# Finger-Vein Image Enhancement and 2D CNN Recognition

Noroz Khan Baloch[1], Saleem Ahmed[2], Ramesh Kumar[2],
[1]Dept. of Electronics Engg. Dawood University of Engineering & Technology Karachi, Pakistan.
[2]Dept. of Computer System Engg. Dawood University of Engineering & Technology Karachi, Pakistan
***Correspondence**: Noroz Khan Baloch, Email ID: b.noroz820@yahoo.com.

Finger vein recognition technology is a novel biometric technology with multiple features such as live capture, stability, difficulty in stealing and imitating, and more in the field of information security that has been utilized in a wide range of applications. In this proposed method, the finger region is separated from the background using a Sobel Edge detector and a Poly ROI which helps shape the finger. The background separation enhancement of low contrast using dual contrast limited adaptive histogram equalization which works on the visual characteristics of the finger-vein image dataset. When dual CLAHE is applied, the finger-vein histogram intensity is separated all across the image. Following the implementation of DCLAHE, an enhanced 2D-CNN model is utilized to recognize objects with the updated dataset. By maximizing the values of a preprocessed dataset, the 2D CNN model learns features. This model has a 94.88% accuracy rate.
**Keywords:** biometric; contrast limited adaptive histogram equalization; Sobel edge detector; poly region of interest and two dimensional convolution neural network.

## Introduction

Personal identification is currently achieved using biometric technology used in a variety of security applications such as mobile phones, computer access, border crossing, banks, and ATMs. Fingerprint, iris, face, voice, and finger-vein recognition are some of the methods used (FV) [1, 2]. Because the vein patterns beneath the skin differ, the finger-vein recognition system has gotten a lot more attention in recent years. As a result, finger-vein recognition suggests a safe and practical technique for human biometric classification [3]. Traditional methods for recognizing finger veins rely on extracting lines from the input image or amplification and feature extraction [4]. Near-infrared (NIR) light is used to capture vein images on the finger, and a charged coupled device (CCD) camera is used to record vein images beneath the finger [5]. Because the dataset needed to be normalized for the preprocessing step, the bitmap image was turned into a grayscale dataset with a single dimension for preprocessing and 2D CNN implementation [15]. When NIR penetrates the dorsal side of the finger and a CCD camera penetrates the palmar side, the region of the finger vein appears darker [6].

Finger vein datasets can be found in a variety of formats. The proposed approach for finger vein recognition was evaluated using the SDUMLA-HMT database [7]. Other finger vein datasets include HKPU-FV [8], UTFV [9], MMCBNU 6000 [10], THU-FV [11], and others. Local binary pattern (LBP) [12], Gabor filters (GF) [13], local derivative pattern (LDP) [12], and other techniques have been utilized to recover finger-vein images from the backdrop. Because these techniques have a weakness in similar settings such as translation and rotation during the capture of finger vein images, the deep learning technique known as convolutional neural network (CNN) is used for the classification of finger vein images. CNN/ConvNet is a deep learning algorithm that is faster than standard techniques [14].

Our suggested technique starts with a preprocessed Dual CLAHE [7] image dataset and feeds these preprocessed finger-vein images into a two-dimensional CNN model. Before classifying finger veins and applying enhancement techniques, our model uses a Sobel edge detector and a polygonal region of interest (ROI) to extract the background of the finger region [15]. As a result, we employ our image for histogram equalization follow extraction, which improves the image dataset. A histogram is a graphic representation of the distribution of greyscale values of pixels in a greyscale image [16]. The histogram's equalization distributes grey values uniformly throughout the image. Dual contrast limited adaptive histogram equalization (CLAHE) is used twice in the suggested technique to boost grey values in a finger vein image. Based on the folder names, the processed dual CLAHE finger vein images are classified. These pictures are then organized and utilized in the suggested 2D CNN model. In contrast to prior methods for extracting finger veins from picture datasets, 2D CNN algorithms can detect finger veins on a large scale of the dataset by learning features and recognizing them without removing background characteristics. According to the findings, 2D CNN is capable of locating finger veins with a high degree of accuracy.

The paper is organized as follows. The related study on finger-vein detecting technology is described in Section 2. Section 3 explains the finger-vein background extraction, Dual CLAHE, and 2D CNN architecture. The model's result and explanation are shown in section 4, and the conclusion is presented in section 5.

## Related Work

According to prior studies [1], [4], preprocessing and feature deduction approaches are commonly used in the detection of finger veins. The Gabor Filters have variable forms and directions to locate the finger-vein pattern [17]. The researcher utilized the local binary pattern (LBP) [18] and local derivative pattern (LDP) techniques to extract finger-vein patterns in [12]. The Gabor filter is used to match the SIFT features of vein patterns in the finger-vein classification [19, 20]. The principal component analysis (PCA) technique is used to create different traits [21]. Finger-vein feature extraction using sparse representation, the maximum curvature approach, HOG, and SVM are some of the documented strategies [23].

The algorithms for recognizing finger veins described above are standard. To improve the performance of finger-vein recognition, the researcher uses deep learning CNN-based algorithms instead of standard approaches [1]. In [24], the author developed a CNN technique for finger recognition that combines a reduced-complexity four-layer with fused convolutional-subsampling. In [24] suggested a CNN approach that consists of seven layers, five of which are convolutional layers and two of which are fully connected layers. The challenge of irregular size for features is acceptable resolved with a PCA-based dimensionality reduction strategy in the [26] paper, which employs a method that uses a multimodal finger fusion based on CNN. These procedures, on the other hand, are either more difficult or incapable of delivering the required results. On the other extreme, these treatments are either more complex or incapable of producing the desired effects.

## The Proposed Method
## Overview

In this study, finger-vein recognition is broken down into six steps, as shown in the flow diagram in Figure 1. The SDUMLA-HMT picture database [7], which is a free collection of finger-vein images, is used to read the first photos. As observed in Figure 2(a) and Figure 2(b), the finger-vein images in the dataset are standardized, which is normalized in the second phase, which changes the image dataset into greyscale images of size, as seen in Figure 2(a) and Figure 2(b). The third step is to extract our backdrop from the finger-vein pictures dataset (see Figure 2(c)). The Sobel edge detector and polygonal ROI are used to obtain the finger region. The finger-vein picture dataset is first run through the dual CLAHE, which produces dark vein ridges, before being fed into the 2D CNN deep learning model. The architecture of the 2D CNN with twenty layers, as shown in Figure 9, is shown below.
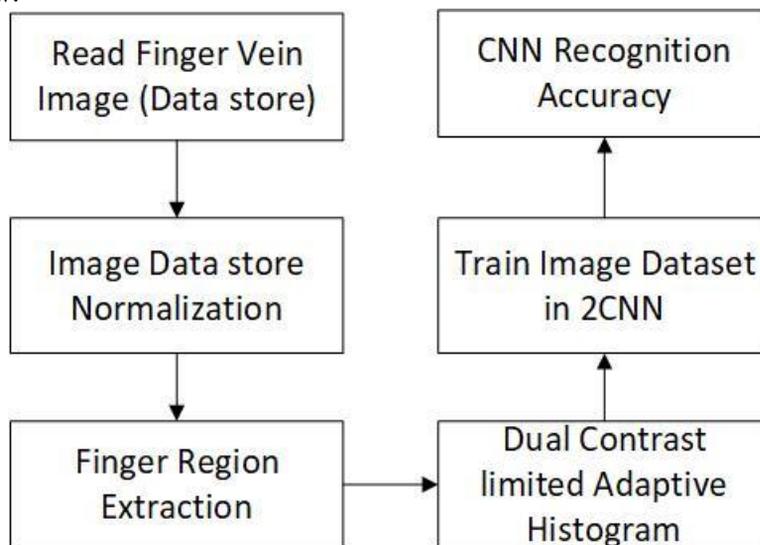


**Figure 1.** The proposed method's flow diagram.

## Sobel Edge Detector and Poly ROI.

At the points where the gradient is the highest, it appears ahead of the edges. By performing a 2-D spatial gradient quantity on a picture, the Sobel method accentuates regions of high spatial frequency that correlate to edges. It only takes the finger boundary of the finger-vein image [31]. So after the Sobel edge detector detects some areas of the finger and detects some of the backgrounds at the corner of the image, it cannot discriminate the finger region, so for this, it uses polygonal ROI to shape the finger part. Polygonal ROI creates a binary mask for the finger region to subtract the finger from the background portion and take the finger to detect finger-vein edges [32]. Figure 2(c) illustrates the result of separating the finger region from the background part.

## Dual CLAHE

Considering that the contrast of the dataset's finger-vein images is too low, the finger-vein cannot be distinguished from the background or skin region, we utilized a dual CLAHE enhancement technique [7], in which it is applied twice to the finger-vein image. The finger-vein image is cropped and reduced using this technique from the black region, which is the background portion of the original image that has been removed. We used dual CLAHE to enhance the vein region twice after cropping the finger-vein image, as shown in Figure 3. CLAHE is a modified form of Adaptive Histogram Equalization (AHE) that works on small parts of the contextual region [28] in the finger-vein picture known as tiles 22 on both CLAHE and AHE. CLAHE with the same exponential distribution [7], [27], with clipping limits of 0.03 on the first and 0.04 on the second.

$$N_{avg} = \frac{N_r X \times N_r Y}{N_{gray}} \qquad (1)$$

The average number of pixels is represented by $N_{avg}$, the number of gray-scale pixels is represented by $N_{gray}$, and pixel amounts in the $X$ and $Y$ directions are represented by $N_r X$ and $N_r Y$, respectively, as indicated in Eq. (1).

The CL (clipped limit) is a value that is limited to a specific threshold where grayscale is darker and is considered finger-vein as follows:

$$N_{CL} = N_{clip} \times N_{avg} \qquad (2)$$

$$\text{Navggray} = \frac{N\Sigma\text{clip}}{N_{gray}} \qquad (3)$$

Within the range of the [0 1] array, the actual CL is $N_{clip}$, where the real CL is $N_{CL}$. Only when it exceeds the pixel number $N_{CL}$ is it clipped. $N\Sigma\text{clip}$ is the total number of pixels clipped, and the remaining average grey-level pixels are distributed as in Eq. (3) [7], [27].

## 2D CNN Network Architecture.

The image input layer has a size of pixels, and there are four 2D convolutional layers, four batch normalization layers, and four rectified linear units (ReLU) with three max-pooling layers, two fully connected layers, one softmax layer, and a classification output layer. Learning characteristics from finger veins is 94.38% accurate.

The convolutional layer concatenates different sizes of filters on the finger-vein image dataset to learn features of the finger-vein image for the image recognition system, as demonstrated in Figure 4 which shows the initial convolutional layers. The first convolutional layer employs five filters and [2 2 2 2] padding to maintain the image size by adding two zeros to the top, bottom, left, and right of the finger-vein image matrix. Ten, twenty, and thirty filters of various sizes are used in the convolutional layers, as shown in Figure 9.

$$y = \sum_{c=1}^{N} w\text{c} * x_c + b \qquad (4)$$

$\omega_c$ is the weight array of a 2D filter in the $c^{th}$ input channel, $x_c$ is the 2D input of the $c^{th}$ channel filter, and b is the bias of a filter in the above Eq. (4). As seen in Figure 5, batch normalization substitutes this bias with a $\beta-$shift factor. The sizes of channels may fluctuate with the bias factor after convolving the finger-vein input images with different filters, hence batch normalization normalizes the channels for ReLU activation [28].

The ReLU layer, as shown in Figure 6 [5], is an activation function that increases nonlinearity after batch normalization. ReLU employs the maximum pixel value in this model. Pixels less than zero are referred to as 0, whereas pixels larger than zero are referred to as y. Non-linear functions such as tanh and sigmoid were previously employed. Eq. (5) [1], [5] shows how the researcher determines the ReLU:

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

The max-pooling layer with a filter matrix will take the maximum values, taking only the maximum number of channel values. The stride of the max-pooling layer is [2 2] with padding. The maximum pooling output is shown in Figure 7.

(a)



(b)



(c)
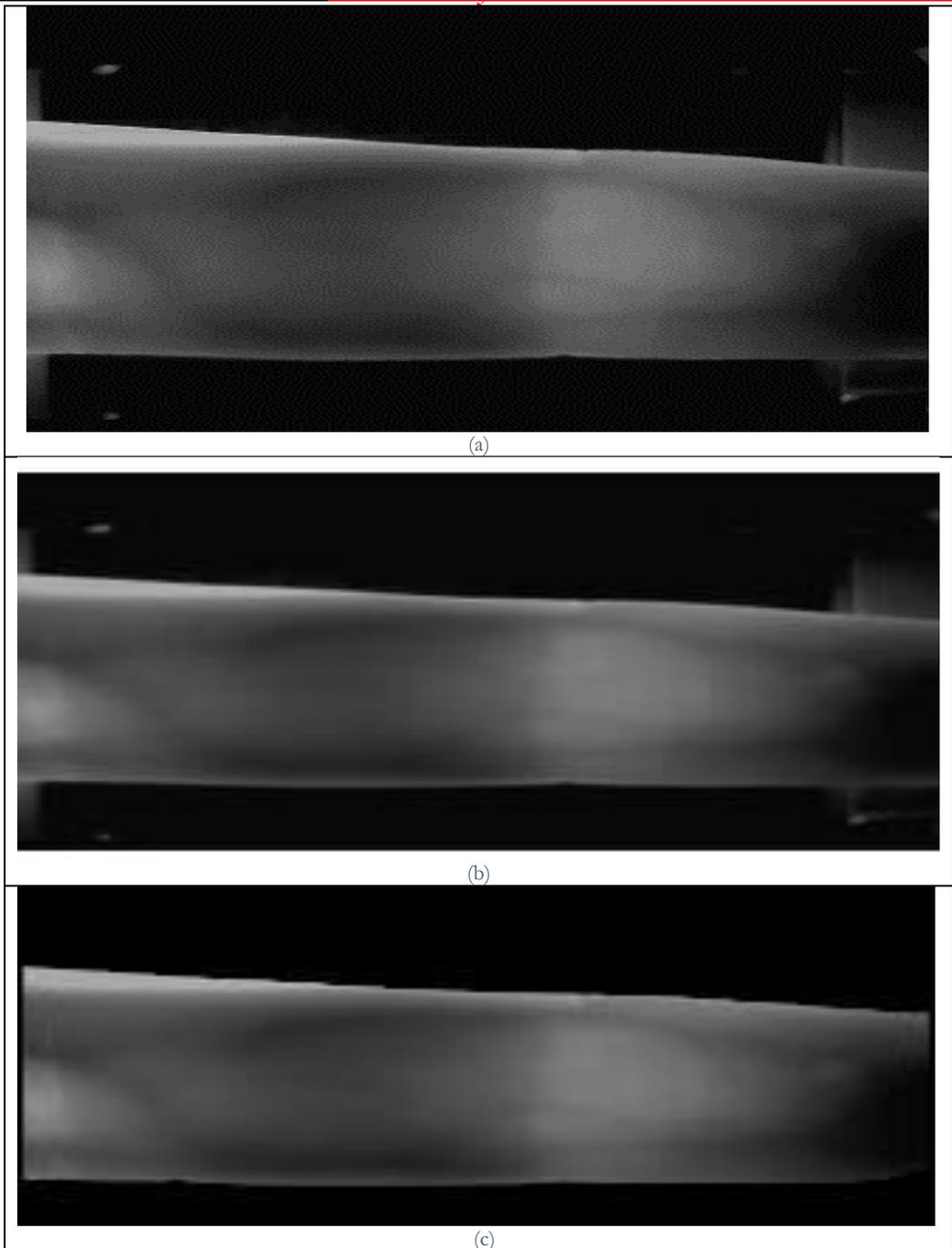
**Figure 2.** (a) Original Finger Vein 240×320×3, (b) Normalized Image 120×160×1, (c) Finger Region Extraction using Sobel edge detector and polygonal ROI.
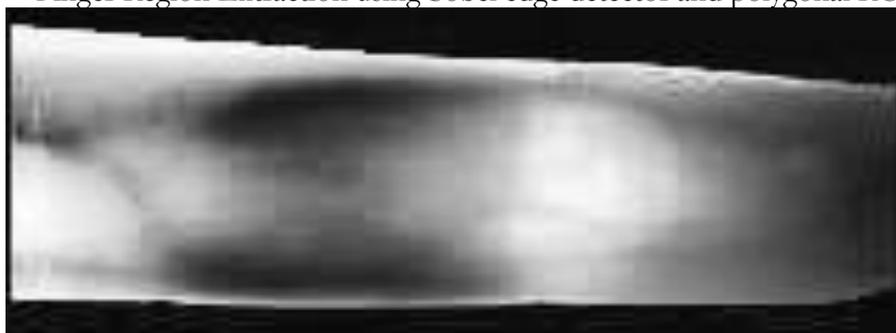


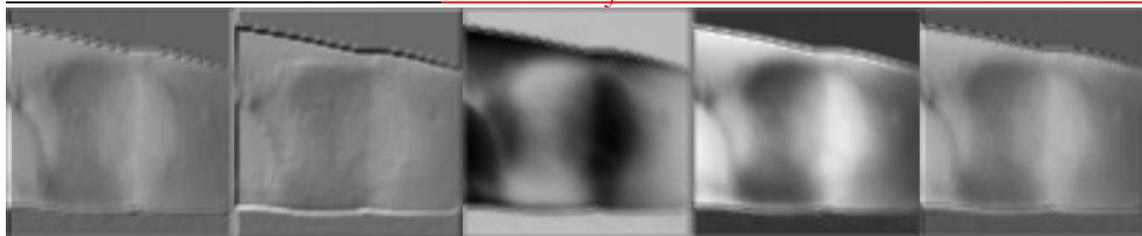**Figure 3.** Dual CLAHE implementation.
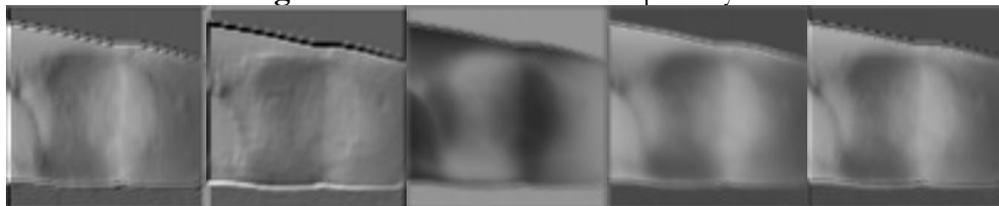
**Figure 4.**First Convolution Output Layer.



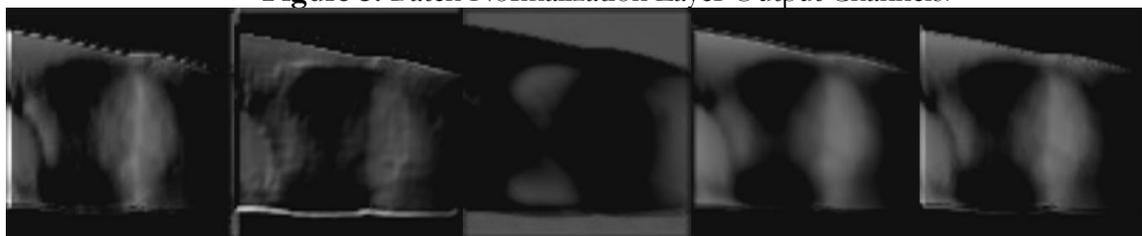**Figure 5.** Batch Normalization Layer Output Channels.



**Figure 6.** ReLU Layer Output Channels.


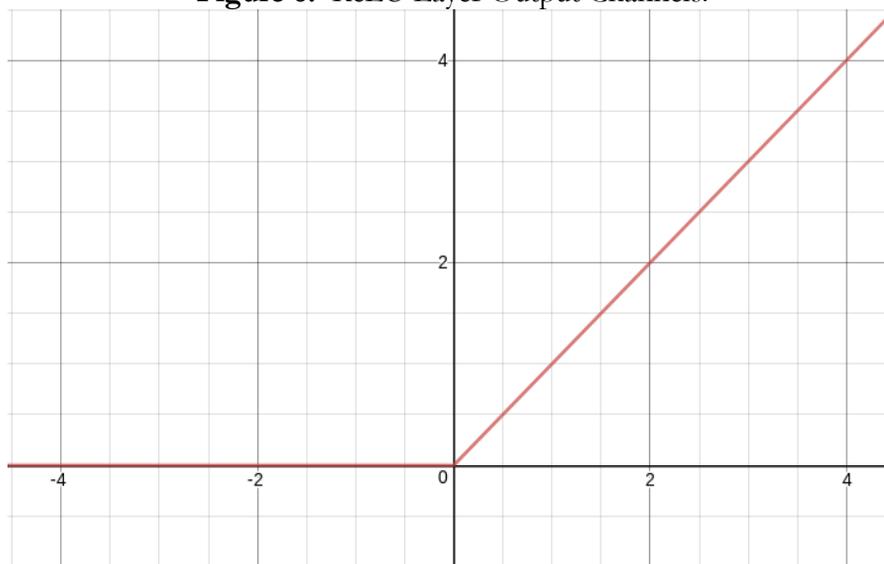
**Figure 7.** Graph of ReLU Layer.



**Figure 8.** Layer Output Channels of Max-pooling.

We apply this function of the fully connected layer two times $t = f_c(x, \omega, b)$, where $x$ represents the input of this layer, w represents the weight matrix, b represents the bias, and t represents the output [29], which is as described in Eq. (6):

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

For each image location used to classify pixels into one of the initial labels where the linked object classes belong, the second final layer of 2D CNN generates a probability distribution. For multi-elegance category issues, the category output layer employs the pass-entropy lack of reciprocally absolute instruction [30].
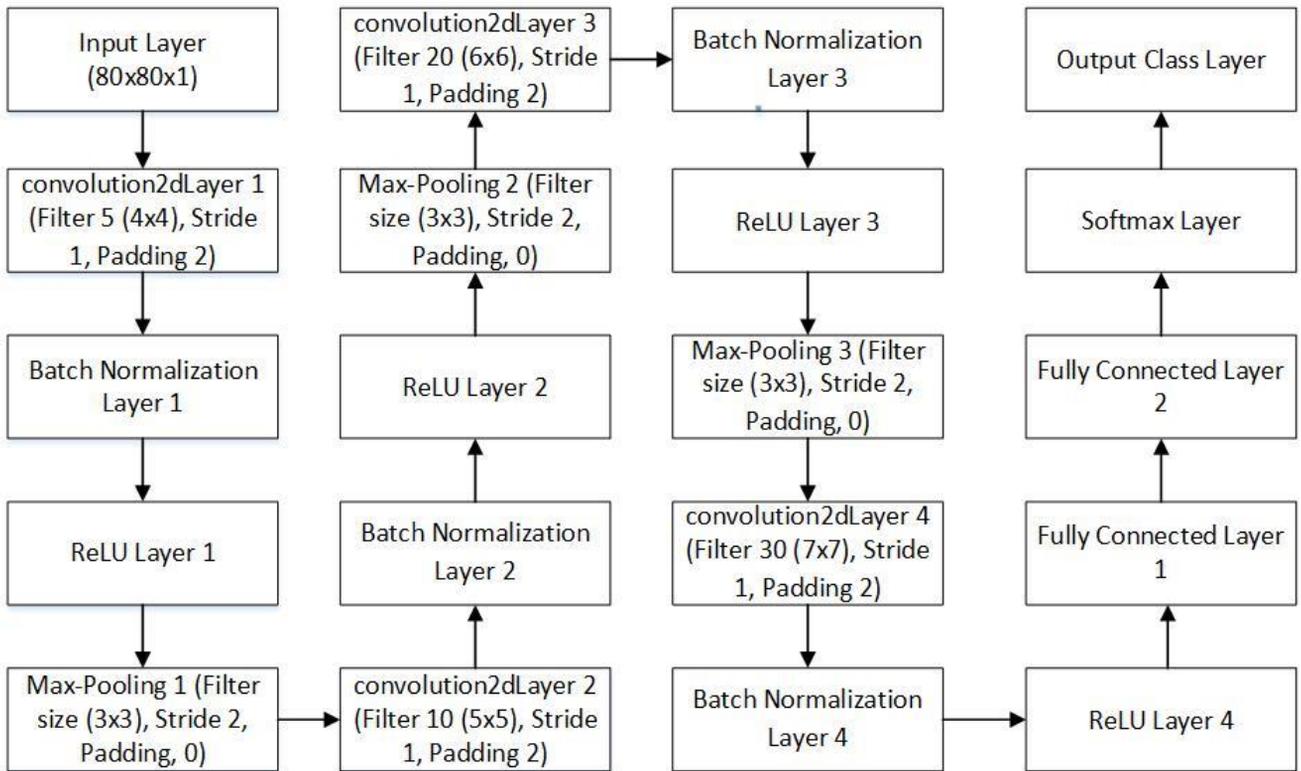


**Figure 9.** Basic architecture of 2D CNN.

### Result

The preprocessed dataset of the dual CLAHE finger-vein picture dataset is randomly divided into 80 percent training, 20 percent validation, and 20 percent test images, with four index, middle, and ring finger classes in each folder. These divided data are fed into an Adam optimizer-based train network, which optimizes the learned values from our CNN model and detects features in each image. In this version, we extract four preprocessed finger-vein pictures from each folder of the left and proper hands, labeling them with the names of the index, center, and ring palms. The training option employs an Adam optimizer with a maximum epoch of 80, with one epoch equaling one complete cycle and twelve iterations. The training progress report in Figure 10 shows that after 960 iterations, the validation accuracy for learning the dual CLAHE preprocessed image dataset with various classes is 94.88 %.

The Adam optimizer is used to train the 2D CNN model and update network weights. The epoch values, iteration, iteration per epoch, and validation frequency are shown in table (1) below. Figure 11 depicts a model loss in which the weight may be replenished and the loss reduced on subsequent review. The blue line in Figure 10 depicts the training (smoothed), which is less noisy. The light blue line represents training, which classifies individual mini-batches, while dotted or dashed lines indicate validation, which classifies the complete collection. In Figure 10, the blue line illustrates the training (smoothed), which is less noisy. The light blue line is the training that classifies individual mini-batches, and validation is represented by dotted or dashed lines that classify the entire set. The training loss on each mini-batch is shown in Figure 11 in light orange, the smoothed training loss in orange, and the validation loss in dotted or dashed lines.
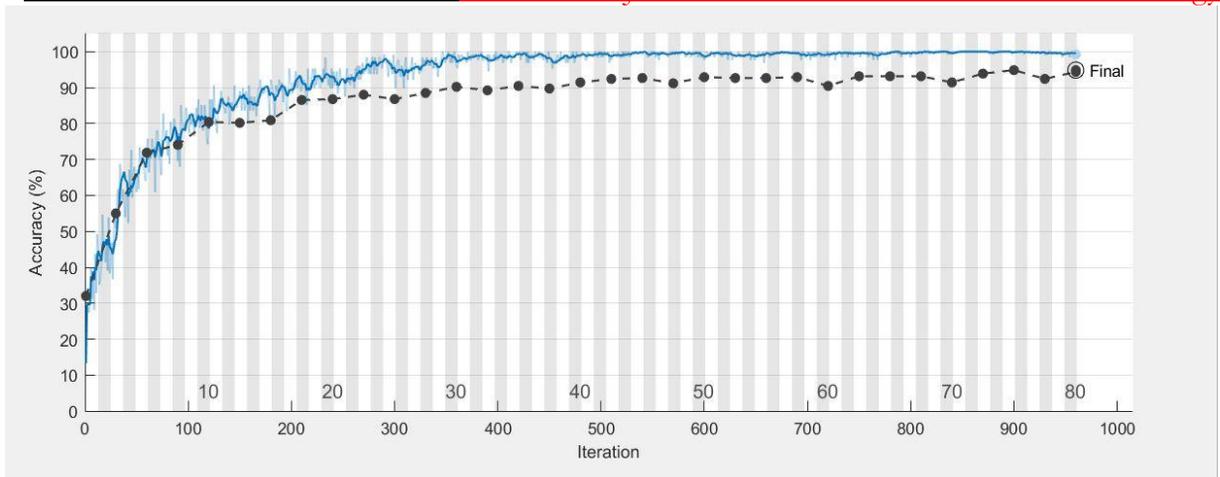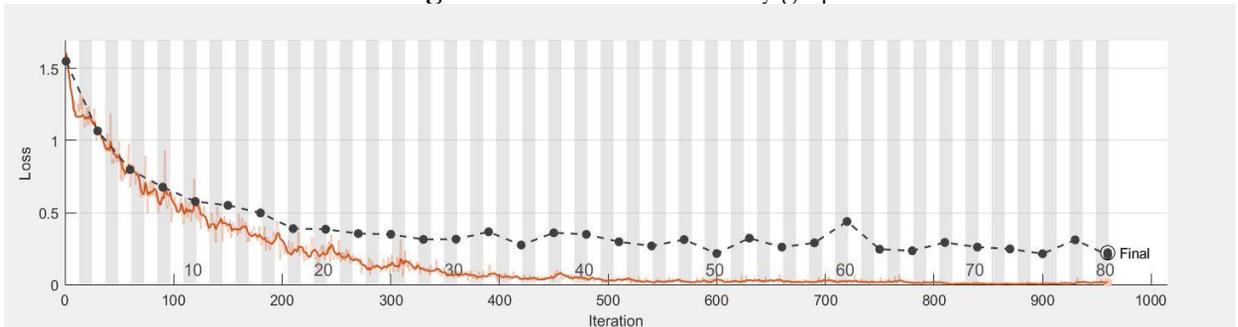
**Figure 10.** 2D CNN accuracy graph.



**Figure 11.** 2D CNN loss graph.

**Table 1**. Table describing sample of IJIST…..

| # | Parameter | Values |
|---|---|---|
| 1 | Validation accuracy | 94.87% |
| 2 | Epoch | 80 |
| 3 | Iteration | 960 |
| 4 | Iteration per epoch | 12 |
| 5 | Validation Frequency | 30 iteration |

**Conclusion**

In this research work, we used 2544 images to separate the finger region from the background using a Sobel edge detector and poly ROI, which shaped the acquired edges of the finger and created a binary mask to separate out the finger. The finger-vein dataset is enhanced using dual CLAHE and after enhancing the finger-vein images, these datasets are further utilized in 2D CNN where the model learns the features of every segmented finger-vein. The finger-vein images are divided into distinct classes of folders such as index, middle, and ring folders, with 2D CNN taking four images from each folder. Hence, our version became capable of examining the traits of every photo as well as distinguishing every photograph inside folders. The suggested model's accuracy is 94.88%.

## 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, EmanuelaPiciucco, 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, 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.

7. Baloch, Noroz Khan, Zuhaibuddin Bhutto, Abdul Sattar Chan, Mudasar Latif Memon, Kashif Saleem, Murtaza Hussain Shaikh, 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. Muthukrishnan, Ranjan, and Miyilsamy Radha. "Edge detection techniques for image segmentation." International Journal of Computer Science & Information Technology 3, no. 6 (2011): 259.

32. Li, Yongxiao, and Woei Ming Lee. "PScan 1.0: flexible software framework for polygon based multiphoton microscopy." In SPIE BioPhotonics Australasia, vol. 10013, p. 1001333. International Society for Optics and Photonics, 2016.