Digital Object Identifier

Multi-tasks Loss Function is All You Need

JIASHU LOU[1]
[1]Shenzhen University, No. 3688, Nanhai Avenue, Shenzhen, Guangdong, China (e-mail: loujiashu@163.com)

ABSTRACT With the improvement of arithmetic power and algorithm accuracy of personal devices, biological features are more and more widely used in personal identification, and palm features possess richer feature points than fingerprints, for example. In this paper, based on the infrared palm vein dataset, we first extract features using optical methods. And a convolutional neural network based on VGG-16 migratory learning fused attention mechanism is proposed. The resulting model is finally applied to palm vein matching, and a matching goodness-of-fit index is introduced for evaluation. Based on this, a **multi-tasks loss function** is proposed to fuse the classification task and the matching task. The results show that the model of this paper obtains 97.39% matching accuracy. And the model still performs well in different data sets. We used clustering to determine the matching threshold adaptively and achieved an accuracy rate of 98.89% in different data sets.At the same time, the training time is short and the matching speed is fast, which has some application value.

INDEX TERMS Palm vein recognition, Deep learning, Transform learning, Multi-tasks loss function

I. INTRODUCTION

A. PALM FEATURE MATCHING

Biometric identification technology refers to the use of an individual's unique biometric characteristics to uniquely determine an individual's identity [1] [2]. With the development of mobile Internet and the increase of computing speed of personal electronic devices, technologies such as facial recognition [5] [6], fingerprint recognition [3] [4], and even iris recognition [7] have been integrated into every personal terminal. And palmprint recognition as an emerging biometric technology has recently appeared in the public eye. Palm prints have richer features compared to fingerprints which is not easy to change and has better identification and uniqueness.

Palmprint research uses high resolution or low resolution images. High-resolution images are suitable for forensic applications such as criminal investigation, e.g., Jain, AK (2009) [11] developed latent palmprint matching which mainly deals with low-resolution palmprints. Since about 30% of the biopsies recovered from crime scenes come from the palm of the hand, the evidentiary value of palm prints in forensic applications is evident. Low-resolution images, on the other hand, are more suitable for civilian and commercial applications such as access control. Generally, high resolution is defined as 400 dpi or higher, and low resolution is defined as 150 dpi or lower. Ridges, singularities and detail points are generally extracted as features in high-resolution images. While in low resolution images, main lines, wrinkles and textures are usually extracted. Initially, palmprint research focused on high-resolution images [29] [30]. However, recent research hotspots have shifted to commercial and residential low-resolution palmprint recognition and matching [31]

palm vein features are similar to it, and its application is better [38] because the features of veins are more obvious and unique. Moreover, compared to palm prints, palm veins rely on biometric information inside the body and are therefore less susceptible to destruction, alteration or tampering. Therefore, vein recognition is becoming one of the most reliable methods in biometrics and has attracted wide interest from biometric researchers. Veins are huge networks of blood vessels under the human skin that are almost invisible to the human eye and are more difficult to replicate than other biometric features [40]. The shape of the vascular pattern is thought to be unique across individuals (Wilson, 2010) [39] and stable over time.

For now, the academic community mainly focuses on the extraction and localization of palmprint features using mathematical means, such as Huang (2008) et al [9] and Wu (2006) et al [10] using intrinsic features of palmprints, such as main lines and wrinkles, for palmprint recognition. ; Lu, GM (2003) [8] proposed a method to extract palmprint

feature vectors by Karhunen-Loeve transformation; Zhang, D (2010) extended the feature space to three dimensions, extracted depth features using structured light techniques, and developed a multilevel framework for personal identity verification. Palmprint matching has been richly used in various fields. In addition, with the popularity of deep learning and the increase of computing hardware arithmetic power, deep learning networks are gradually used in palmprint recognition. For example, Zhao, SP (2022) [12] used deep convolutional neural networks to extract palmprint features and developed a joint constrained least squares regression framework for palmprint recognition; Trabelsi, S (2022) [13] used a simplified PalmNet-Gabor method to improve PalmNet and obtained a higher high accuracy, reducing the number of features and saving computation time.

B. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks use the original image as input, which can effectively learn the corresponding features from a large number of samples and avoid the complicated feature extraction process. Unlike fully connected neural networks, convolutional neural networks (CNNs) can directly process two-dimensional images, while the former loses dimensional information in the process of image spreading. Therefore, CNNs have been widely used in image processing. For example, R Vinoth (2014) [27] et al. used CNN networks for tumor image recognition and segmentation in magnetic resonance imaging and compared them with traditional methods such as SVM; Xiao Huang (2019) [28] et al. used two convolutional neural networks to extract visual and textual features from social media posts and identify disasters associated with related social media for fast response. It can be seen that convolutional neural networks have made outstanding contributions in the field of image recognition.

However, a deep convolutional neural network contains a large number of parameters to be trained (e.g., ResNet-50 [17] has 23 million parameters), and we need a large amount of labeled data to train it, but manually labeling the data would be a very large amount of work. One idea is to use unsupervised learning to annotate the data, as in Kiselev, VY (2019) [15] Unsupervised learning of single-cell RNA-seq data using clustering. Another direction is migration learning [16], which can take knowledge from one domain (source domain) and migrate it to another domain (target domain), enabling the target domain to achieve better learning results. This is especially true for scenarios where the source domain has sufficient data and the target domain has a small amount of data. In the field of image recognition and classification, migration learning has achieved good performance.Shin, HC (2016) [14] used a network pre-trained on ImageNet dataset for medical image recognition by migration learning for thoracoabdominal lymph node (LN) detection and interstitial lung disease (ILD) classification, and reported achieving the best performance. The combination of convolutional neural networks and migration learning allows for a much broader range of applications of deep learning.

C. THIS PAPER

In this paper, we will focus on feature extraction of palm vein images. Since the traditional optical feature extraction is characterized by strong interpretability and clear mathematical and physical principles; while the deep convolutional extraction features are characterized by adequate feature mining and better performance. In this paper, we propose to combine the two, and firstly, optical enhancement is achieved by histogram equalization and Gabor filtering. The resulting results are then fed into a partially pre-trained VGG-16 [18] deep convolutional network for feature extraction. An attention mechanism was also introduced to improve the prediction accuracy. In terms of training, we first train the classification network in the traditional way, and finally achieve the matching degree calculation for two palm vein images of any input model by zero-shot learning, and can give an acceptance or rejection judgment. Meanwhile, we propose a **multi-tasks loss function**. The training results under this loss function are given and compared with the previous one. Finally, we extend the model to palm vein datasets obtained by different methods. An adaptive threshold adjustment strategy is proposed to obtain the best matching effect.

II. MATERIALS AND METHODS

The model in this paper will be divided into two aspects, training and testing, since ultimately the task to be solved in this paper is not to classify the given palm vein images, but to determine whether the given pair of palm vein images belong to the same person. If reconstructing the dataset into pairs of palm vein images and labels of whether they belong to the same person, it will bring a large workload and is not conducive to migration learning using existing convolutional networks. Therefore, after extracting the optical features, we use VGG-16 for feature extraction, and then send the data to the linear classification layer during training, and use the output given by this layer to calculate the loss in combination with the training labels for training. After the training, the test data is sent into the VGG-16 feature extraction layer to extract features, and then the extracted features are spread into a feature vector, and the similarity of the two image feature vectors is calculated using the similarity function. And a certain threshold value is used to decide whether these two images belong to one person or not. The Figure1 gives the general structure of the model in this paper, and each link will be introduced in this paper.

A. THE DATA SETS

The dataset used in this paper is the palm vein dataset from HongKong Poly university [34] [43]. To fit the input size of the convolutional network, we first cropped the image to 224*224 size centered on the palm of the hand to be the ROI(region of interest). The images were sourced from the same person by recording 6 consecutive images and again 6 images 10 days later. That is, there are 12 pictures from the same hand each. Therefore, for the classification task, we use the former as the training set, counting as the set TR.

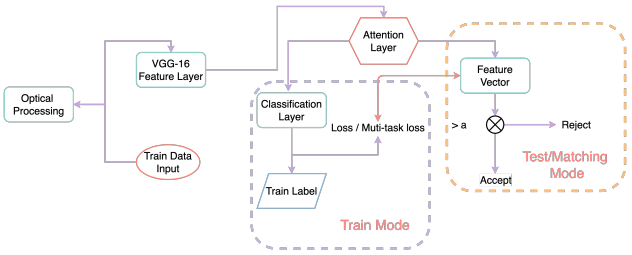

FIGURE 1. Model Structure

And every 9 photos are taken out as the validation set, which is counted as the set V. For the matching process, in order to simulate the entry-comparison process in real application scenarios, we extract the last 10 sets of photos (i.e., 10 different palms) from the dataset as the test set for zero-experience learning matching to test the matching effect, and match the photos in the two datasets one by one to form a 60*60 dataset, which is counted as The set TE_m.

B. OPTICAL PRE-PROCESSING

Due to the influence of environment, light, etc., the clarity and contrast of the photographed images are relatively low, which cannot highlight the features in the palm vein images. Therefore, we first perform optical image enhancement to enhance the contrast of the image by certain means to make the features more obvious, which is beneficial for the later recognition and classification.

1) histogram equalization

Histogram equalization is often used when the grayscale image is too concentrated in the grayscale range, resulting in the subject and background being very similar, and it is not easy to distinguish features from noise [19]. Changing the grayscale of each pixel in an image by changing the histogram of the image is mainly used to enhance the contrast of images with a small dynamic range. The following figure shows a randomly selected image of a palm vein with its grayscale histogram.

It can be seen that the grayscale values are mostly concentrated between 50 and 100, with a very concentrated distribution. From the palm vein images, we can also see that the color difference between the palm vein and the rest of the palm is very small due to the illumination, which makes it difficult to identify the main lines of the palm. Therefore, we apply histogram equalization using the following equation.

$$s_k = T(r_k) = \sum_{i=0}^{k} p_r(r_i) = \sum_{i=0}^{k} \frac{n_i}{N} \tag{1}$$

The result of the equalization is shown in the figure below, which shows that the features of the palm veins are effectively enhanced, while the gray values are more even.

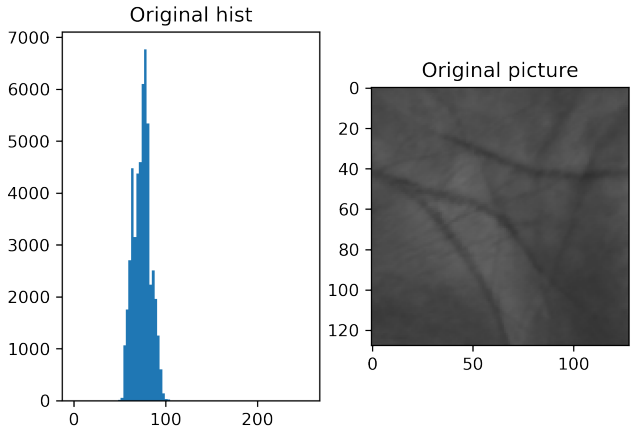

FIGURE 2. Original grayscale histogram with images

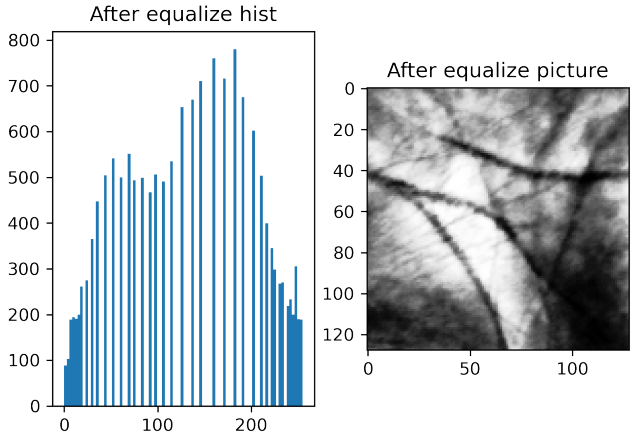

FIGURE 3. Grayscale histogram with images after equalization

2) Gabor filter

In the field of image processing, the Gabor filter, named after Dennis Gabor, is a linear filter for texture analysis, i.e., it analyzes, among other things, whether an image has specific frequency content in a particular direction in a particular region [35]. It is found that the Gabor filter is particularly suitable for texture representation and discrimination [36]. the Gabor filter is implemented using the following Gabor kernel function.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \tag{2}$$

Its frequency and directional expression is similar to that of the human visual system, which can provide good directional and scale selection properties, and is insensitive to light changes, making it well suited for texture analysis.

C. VGG TRANSFER LEARNING

1) Introduction of VGG structure

The VGG deep learning network was proposed by K Simonyan, A Zisserman in 2014 [18], using an architecture with very small (3x3) convolutional filters for a comprehensive evaluation of networks with increasing depth, and introducing a 1*1 convolutional kernel in the convolutional structure of VGG, without affecting the input-output dimensionality, and introducing nonlinear transformations to increase the expressiveness of the network and reduce the computational effort. They demonstrate that proving increasing the depth to 16-19 weight layers can achieve a significant improvement on the existing technical configuration.

VGG-16 has a total of 16 layers, divided into 13 convolutional layers and 3 fully connected layers. The first time, after two convolutions of 64 convolutional kernels, one pooling is used, the second time, after two convolutions of 128 convolutional kernels, another pooling is used, and two more repetitions of three 512 convolutional kernels are convolved and then pooled, and finally, three full connections are used to obtain the classification output The results. The structure schematic is shown below.

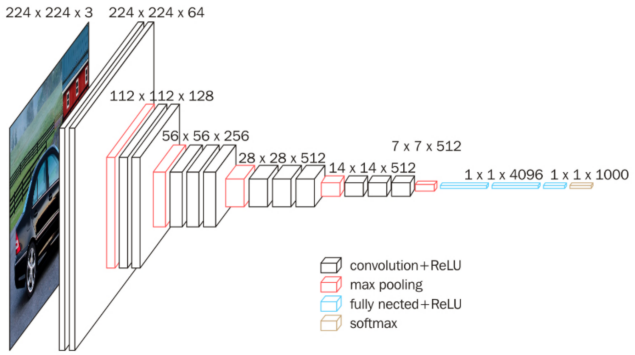

FIGURE 4. VGG-16

In the convolution layer, we use the convolution kernel to slide over the original image. For a two-dimensional image input, we perform the convolution operation using the following equation.

$$S(i,j) = \sum_m \sum_n I(m,n)k(i-m, j-n) \qquad (3)$$

This can effectively learn the corresponding features from a large number of samples and avoid the complicated feature extraction process.

The activation function is a node after the convolution layer that performs a nonlinear transformation of the input signal. The rectified linear unit activation function (ReLU) is a partially linear function. It will suppress the negative part of the signal to zero and output the positive part of the signal. Its function image is shown below.

However, since the convolutional layer records the exact positions of the features in the input, this means that cropping, rotating, or any other minor changes to the input

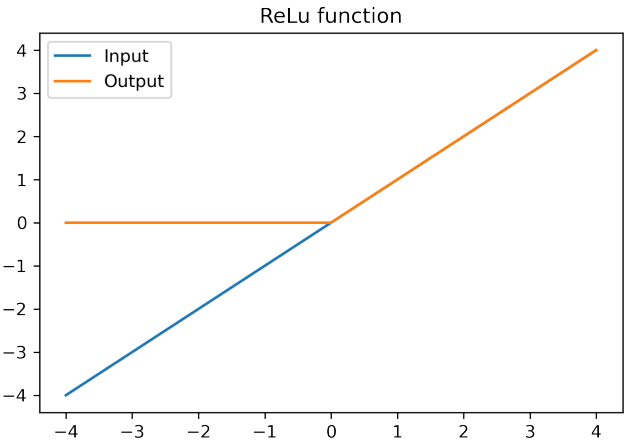

FIGURE 5. Relu function

image will result in a different feature output altogether. To solve this problem, we downsample the convolutional layer. Downsampling sampling can be achieved by applying a pooling layer after the nonlinear layer [20]. Pooling helps to make become approximately invariant to small amplitude panning of the input. Translational invariance means that if we translate the input by a small amount by a small amount, the value of the output of most pooling layers will not change. This has a large improvement on the robustness of image recognition.

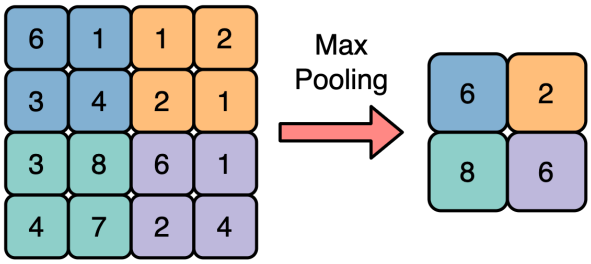

FIGURE 6. Maximum pooling

At the end of the network, VGG uses a set of fully connected nodes for the output, where the final output dimension is the number of categories of images in the dataset. In the middle of each fully connected layer, we add a Dropout layer to randomly deactivate a certain percentage of neurons, which avoids the occurrence of model overfitting [42]. Finally, the obtained results are fed into the $Softmax$ function (4) to find the probability distribution, and finally the loss is calculated and the error is back-propagated by the cross-entropy loss function (5).

$$\text{Softmax}\,(z_i) = \frac{e^{z_i}}{\sum_{c=1}^{C} e^{z_c}} \qquad (4)$$

$$L = \frac{1}{N}\sum_{i} L_i = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic}\log\left(p_{ic}\right) \tag{5}$$

2) Transfer Learning

As more and more machine learning application scenarios emerge, and existing supervised learning that performs better requires large amounts of labeled data, labeling data can be a tedious and costly task, so transfer learning is receiving more and more attention. The goal of transfer learning is to apply the knowledge or patterns learned on a domain or task to a different but related domain or problem [22]

For the palm vein recognition task in this paper, we use a VGG-16 network pre-trained on the ImageNet [21] dataset for transfer learning. imageNet is pre-trained on more than one million images containing more than 20,000 categories. The data volume of the palm vein dataset involved in this paper is small, but also because there is still some difference between object recognition and palm vein recognition. Considering the above, we freeze the first 30 convolutional layers in VGG-16 with gradient, i.e., they do not generate gradient information during the training process, and thus the parameters are not updated. Instead, the remaining 34 convolutional layers are trained with fully connected layers, which greatly shortens the training time and increases the convergence rate while ensuring the prediction accuracy.

3) Attention mechanism

When a human sees a set of information, human neurons will automatically scan the global image based on past experience and the current target, and obtain the target area that needs to be focused on, that is, the focus of attention. More attention is then devoted to this region to obtain more details about the target to be focused on and to suppress other useless information. The following figure is a schematic diagram of the application of the attention mechanism in the field of image processing. For such a journal screenshot, attention is focused on the image and the caption, respectively, in accordance with our life experience. [23].

The essence of the attention mechanism is to assign a weighting factor to each factor to measure its importance, which is calculated as follows.

First, two hidden states are created, Encoder and Decoder, which are equivalent to the input and output of the structure. Subsequently, a dot product is performed with each hidden state in Decoder and Encoder, and the result is noted as $Score$. Then all the scores are sent to the softmax layer (4), so that the higher the original score of the hidden state, the higher its corresponding probability, thus suppressing those invalid or noisy information.

Next, the hidden state of each Encoder is multiplied by the $Score$ after softmax, and then all aligned vectors are accumulated and the context vectors are sent to the Decoder to obtain the decoded output. the internal structure of the Attention layer is as follows.

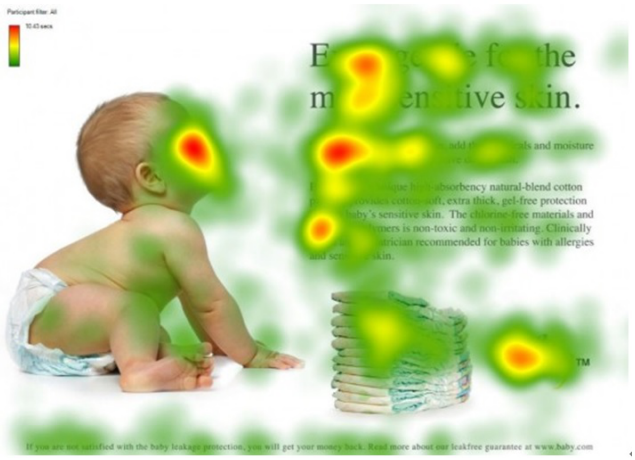

FIGURE 7. Attention

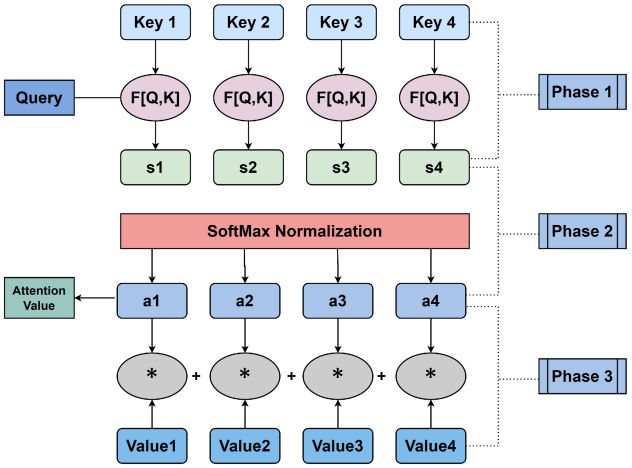

FIGURE 8. Attentional processing

For the palm vein recognition task in this paper, we want the neural network to focus more on the features and patterns of the hand, and give a relatively low weight to secondary factors such as shadows and illumination. Therefore, we use the spatial Attention mechanism before image input and the channel Attention mechanism for image feature extraction [24]. Another advantage of the Attention layer is that it is a plug-and-play adaptive module, which does not change the dimensionality of the image and does not require changes to the subject model. It can also be applied to the learning and training of migration models if it is applied before the input or after the output of the subject model.

D. TRAINING AND MATCHING ALGORITHMS

1) Framework and optimization

In this paper, we build and train the model based on the Pytorch framework [26], and optimize the model using the Adam algorithm [25]. Adam algorithm is different from the

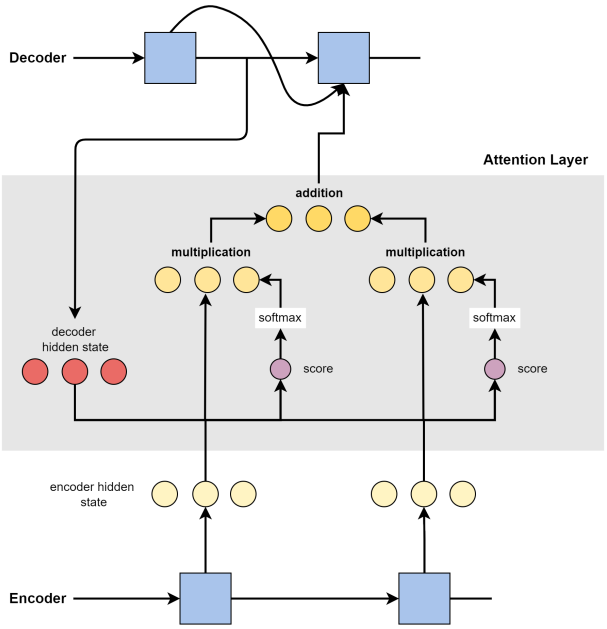

FIGURE 9. Attention layer structure

traditional stochastic gradient descent. Stochastic gradient descent maintains a single learning rate to update all the weights, and the learning rate does not change during the training process. In contrast, the Adam algorithm designs independent adaptive learning rates for different parameters by computing first-order moment estimates and second-order moment estimates of the gradient. This allows the loss to drop faster in the early learning period without jaggedness in the late learning period due to excessive learning rate.

Meanwhile, we use the (7)L_2 penalty term for the linear output layer, which enables the network to output a relatively sparse feature vector, which is more helpful for the subsequent similarity calculation. The L_2 penalty term can limit the size of the secondary elements in the weights, and in terms of optimization difficulty L_2 penalty is more convex compared to L_0 and L_1 penalties [32], so the optimization is less difficult and the results obtained are better. Also, the inclusion of the regular term helps to avoid overfitting because it compresses part of the parameters to a number close to 0, making the model simpler.

$$penalty = \|w\|_2 \tag{6}$$

The expression of the loss function after adding the canonical term is as follows.

$$loss = Crossentropy(X, y) + \lambda \|w\|_2 \tag{7}$$

2) Matching algorithm

After feature extraction by the above deep learning network, for the input ith palm vein image, we can get the feature vector V_i. Subsequently, to determine whether the ith and jth images are from the same person, we use the cosine similarity function for matching comparison, whose expression is shown in (8).

$$s = \cos(\theta) = \frac{V_i \cdot V_j}{\|V_i\|\|V_j\|} = \frac{\sum_{k=1}^{n} V_{ik} \times V_{jk}}{\sqrt{\sum_{k=1}^{n} (V_{ik})^2} \times \sqrt{\sum_{k=1}^{n} (V_{jk})^2}} \tag{8}$$

When $s > \alpha$, we decide that the two palm vein pictures originate from the same person, and reject them in the opposite case. Obviously, in the most ideal case, for any V_i and V_j coming from the same person, there should be:

$$s = \cos(V_i, V_j) = 1 \tag{9}$$

And for any V_i and V_j from different persons, there should be:

$$s = \cos(V_i, V_j) = 0 \tag{10}$$

But in practical scenarios, we can relax this judgment condition. We usually use the threshold α to determine whether it is the same person or not. It follows that as long as the difference between the similarity s_i for any pair of palm veins from the same person and the similarity s_j for any pair of palm veins from different people is large enough, then we can always find a suitable α that allows us to distinguish whether a given input is the same person or not. That is, the difference between the similarity values computed by the similarity function from the same person and different persons should be as large as possible, i.e., equation (11) should be as large as possible. We call this equation **matching goodness of fit**, which is counted as MG.

$$MG = Average(\cos(V_{is}, V_{js})) - Average(\cos(V_{id}, V_{jd})) \tag{11}$$

This facilitates us to pick a better threshold α more easily, making the rate of wrong rejections and wrong acceptance decrease.

3) multi-tasks loss function

For the traditional classification task, our loss function is shown in (7). That is, we solve the problem posed by (11) by solving the (7)-style. However, there is no research to prove that there is a relaxation optimization relationship between the two, and it may not be effective if the classification model alone is more indirect for the matching task. Therefore, we would like to add (11) to the loss function as well for optimization together. The reconstructed loss function is shown as follows.

$$loss = \theta * Crossentropy(X, y) + (1-\theta)(1-MG) + \lambda \|w\|_2 \tag{12}$$

This is because although we want the value of equation (11) to be the largest for all data (i.e., $1 - MG$ is the smallest), it is very detrimental to training because it takes

too much time to compute MG. However, since we divide the training batches randomly, so that each training batch $1 - MG$ minimum is equivalent to the overall $1 - MG$ minimum.

The parameter θ controls the weight between two tasks, $\theta = 1$ for a pure classification task and $\theta = 0$ for a pure matching task. It is worth noting that the premise of computing the batch MG is that there must be both data of the same category and data of different categories in each batch. To achieve this, we may need to increase the amount of data in each batch, but this inevitably leads to a decrease in training accuracy. We will elaborate on the specific parameter selection in Chapter 3.

4) Judging criteria

For a single pair of samples, we can use the deviation from the ideal state equation (9, 10) to measure the deviation. For a given multi-treatment matching sample, since the matching result is essentially a dichotomous variable of "acceptance" and "rejection", we use the AUC index to judge the matching merit. It is calculated by the following formula.

$$AUC = \frac{\sum pred_{pos} > pred_{neg}}{positiveNum * negativeNum} \tag{13}$$

It is noted that positive and negative cases may be unbalanced in this problem. In contrast, the calculation of AUC takes into account the classification ability of positive and negative cases and is able to make a reasonable evaluation of the classifier despite the unbalanced sample. Therefore, AUC is not sensitive to whether the sample categories are balanced or not, and is suitable as a scoring criterion for this work.

III. RESULT

A. RESULTS BASED ON TRADITIONAL CLASSIFICATION-MATCHING METHODS

1) classification results

In this paper, we base on Pytorch 1.11.0, Python 3.9.12, CUDA 11.6, and use Tesla V100 for GPU acceleration for training. Batch size = 32 is set, 20 rounds are trained, and a regularization strength of $\lambda = 0.001$ is used. The resulting classification accuracies and loss functions for the training set, validation set and test set are shown in Figure 10 and Figure11, and the training was stopped at the 7th round due to the fast convergence rate.

The training took a total of 173 seconds for 7 rounds, and it can be found from the figure that the model converged well. The accuracy is 100% on the training set and 99.85% on the validation set. We also tried different combinations of parameters, and the results are shown in the table 1.

The results show that with the regularization strength of $\lambda = 0.001$, the final loss and accuracy are not sensitive to the number of batches and the number of training sessions. However, for $\lambda = 0.01$ the model appears to be underfitted.

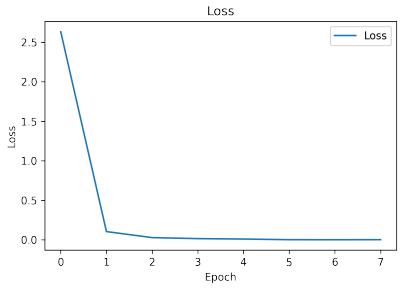

FIGURE 10. Train loss

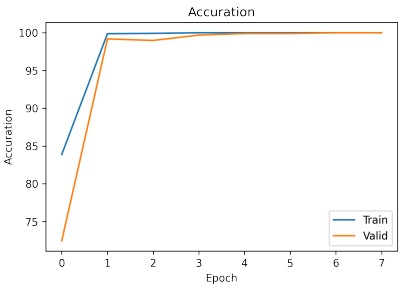

FIGURE 11. Train accuracy

2) matching results

Next, the matching effect is tested on the dataset TE_m. Also, to get a better matching effect, we test thresholds of 0.6, 0.65, 0.7, 0.75, and 0.8, calculate their AUCs and compare them. Figure 12 shows the comparison graph of AUC, and it can be found that the best result is obtained when the threshold is 0.6. Figure 13 shows the prediction when the threshold is 0.6, where yellow is the correct prediction and blue is the wrong prediction.

Batch Size	Epoch	Regularization strength	Loss	Train Accuration	Test Accuration	Early stop
32	**10**	**0.001**	0.000385	100%	99.85%	7
64	**10**	**0.001**	0.000461	100%	98.73%	No
32	**5**	**0.001**	0.000612	99.63%	98.77%	No
64	**5**	**0.001**	0.001324	96%	95.87%	No
32	**10**	**0.01**	0.00836	95.42%	94.68%	8
64	**10**	**0.01**	0.00972	95.11%	94.32%	No
32	**5**	**0.01**	0.0132	94.91%	89.23%	No
64	**5**	**0.01**	0.0187	91.61%	85.09%	No

TABLE 1. Experimental results of the original loss function

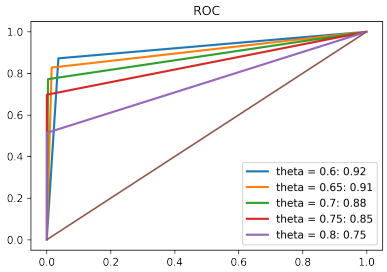

FIGURE 12. Comparison of AUC at different thresholds

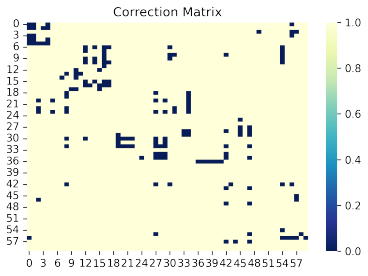

FIGURE 13. Prediction Accuracy

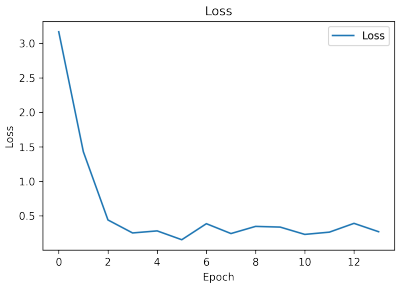

FIGURE 14. multi-tasks loss function loss

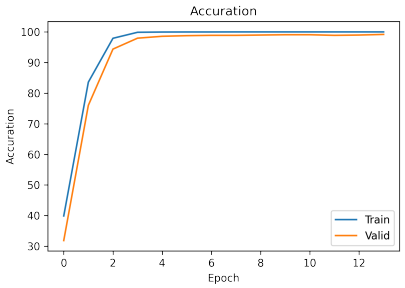

FIGURE 15. multi-tasks loss function accuracy

The overall correct rate was 95.08%. Among them, 279 pairs were correctly accepted, 121 pairs were incorrectly rejected, 56 pairs were incorrectly accepted, and 3144 pairs were correctly rejected.

Meanwhile, we used the (11) equation to calculate the goodness-of-fit under the matching task, i.e., the difference between the average similarity from the same sample and the average similarity from different samples were calculated separately. The experimental results show that the average similarity of samples from the same palm is 0.7700, and the average similarity of samples from different palms is 0.2531, with a difference of 0.5169. We find that the correct acceptance rate of this method is low and far from the standard for practical use, so we will try our improved objective function next.

B. RESULT BASED ON MULTI-TASKS LOSS FUNCTION

Set Batch size = 128, keep the rest parameters the same as the optimal parameters above, and change the value of θ for several experiments, and take one of the training accuracy and loss variation graphs to show as follows.

We found that the model converged slower due to the improvement of the loss function. The model stopped training at 12 rounds and took a total of 1247 seconds. The training time of the loss function is higher compared to the normal classification task. We tried different combinations of parameters and the obtained experimental results are shown in the table 2.

Where α are used to select the optimal value by the above method and change the value of θ, the results show that the highest matching accuracy is achieved when $\theta = 0.3$, when the loss function is as follows.

$$loss = 0.3 * Crossentropy(X, y) + 0.7 * (1 - MG) + \lambda \|w\|_2 \tag{14}$$

Where α are used to select the optimal value by the above method and change the value of θ, the results show that the highest matching accuracy is achieved when $\theta = 0.3$, when the loss function is as follows.

Theta	Alpha	Correctly accepted	Wrong accepted	Wrong rejected	Correctly rejected	Correct Rate	AUC
0.5	**0.7**	310	132	36	3122	95.33%	0.9442
0.3	**0.7**	345	79	15	3161	97.39%	0.9669
0.1	**0.7**	324	79	36	3161	96.81%	0.9378

TABLE 2. Experimental results of multi-tasks loss function

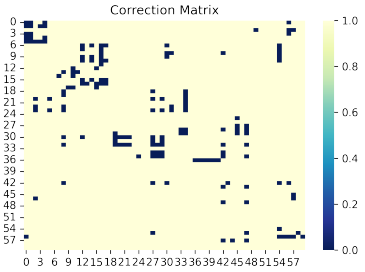

FIGURE 16. Original loss function prediction results

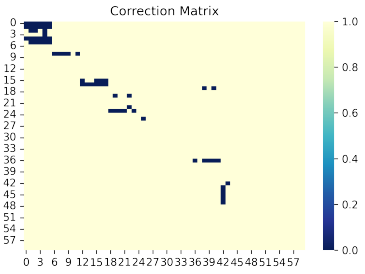

FIGURE 17. multi-tasks loss function prediction results

It can be seen that the multi-tasks loss function proposed in this paper has a higher prediction accuracy. We also test the goodness of matching, which is the difference between the average similarity of samples from different palms and from the same palms. It is 0.8293 which is far more high than the result in section A.

C. THE CASE WHERE THE TRAINING AND TEST SETS ARE FROM DIFFERENT DATASETS

The palm vein images used in the above model training and matching are from the same dataset. In other words, the locations and parameters of the photos taken are approximately the same. Testing these photos alone does not measure the true robustness and transferability of the model. Therefore, we next use photos taken in different ways for matching tests. To improve the accuracy of the model, we use palm vein images taken in two completely different environments for training, and use the dataset from the third case for matching tests. The appearance of the training image and the test image are shown below.

It is clear that the palm veins are more informative in the top left image, while the palm surface is more informative in the bottom left image. And in this case, the determination of the matching threshold is a key. Since the test and training sets are from very different datasets, the positive

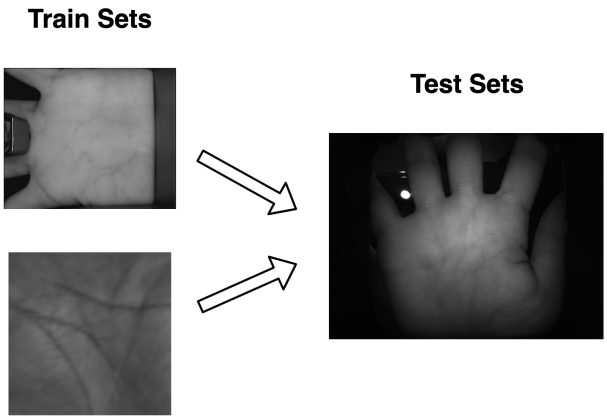

FIGURE 18. Different training and test sets

and negative examples cannot be simply partitioned by a number like 0.6 or 0.7. Therefore, we must try to determine the matching threshold α. From the above discussion, we can find that based on the Multi-tasks loss function, we believe that the final set of obtained feature vectors should satisfy the $MG(11)$ minimum. In this case, we believe that the similarity of positive and negative examples should each be well separated, and although in reality they will partially cross, we can always find a better threshold based on such an assumption. Therefore, we first perform K-mean [45] clustering on the similarity of all samples and get two clustering centers p and n, and we take $\alpha = \frac{p+n}{2}$ as the threshold for matching.

In the case of the experiments in this paper, we finally choose $\alpha = 0.992$. Predictions were performed for a test set with 46 pairs of positive cases and 134 pairs of negative cases. It turns out that out of 180 pairs of predictions, only 2 pairs of palms that should have belonged to the same person were incorrectly rejected. The overall accuracy rate was 98.89%. The following figure shows the confusion matrix of the predicted results.

We find that the model still shows very high accuracy on different datasets, so we consider the training idea of the palm vein matching model we have provided is relocatable.

D. TIME CONSUMPTION

Next, we test the time complexity of the matching process. Considering that most of the application scenarios are not supported by GPU arithmetic, in order to simulate real usage scenarios, the test will be based on Apple M1 Pro platform using CPU for inference. In the test on 3600 pairs of samples,

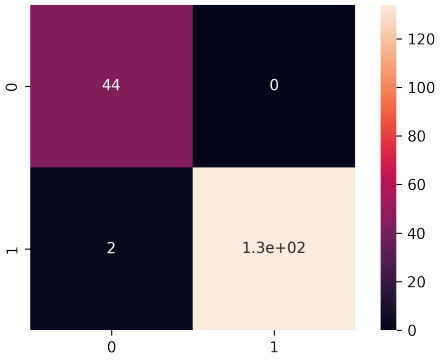

FIGURE 19. Confusion matrix of test results

the average test time per pair of samples was 0.1316 seconds, and the time consumption curve for each pair of samples is shown below.

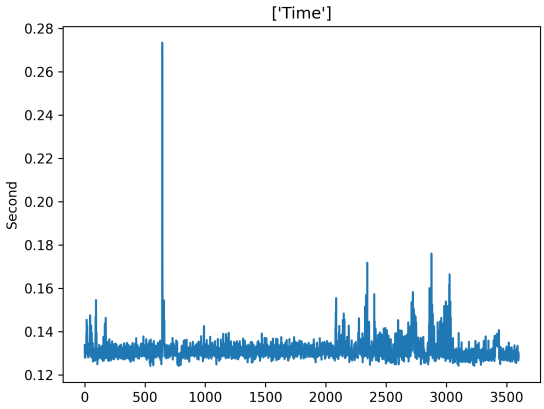

FIGURE 20. Prediction time consumption

Except for individual outliers (twice the time-consuming), the reasoning time is very short. The results show that the algorithm is short time consuming and stable, which has the possibility to be used in real life.

IV. CONCLUSION, DISCUSSION AND FUTURE WORK

A. CONCLUSION AND INNOVATION

In this paper, we combine optical feature extraction with depth feature extraction from the traditional method of optical recognition of palm veins. Through experiments, it is found that this approach can better mine image texture features and provide features that can be multi-dimensional for matching analysis. At the same time, the use of VGG-16 transfer learning improves the learning efficiency and makes the model training overhead much lower; the practice of freezing part of the pre-training parameters makes use of the original image recognition ability of the network on the one hand, and can be better adapted to different tasks on the other. At the same time, we introduce the Attention mechanism, which enables the model to distinguish between primary and secondary features after feature extraction, which can improve the model robustness and further increase the model accuracy. Finally, we use the trained model for the palm vein matching task and get an accuracy rate of more than 96%.

For the superiority test of the matching task, in addition to using the traditional AUC as a statistical index, we also introduced the concept of matching fit superiority. For all other methods we can use this metric for the goodness-of-fit test. In the subsequent study, we can use this metric to evaluate and compare traditional matching solutions, such as using PCA to extract features by dimensionality reduction and using SVM support vector machine to match [41]

For the analysis of the classification-matching results, we found that the correct acceptance rate of the model was satisfactory with respect to the overall accuracy, but the error acceptance rate was relatively high, which on the one hand is related to the low overall threshold setting. On the other hand, the overall amount of work involved in matching predictions by training the classification task is low, the dataset preprocessing is simple, and it also facilitates the use of off-the-shelf networks for transfer learning. However, the drawback is also obvious, i.e., the training target is not consistent with the final task, and matching palm veins by training classification is more indirect and does not end up with a very good result. Therefore, we propose a multi-task loss function, fusing the judging metric, the matching fit merit, into the loss function, and through a series of parameter adjustments, we demonstrate that this improvement is feasible and effective. This idea is also informative in that for a specific machine learning task, we can construct a function related to the task and combine it linearly with the loss function of the classification task to form a new optimization objective.

A similar improvement to this idea was proposed by Google at CVPR 2015 [37], which reconstructs the face classification dataset and proposes Triplets Loss for training. This network is called FaceNet, and its ultimate goal is to embed face images into a 128-dimensional Euclidean space and get a better matching result by minimizing the Euclidean distance between individual and maximizing the Euclidean distance between different individuals.

Finally, to satisfy the migration of the model to other datasets. We determine the adaptive matching threshold based on a clustering algorithm, and the experimental results prove that the method is simple and feasible with high accuracy.

B. SHORTCOMINGS AND FUTURE WORK

However, in this paper, no special treatment is done for the dataset. A random method is still used for the division of batches. In future work, we can reconstruct the dataset so that the number of data from the same sample and different samples in each batch is approximately the same. This would be

more representative and reliable for the MG calculation, thus further improving the model performance. Meanwhile, the VGG network used for the migration learning in this paper is not the best image recognition network available. There are well-performing networks such as Resnet . The latest results are given by NFNet [44] proposed by DeepMind.

Furthermore, due to the lack of infrared devices, this paper has not yet conducted demo production and application tests on the matching of the model, i.e., its performance in continuous and real-time recognition has not been tested. These aspects need to be further explored and studied.

REFERENCES

[1] D. Zhang, Automated Biometrics: Technologies and Systems, New York, NY, USA:Springer, 2013.

[2] A. K. Jain, A. Ross and S. Prabhakar, "An introduction to biometric recognition", IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 1, pp. 4-20, Jan. 2004.

[3] F. Chen, X. Huang and J. Zhou, "Hierarchical minutiae matching for fingerprint and palmprint identification", IEEE Trans. Image Process., vol. 22, no. 12, pp. 4964-4971, Dec. 2013.

[4] Y. Ding, D. Zhuang and K. Wang, "A study of hand vein recognition method", Proc. IEEE Int. Conf. Mechatronics Autom., pp. 2106-2110, 2005.

[5] X. Wang and X. Tang, "A unified framework for subspace face recognition", IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 9, pp. 1222-1228, Sep. 2004.

[6] M. I. Ahmad, W. L. Woo and S. Dlay, "Non-stationary feature fusion of face and palmprint multimodal biometrics", Neurocomputing, vol. 177, pp. 49-61, Feb. 2016.

[7] Galbally, J., et al. (2014). "Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition." Ieee Transactions on Image Processing 23(2): 710-724.

[8] Lu, G. M., et al. (2003). "Palmprint recognition using eigenpalms features." Pattern Recognition Letters 24(9-10): 1463-1467.

[9] D.-S. Huang, W. Jia and D. Zhang, "Palmprint verification based on principal lines", Pattern Recognit., vol. 41, no. 4, pp. 1316-1328, Apr. 2008.

[10] X. Wu, D. Zhang and K. Wang, "Palm line extraction and matching for personal authentication", IEEE Trans. Syst. Man Cybern. A Syst. Humans, vol. 36, no. 5, pp. 978-987, Sep. 2006.

[11] Jain, A. K. and J. J. Feng (2009). "Latent Palmprint Matching." Ieee Transactions on Pattern Analysis and Machine Intelligence 31(6): 1032-1047.

[12] Zhao, S. P. and B. Zhang (2022). "Joint Constrained Least-Square Regression With Deep Convolutional Feature for Palmprint Recognition." Ieee Transactions on Systems Man Cybernetics-Systems 52(1): 511-522.

[13] Trabelsi, S., et al. "Efficient palmprint biometric identification systems using deep learning and feature selection methods." Neural Computing & Applications.

[14] Shin, H. C., et al. (2016). "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning." Ieee Transactions on Medical Imaging 35(5): 1285-1298.

[15] Kiselev, V. Y., et al. (2019). "Challenges in unsupervised clustering of single-cell RNA-seq data." Nature Reviews Genetics 20(5): 273-282.

[16] Yosinski, J., et al. (2014). "How transferable are features in deep neural networks?" Advances in neural information processing systems 27.

[17] He, K., et al. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.

[18] Simonyan, K. and A. Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.

[19] Pizer, S. M., et al. (1987). "Adaptive histogram equalization and its variations." Computer vision, graphics, and image processing 39(3): 355-368.

[20] LeCun, Y., et al. (1998). "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86(11): 2278-2324.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. IEEE Computer Vision and Pattern Recognition (CVPR), 2009.

[22] Pan, S. J., et al. (2010). "Domain adaptation via transfer component analysis." IEEE transactions on neural networks 22(2): 199-210.

[23] Vaswani, A., et al. (2017). "Attention is all you need." Advances in neural information processing systems 30.

[24] Choi, M., et al. (2020). Channel attention is all you need for video frame interpolation. Proceedings of the AAAI Conference on Artificial Intelligence.

[25] Kingma, D. P. and J. Ba (2014). "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980.

[26] Paszke, Adam and Gross, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam (2017). "Automatic differentiation in PyTorch." NIPS-W

[27] Vinoth, R. and C. Venkatesh (2018). Segmentation and Detection of Tumor in MRI images Using CNN and SVM Classification. 2018 Conference on Emerging Devices and Smart Systems (ICEDSS), IEEE.

[28] Huang, X., et al. (2019). "Identifying disaster related social media for rapid response: a visual-textual fused CNN architecture." International Journal of Digital Earth.

[29] N. Duta, A.K. Jain, K.V. Mardia Matching of palmprints Pattern Recognition Letters, 23 (4) (2002), pp. 477-485

[30] W. Shu, D. Zhang Automated personal identification by palmprint Optical Engineering, 38 (8) (1998), pp. 2359-2362

[31] Kong, A., et al. (2009). "A survey of palmprint recognition." Pattern Recognition 42(7): 1408-1418.

[32] Candès, E. J. and B. Recht (2009). "Exact matrix completion via convex optimization." Foundations of Computational mathematics 9(6): 717-772.

[33] Jia, Wei & Xia, Wei & Zhao, Yang & Min, Hai & Chen, Yan-Xiang. (2021). 2D and 3D Palmprint and Palm Vein Recognition Based on Neural Architecture Search. International Journal of Automation and Computing. 18. 10.1007/s11633-021-1292-1.

[34] A. Genovese, V. Piuri, K. N. Plataniotis, F. Scotti. PalmNet: Gabor-PCA convolutional networks for touchless palmprint recognition. IEEE Transactions on Information Forensics and Security, vol.14, no.12, pp.3160–3174, 2019.

[35] Han, W.-Y. and J.-C. Lee (2012). "Palm vein recognition using adaptive Gabor filter." Expert Systems with Applications 39(18): 13225-13234.

[36] Weldon, T. P., et al. (1996). "Efficient Gabor filter design for texture segmentation." Pattern Recognition 29(12): 2005-2015.

[37] Schroff, F., et al. (2015). Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on computer vision and pattern recognition.

[38] Fanjiang, Y. Y., et al. (2021). "Palm Vein Recognition Based on Convolutional Neural Network." Informatica 32(4): 687-708.

[39] Wilson, C. (2010). Vein pattern recognition: a privacy-enhancing biometric, CRC press.

[40] Wu, K. S., et al. (2013). "A secure palm vein recognition system." Journal of Systems and Software 86(11): 2870-2876.

[41] Faruqe, M. O. and M. A. M. Hasan (2009). Face recognition using PCA and SVM. 2009 3rd international conference on anti-counterfeiting, security, and identification in communication, IEEE.

[42] Baldi, P. and P. J. Sadowski (2013). "Understanding dropout." Advances in neural information processing systems 26.

[43] PolyU Palmprint database,http://www.comp.polyu.edu.hk/ biometrics/

[44] Brock, A., et al. (2021). High-performance large-scale image recognition without normalization. International Conference on Machine Learning, PMLR.

[45] MacQueen, J. (1967). Classification and analysis of multivariate observations. 5th Berkeley Symp. Math. Statist. Probability.