An Effective Method For Finger-Vein Verification Using Convolutional Auto-Encoder And SVM Method

N Vijila[1], X Binisha[2], H Riyaz Fathima[3]
[1, 2, 3] PET Engineering College

Abstract- *This project presents a novel deep learning-based method that integrates a Convolutional Auto-Encoder (CAE) with support vector machine (SVM) for finger vein verification. The CAE is used to learn the features from finger vein images and the SVM is used to classify finger vein from these learned feature codes. The CAE consists of a finger vein encoder, which extracts high-level feature representation from raw pixels of the images, and a decoder which outputs reconstruct finger vein images from high-level feature code. As an effective classifier, support vector machine (SVM) is introduced in this project to classify the feature code which is obtained from CAE. Experiments prove that the proposed deep learning-based approach has superior performance in learning features than traditional method without any prior knowledge, presenting a good potential in the verification of finger vein.*

Keywords*-* Biometrics; finger-vein; deep learning; convolutional auto-encoder ; support vector machine .

I. INTRODUCTION

Automatic personal verification using biometrics has received widespread attention and increasing demand for border control, access control, attendance systems and financial security. Biometrics is a highly safe and reliable technique, which utilizes physical or physiological characteristics to perform personal verification. Up to data, various types of biometrics have been proposed, including face, iris, fingerprint, palm print, posture, ECG, finger vein and so on.

As an intrinsic biometric method, finger vein is much harder to forge as it is located inside the subcutaneous layer. As compared to other biometric technologies, finger vein verification has some advantages: (1) Fast: small image data for fast processing (2) Non-contact: not affected by skin conditions. (3) Biometric: finger vein patterns can be identified on the living body. (4) High safety: vein pattern is internal features, not easy to replicate.

Like other biometric verification method, a finger-vein- based verification method mainly has four steps, data acquisition, data pre-processing, feature extraction and data verification. Usually, data pre-processing consists of several steps and they are image normalization, region of interest (ROI) extraction and Image enhancement. Finger vein verification methods can be divided into two categories: template matching and feature classification. In data pre-processing, various operations on finger vein are performed such that finger vein structure is enhanced and ROI images are extracted from the original images. After feature extraction, suitable features from the veins are extracted and then verified in data verification.

In data pre-processing, various filters and transforms has been introduced in finger vein enhancement, such as Gabor filter, and Curvelet transform. Gabor filters was introduced in finger-vein image enhancement and then the Phase-Only-Correlation (POC) measure method was used to verify finger-vein image. While in, a Gabor filter-based finger-vein feature extraction approach was proposed, where a set of Gabor filters was introduced to exploit the finger-vein features at different orientations and scales, and then a finger-vein code was constructed through these features for identification. Based on Curvelet transform, a multiscale enhancement method was used to perform on the finger-vein image, and then a local interconnection structure neural network is applied to identify finger-vein images. However, these enhancement methods are not self-adaptive to variable width veins, while noise also gets enhanced when it presents closely resembles the vein structure. As a self-adaptive enhancement method, a repeated line tracking method based on valley feature detection was proposed to extract the vein network by calculating difference between the center pixel value and the pixel value in the corresponding range. And in, the repeated line tracking method has been modified by calculating the maximum curvature information during vein tracking. Furthermore, a sliding window-based image enhancement method was proposed to enhance finger vein network, and such a method was improved using a dual-sliding window. But it is still observed that thin veins are not effectively enhanced due to small width.

After image enhancement, features were extracted for data verification, such as random transform, local line binary pattern, Pattern map, PCA and hybrid methods. For example, random transform was introduced, while random transform was used to extract features and then used as input to an artificial neural network classifier for image identification. A finger-vein pattern matching approach based on Local Line Binary Pattern (LLBP) was proposed in, where the LLBP features were extracted and then the matching scores were calculated by Hamming distance. Pattern map based on pixel-pattern-based texture feature (PPBTF), together with principal component analysis (PCA), was also applied for finger- vein recognition, where the PPBTF was used to represent finger vein pattern information, and PCA was introduced to further reduce the dimension of the PPBTFs before sending to a nearest neighbor classifier. As a popular dimensionality reduction and feature extraction technology, PCA and (2D) PCA also introduced in finger vein feature extraction and then trained a neural network for verification, which result in a high recognition rate. Finger vein feature extraction was also accomplished by various ways. Combined PCA and LDA, a hybrid finger vein recognition method was proposed, PCA and LDA were applied to extract features and SVM model was trained for recognition. In, local binary pattern was introduced to extract local features of finger vein images while wavelet transform was used to collect global features. However, these feature extraction approaches need to assume shape and structure of the finger vein conform to a particular pattern, such as the distribution of valleys and segments, and it is not easy to design an effective finger vein feature description model.

II. LITERATURE SURVEY

Biometrics is an automatic user authentication technology, which uses human physiological and/or behavioral characteristics with several desirable properties like universality, distinctiveness, permanence and acceptability. In recent years, there has been an increasing interest in finger vein recognition, which is motivated by the advantages of finger vein in living-body identification, noninvasive and non-contact image capture, and high security over other biometric recognition techniques Soft biometric trait has been used as ancillary information to enhance the recognition accuracy for face, fingerprint, gait, iris, etc. In this paper, we present a new investigation of soft biometric trait to improve the performance of finger vein recognition. We first propose some extraction criteria of soft biometric trait for comprehensively understanding this kind of ancillary information. And then based on these criteria, the width of phalangeal joint is employed as a novel soft biometric trait, which can be directly extracted from finger vein image.

Automated human identification using physiological and/or behavioural characteristics, i.e. biometrics, is increasingly mapped to new civilian applications for commercial use. The tremendous growth in the demand for more user friendly and secured biometrics systems has motivated researchers to explore new biometrics features and traits. The anatomy of human fingers is quite complicated and largely responsible for the individuality of fingerprints and finger veins. Several biometrics technologies are susceptible to spoof attacks in which fake fingerprints, static palmprints, static face images can be successfully employed as biometric samples to impersonate the identification. This paper presents a new approach to improve the performance of finger vein identification systems presented in the literature. The proposed system simultaneously acquires the finger vein and low resolution fingerprint images and combines these two evidences using a novel score level combination strategy. We examine the previously proposed finger vein identification approaches and develop a new approach that illustrates it superiority over prior published efforts. The utility of low resolution fingerprint images acquired from a webcam is examined to ascertain the matching performance from such images. We develop and investigate two new score level combinations, i.e., holistic and nonlinear fusion, and comparatively evaluate them with more popular score level fusion approaches to ascertain their effectiveness in proposed system. The rigorous experimental results presented on the database of 6,264 images from 156 subjects illustrate significant improvement in the performance, both from the authentication and recognition experiments.

Finger-vein biometrics has been extensively investigated for personal verification. Despite recent advances in finger vein verification, current solutions completely depend on domain knowledge and still lack the robustness to extract finger-vein features from raw images. This paper proposes a deep learning model to extract and recover vein features using limited a priori knowledge. Firstly, based on a combination of known state of the art handcrafted finger-vein image segmentation techniques, we automatically identify two regions: a clear region with high separability between finger-vein patterns and background, and an ambiguous region with low separability between them. The first is associated with pixels on which all the segmentation techniques above assign the same segmentation label (either foreground or background), while the second corresponds to all the remaining pixels. This scheme is used to automatically discard the ambiguous region and to label the pixels of the clear region as foreground or background. A training dataset is constructed based on the patches centered on the labeled pixels. Secondly, a Convolutional Neural Network (CNN) is trained on the resulting dataset to predict the probability of each pixel of being foreground (i.e. vein pixel) given a patch centered on it. The CNN learns what a finger vein pattern is by learning the difference between vein patterns and background ones. The pixels in any region of a test image can then be classified effectively. Thirdly, we propose another new and original contribution by developing and investigating a Fully Convolutional Network (FCN) to recover missing finger vein patterns in the segmented image. The experimental results on two public finger-vein databases show a significant improvement in terms of finger-vein verification accuracy.

Finger vein recognition has received a lot of attention recently and is viewed as a promising biometric trait. In related methods, vein pattern-based methods explore intrinsic finger vein recognition, but their performance remains unsatisfactory owing to defective vein networks and weak matching. One important reason may be the neglect of deep analysis of the vein anatomy structure. By comprehensively exploring the anatomy structure and imaging characteristic of vein patterns, this paper proposes a novel finger vein recognition framework, including an anatomy structure analysis-based vein extraction (ASAVE) algorithm and an integration matching strategy. Specifically, the vein pattern is extracted by the orientation map-guided curvature based on the valley- or half valley-shaped crosssectional profile. In addition, the extracted vein pattern is further thinned and refined to obtain a reliable vein network. In addition to the vein network, the relatively clear vein branches in the image are mined from the vein pattern, referred to as the vein backbone. In matching, the vein backbone is used in vein network calibration to overcome finger displacements. The similarity of two calibrated vein networks is measured by the proposed elastic matching and further recomputed by integrating the overlap degree of corresponding vein backbones. Extensive experiments on two public finger vein databases verify the effectiveness of the proposed framework.

Biometric techniques have attracted increased attention in various applications where correct identity assessment is crucial. In general, by taking full advantage of Intrinsic physiological or extrinsic behavioral characteristics of humans for robust personal authentication, biometric techniques have shown significant advantages over traditional authentication mechanisms such as passwords, keys, personal identification numbers, and smart cards Finger vein recognition is an emerging biometric technique for personal authentication that has garnered considerable attention in the past decade. Although shown to be effective, recent studies have revealed that finger vein biometrics is also vulnerable to presentation attacks, i.e., printed versions of authorized individual finger vein images can be used to gain access to facilities or services. In this paper, given that both blurriness and the noise distribution are slightly different between real and forged finger vein images, we propose an efficient and robust method for detecting presentation attacks that use forged finger vein images (print artifacts). First, we use total variation (TV) regularization to decompose original finger vein images into structure and noise components, which represent the degrees of blurriness and the noise distribution. Second, a block local binary pattern (LBP) descriptor is used to encode both structure and noise information in the decomposed components. Finally, we use a cascaded support vector machine (SVM) model for classification, by which finger vein presentation attacks can be effectively detected. To evaluate the performance of our approach, we constructed a new finger vein presentation attack database.

III. PROPOSED SYSTEM

The This project presents a convolutional auto-encoder (CAE)-based deep learning model for finger-vein verification. The CAE network is new designed and applied to learning feature codes following a certain distribution, which are then processed by SVM for classification.

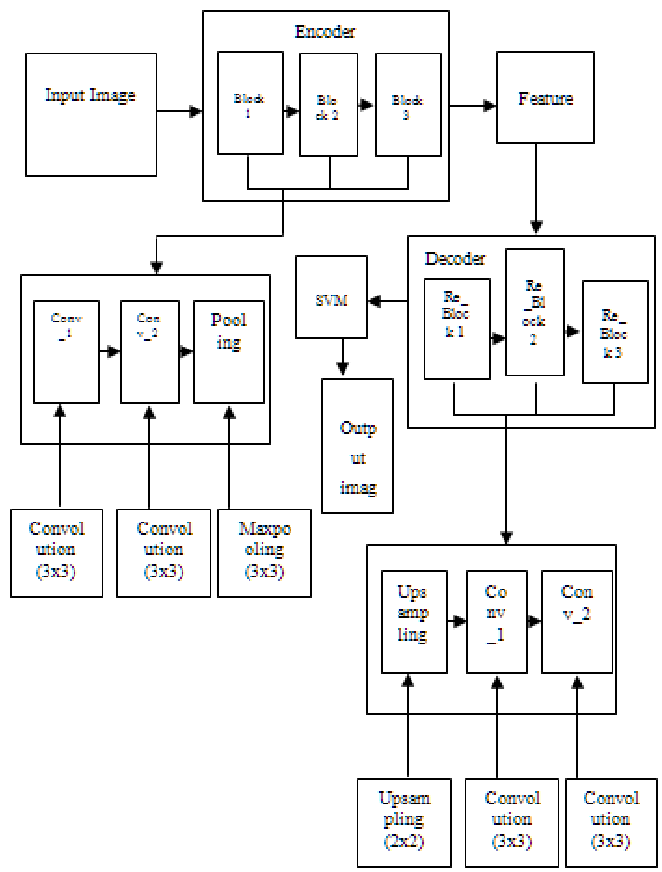


Fig 1 Block Diagram

Convolutional neural network has demonstrated its powerful ability to represent features. It uses multiple filters which shares different parameters to extract image features. In general, CNN includes two kinds of connections: convolution and pooling. Finally support vector machine (SVM) used to classify the feature code which is obtained from CAE. The size of the extracted deep feature codes has been greatly decreased in the proposed method.

CONVOLUTIONAL NEURAL NETWORKS

Successfully applied in the field of computer vision, convolutional neural network has demonstrated its powerful ability to represent features. It uses multiple filters which shares different parameters to extract image features. In general, CNN includes two kinds of connections: convolution and pooling as illustrated in fig. 2.

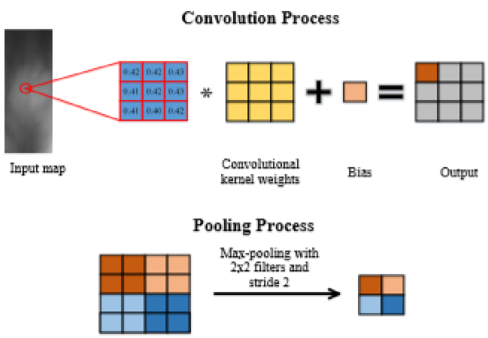


Fig 2 The example of two connections: convolution and pooling

For convolution connection, the features are extracted by performing a two-dimensional convolution of the input map and the convolutional kernel. Here, the map refers to an image with multiple channels. Let x_i^m represent the m-th channel of input map at layer i, the output map y_i^n of n-th at layer i can be represented as

$$y_i^n = ReLU\left(\sum_{m}^{M^{i-1}} w_i^{n,m} * x_i^m + b_i^n\right)$$

Where $\omega in,m$ is the convolutional kernel between the $yin$and xim, the symbol * represents convolutional operation, $Mi{-}1$ is the number of input maps, and the bias of the n-th output map is $bn\ i$. As for the activation function, Rectified Linear Units (ReLU) (y = max (0, x)) is chosen. The rectified linear units (ReLU) instead of traditional sigmoid units is introduced as the activation function of hidden layers, because the ReLU works better on capturing patterns in natural images and improves the ability of neural network on solving image de-noising problem. In pooling connection, the filter response can be reduced to lower dimensions and the input representation can be more compact. In general, there are two kinds of operation: maxpooling and averaging-pooling. Max-pooling decreases the dimension of data simply by taking only the maximum data, while average-pooling works in a similar way by taking the average of inputs instead of maximum. Based on the conceptual difference between the two methods, max-pooling is sensitive to texture information of the image, and average-pooling retains more background information of the image. Therefore, maxpooling is more conducive to extract the feature information of the image. In this study, max-pooling is employed after the ReLU output is calculated.

CONVOLUTIONAL AUTO-ENCODER

An auto-encoder consists of two parts: encoder and decoder as illustrated in Fig. 3.3. The encoder converts the input x to a hidden representation y (feature code) using a deterministic mapping function. Typically, it is an affine mapping function followed by non-linearity:

$$y = f(Wx + b)$$

Where, W is weight between input x and hidden representation y and b is bias.

The decoder implements the process of reconstructing the output z by y, which can be expressed as:

$$z = f'(W'y + b')$$

Where W' is weight between hidden representation y and output z and b' is bias. Similar to the input x, z is considered as the reconstruction of x.

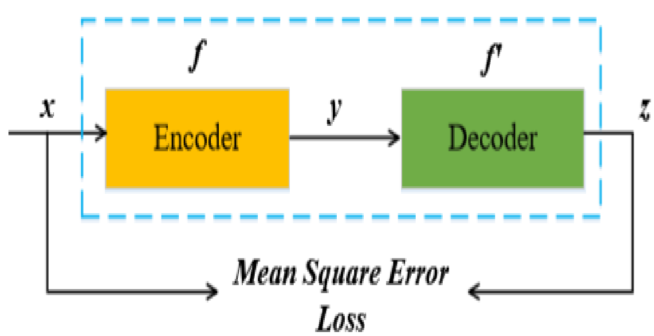


Fig 3 The architecture of an Auto-Encoder

The principle of training an auto-encoder is to minimize the reconstruction error, which can be realized by minimizing the following cost function J_{AE}

$$J_{AE} = \frac{1}{p}\sum_{i=1}^{p} L[x_i, y_i]$$

where p is the number of input images, xi is the i-th input image, and zi is the reconstructed image corresponding to xi. $L[xi,zi]$ represents reconstruction error of the input image xi, which can be measured by mean square error or cross entropy. In this study, the mean square error between the input image xi ,(i = 1,2,⋯p) and the reconstructed patch of image zi (i = 1,2,⋯p) is used. Correspondingly, $L[xi,zi]$can be expressed as:

$$LAE[xi,zi] = \|xi - zi\|^2$$

Convolutional auto-encoder combines the local convolution connection with the auto-encoder, which is a simple step that adds convolution operation to inputs. Correspondingly, a convolutional auto-encoder is consisted of convolutional encoder and convolutional decoder. The convolutional encoder realizes the process of convolutional conversion from the input to feature maps, while convolutional decoder implements the convolutional conversion from feature maps to the output. In CAE, the extracted features and the reconstructed output are calculated through CNN.

$$y = ReLU(\omega x + b)$$

$$z = \mathrm{ReLU}(\omega' y + b')$$

where ω represents the convolutional kernel between the input and the code y, ω' represents the convolutional kernel between the code y and the output. b and $b'$are bias. Moreover, the parameters of the encoding and decoding operations can be computed using unsupervised greedy training.

SUPPORT VECTOR MACHINE

SVM, which is derived from learning theory, has been widely applied and has the advantage in automatic complexity control to avoid over-fitting. The main conception of SVM is to find a hyperplane in a high-dimensional space by maximizing the minimum distance between the hyperplane and training samples or by separating the training samples of each class. Originally designed for binary classification, one-against-one SVM training scheme is proposed to deal with the multi-class classification problem. The multiclass classification problem can be decoupled to several two-class classification problems and a voting strategy is introduced. Every binary classification is considered to be a voting and the maximum number of votes decides the class of sampling. The SVM is defined as:

$$f(ycode) = sign[\omega T\varphi(ycode) + b]$$

where ω is weight vector and b is a bias, and $\varphi(ycode)$ represents the feature vector that maps feature $ycode$ into the high dimension feature space. Given a training sample ($ycode^i$,$Lcode^i$), the function margin can be defined as:

$$\hat{\gamma}_i = Lcode^i(\omega T ycode^i + b)$$

In order to find the maximum geometric margin γ, the following optimization problem is proposed:

$$max_{\hat{\gamma},\omega,b} \frac{\hat{\gamma}}{\|\omega\|}$$

$$s.t.\ \ Lcodei(\omega^T ycode^i + b) \geq \hat{\gamma}, i = 1,\ldots,m$$

Through using the kernel function, SVMs can learn in the high dimensional feature space without having to explicitly find feature vector $\varphi(ycode)$. There are three basic SVM kernels, including linear, poly and RBF. Linear kernel is the simplest kernel function, Poly kernel is a non-stationary kernel, and the polynomial degree is 2 in this paper. The RBF kernel, radial basis function kernel, is the most widely used kernels and usually has good performance. The kernel functions are shown as:

Linear kernel: $K(x,y) = x^T y + c$
Poly kernel : $K(x,y) = (\alpha x^T y + c)^2$
RBF kernel : $K(x,y) = \exp(-\gamma\|x - y\|^2)$

where, x, y refer to $ycode$ in this paper, the c is an optional constant,α and $\gamma$are adjustable parameters. In the experiments, three basic SVM kernels have been used, and the performance is evaluated based on classification accuracy and equal error rate.

METHODOLOGY

By taking advantage of CAE and SVM, a CAE-based deep feature learning integrated with SVM for finger vein verification is proposed in this paper, where the CAE network is used to learn features from the finger vein image, and then sent to SVM classifier to identify different finger veins.

Specifically, the finger vein image x is inputted to the CAE. Convolution operation is used to extract the image features, and max pooling worked as down-sampling to obtain the feature code y, which is m-channel (m is the number of convolutional kernels) probability maps, representing finger vein features with the values between 0 and 1. The output image z is then reconstructed through a combination of convolution and upsampling operations. The target is to minimize the mean square error between the output image z and the input image x. The smaller the loss function value is, the more similar the output image z will be to the input image x. It also shows that the feature code y can well represent the image x. After extracting finger vein features, SVM is used for classification. The input to the SVM comes from the feature code y. The SVM classifier has been widely used in pattern classification for same problem domain and achieved good results.

After image acquisition, ROI images were extracted from finger vein images. The ROI of the finger vein has been provided in FV_USM dataset, while in SDUMLA dataset the method mentioned has been introduced to extract the ROI image. For the purpose of reducing the computational cost, the image size is normalized to 48x144x3 pixels in this experiment for these two datasets. And pixel scale of the image is also normalized and contrast enhanced to eliminate the effects of uneven brightness. If a pixel belongs to the vein pattern, the gray level of the pixel is similar to other part of the same vein pattern. Through normalization and contrast enhancement, the finger vein details are more clearly which more reliable for training.

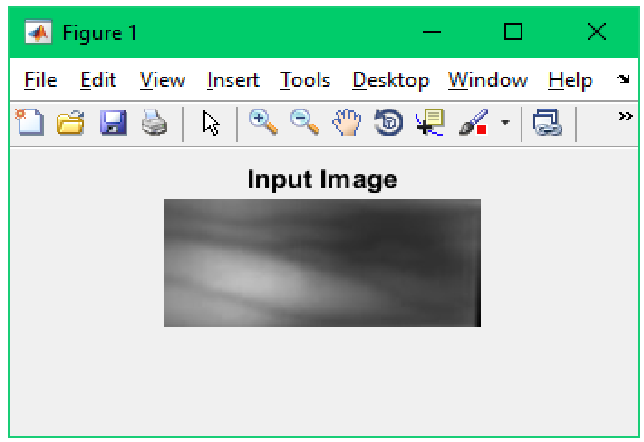
Then data augmentation method is also applied in the experiments. Condition information, such as different orientation, location, scale, brightness, is added to the underlying data through data augmentation. In this project, data augmentation methods, such as flip, rotation, shift, shear and zoom, were randomly introduced to these two datasets, and thus the training samples of each class has been increased after data augmentation. For FV-USM dataset, there are 123x4 classes and the number of samples of each class has been increased from 12 to 60, while there are 106x6 classes in SDUMLA datasets and the number of samples of each class has been increased from 6 to 60.

Firstly, images of dataset are used to train the CAE, and then features learned are employed in SVM classification. To verify the proposed method, ten-fold cross-validation approach is employed in this work. For ten-fold cross-validation approach, the total datasets are randomly divided into ten equal parts. Nine out of ten portions are used for training and the rest (one-tenth) of the data is used for testing. This procedure is repeated 10 times by rotating the test data. Then, the performance measure (accuracy, and equal error rate) is evaluated in each approach. The average of all ten-folds measures the overall performance of the proposed method.

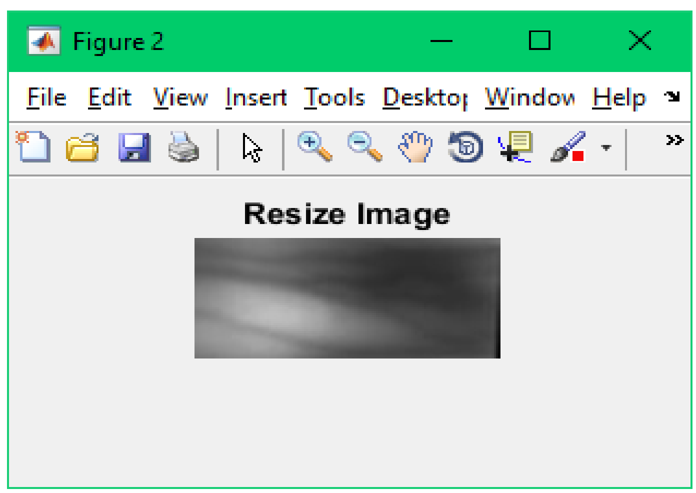
In order to obtain good verification performance, it is important to choose appropriate parameters for the proposed network structures.

A wide range of experiments are conducted to test the effect of network structure parameters on the performance of finger vein verification. Two measures, RR and EER, are used to quantify the verification performance. RR means the rank-1 recognition rate of the dataset in this study. EER is the abbreviation of equal error rate, which is related to False Rejection Rate (FRR) and False Acceptance Rate (FAR). FRR is calculated by genuine scores while the FAR is computed by impostor scores.

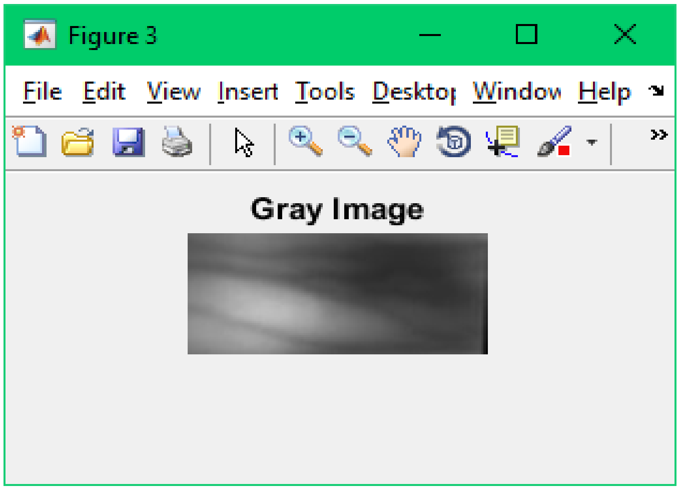
IV. SCREEN SHOTS



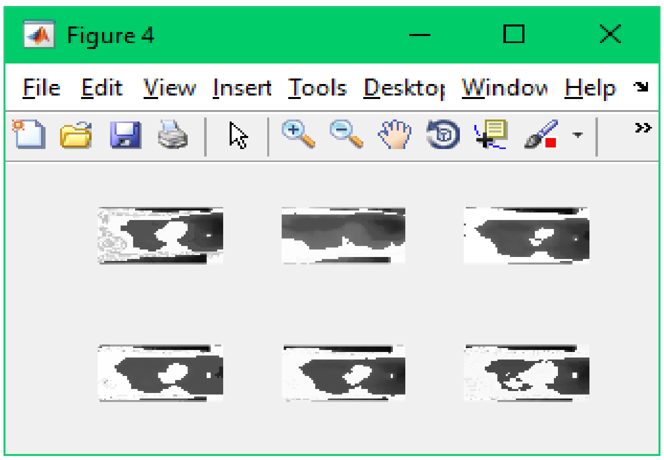
Input Image

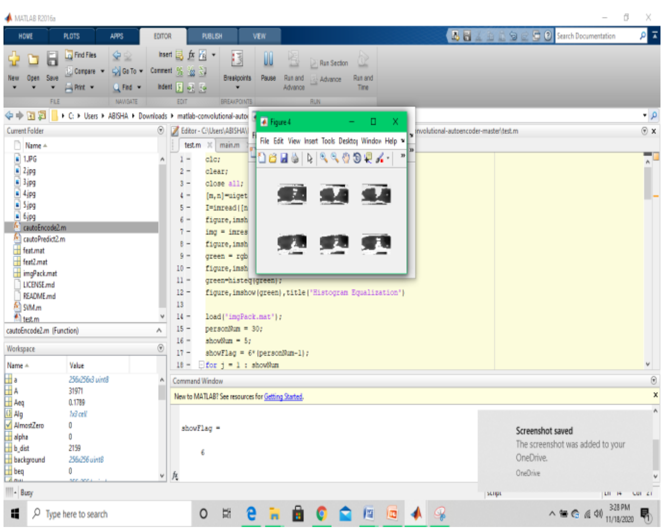


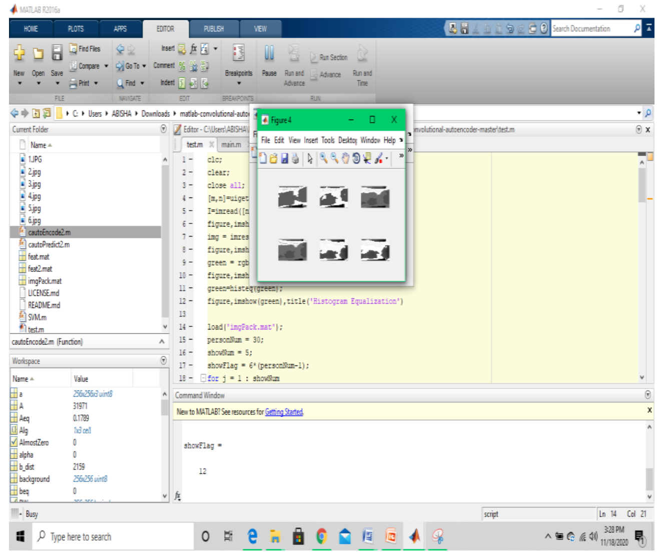
Resize Image

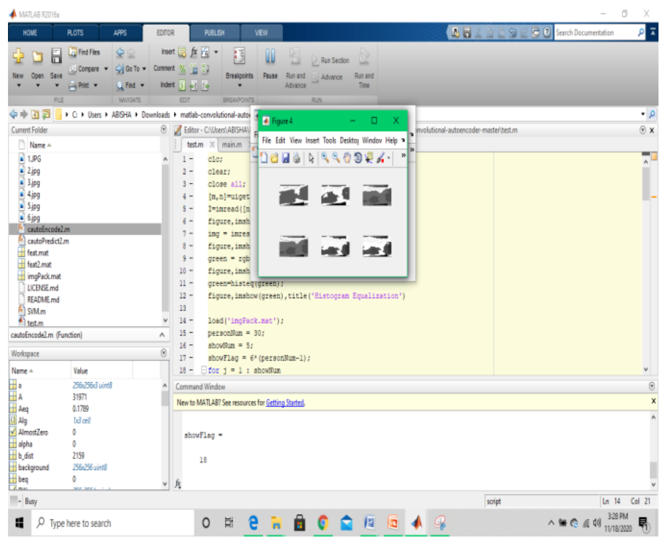


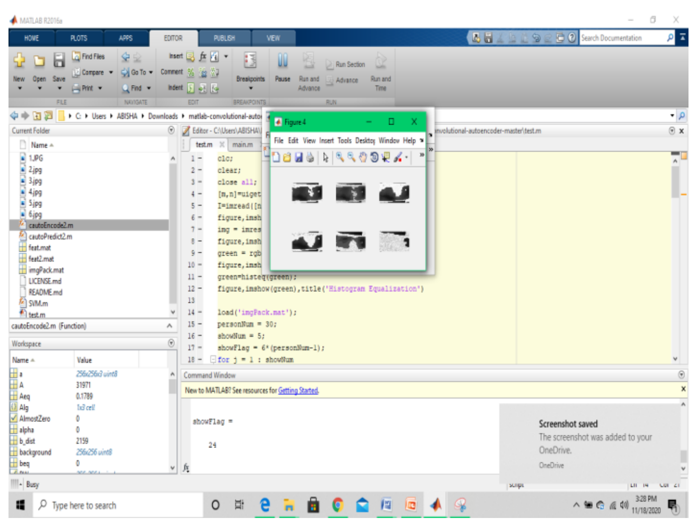
Gray Image



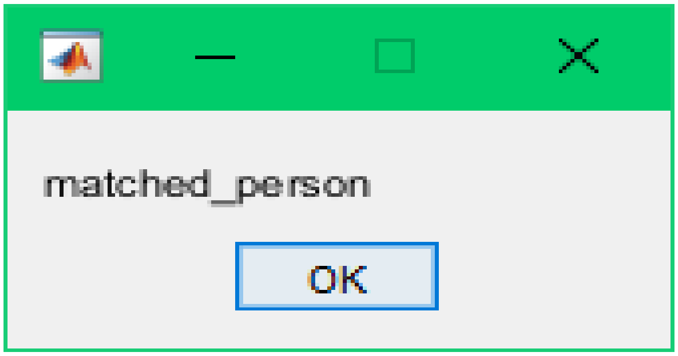








Feature Values



SVM Recognition Result

V. CONCLUSION

Thus, a deep learning-based approach is presented in verification of finger vein. Data augmentation method is applied in this paper to increase the number and the diversity of data. The convolutional auto-encoder can effectively learn finger vein features, preserve the main information of the image, reduce redundant information, and improve the recognition efficiency with the help of SVM classifier. Experimental study on the FV_USM and SDUMLA dataset using the proposed method has shown that, in comparison with other methods in the literature, the proposed model has better performance, and can work more accurately and effectively. The information of the finger vein images has been further compressed, and it makes the proposed method more advantageous for practical applications. Further research can be extended to reduce the response time of finger vein verification when applied to a large database and data generative model will be introduced in data augmentation.

REFERENCES

[1] Al- Aglika Gyaourova, Arun Ross, (2012), "Index Codes for Multibiometric Pattern Retrieval", IEEE Transactions on Information Forensics and Security, vol. 7, No. 2 , April. pp. 518 – 529.
[2] Feifei CUI, Gongping YANG., (2011), "Score Level Fusion of Fingerprint and Finger Vein Recognition", Journal of Computational Information Systems 7: 16 5723-5731.
[3] Huafeng Qin Q., Mounim A and El-Yacoubi., (2018) "Deep Representation for Finger-Vein Image-Quality Assessment", IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, No. 8, Aug, pp. 1677 – 1693.
[4] Hyeon Chang LEE, Byung Jun KANG, Eui Chul LEE, and Kang Ryoung PARK., (2010) "Finger vein recognition using weighted local binary pattern code based on a support vector machine", Journal of Zhejiang University-SCIENCE C (Computers & Electronics) 11(7):514-524.
[5] Kumar A and Zhou Y., (2012) "Human identification using finger images", IEEE Trans. Image Process., vol. 21, no. 4, pp. 2228–2244.
[6] Liukui Chen C., Jing Wang A., Shiyu Yang E and Haibo He F., (2017) "A Finger Vein Image-Based Personal Identification System with Self-Adaptive Illuminance Control", IEEE Transactions on Instrumentation and Measurement vol. 66, No. 2, Feb, pp. 294 - 304.
[7] Naoto Miura, Akio Nagasaka and Takafumi Miyatake., (2004) "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification", Machine Vision and Applications 15: 194–203.
[8] Qin H., El-Yacoubi M. A., (2017) "Deep representation based feature extraction and recovering for finger-vein verification," IEEE Trans. Inf. Foren. Sec., vol. 12, no. 8, pp. 1816–1829.
[9] Qiu X., Kang W., Tian S., Jia W and Huang Z., (2017) "Finger vein presentation attack detection using total variation decomposition," IEEE Trans. Inf. Foren. Sec., vol. 12, no. 2, pp. 465–477.
[10] Wencheng Yang, Song Wang and Jiankun Hu., (2019), "Securing Deep Learning Based Edge Finger Vein Biometrics with Binary Decision Diagram", IEEE Transactions on Industrial Informatics, vol. 15, No. 7, Jul. pp. 4244 – 4253.
[11] Wenming Yang, Changqing Hui, Zhiquan Chen., (2018) "FV-GAN: Finger Vein Representation Using Generative Adversarial Networks", IEEE Transactions On Information Forensics And Security, No.28, JUNE. pp. 2512 – 2524.
[12] Yang J., Shi Y and Jia G., (2017) "Finger-vein Image Matching Based on Adaptive Curve Transformation," Pattern Recogn.
[13] Yang L., Yang G., Yin Y. and Xi X., (2014) "Exploring soft biometric trait with finger vein recognition", Neurocomputing, vol. 135, pp. 218 - 228.
[14] Yang L., Yang G., Yin Y. and Xi X., (2018) "Finger Vein Recognition with Anatomy Structure Analysis", IEEE Trans. Circuits Syst. Video. Technol., vol.28, no.8, pp.1892-1905.
[15] Zhi Liua,, Yilong Yinb, Hongjun Wanga, Shangling Songa., (2010), "Finger vein recognition with manifold learning", Journal of Network and Computer Applications 33 275–282.