Vol. 6, No. 1 | January – June 2022

Finger-Vein Image Dual Contrast Adjustment and Recognition Using 2D-CNN

Noroz Khan Baloch[1], Saleem Ahmed[2], Ramesh Kumar[2], DM Saqib Bhatti[3], Yawar Rehaman[4]

Abstract:

The Recommended steps improve the low contrast of the finger-vein image by employing dual contrast adaptive histogram equalization (DCLAHE) for visual attributes. The dual CLAHE improves image histogram and it is split up throughout the image pixels. The dataset containing finger-vein images is from the SDUMLA-HMT finger-vein database. Following the deployment of DCLAHE. The new version of the dataset is used to detect objects by employing an improved 2D-CNN method. The learning of features in the proposed model is based on optimizing the values of a preprocessed dataset. The proposed model produces 95.35% accuracy.

Keywords: *adaptive histogram equalization; convolutional neural network.*

1 Introduction

At present, the recognition systems are based on biometric technologies which can be used for various security purposes like computer access, banks, phones, ATMs. Such recognition approaches are known as finger-vein, iris, fingerprint, iris, [1]-[3]. In recent years, the recognition system which got much attention is known as the finger-vein recognition system due to vein patterns under the skin are distinct makes it a suitable method to be used for human classification based on the biometrics method [3]. The previous approaches for finger-vein recognition are based on highlighting the lines from the image or it improves the quality of the image and then performing feature extraction [4]. The vein of the fingers is usually extracted by using near-infrared (NIR) light and a charged coupled device (CCD) camera beneath the finger to capture veins is used for image acquisition [5]. The first step is to normalize data as a preprocessing requirement. Therefore, each image of 240×320×3 is transformed into single dimension grayscale image of 128×160×1 before feeding into preprocessing unit and then forwarded to 2D CNN model [5]. Hence finger-vein area becomes darker on the palmer side due to CCD camera on the palmar side of the finger and also due to the penetration of NIR on the dorsal side [6].

There are several datasets to be used for the finger-vein detection method. We employed one of the most common databases known as SDUMLA-HMT [7] to analyze the feasibility of the proposed approach for finger vein recognition. Few of the known datasets are

[1]Dept. of Electronics Engg. Dawood University of Engineering and technology Karachi, Pakistan.
[2]Dept. of Computer System Engg. Dawood University of Engineering and technology Karachi, Pakistan..
[3]Dept. of Telecommunication Engg. Dawood University of Engineering and Technology Karachi, Pakistan.
[4]Dept. of Electronic Engg., NED UET Karachi, Pakistan.

MMCBNU_6000, HKPU-FV, THU-FV UTFV, THU-FV [8]-[11]. There are several approaches used to extract the finger vein part from image background like Gabor filters (GF), local binary pattern (LBP), and local derivative pattern (LDP) [12]-[14]. However, these methods' performance is lower due to limitations in familiar scenarios such as rotation during the acquisition time of the finger vein. Therefore, for such reason the classification of finger-vein images is performed by employing Artificial Intelligence techniques known as convolutional neural network (CNN). CNN/ConvNet is deep learning (DL) method that has the advantage of more accurate and rapid results compared to traditional algorithms [1].

In our proposed method we employ Dual CLAHE [7] based preprocessed dataset and then fed them to CNN model which employs 2D-CNN model for classification. Prior to classification, our proposed model uses Sobel edge detector and polygonal region of interest (ROI) methods to extract the background of the finger region. Therefore, we employ histogram equalization to enhance the performance of the proposed method. The histogram illustrates the grayscale distribution of pixels in an image and distributes grey values in a whole image [16]. In our proposed method we employ dual contrast limited adaptive histogram equalization (CLAHE) to increase grey values in an image. The output images of dual CLAHE are prearranged as classes concerning their folder names. Then these finger vein images are fed into the proposed 2D-CNN model. As compared to traditional approaches the 2D-CNN method extracts the finger-vein from image datasets by convolving different filters on a large scale of the dataset by learning features and recognizing them without extracting features from the background. It is observed that 2D CNN can produce results of finger vein detection rate reaching near 95%.

The paper is organized as follows. Section 2 contains the work done in the field of finger-vein detection. Then the proposed model and relevant approaches such as background extraction, Dual CLAHE, and 2D CNN architecture are provided in section 3. In section 4, the simulation results and model and its parameters are provided and a conclusion is given in section 5.

2 Related work on Finger-vein Detection

Previous research on finger vein detection is based on feature extraction and preprocessing [1]-[4]. The Gabor filters are used in order to extract different shapes and directions to detect the finger-vein pattern [17]. In [12], the author used a local derivative pattern (LDP) [12] and a local binary pattern (LBP) [18] to extract finger–vein patterns. By using the Gabor filter, the SIFT features of finger vein patterns are matched in the finger-vein classification method proposed in [19]-[20]. The distinguishable features are generated by applying the principal component analysis (PCA) technique for feature extraction [21]. Other few notable techniques for finger-vein feature extraction are SVM, HOB, and sparse representation techniques [23].

Instead of above mentioned traditional schemes, researchers are putting efforts to utilizes deep learning for finger-vein detection for better performance in the finger-vein detection [1]-[2]. In [24], CNN based approach with four-layer employing fused convolutional sampling is used for finger vein detection where they propose to reduce the complexity by applying mentioned approach. In [25], a seven-layer CNN model with five convolution layers and two fully connected layers is proposed. The CNN with multimodal finger fusion method is proposed in [26]. They also worked on the uneven size for feature extraction by applying PCA based dimensionality reduction method. However, conventional methods suffer from a complexity-performance tradeoff.

3 The Proposed Finger Vein Detection Method

3.1 Overview

Figure 1 shows the proposed finger vein detection method contains six steps. First, we read the images from an available database used in our study SDUMLA-HMT [7], which is an open source dataset that can be used for study based on finger-vein images. Each image has a size of 240×320×3 is normalized in the second step and it became 128×160×1 size image as shown in Fig. 2(a & b), respectively. Fig. 2(c) contains the background separation of an image which is the third step of our proposed method. The polygonal ROI and Sobel edge detector are used for finger region extraction. Then the dual CLAHE is used for yielding dark vein ridges. In the last step, the images are passed through 2D-CNN model. Table 1 shows the 2D CNN architecture containing 24 layers.

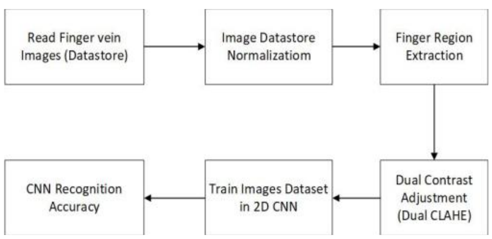


Fig. 1. Steps involved in the proposed scheme.

3.2 Dual CLAHE

Due to the low contrast of the finger vein images, it becomes difficult to separate the finger vein from the background or other regions. To resolve this issue, we employed the dual CLAHE technique on the finger-vein image. In the dual CLAHE method, first, the finger-vein image is cropped from the dark region. After cropping the image, we employ dual CLAHE to enlighten the vein region twice as shown in Fig. 3. CLAHE is derived from Adaptive Histogram Equalization (AHE) [28]. AHE is employed on small areas of the contextual region in the image known as tiles 2×2 on both CLAHE. With clipping limits of 0.03 and 0.04 on the first and second, respectively having the CLAHE with the same exponential distribution [7] [27].

$$N_{avg} = \frac{N_r X \times N_r Y}{N_{gray}} \tag{1}$$

Where N_avg is the average number of pixels, and a number of pixels in the X & Y directions are shown as N_r X & N_r Y, respectively, and N_gray is the number of gray-scale pixels.

The CL (clipped limit) is used to limit the values at a certain specified level in a grayscale having values darker which are recognized as finger-vein as given below:

$$N_{CL} = N_{clip} \times N_{avg} \tag{2}$$

$$N_{\text{avggray}} = \frac{N_{\Sigma\text{clip}}}{N_{gray}} \tag{3}$$

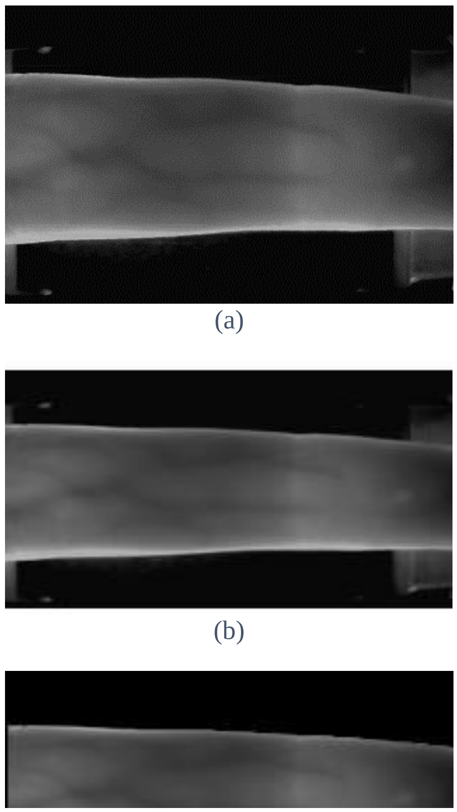
Where the real CL is N_CL, the actual CL is N_clip in the [0-1] range. If pixel number N_CL is lower than CL, then it is clipped. The NΣclip is the number of clipped pixels. Then the remaining average grey-level pixels are distributed as shown in (3) [7] [27].

3.3 2D-CNN Architecture

The 2D-CNN architecture has twenty-four layers. Initially, the image is input in the first layer having 50×50×1 size. Then there is a total of five 2D convolutional layers which apply different filters on images. The proposed model also contains five ReLU layers and five batch normalization layers. The proposed architecture contains a total of 3 max-pooling layers. There are 2 layers of fully connected layers followed by 1 softmax layer and finally, the output layer is known as the classification layer. The proposed model achieves an accuracy of 95.35 %.

The finger vein dataset is convolved with filters when they are passed through convolution layers. Where the features are learned for the given images and a feature matrix is generated which is given in Fig. 4

which shows the output of 1st convolutional layer. The five filters of size 4×4 are employed in the first convolutional layer with padding of [2 2 2 2] zeros from each direction. The other convolutional layers we employed 10, 20, 30, 40 filter size on each second, third, fourth, and fifth convolutional layer, respectively, as shown in table (1).



(a)

(b)

Fig. 2. (a) Original image of size 240×320×3, as of dataset SDUMLA-HMT (b) dimension reduced Image 128×160×1, (c) Region extracted image by applying polygonal ROI and Sobel edge detector.

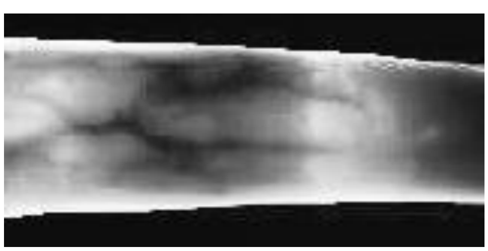


Fig. 3. Dual CLAHE implementation.

$$y = \sum_{c=1}^{N} w_c * x_c + b \quad (4)$$

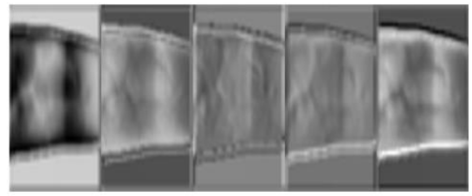


Fig. 4.Output of the convolutional layer.

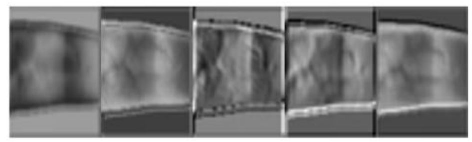


Fig. 5. Output from Batch Normalization Layer.

In (4) the ω_c , x_c and b are the weight in the c^{th} input channel, 2D input of the c^{th} channel filter, and filter bias, respectively. The bias factor can neglected by apply batch normalization having β– shift factor as depicted in Fig. 5. The output of the convolutional layer, the bias factor can change the sizes of channels. Therefore, batch normalization is employed for channels normalization for ReLU activation [28].

The output of batch normalization is fed to the activation function in the ReLU layer that introduces nonlinearity as depicted in Fig. 6 [5]. In our proposed system, we employ the maximum value of pixels. Pixels are considered as zero if pixels have a value less than zero and if the value is greater than zero they are considered y. The nonlinear functions employed during the ReLU layer are the tanh and sigmoid activation functions. The ReLU is well known established approach and it is as given in (5) [1] [5]:

$$f(y) = \max(0, y) \quad (5)$$

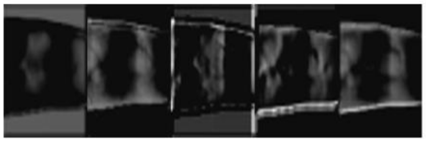


Fig. 6. ReLU layer output.

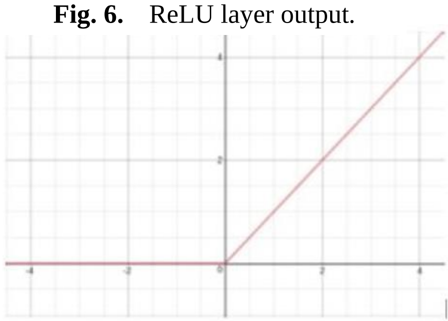


Fig. 7. ReLU Layer Graph.

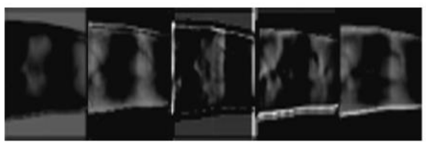


Fig. 8. Output Channels of Max-pooling layer.

The max-pooling layer will take the maximum values with a filter matrix of 2×2 which will only take the maximum number of values of channels. The stride of the max-pooling layer is [1 1] with padding 0. The max-pooling output is shown below in Fig. 7.

The fully connected layer we use two times t=f_c (x,w,b) this function of the fully connected layer, where x represents the input of this layer, where w represents the weight matrix, b is the bias, and t is the output [29] which is as mentioned in (6):

$$t_{i'} = \sum_i w_{ii'} x_i + b_i \quad (6)$$

The second last layer of 2D CNN produces a probability distribution for every image location used to classify pixels into one of the initial labelled where the related object classes belong to. The classification output layer uses the cross-entropy loss of reciprocally absolute classes for multi-class classification problems [30].

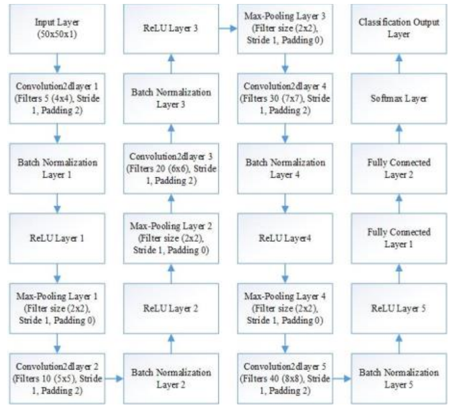


Fig. 9. Basic architecture of 2D CNN.

4 Result

In the proposed method we have divided the dataset as 80% for training and 20% for validation and testing of the model. The Adam optimizer-based train network is fed with the partitioned dataset in order to optimize the learned values and extract features in fed images from the proposed 2D-CNN model. In the proposed method, four images were selected for preprocessing from left-hand and right-hand folders and tagging them with the designations of the index, middle, and ring fingers sub-folders. During the training, the process epoch was set to 80, where an epoch is a full cycle with 12 iterations. In Fig. 10, the training process illustrates validation accuracy of 95.35% after completing the 960 iterations. The 2D-CNN model is trained and the update of weights is done by using the Adam optimizer. Table (II) shows the various parameter like iterations, epoch size, and iterations per epoch. It also contains the validation accuracy achieved during these parameters used in the model. Fig. 11 shows the loss of model in which weight can be renewed and to decrease the loss following evaluation

TABLE I. THE 2D CNN ARCHITECTURE FOR THE PROPOSED METHOD.

#	Layers Name	Input Size	Kernel size	Stride	Padding	Filters
1	Image Input layer	$50 \times 50 \times 1$				
2	Convolution2dlayer 1		4×4	[1 1]	[2 2 2 2]	5
3	Batch Normalization layer 1					
4	ReLulayer 1					
5	Max Pooling 1		2×2	[1 1]	[0 0 0 0]	
6	Convolution2dlayer 2		5×5	[1 1]	[2 2 2 2]	10
7	Batch Normalization layer 2					
8	ReLulayer 2					
9	Max Pooling 2		2×2	[1 1]	[0 0 0 0]	
10	Convolution2dlayer 3		6×6	[1 1]	[2 2 2 2]	20
11	Batch Normalization layer 3					
12	ReLulayer 3					
13	Max Pooling 3		2×2	[1 1]	[0 0 0 0]	
14	Convolution2dlayer 4		7×7	[1 1]	[2 2 2 2]	30
15	Batch Normalization layer 4					
16	ReLulayer 4					
17	maxPooling2dLayer		2×2	[1 1]	[0 0 0 0]	
18	Convolution2dlayer 5		8×8	[1 1]	[2 2 2 2]	40
19	Batch Normalization layer 5					
20	ReLulayer 5					
21	Fully Connected Layer 1					
22	Fully Connected Layer 2					
23	Softmax					
24	Classification Output					

TABLE II. 2D CNN TRAINING PARAMETERS

#	Parameter	Values
1	Validation accuracy	95.35%
2	Epoch	80
3	Iteration	960
4	Iteration per epoch	12
5	Validation Frequency	30 iteration

The proposed model [31] results are compared with a twenty-layer 2D-CNN in which the features of finger-vein images are learned with a validation accuracy of 94.87%, as illustrated in Fig. 12–13 below;

This space is intentionally left blank to adjust the images

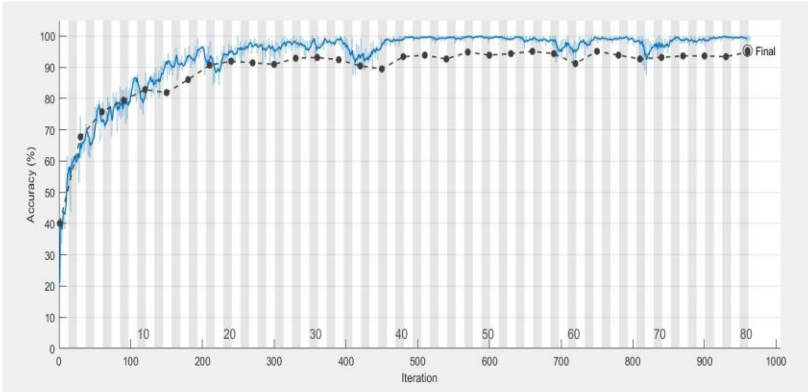


Fig. 10. Accuracy graph of 2D CNN using 24 layers.

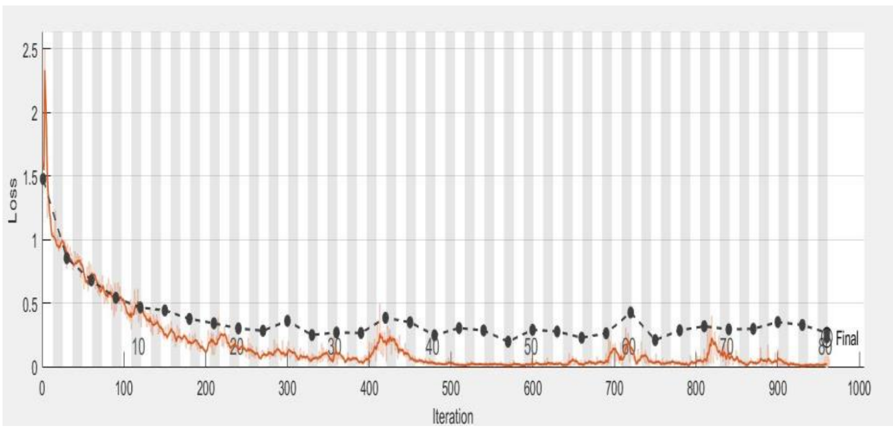


Fig. 11. Loss graph of 2D CNN using 24 layers.

This space is intentionally left blank to adjust the images

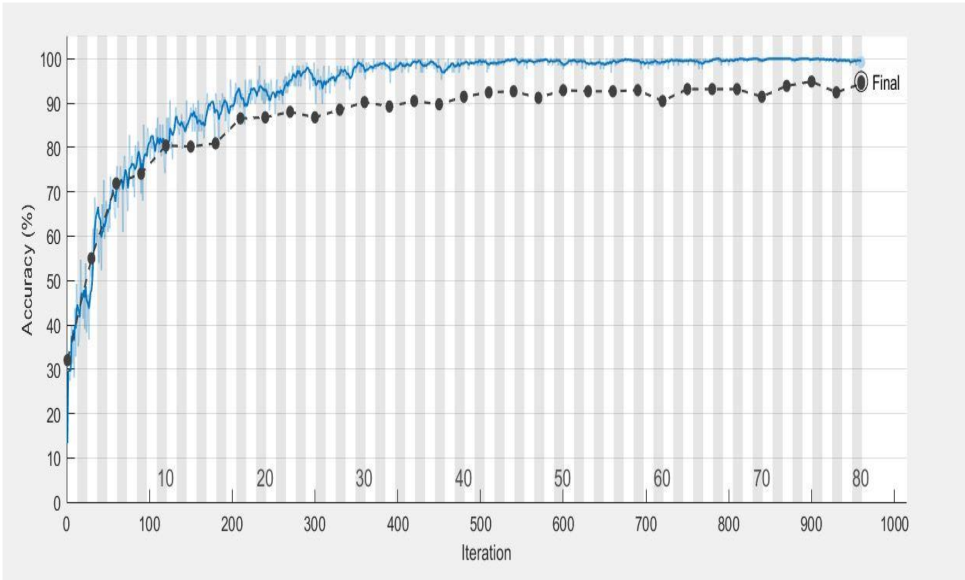


Fig. 12. 2D-CNN accuracy graph using 20 layers.

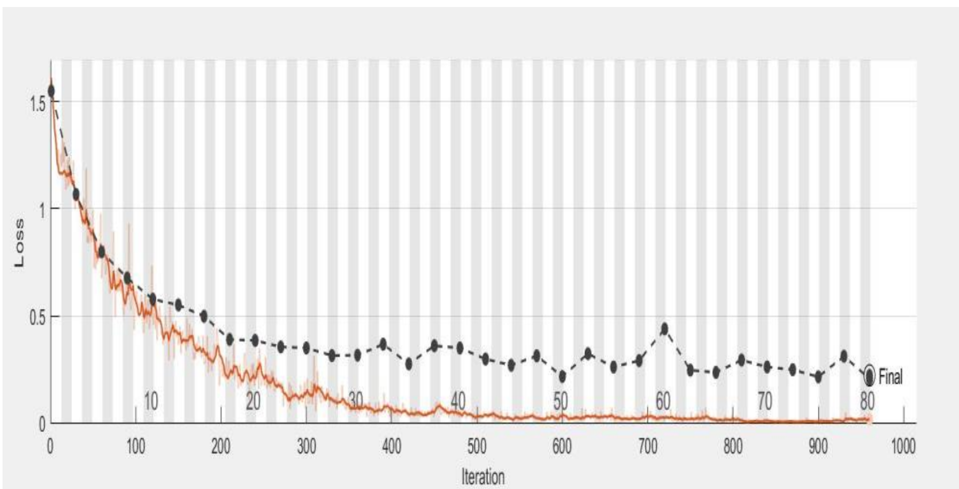


Fig. 13. The loss graph 2D-CNN using 20 layers [31].

Fig. 10 and 12 show that the validation accuracy of the proposed model using 24 layered 2D CNN yields much better results with a validation accuracy of 95.35%, while the previous results have a validation accuracy of 94.87%. In this proposed model, we use four max-pooling layers with a kernel size of 2×2 with a stride of 1, while the previous model used three max-pooling layers with a filter size of 3×3 and stride of 2. Fig. 11 and 13 compare the losses between the proposed model and the previous model of 2D-CNN in which validation losses are minimized as compared to the previous 2D-CNN model in [31].

5 Conclusion

1. We deployed a 2D CNN for finger-vein recognition in this study, which uses preprocessed dual CLAHE augmented image features and takes 2544 finger-vein images. To distinguish the background from the finger region, the pre-enhanced finger-vein images are run via the Sobel edge detector and polygonal ROI. The dataset is then run through 2D CNN to recognize the features of finger-vein images with distinct classes of folders like the middle, index, and ring folders. The 2D-CNN uses 4 images from each. As a result, our model was able to learn the characteristics of each image as well as recognize each image within folders. The accuracy of this proposed model is also near 96%.

REFERENCES

[1] Liu, Wenjie, Weijun Li, Linjun Sun, Liping Zhang, and Peng Chen, "Finger vein recognition based on deep learning," In 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), p. 205-210. IEEE, 2017

[2] Pham, Tuyen Danh, Young Ho Park, Dat Tien Nguyen, Seung Yong Kwon, and Kang Ryoung Park, "Nonintrusive finger-vein recognition system using NIR image sensor and accuracy analyses according to various factors," Sensors 15, no. 7 (2015): 16866-16894.

[3] Mulyono, David, and Horng Shi Jinn, "A study of finger vein biometric for personal identification," In 2008 International Symposium on Biometrics and Security Technologies, pp. 1-8. IEEE, 2008.

[4] Hong, Hyung Gil, Min Beom Lee, and Kang Ryoung Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors," Sensors 17, no. 6 (2017): 1297.

[5] Das, Rig, Emanuela Piciucco, Emanuele Maiorana, and Patrizio Campisi, "Convolutional neural network for finger-vein-based biometric identification," IEEE Transactions on Information Forensics and Security 14, no. 2 (2018): 360-373.

[6] Shin, Kwang Yong, and Kang Ryoung Park, "Finger-vein image enhancement using a fuzzy-based fusion method with gabor and retinex filtering," Sensors 14, no. 2 (2014): 3095-3129.

[7] Noroz Khan and Saleem Ahmed, "Finger-vein Image Dual Contrast Enhancement and Edge Detection," International Journal of Computer Science and Network Security 19, no. 11 (2019): 184-192.

[8] X. Qiu, W. Kang, S. Tian, W. Jia, and Z. Huang, "Finger Vein Presentation Attack Detection Using Total Variation Decomposition," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 2, pp. 465-477, Feb. 2018.

[9] B. T. Ton and R. N. J. Veldhuis, "A high-quality finger vascular pattern dataset collected using a custom-designed capturing device," International Conference on Biometrics (ICB), pp. 1-5, Madrid, 2013.

[10] Y. Lu, S. J. Xie, S. Yoon, Z. Wang and D. S. Park, "An available database for the research of finger vein recognition," 6th International Congress on Image and Signal Processing (CISP), pp. 410-415, Hangzhou, 2013.

[11] W. Yang, X. Huang, F. Zhou, Q. Liao, "Comparative competitive coding for personal identification by using finger vein and finger dorsal texture fusion", Inf. Sci., vol. 268, pp. 20-32, Jun. 2014.

[12] Lee, Eui Chul, Hyunwoo Jung, and Daeyeoul Kim. "New finger biometric method using near infrared imaging." Sensors 11, no. 3 (2011): 2319-2333.

[13] Kejun, Wang, Liu Jingyu, P. Popoola Oluwatoyin, and Feng Weixing, "Finger vein identification based on 2-D gabor filter," In the 2nd International Conference on Industrial Mechatronics and Automation, vol. 2, pp. 10-13. IEEE, 2010.

[14] Ezhilmaran, D., and P. Rose Bindu Joseph, "Fuzzy inference system for finger vein biometric images." In 2017 International Conference on Inventive Systems and Control (ICISC), pp. 1-4. IEEE, 2017.

[15] Lin, Cheng-Jian, Shiou-Yun Jeng, and Mei-Kuei Chen, "Using 2D CNN with Taguchi parametric optimization for lung cancer recognition from CT images," Applied Sciences 10, no. 7 (2020): 2591.

[16] Garg, Priyanka, and Trisha Jain, "A comparative study on histogram equalization and cumulative histogram equalization," International Journal of New Technology and Research 3, no. 9 (2017): 263242.

[17] Shin, Kwang Yong, Young Ho Park, Dat Tien Nguyen, and Kang Ryoung Park, "Finger-vein image enhancement using a fuzzy-based fusion method with gabor and retinex filtering," Sensors 14, no. 2 (2014): 3095-3129.

[18] Pham, Tuyen Danh, Young Ho Park, Dat Tien Nguyen, Seung Yong Kwon, and Kang Ryoung Park, "Nonintrusive finger-vein recognition system using NIR image sensor

and accuracy analyses according to various factors," Sensors 15, no. 7 (2015): 16866-16894.

[19] Peng, Jialiang, Ning Wang, Ahmed A. Abd El-Latif, Qiong Li, and Xiamu Niu, "Finger-vein verification using Gabor filter and SIFT feature matching," In 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 45-48. IEEE, 2012.

[20] Pang, Shaohua, Yilong Yin, Gongping Yang, and Yanan Li, "Rotation invariant finger vein recognition," In Chinese Conference on Biometric Recognition, pp. 151-156. Springer, Berlin, Heidelberg, 2012.

[21] Wu, Jian-Da, and Chiung-Tsiung Liu, "Finger-vein pattern identification using principal component analysis and the neural network technique," Expert Systems with Applications 38, no. 5 (2011): 5423-5427.

[22] Xin, Yang, Zhi Liu, Haixia Zhang, and Hong Zhang, "Finger vein verification system based on sparse representation," Applied optics 51, no. 25 (2012): 6252-6258.

[23] Syarif, Munalih Ahmad, Thian Song Ong, Andrew BJ Teoh, and Connie Tee, "Enhanced maximum curvature descriptors for finger vein verification," Multimedia Tools and Applications 76, no. 5 (2017): 6859-6887.

[24] Radzi, Syafeeza Ahmad, Mohamed Khalil Hani, and Rabia Bakhteri, "Finger-vein biometric identification using convolutional neural network," Turkish Journal of Electrical Engineering & Computer Sciences 24, no. 3 (2016): 1863-1878.

[25] Gopinath, P., and R. Shivakumar. "Exploration of finger vein recognition systems." (2021).

[26] Wang, Li, Haigang Zhang, and Jingfeng Yang, "Finger Multimodal Features Fusion and Recognition Based on CNN," In 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 3183-3188. IEEE, 2019.

[27] Ganesan, Thenmozhi, Anandha Jothi Rajendran, and Palanisamy Vellaiyan, "An Efficient Finger Vein Image Enhancement and Pattern Extraction Using CLAHE and Repeated Line Tracking Algorithm," In International Conference on Intelligent Computing, Information and Control Systems, pp. 690-700. Springer, Cham, 2019.

[28] Sledevic, Tomyslav, "Adaptation of convolution and batch normalization layer for CNN implementation on FPGA," In 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), pp. 1-4. IEEE, 2019.

[29] Liu, Kui, Guixia Kang, Ningbo Zhang, and Beibei Hou, "Breast cancer classification based on fully-connected layer first convolutional neural networks," IEEE Access 6 (2018): 23722-23732.

[30] Borra, Surya Prasada Rao, N. V. S. S. Pradeep, N. Raju, S. Vineel, and V. Karteek, "Face recognition based on convolutional neural network," International Journal of Engineering and Advanced Technology 9, no. 4 (2020): 156-162.

[31] Noroz, Noroz Khan Baloch, Saleem Ahmed Saleem, and Ramesh Kumar Kumar. "Finger-vein Image Enhancement and 2D CNN Recognition." International Journal of Innovations in Science & Technology 3 (2021): 33-44