2017**, *17*, 1297. [CrossRef]
24. Zeng, J.; Wang, F.; Deng, J.; Qin, C.; Zhai, Y.; Gan, J.; Piuri, V. Finger vein verification algorithm based on fully convolutional neural network and conditional random field. *IEEE Access* **2020**, *8*, 65402–65419. [CrossRef]
25. Wang, K.X.; Chen, G.H.; Chu, H.J. Finger vein recognition based on multi-receptive field bilinear convolutional neural network. *IEEE Signal Process. Lett.* **2021**, *28*, 1590–1594. [CrossRef]
26. Li, Q.; Shen, L.; Guo, S.; Lai, Z. WaveCNet: Wavelet integrated CNNs to suppress aliasing effect for noise-robust image classification. *IEEE Trans. Image Process.* **2021**, *30*, 7074–7089. [CrossRef]
27. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. CoRR 2019, abs/1905.05055. Available online: https://arxiv.org/abs/1905.05055 (accessed on 16 May 2019).
28. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [CrossRef]
29. Degardin, B.; Proença, H. Human behavior analysis: A survey on action recognition. *Appl. Sci.* **2021**, *11*, 8324. [CrossRef]
30. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 213–229.
31. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 10347–10357.
32. Chen, B.; Li, P.; Li, C.; Li, B.; Bai, L.; Lin, C.; Ouyang, W. Glit: Neural architecture search for global and local image transformer. In Proceedings of the International Conference on Computer Vision (ICCV), Virtual Event. 11–17 October 2021; pp. 12–21.
33. Lu, Y.; Xie, S.J.; Yoon, S.; Wang, Z.; Park, D.S. An available database for the research of finger vein recognition. In Proceedings of the IEEE International Congress on Image and Signal Processing (CISP), Dalian, China, 14–16 October 2014; pp. 410–415.
34. Yin, Y.L.; Liu, L.L.; Sun, X.W. SDUMLA-HMT: A multimodal biometric database. In Proceedings of the Chinese Conference on Biometric Recognition (CCBR), Shanghai, China, 10–12 September 2011; pp. 260–268.
35. FV-USM Finger vein Image Database (DB/OL). Available online: http://drfendi.com/fv_usm_datae (accessed on 22 January 2021).
36. Lu, H.; Wang, Y.; Gao, R.; Zhao, C.; Li, Y. A novel ROI extraction method based on the characteristics of the original finger vein image. *Sensors* **2021**, *21*, 4402. [CrossRef]
37. Qin, H.F.; Mounim, A. Deep representation for finger-vein image-quality assessment. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 1677–1693. [CrossRef]
38. Kang, W.; Liu, H.; Luo, W.; Deng, F. Study of a full-view 3D finger vein verification technique. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1175–1189. [CrossRef]
39. Yao, Q.; Song, D.; Xu, X.; Zou, K. A novel finger vein recognition method based on aggregation of radon-like features. *Sensors* **2021**, *21*, 1885. [CrossRef] [PubMed]
40. Tao, Z.; Wang, H.; Hu, Y.; Han, Y.; Lin, S.; Liu, Y. DGLFV: Deep generalized label algorithm for finger-vein recognition. *IEEE Access* **2021**, *9*, 78594–78606. [CrossRef]