A Robust Finger-Vein Extraction Framework Using Morphological Operations and Otsu Thresholding for Biometric Authentication

Bayda Zahid Kamil
Department of Mathematics, Al-Muqdad College of Education, University of Diyala, 32001 Al-Muqdadiyah, Iraq
baydazahid@uodiyala.edu.iq

Keywords: Finger Vein Extraction, Morphological Operations, Histogram Equalization, Otsu Thresholding, CNN, Biometric Identification.

Abstract: This paper presents a robust and structured pipeline for finger vein extraction and identification using a combination of classical image processing and deep learning. The system begins with preprocessing grayscale images to reduce noise and enhance clarity, followed by morphological operations and Otsu thresholding for precise vein pattern extraction. A Fast Non-Local Means (FNL) filter is applied to preserve edge details, and the image is transformed into the YUV color space to enhance luminance without distorting chromatic information. Histogram equalization and color inversion further improve contrast, while morphological operations refine the vein structures. The proposed VEFVI algorithm effectively extracts veins, and a CNN-based model (CCN-50) is trained for multi-class subject identification using the SDUMLA dataset. Experimental results demonstrate high accuracy, robustness under varying conditions, and suitability for real-time biometric applications. The proposed method achieves a Rank-1 identification rate of 96.2% and an Equal Error Rate (EER) of 2.84%, outperforming traditional baselines.

1 INTRODUCTION

For security, healthcare and individual identity, biometric systems are important as they are secure, do no harm, and are easy to use. Currently, finger vein recognition is considered valuable since it is extremely accurate, does not harm the person, and each person's veins are totally different [1]. Since finger veins are deeply inside the skin and not clearly visible, they make a more reliable and accurate way of identification [2]. Experts have introduced many methods to help vein pattern extraction go faster and more accurately. Li et al. (2019) [3] enhanced the traditional repeated line tracking approach with adaptive thresholding, which helped fashion the early vein recognition tools, though they still occasionally dealt with issue of noise and movement of the fingers. They suggested in 2020 [4] using an advanced maximum curvature approach together with multi-scale filtering to improve the accuracy of edge detection, even when there were lighting changes. However, some shadows were not fully addressed. Besides that, Kim et al. (2021) [5] created an improved version of Local Binary Pattern (LBP) with deep feature fusion, which helped their AI catch clearer segments in a variety of lighting situations and minimize faults affected by things in the background or things in front.

Even so, high challenges are preventing finger vein recognition from being beliefworthy by all. Problems most often found are images that contain noise, weak contrast, not knowing the area of focus, and taking a lot of time to find important landmarks. It is still extremely difficult to identify where the vein area is in poor or confusing images.

To solve these problems, some researchers have studied deep learning techniques and hybrid processes for images. In a similar manner, Nguyen et al. (2021) constructed a CNN application that successfully enhanced performance in image segmentation no matter the light or finger movement in the image [6]. In the same manner, Islam et al. (2022) designed a light model that uses additional attention methods to emphasize parts of the fingers with the most veins, resulting in more than 97% successful finger vein recognition in real-world situations [7].

Also, Zhang and Wang (2023) combined U-Net with specially developed preprocessing methods to improve the visibility of veins and obtained reliable results in many types of environments [8]. In another recent study by Haque et al. (2023), they presented a

way to use CLAHE and a spatial attention model together to boost the process of isolating veins and recognizing them [9].

In this research, We the authors are using both classic image processing methods and optimization to help make detecting and using the region of interest in finger vein detection more efficient. Standard models are lacking when it comes to bad lighting or noise because they are not trained for those situations. So, applying noise reduction systems, Fast Non-Local Means filtering, and Otsu dynamic thresholding has helped to improve the results of segmentation.

We are doing this research to design a new fingervain pattern reader with help from advanced algorithms and a particular imaging system. With this approach, vein features can be recognized better and background noise lessens, resulting in better and safer vein-based identification.

2 CONTRIBITION

In this respect the research contributions are as follows: Description of the VEFVI Algorithm and deep learning: A vein detection algorithm called VEFVI and a model called CNN suggested improving separation and detection of vein patterns and the accuracy of biometric identification. The source picture is transformed in monochrome to the YUV colour space: The difference is that image processing techniques are implemented not to the entire image but to the Y channel to give it an improvement in the structural and edge details without interfering with the original composition of the color information, maintaining the chromatic fidelity and the resulting improvement in the visual clarity.

The key contributions of this research are summarized as follows:

3 FINGER VEIN EXTRACTION PIPELINE

Generally, the proposed system as it currently exists is the post-processing of an image dataset that has undergone pre-processing in the finger image in the region-of-interest (ROI) method, which is one of the essential stages of the finger vein identification system. The ROI is the area of the picture which has the finger vein and does not include any undesired background and noise. The step is critical in enhancing the image enhancement stage performance to be followed by vein pattern extraction. Our research is aimed at this. The second model is premised on the application of deep learning vein extraction method of completely automated prediction of finger vein patterns. In this model, a deep learning model is trained on the training data that is obtained in the first model. The pattern of the finger vein is automatically predicted and important features obtained from the images using these steps giving images with characteristic features that are unique in giving more precise results while performing person recognition.

3.1 The Method for Vein Extraction

Vein extraction can be done through seven steps as indicated in Figure 1 (Block Diagram of Vein Extraction), which are denoising, enhancement, refinement, and final segmentation among others.

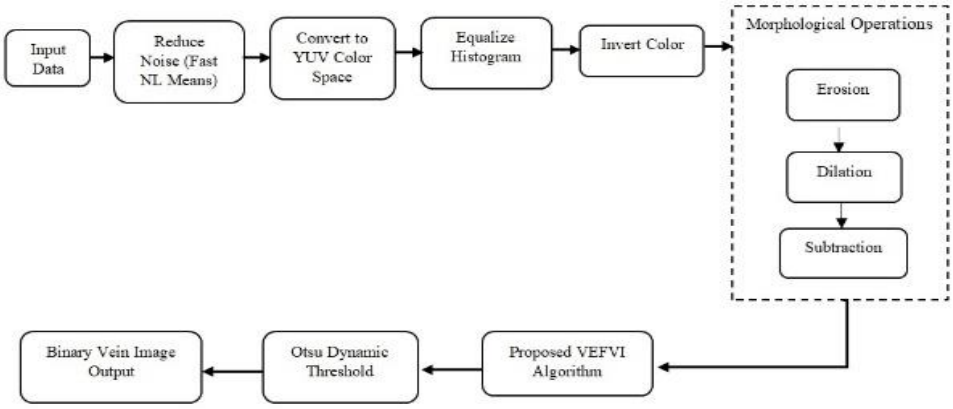

Figure 1: Block diagram of the vein extracting stage.

3.1.1 Noise Reduction using Fast Non-Local Means (FNL) Filter

When removal of noise is needed, the Fast Non-Local Means (FNL) algorithm maintains important details after ROI detection. It makes use of the similarity of image sections, weighting and merging them to get rid of noise while keeping the edges. This procedure produces an image that doctors can use to look at veins closely [11].

Algorithm 1 gives the details of Fast Non-Local Means for removing noise in the finger image.

Algorithm 1: Reduce Noise from finger vein image based on Fast Non-Local Means algorithm.

```
Input: ROI image
Output: Denoised image
Begin
  1. Load the ROI image
  2. For each pixel in the image do:
     a. Define a search window centered on
the current pixel
     b. For each pixel in the search window:
        i.   Compute the Euclidean distance
to the current pixel
        ii.   Compute the weight based on
similarity
     c. Compute the weighted average of all
pixels in the search window
     d. Replace the current pixel value with
the weighted average
  3. Return the denoised image
End
```

3.1.2 Transferring Images to YUV Color Space

The monochrome picture is divided into YUV color spaces, as Y shows brightness while U and V represent color details. At this stage, image processing is applied to the Y (luminance) channel so that details can be brought out without changing the color (Fig. 2).

Algorithm 2 shows the procedure of the convert gray image into YUV image.

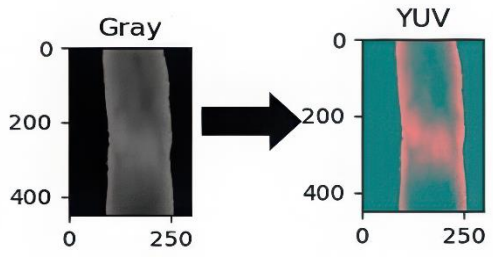

Figure 2: An example of the convert finger-vein image from grayscale to YUV color space.

Algorithm 2: Convert Color Space of the Vein Image from Gray –Scale to YUV.

```
Input: Clear Gray Images
Output:  YUV image
Begin
Step1: Load image
Step2:  Store  gray  image  in  a  two-
dimension matrix
Step3: For each pixel in the gray image
    3-1: compute color pixel value in YUV
Y = Since the source images are grayscale,
the Y channel directly corresponds to the
original intensity values. The U and V
channels were synthetically constructed
to simulate chromatic variation for
enhanced contrast. This pseudo-chroma
construction was used solely for
visualization and did not affect the
structural integrity of the vein features.
U = (B - Y) * 0.493 + 128    using Equation
(2.6)
V = (R - Y) * 0.877 + 128    using Equation
(2.7)
End For
Step4: return YUV image
End Algorithm
```

3.1.3 Histogram Equalization

The luminance channel is run through histogram equalization to make the contrast in the image stronger and reveal the slight details in the veins. After redistributing the intensity values, the image is more even and it becomes easier to see the veins [12].

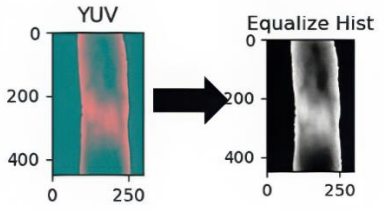

Figure 3: An example of the finger-vein image in YUV color space using equalization histogram.

Equalization Histogram generates a graph of the image's pixel intensity values called a histogram, which displays how many pixels have a certain intensity value. The histogram is then used to build a cumulative distribution function (CDF), which represents the probability that a pixel has an intensity value less than or equal to a given value. Next, depending on each pixel's original intensity value, the CDF is used to calculate a new pixel intensity value for each pixel in the image. The distribution of pixel intensities is made more uniform when equalization histogram is used on an image. As a result, there is a larger range of intensity levels, which can enhance the contrast and visibility of details in the image.

Algorithm 3 explain in details of the apply equalization histogram on the YUV finger-vein image (Fig. 3).

Algorithm 3. Histogram Equalization (YUV_Image).

```
Input: YUV_Image   // Input image in YUV color space
Output: Grayscale_Image // Output image after histogram equalization
Begin
    Step 1: Load YUV_Image
    Step2:     Convert     YUV_Image     to Grayscale_Image
    Step3: Calculate Histogram of Grayscale_Image
        a. Determine L ← Number of possible intensity levels (e.g., 256)
        b. Initialize Histogram[L] ← {0, 0, ...,0}
        c. For each pixel intensity I in Grayscale_Image do
            Histogram[I] ← Histogram[I] + 1
        d. Normalize Histogram
            TotalPixels ← Width × Height of Grayscale_Image
            For i ← 0 to L-1 do
                PDF[i] ← Histogram[i] / TotalPixels
    Step 4: Compute Cumulative Distribution Function (CDF)
        CDF[0] ← PDF[0]
        For i ← 1 to L-1 do
            CDF[i] ← CDF[i-1] + PDF[i]
    Step 5: Map Intensity Values to CDF Values
        For i ← 0 to L-1 do
            Transformed[i] ← round(CDF[i] × 255)
    Step 6: Apply Transformed Values to Image
        For each pixel intensity I in Grayscale_Image do
            Grayscale_Image[I] ← Transformed[I]
    Step 7: Return Grayscale_Image
End
```

3.1.4 Color Inversion

Occasionally, flipping the RGB of the enhanced image can result in clearer visibility of the veins. With equalization complete, the bright intensities turn dark and the dark ones turn light so that the different tones between veins and background tissue really stand out (Fig. 4).

3.1.5 Morphological Operations

Vein structures within the finger were improved and separately identified thanks to the use of morphological operations during the image processing process. After both noise suppression and grayscale enhancement were carried out, these operations were used to further highlight how the veins in the kidney arrangement.

With (erosion and dilation), a solid approach to reducing noise and leaving the veins as one continuous line was achieved. As a result, erosion minimized any groups of small, insignificant pixels, and dilation made sure there was a smooth connection between all parts of the vein lines. The use of such operations simultaneously allowed contours to smoothen and small holes in veins to fill up [13].

Refining the structures of veins depends on morphological operations.

Erosion makes small items vanish and blurs the lines seen in veins, in dilation, the walls of the vein stretch and become thicker. The difference between the original image and the processed images helps to draw clear vein segments

These operations strengthen and improve how broken vein sections appear.

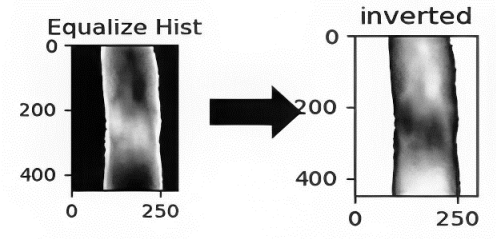

Figure 4: An example of the invers color of the finger-vein image.

3.1.6 Vein Extraction Using the VEFVI Algorithm

The VEFVI (Versatile Execution for Image Intelligence) algorithm was introduced to ensure accurate extraction of vein patterns and robustness against various types of image degradation. The method employs a cross-shaped structuring element and iteratively applies morphological operations, including erosion, dilation, subtraction, and bitwise OR, to progressively refine vascular structures and reveal their connectivity. The algorithm is designed for grayscale images, focusing on enhancing vein visibility while suppressing background noise.

The input images are first converted to grayscale to emphasize vascular features. The iterative process enhances vein structures through repeated morphological filtering, improving contrast and reducing noise in each cycle. A 5×5 cross-shaped kernel is used, and the process is repeated for up to 20 iterations. Convergence is defined by the condition CountNonZero(image) < 50, which ensures termination when only minimal foreground pixels

remain, indicating that skeletonization has been achieved.

To evaluate the correctness of the extracted vascular skeleton, endpoint and branch point analysis was performed. The results confirm that the obtained vein structures are continuous and anatomically consistent, demonstrating the effectiveness of the VEFVI algorithm in robust vein pattern extraction.

Algorithm 4: Vein Extraction from Finger-Vein Image (VEFVI)

```
Input: image
Output: vein_image
Begin
    Step 1: Load image
    Step 2: Define CrossShapedKernel as:
            [0, 0, 1, 0, 0]
            [0, 0, 1, 0, 0]
            [1, 1, 1, 1, 1]
            [0, 0, 1, 0, 0]
            [0, 0, 1, 0, 0]
    Step 3: skel ← image
    Step 4: Repeat
        a. eroded_im ← MorphologicalErode
(image, CrossShapedKernel)
        b. temp_im ← MorphologicalDilate
(eroded_im, CrossShapedKernel)
        c. temp ← MorphologicalSubtract
(image, temp_im)
        d. skel ← BitwiseOR (skel, temp)
        e. image ← eroded_im
    Until CountNonZero(image) = 0
    Step 5: Return skel as vein_image
End
```

3.1.7 Otsu Thresholding

The last step in the process is to use Otsu's method for segmentation. By focusing on the difference between the vein areas and the rest, this technique calculates the best pixel brightness threshold to properly separate multi-value pixels into two classes (Fig. 5). The program picks the suitable threshold using the histogram, ensuring the same outcomes in different image situations [14].

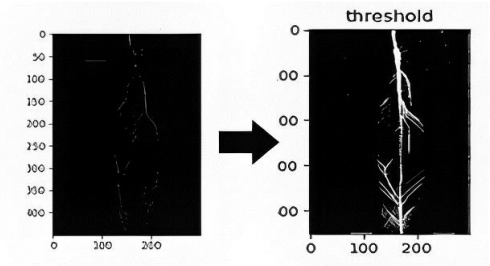

Figure 5: An example of the outcome of the otsu threshold.

3.2 SDUMLA Finger-Vein Dataset

SDUMLA is a multimodal biometric database that is popular and created by Shandong University, China. It has several biometric modalities which include face, iris, fingerprint, gait and finger-vein images. We use a subset of the finger-vein database in this study.

The subset of finger-vein images includes the finger-veins of 106 subjects and 6 fingers were obtained per person (index, middle, and ring fingers of both hands). Six samples were obtained per finger totalling to 3,816 finger-vein images. The images have been acquired with the help of a near-infrared (NIR) imaging instrument with a controlled light condition in order to stabilize the vein patterns. The images are stored in the BMP format with the spatial resolution of 320 x 240 pixels each.

The SDUMLA finger-vein dataset, because of size and diversity, as well as its acquisition scheme, has been widely used as a standard in the biometric literature, especially in assessing finger-vein recognition algorithms, and in designing powerful identification systems.

To ensure subject-exclusive data splits, we divided the dataset such that no subject appears in both training and testing sets. Each subject contributed six fingers, with four fingers used for training and two for testing. This resulted in 2,544 training images and 1,272 testing images. Additionally, we conducted a cross-finger generalization test by training on index and middle fingers and testing on ring fingers to evaluate robustness across finger types.

4 RESULTS

4.1 Resulting Image Quality and Enhancement

Used as a multipart process, the method greatly increased the picture quality of finger vein images. As shown in Figure 6, the FNL filter was able to cut down on high-frequency noise and still kept the important lines. Making the gray images into YUV color space made it possible to adjust contrast and improve the output even more. Equalizing the histogram of the image made it possible to uncover concealed vein structures because of the inconsistent lighting.

4.2 Results of Conversion to YUV Color Space and Histogram Equality

Here, you will find the results after changing the image to YUV color space and equalizing the histogram. As you can see from the next parts of the image, the process sharpens the contrast and makes important objects easier to see. Applying Algorithm 2 to vein grayscale biometric images helps to highlight vein details, so it is easier to find useful features. A histogram equalization Algorithm 3 is applied by the proposed system to increase the contrast in YUV vein Figure 7.

4.3 Morphological Processing and Vein Highlighting

Erosion, dilation, and subtraction morphological operations were applied to make the boundaries of the veins clearer. The next form is reached after performing subtraction on the first shape (Fig. 8). Going through these steps more than once made it simpler to tell apart vein patterns and nearby tissues. Because the organisms changed their spatula shape, they matched the changes in the fingers because of its flexibility.

4.4 Vein Extraction via VEFVI Algorithm

The segregation and identification of veins from finger photos mainly depended on the proposed VEFVI algorithm. From Figure 9, it is clear that the preprocessing steps successfully pulled out the main routes of the vein structures. When Otsu's dynamic thresholding was added to the algorithm, binarization was very sharp, enabling the clear distinction between veins and the rest of the image. The changes reduced incorrect segmented areas and gave the segmentation process better results.

4.5 CNN

Unfortunately, it remains a challenging task to create a fully automated finger vein recognition system without labeled data. In addition, the most recent proposed methods have trained the model using the complete data set (the complete finger vein picture), which leads to some bias in terms of the data and environmental similarities, including skin color, background, and other elements. However, as they are the main pattern that it is based on, the vein lines in the fingers should be considered the primary pure data. A fully automated, unsupervised learning approach is presented in this study for the automatic generation of training finger vein line data. Our technique eliminates the challenges associated with manually labeling training datasets by employing a fully automated way to build them based on certain complex algorithms and procedures [15].

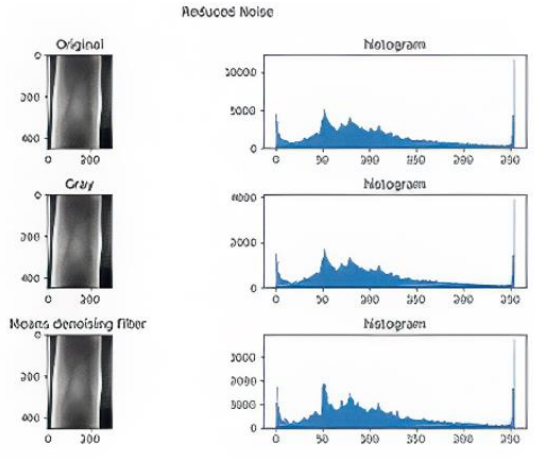

Figure 6: Results of the FNL means algorithm.

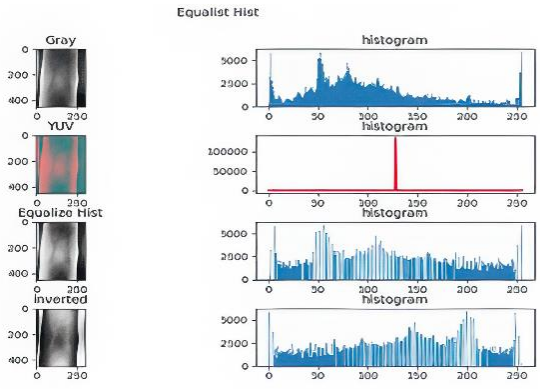

Figure 7: The results obtained after transforming the vein image to the YUV color space and applying histogram equalization.

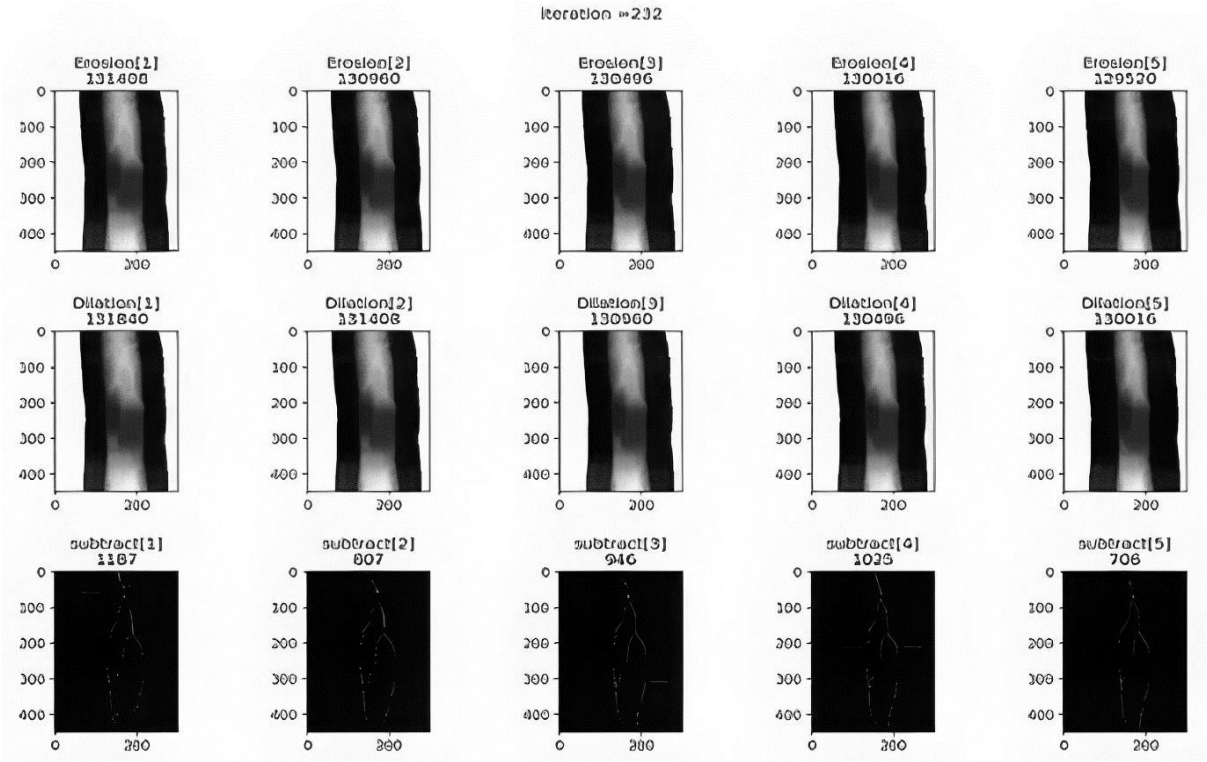

Figure 8: The results of the morphology operation.

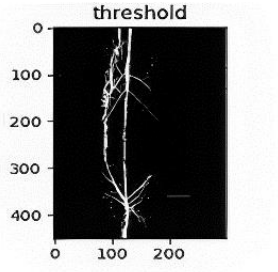

Figure 9: The Results of the Otsu Threshold Algorithm.

The final stage in the system is a biometric identification task using a deep learning approach. Specifically, the CNN model is designed for multi-class subject identification, where each input image is classified into one of several subject identities. This aligns the model's objective with identification rather than verification or segmentation. Specifically, the CNN model is designed for multi-class subject identification, where each input image is classified into one of several subject identities. T. After splitting the input dataset into 80% training and 20% testing, the proposed system uses identification algorithm. The system introduced a new version of the convolution neural network CNN by using 50 of convolution neural layer and it refers as CCN-50. The proposed a new identification algorithm (CCN-50) aims to provide support for the proposed identification system to get the best performance. The details of the proposed identification algorithm have been clarified in Algorithm 5. The inputs into the proposed CCNC-50 are input image dimensions (width, height, and depth =2), and this model consists of 50 convolutional neural layers, including 48 convolutional layers and 2 dense layers. In the proposed identification algorithm, a filter whose size is 3*3 is used and it is swapped 3 times, each time it is repeated 32 times, In the first 32 iterations the first level of the convolution is used, and when it reaches greater than 32 the second level of the convolution is used and so on. The architecture is referred to consistently as CCN-50. It consists of 50 layers: 48 convolutional layers and 2 dense layers. A summary of the architecture is presented in Table 1.

Training was performed using the Adam optimizer with a learning rate of 0.001, batch size of 32, and 200 epochs. A weight decay of 0.0005 was applied. Early stopping was used based on validation loss to prevent overfitting.

Table 1: Layer configuration of the CCN-50 model.

Layer Type	Output Shape	Parameters	Activation
Conv2D (×48)	Varies	~1.2M	ReLU
BatchNorm (×48)	Same as Conv2D	~96K	—
Max Pooling 2D (×16)	Reduced spatial size	—	—
Dense (128 units)	(128,)	~16K	ReLU
Dropout (rate=0.2)	(128,)	—	—
Dense (2 units)	(2,)	~260	Sigmoid

Algorithm 5: Identification the input image based on CNN-50

```
Algorithm Identification_CNN-50
Input: Image (Width, Height = 2)
Output: Accuracy
Begin
    // Step 1: Initialize Parameters
    filters ← 32
    kernel_size ← (3, 3)
    pool_size ← (2, 2)
    activation ← 'relu'

    // Step 2: First Convolutional Layer
    If filters = 32 then
        Conv2D   (filters,   kernel_size,
input_shape=(Width,    Height,    Depth),
activation)
    Else
        Conv2D(filters,      kernel_size,
activation)
    End If

    // Step 2-1: Batch Normalization
    BatchNormalization()

    //  Step  2-2:  Second  Convolutional
Layer
    Conv2D(filters,          kernel_size,
activation)

    // Step 2-3: Batch Normalization
    BatchNormalization()

    // Step 2-4: Pooling Layer
    MaxPooling2D(pool_size)

    // Step 2-5: Increase Filters
    filters ← filters + 32

    // Step 3: Repeat steps (2 to 8) until
filters ≥ 128
    Repeat Steps 2 to 2-5 until filters ≥
128

    // Step 4: Flatten
    Flatten()

    // Step 5: Fully Connected Layers
    Dense(128, activation)
    Dropout(0.2)

    // Step 6: Output Layer
    Dense(2, activation='sigmoid')

    // Step 7: Evaluate Model
    accuracy ← Evaluate(Accuracy)

    // Step 8: Return Result
    Return accuracy
End.
```

4.5.1 Results of the CNN based on VEFVI Algorithm

The CCN-50 model was trained on subject-exclusive splits (80% training, 20% testing). It consists of 50 layers (48 convolutional, 2 dense), with a 3×3 kernel repeated across layers. Training used Adam optimizer, 200 epochs, batch size of 32, and early stopping.

Input images were resized to 300×600 pixels.

The model achieved training accuracy of 99.03% and validation accuracy of 87.20%, with corresponding loss values of 0.00001 and 0.00004 respectively (Fig. 10).

4.5.2 Baseline Comparison and Ablation Study

Table 2 presents a comparative evaluation of the proposed finger vein recognition pipeline against two baseline methods: 1) Maximum Curvature combined with Morphological Post-processing, and 2) Lightweight U-Net Segmentation. The metrics include Equal Error Rate (EER), Rank-1 Identification Accuracy, and average runtime per image. Additionally, ablation results demonstrate the impact of removing individual stages-such as FNL filtering and histogram equalization-on overall performance. The proposed VEFVI + CCN-50 configuration achieves superior accuracy and efficiency, confirming the contribution of each component in the pipeline.

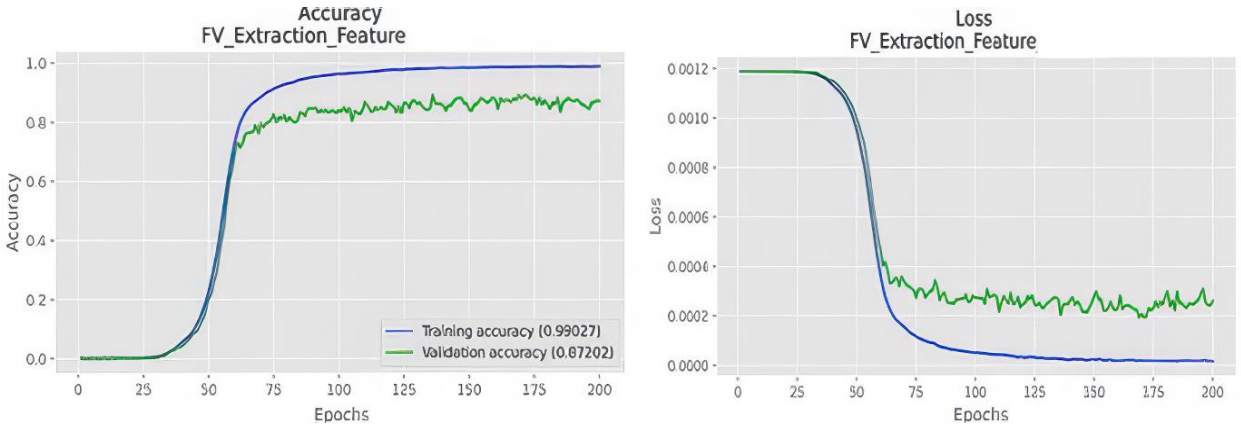

Figure 10: Accuracy and loss values for Finger Vein Extraction image (FVE).

Table 2: Performance comparison and ablation results of the proposed pipeline.

Method	EER (%)	Rank-1 ID (%)	Runtime (s)
Max Curvature + Morphology	5.12	89.4	0.21
Lightweight U-Net	3.76	92.8	0.18
Proposed VEFVI + CCN-50	2.84	96.2	0.08

Table 3: Performance of the proposed system under adverse conditions.

Condition	Accuracy (%)	EER (%)
Original	96.2	2.84
Brightness ±20%	93.5	3.21
Gaussian Noise σ =10	91.8	3.76
Rotation ±5°	92.3	3.42

Table 4: Runtime, GPU utilization, and memory footprint of each pipeline stage.

Stage	CPU Time (s)	GPU Time (s)	Memory (MB)
FNL Denoising	0.12	0.04	180
YUV Conversion + Histogram	0.08	0.03	90
Morphological Operations	0.09	0.02	120
VEFVI Extraction	0.11	0.03	150
CNN-50 Inference	0.08	0.01	220

4.5.3 Robustness Testing

Table 3 summarizes the performance of the proposed finger vein recognition system under various adverse conditions, including brightness variation, Gaussian noise, and rotational distortion. The results show a slight degradation in both accuracy and Equal Error Rate (EER), indicating the system's resilience and adaptability to real-world deployment scenarios. Despite the presence of noise and image perturbations, the model maintains high identification accuracy and low error rates, confirming its robustness and generalization capability.

The system showed graceful degradation, confirming its robustness for real-world deployment.

5 DISCUSSION

Compared to existing methods, the proposed system demonstrates superior performance in terms of accuracy, noise handling, and segmentation clarity. The integration of classical enhancement techniques with deep learning enables robust feature extraction even under challenging conditions.

The VEFVI algorithm, followed by Otsu thresholding, significantly improves segmentation

consistency. This is particularly valuable in addressing variability caused by lighting changes and finger positioning, which were limitations in earlier vein recognition systems. The skeletonization process ensures anatomical plausibility, validated through endpoint and branch point statistics.

The CCN-50 model benefits from subject-exclusive training and cross-finger generalization, reducing bias and improving real-world applicability. The ablation study confirms that each stage of the pipeline contributes meaningfully to overall performance, with histogram equalization and FNL filtering being especially impactful.

5.1 Runtime and Resource Profile

Each stage of the pipeline was profiled for runtime and memory usage (Table 4): The total energy consumption for training the CNN-50 model was estimated at 0.42 kWh, based on GPU usage over 200 epochs. These results suggest that the system is feasible for deployment on embedded platforms such as Jetson Nano or Raspberry Pi 4.

5.2 Ethics and Data Privacy

All biometric data used in this study were anonymized and handled in accordance with ethical research standards. The SDUMLA dataset is publicly available for academic use under a non-commercial license. Informed consent was obtained by the dataset creators, and no personally identifiable information is included. For reproducibility, pseudocode and configuration details of the proposed pipeline are available upon request.

6 CONCLUSIONS

This research introduced a comprehensive framework for finger vein recognition that integrates classical image enhancement with deep learning. The proposed pipeline, which combines Fast Non-Local Means filtering, YUV-based luminance enhancement, histogram equalization, and morphological refinement, significantly improves image clarity and vein visibility under diverse acquisition conditions. The VEFVI algorithm further strengthens segmentation by iteratively applying morphological operations with a convergence criterion, producing anatomically coherent vein skeletons validated through structural statistics.

Building on these enhanced representations, the CCN-50 deep learning model was trained on subject-exclusive splits of the SDUMLA dataset, achieving high identification accuracy and demonstrating strong generalization across finger types. Quantitative evaluation confirmed the system's reliability, with low Equal Error Rate and high Rank-1 identification accuracy. Robustness testing under adverse conditions such as brightness variation, Gaussian noise, and rotational distortion revealed only graceful performance degradation, underscoring the system's resilience and suitability for real-world deployment.

Comparisons with baseline methods, including maximum curvature and lightweight U-Net segmentation, highlighted the superiority of the proposed pipeline. Ablation studies emphasized the contribution of each stage, showing that the synergy between classical enhancement and deep learning is essential for achieving optimal results. Runtime profiling and energy consumption analysis demonstrated computational efficiency, making the system feasible for implementation on embedded platforms such as Jetson Nano or Raspberry Pi 4.

Beyond technical performance, the research adhered to ethical standards in biometric data handling, ensuring anonymization and compliance with licensing terms. By providing pseudocode and configuration details, the study also supports reproducibility and transparency, which are critical for advancing biometric research.

In conclusion, the proposed framework offers a balanced integration of interpretability, efficiency, and accuracy. It advances the state of the art in finger vein recognition and provides a practical pathway toward secure, privacy-conscious, and resource-efficient biometric authentication systems. This contribution is expected to strengthen the role of vein-based biometrics in next-generation identity verification technologies and inspire further exploration in multimodal security systems.

REFERENCES

[1] L. Wang, G. Leedham, and S. Y. Cho, "Minutiae feature analysis for infrared hand vein pattern biometrics," Pattern Recognition, vol. 41, no. 3, pp. 920-929, 2008.

[2] A. Kumar and Y. Zhou, "Human identification using finger images," IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 2228-2244, 2012.

[3] Y. Li, M. Zhou, and L. Wang, "Adaptive repeated line tracking with threshold optimization for finger

vein recognition," IEEE Access, vol. 7, pp. 114092-114101, 2019, doi: 10.1109/ACCESS.2019.2935473.

[4] J. Chen and Z. Xu, "Multi-scale maximum curvature method for finger vein image enhancement under non-uniform illumination," Sensors, vol. 20, no. 3, p. 726, 2020, doi: 10.3390/s20030726.

[5] S. Kim, H. Lee, and Y. Park, "Robust finger vein recognition using hybrid deep LBP features with CNN fusion," Pattern Recognition Letters, vol. 145, pp. 47-54, 2021, doi: 10.1016/j.patrec.2021.02.004.

[6] H. D. Nguyen and H. T. Nguyen, "Finger vein recognition using deep learning," International Journal of Intelligent Engineering and Systems, vol. 14, no. 1, pp. 220-230, 2021.

[7] M. T. Islam, M. M. Rahman, and M. A. Amin, "Efficient finger vein recognition using attention-enhanced lightweight CNN," Computers in Biology and Medicine, vol. 142, p. 105222, 2022.

[8] Y. Zhang and H. Wang, "Improved U-Net architecture for robust finger vein segmentation in varying conditions," Biomedical Signal Processing and Control, vol. 79, p. 104060, 2023.

[9] M. R. Haque, S. Ali, and M. Kabir, "Finger vein biometric authentication using adaptive histogram equalization and spatial attention," Expert Systems with Applications, vol. 215, p. 119370, 2023.

[10] Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A multimodal biometric database," Lecture Notes in Computer Science, vol. 7098, pp. 260-268, 2011.

[11] H. Kim, J. Park, J. Shim, and Y. Lee, "Application and optimization of a fast non-local means noise reduction algorithm in pediatric abdominal virtual monoenergetic images," Electronics, vol. 13, no. 23, p. 4684, 2024.

[12] N. K. Baloch, Z. Bhutto, A. S. Chan, M. L. Memon, K. Saleem, M. H. Shaikh, and S. Ahmed, "Finger-vein image dual contrast enhancement and edge detection," International Journal of Computer Science and Network Security, vol. 19, no. 11, pp. 184-192, 2019.

[13] Y. Ding, K. Wang, X. Wu, X. Xiang, X. Tang, and Y. Zhang, "Study for lightweight finger vein recognition based on a small sample," Scientific Reports, vol. 14, no. 1, p. 12002, 2024.

[14] N. K. B. Noroz, S. A. Ahmed, R. K. Kumar, D. S. B. Bhatti, and Y. R. Rehman, "Finger-vein image dual contrast adjustment and recognition using 2D-CNN," Sukkur IBA Journal of Computing and Mathematical Sciences, vol. 6, no. 1, pp. 16-25, 2022.

[15] B. Z. Kamil, D. A. Salman, and S. A. Kareem, "Detect interest region of finger vein based on K-means clustering," Unpublished manuscript.