VOL. 11, NO. 5, MARCH 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

USER IDENTIFICATION SYSTEM BASED ON FINGER-VEIN PATTERNS USING CONVOLUTIONAL NEURAL NETWORK

K. S. Itqan, A. R. Syafeeza , F. G. Gong, N. Mustafa, Y. C. Wong and M. M. Ibrahim
Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, Melaka, Malaysia
E-Mail: syafeeza@utem.edu.my

ABSTRACT

Finger-vein biometric identification has gained attention recently due to its several advantages over fingerprint biometric traits. Finger-vein recognition is a method of biometric authentication that applies pattern recognition techniques based on the image of human finger-vein patterns. This paper is focused on developing a MATLAB-based finger-vein recognition system using Convolutional Neural Network (CNN) with Graphical User Interface (GUI) as the user input. Two layers of CNN out of the proposed four-layer CNN have been used to retrain the network for new incoming subjects. The pre-processing steps for finger-vein images and CNN design have been conducted pm different platforms. Therefore, this paper discusses the method of linking both parts from different platforms using MEX-files in MATLAB. Evaluation is carried out using images of 50 subjects that are developed in-house. An accuracy of an average of 96% is obtained to recognize 1 to 10 new subjects within less than 10 seconds.

Keywords: finger-vein, convolutional neural network, biometric identification.

INTRODUCTION AND RELATED WORKS

Nowadays, a reliable personal recognition system is important to secure individual privacy details and information. Biometric authentication has been widely developed in recent years in order to achieve higher security of privacy details and user information. Several biometric methods have been introduced in recent years such as signature dynamics, typing patterns, voice recognition and facial recognition, etc.

Finger-vein biometrics was introduced in the year 2002 [1]. It possesses several advantages over other biometric traits. Firstly, finger-veins are located beneath the skin, which makes forgery almost impossible compared to fingerprint features that are visible to human eyes. Secondly, finger-veins are also user friendly since the user only needs to place their finger on the capture device, in contrast with iris recognition, in which the brightness of the light causes discomfort to the user's eye [2]. Finger-vein recognition also ensures high performance and less sensitivity towards improper environmental conditions as compared to face recognition in which the performance is sensitive to illumination changes, appearance, facial expressions, poses, etc. [3].

Finger-vein recognition is categorized under complex pattern recognition tasks. One of the candidates for handling the task's complexity is the Convolutional Neural Network (CNN), one of the machine-learning techniques. CNN is a variant of multilayer perceptron (MLP) that possesses build-in invariance. It utilizes 2-dimensional (2D) topology of image data that is robust to any changes of input patterns. This is a first attempt at applying finger-vein recognition using CNN. There are other approaches using machine learning techniques. For example, Zhang *et al*. [4] apply MLP, and an accuracy of 99.87% is obtained on a database of 400 subjects. Wu and Ye [2] used probabilistic neural network technique, with 99% obtained on 20 samples. Wu and Liu [3] apply a support vector machine (SVM) to 10 subjects, with 10 samples each finger resulting in 98% accuracy.

The remainder of this paper is organized as follows. Section II discusses related theories. This is followed by the proposed methodology in section III. Experimental results and analysis are discussed in section IV. Ultimately, the final section concludes the work.

THEORY

Convolutional Neural Network

Convolutional Neural Network (CNN) was proposed by LeCun in 1989 through LeNet-5 architecture. The proposed method has been widely used for several applications such as face detection, face recognition, gender recognition, object recognition, character recognition and texture recognition.

LeNet-5 contains several important features; namely, convolution layers, subsampling layers and two or three fully connected layers. Convolution and subsampling layers are interleaved twice among each other and become the feature extraction layer, while the two or three fully connected layers become the MLP for the classification process. CNN combines segmentation, feature extraction and classification in one trainable module with minimal preprocessing steps. CNN is normally trained with standard backpropagation algorithm.

Finger-Vein image acquisition and preprocessing flow

The veins are captured by placing a finger between an infrared light source and a camera. A near infrared light (NIR) with a wavelength of 760-850 nm is transmitted from the back of the finger, penetrating into skin, while the radiation of light is absorbed by the deoxyhemoglobin [5]. When hemoglobin absorbs light, the finger-vein appears as a pattern of shadows. These vein patterns are enhanced in the preprocessing state of the

design. General preprocessing flow for finger-vein images is shown in Figure-1.

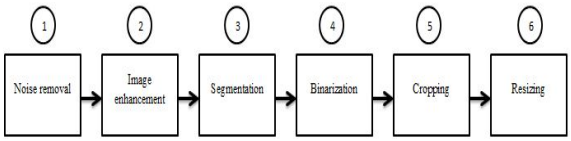


Figure-1. General preprocessing flow in finger-vein biometrics.

METHODOLOGY

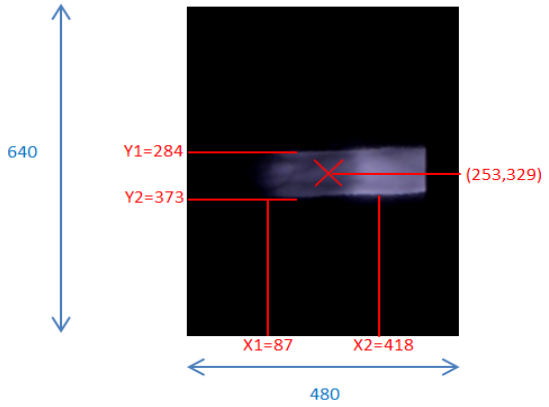
The finger-vein image acquisition detail process is out of the scope of this paper. This section discusses the database used, pre-processing and CNN design.

Preparation of database

The finger-vein images have been collected in-house using 6 different fingers. The participants range from staff to students of the Universiti Teknikal Malaysia Melaka (UTeM). There are 60 participants altogether, with 10 samples from each participant. Hence, there is a total of 600 samples.

Preprocessing in MATLAB

The raw images are in JPEG format, and conversion to grayscale is required to ease the burden of CNN in the subsequent process. All the grayscale images will be stored as .mat files which will be used to merge with the existing database in the same format for later processes. Once the images are converted to grayscale, the next process is to crop the images to the desired size, which is 55 × 67. In the cropping process, there are some calculations required to ensure that the finger-vein images are located correctly. The calculation in the box aids in finding the center point of the finger-vein in the images, and this process ensures that all the images are standard. The cropping process is executed in MATLAB. Figure-2 shows the calculation of cropping a finger-vein image.



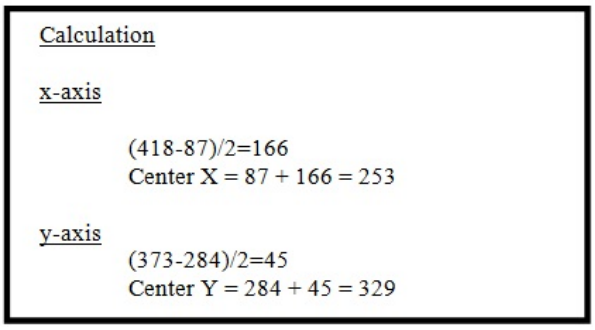


Figure-2. Identifying the target area and pixel location of the finger-vein image.

When the image has been captured, the images must undergo pre-processing steps in MATLAB as shown in the Figure-3. The original image captured is in the size of 640 × 480 pixels and will be cropped to 70 × 130 pixel size. Then, the finger-vein image will be set to the center pixel, and this is fixed for all images since the position of the subject's finger is guided. The image is then resized to 55 × 67 pixels to reduce some information content so as to ease subsequent CNN training.

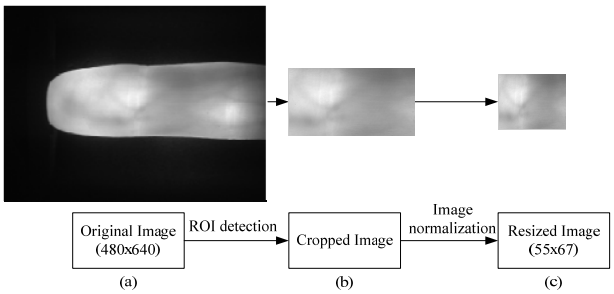


Figure-3. Preprocessing steps for this finger-vein recognition system.

CNN Design

The proposed CNN design for the finger-vein recognition process consists of four layers as shown in Figure-4. The proposed design is derived from LeNet-5 architecture and possesses a smaller neural network size. The convolution and subsampling layers are fused, and two fully connected single nodes are used as the classifier. Detail of the fusion process can be referenced to [6].

The four-layer CNN was used to train the images from 50 subjects. However, whenever there are new incoming subjects, retraining the four-layer CNN will become time consuming. Therefore, one way to reduce the training time is to follow these steps:

- Train the images from 50 subjects. Once optimum accuracy is obtained, save the weights.
- Pass all the image samples for 51-60 subjects and use all the weights from layer C1 to C3 to produce new feature maps at layer C3.
- Now, C3 becomes the new input to the system, and only the last two layers (C3 and output layers) are involved in training the new total subjects.

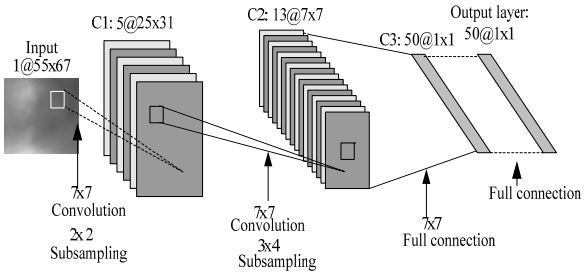


Figure-4. The proposed CNN design

Integrate the preprocessing and CNN design to become a system

- The algorithm for preprocessing is written in MATLAB software, while the algorithm for the CNN is written in Linux-based C language (GCC compiler). Therefore, both of these parts must be linked together in order to make the system operate well. This can be done by using MATLAB executable (MEX) function in MATLAB to call the C program algorithm.

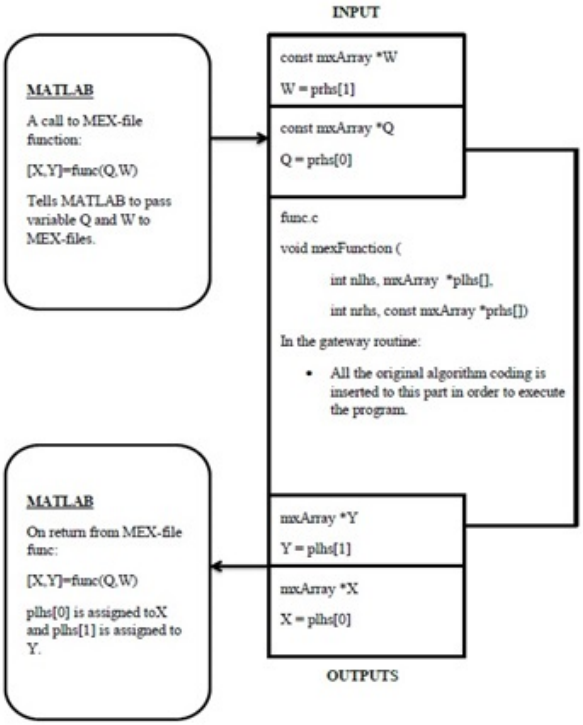


Figure-5. C Mex Cycle.

RESULT AND DATA ANALYSIS

A GUI is designed as shown in Figure-6. The GUI contains several pieces of information such as the number of current subject sand the number of total subjects after a new subject has been added to the system. Once the number of total subjects is confirmed, the image samples of the subjects will go through the preprocessing stage. A total of 80% of the total images become the training samples, while the remaining 20% becomes the test samples. After the preprocessing stage, the samples are trained by the CNN and accuracy is displayed. Once the training process is completed, optimum CNN weights are obtained. These weights are used in the evaluation stage in which an image is selected from the database and the identification process is invoked.

Table-1 shows the time, the number of epochs and the accuracy obtained when the system adds and trains new subjects into the database. The results show that system duration can differ for training different numbers of new subjects.

The results shown in Table-1 can be compared with the results reported in [7]. The same image samples as [7] have been used for the first 50 subjects. The only difference is the platform on which the system is implemented. The recognition result reported in [7] is 100% accuracy for 100 test samples from 50 subjects. However, the result obtained for recognizing the same samples using Windows-based MATLAB platform is 96%. Hence, it can be concluded that by implementing the whole system in Windows-based MATLAB, the recognition accuracy degraded as compared to the Linux-based C language platform implemented in [7].

In addition, the accuracy obtained in Table-1 is due to the low quality of image samples captured by a simple and low-cost finger-vein capture device developed in-house. However, the information about the capture device is out of the scope of this paper. The results obtained in Table-1 are not expected and this proves that type of platform does play a role in affecting recognition accuracy.

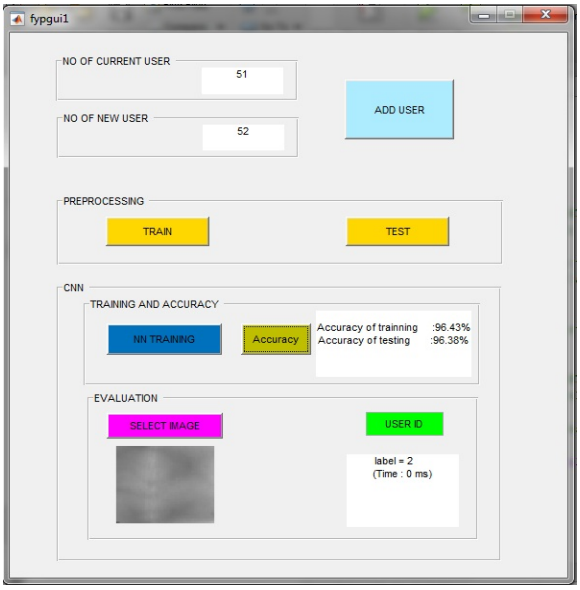


Figure-6. System GUI after adding 1 user.

Table-1. The result of adding 1-10 new subjects.

No. of new subject(s)	Total no. of subjects	No of epochs	Average time to train(s)	Accuracy (%)
1	51	3	5.099	96.39
2	52	6	6.479	96.63
3	53	4	7.746	96.56
4	54	3	7.257	96.50
5	55	6	7.682	96.78
6	56	3	7. 216	96.34
7	57	4	7. 489	96.47
8	58	3	7. 387	96.40
9	59	6	7. 360	96.58
10	60	4	7. 893	96.43

Besides the type of platform, it can also be concluded that only simple pre-processing is required for finger-vein recognition using CNN. The absence of steps 1, 2 and 4 from Figure-1 has proven that CNN can benefit from learning raw image. Figure-7 shows the flow of the system.

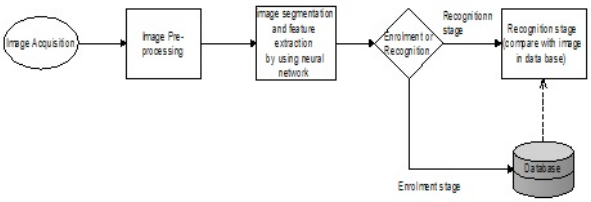


Figure-7. System flow diagram.

CONCLUSIONS

In conclusion, a MATLAB-based finger-vein recognition system using CNN is developed. All of the standalone C program algorithms have been integrated into the system by using MEX-files, a type of MATLAB function. A user-friendly GUI has been developed as an interaction between the user and the system. The result obtained can be improved by enhancing the quality of the image captured. For future work, a finger-vein real-time system is to be developed.

ACKNOWLEDGEMENTS

This work is supported by (UTeM) under the grant PJP/2014/FKEKK(5B)/S01334.

REFERENCES

[1] Dunstone, T. and N. Yager. 2009. Biometric System and Data Analysis. Springer.

[2] Wu, J.-D. and S.-H. Ye. 2009. Driver identification using Finger-Vein Patterns with Radon Transform and Neural Network. Expert Systems with Applications. 36(3): 5793-5799.

[3] Wu, J.-D. and C.-T. Liu. 2011. Finger-Vein Pattern Identification using SVM and Neural Network Technique. Expert Systems with Applications. 38(11): 14284-14289.

[4] Zhongbo, Z., M. Siliang, and H. Xiao. 2006 Multiscale Feature Extraction of Finger-Vein Patterns Based on Curvelets and Local Interconnection Structure Neural Network. 18th International Conference on Pattern Recognition (ICPR).

[5] Miura, N., A. Nagasaka, and T. Miyatake. 2004. Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. Machine Vision and Applications,. 15(4): 194-203.

[6] Syafeeza, A., *et al*. 2014. Convolutional Neural Network for Face Recognition with Pose and Illumination Variation. International Journal of Engineering & Technology. 6(1): 0975-4024

[7] Radzi, S.A., *et al*. 2014. Finger-vein biometric identification using convolutional neural network. Retrieved from file:///C:/Users/user/Downoads/LK-1311-43_manuscript_2. pdf.