CHAPTER 4

FINGER VEIN IMAGE LABELLING USING HYBRID ALGORITHM

4.1 INTRODUCTION

Deep Learning techniques have proved to provide higher accuracy, as supported by the literature review. However, the effectiveness of these models depends on the quality of the input FV images. The FV image dataset captured from the FV image scanner may contain noise and other artifacts that may degrade the model's performance. To address these challenges, the images in the dataset can be labelled, which involves classifying every pixel in the image as a vein pattern pixel or a background pixel. This will enable the DL model to learn more detailed features relevant to vein patterns. Labelling also helps in reducing the training time as it removes irrelevant background information from the images, which the DL model otherwise needs to learn as non-vein regions.

4.2 LABELLING OF FINGER VEIN IMAGES

FV image labelling involves classifying each pixel in an FV image as either part of the vein or the background. This detailed labelling allows DL models to accurately learn and understand the complex patterns of the vein. By labelling, the model becomes good at distinguishing vein structures from background regions. Labelled images help in reducing ambiguity during the training, leading to faster convergence of the model. Labelling can be performed either manually or automatically. Each method has advantages and challenges, which are essential to understand for effective implementation in biometric systems (Sager C. et al., 2021).

a. Manual Labelling

Manual labelling involves human annotators identifying and marking pixels that correspond to vein structures within the images. This is usually done using specialized software tools designed for image annotation. However, there are some challenges in manual labelling, which are discussed below**:**

- **Manually Identify and Mark Pixels**: Annotators use software tools to manually trace and mark the vein patterns in each image. This requires a

thorough examination of the image to distinguish between vein and background pixels.

- **Time-Consumption to Annotate**: Since the FV patterns are complex and detailed, manual labelling is highly time-consuming. Annotators must carefully review each image, which makes the process labour-intensive and slow, especially when dealing with large datasets.
- **Accuracy Concerns**: Manual labelling may not always be accurate. Human annotators are prone to errors and inconsistencies, which can lead to errors in the labelled data. Factors such as fatigue, attention span and subjective judgment can influence the quality of annotations.

b. Automated Labelling

Automated labelling uses algorithms and machine learning techniques to assign labels without human involvement. This approach is increasingly popular due to its efficiency and scalability. Automated labelling has the following advantages:

- **Reduced Time and Effort**: Automated labelling reduces the time and effort required for the labelling process. Algorithms can process large amounts of data quickly, which is important when dealing with large datasets.
- **Cost-Effective**: Automated labelling is cost-effective as it reduces the need for human annotators. The long-term savings in the labour cost can compensate for the initial investment in the software.
- **Consistency across the Dataset**: Algorithms used to assign labels are applied consistently across the entire dataset. This ensures uniformity across the dataset and reduces the variability that can arise from manual labelling.

4.3 FINGER VEIN IMAGE LABELLING USING AUTOMATED LABELLING

Automated labelling is used in this work to annotate vein patterns before it is fed to the DL model. By applying automated labelling, large-scale datasets can be labelled rapidly. Three prominent algorithms were used for the automated labelling of FV images: the Local Maximum Curvature (LMC) algorithm (Miura et al., 2007), the Wide Line

Detector (WLD) (Liu L. et al., 2007), and the Repeated Line Tracking (RLT) algorithm (Miura N. et al., 2004). The selection of the LMC, WLD, and RLT algorithms for automated labelling of FV images was based on their demonstrated effectiveness in identifying and extracting vein patterns as supported by the previous research (Miura N et al., 2004, Miura N. et al., 2007, Huang B. et al. 2010, Yang W. et al., 2019). The results from these algorithms are combined to produce an improved labelling algorithm.

4.3.1 Local Maximum Curvature Algorithm (LMC)

The Local Maximum Curvature (LMC) algorithm (Miura et al., 2007) is designed to identify the central lines of veins in FV images. The LMC algorithm is particularly effective in extracting fine details of vein patterns, which are crucial for accurate FVR.

The curvature of intensity profiles is calculated for every pixel, which evaluates the number of bends at each point, which serves as an indicator of the presence of vein centres. Since veins typically appear as darker patterns against the lighter tissue, the intensity profile curvature helps identify areas where veins are most likely located. Local maxima of the curvature profile are then identified, as they represent the highest point of curvature, which is an indication of vein centres. The vein centres are then assigned a score equivalent to the magnitude of the curvature. A larger curvature score implies a stronger likelihood that the pixel represents the centre of a vein. This helps in filtering noise and irrelevant details.

The algorithm also analyzes the score of vein centres in multiple directions to ensure that the vein patterns are detected consistently across the entire image. This helps in adapting to variations in image orientation and positioning of the finger during scanning. The vein centres are then connected to form continuous lines representing the vein patterns. Thus, the algorithm produces a labelled image by marking each pixel corresponding to a vein centre or a connected vein line. The step-by-step procedure is presented in the algorithm 4.1.

Algorithm 4.1: Local Maximum Curvature

Step 1. Initialize Parameters:

Set parameters for curvature calculation: profile length L and directions D.

Step 2. For every pixel (x, y) in the Image:

Step 2.1. Calculate the Curvature of Profiles:

For each direction d ε D:

Step 2.1.1. Extract the intensity profile I^d of length L centred at (x, y).
Step 2.1.2. Calculate the second derivative I''_d of the intensity profile I^d

$$I''_d = \frac{\partial^2 I_d}{\partial d^2}$$

Step 2.1.3. Compute the curvature K_d (x, y) as the magnitude of the second derivative:

$$K_d\ (x, y) = |I''_d|$$

Step 2.2. Detect Local Maxima:z

Identify local maxima in the curvature profiles K_d (x, y) across all directions.

Let LM (x, y) represent the set of local maxima:

LM (x, y) = {(x', y') | K_d (x', y') > K_d (x, y) for all (x', y') in the neighbourhood of (x, y)}

Step 2.3. Assign scores to the centre positions:

Assign a score S (x, y) to each detected vein centre based on the curvature value:

$$S\ (x, y) = \max_{d \varepsilon D} K_d\ (x, y)$$

Step 2.4. Determine the vein patterns in all directions:

Analyze the scores S (x, y) to confirm consistent vein patterns across different directions. Consider vein centres with high scores in multiple directions as valid vein centres.

Step 3. Connect the Vein Centres:

Connect the identified vein centres (x, y) to form continuous vein lines using a connectivity algorithm. Let C (x, y) represent the connected vein centres:

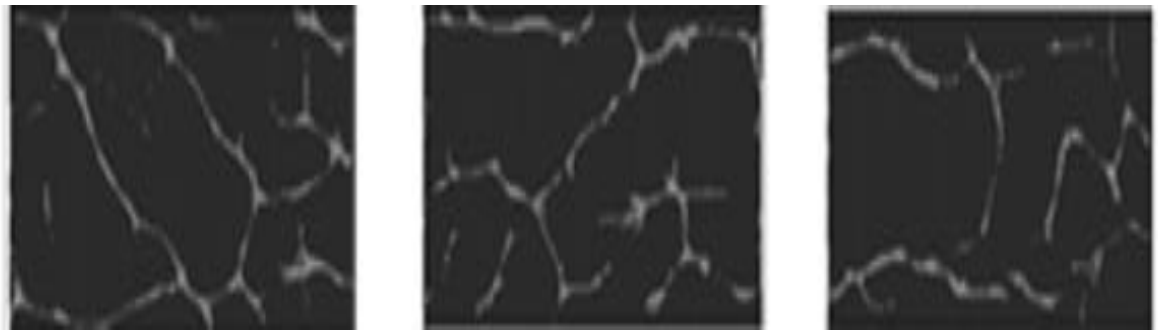
$$C\ (x, y) = \{(x', y') \mid distance\ (x, y, x', y') \leq threshold\}$$

Step 4. Label the Image:

Create a labelled image L (x, y)

$$L\ (x, y) = \begin{cases} 1\ if\ (x, y)\ \varepsilon\ C(x, y), \\ 0 \qquad otherwise \end{cases}$$

Figure 4.1 shows the sample output of the LMC algorithm for three different FV images.



a.THUFV1_Train1.bmp b. THUFV1_Train2.bmp c. THUFV1_Train3.bmp

Figure 4.1 Sample Outputs of Local Maximum Curvature Algorithm

4.3.2 Wide Line Detector (WLD) Algorithm

The WLD algorithm (Liu L. et al., 2007) is designed to detect veins by examining the local brightness contrast within a specified neighbourhood around each pixel. The process starts by loading the input image. An output image is then initialized, which finally holds the detected vein structure. For each pixel in the input image, a circular mask with a defined radius is created to specify the neighbourhood around that pixel. This neighbourhood is used to assess the local contrast. The algorithm then calculates a similarity measure for each pixel by comparing its brightness with the brightness of the central pixel within the mask. If the difference in brightness falls within a given threshold, the algorithm considers it a similar pixel. The sum of these similarity values, called mass, is computed for each central pixel.

The mass indicates how much local similarity exists between the central pixel and the surrounding pixels. It then checks whether the mass exceeds a predefined threshold value, which represents the presence of a vein in that location. The threshold is calculated as a percentage of the neighbourhood size, as shown in equation 4.1:

$$g = \alpha \times N \tag{4.1}$$

where $\alpha \in [0.7, 0.9]$, the similarity value set by the user, and N is the number of pixels in the circular mask. The output pixel is then set to a high value to indicate the presence of a vein. But if the mass is below the threshold, the pixel is set to zero, eliminating non-vein areas. The output image contains the detected vein structures. The method is described in Algorithm 4.2.

Algorithm 4.2 Algorithm for Wide Line Detector (WLD)

Step 1. Initialization:

Step 1.1 Load the input finger vein image V.

Step 1.2 Initialize the output image O with the same dimensions as V.

Step 2. Neighbourhood Analysis:

Step 2.1. For each pixel (x_c, y_c) in V:

Step 2.1.1 Define a circular mask centred at (x_c, y_c) with radius r.

Step 3. Similarity Calculation:

Step 3.1 For each pixel (x_i, y_i) in the neighbourhood of (x_c, y_c):

Step 3.1.1 Calculate the similarity measure S using the equation:

$$S = \begin{cases} 1\,, if\ |V(x_i, y_i) - V(x_c, y_c)| \leq t \\ 0\,, \qquad\qquad Otherwise \end{cases}$$

Step 3.1.2 Sum the similarity values to calculate the mass m (x_c, y_c):

$$m\ (x_c, y_c) = \sum (x_i, y_i) \in neighbourhood\ S$$

Step 4. Output Image Calculation:

Step 4.1 for each pixel (x_c, y_c) in V:

Step 4.1.1 Calculate the output pixel value $O(x_c, y_c)$ using the following equation

$$O\ (x_c, y_c) = \begin{cases} 255\,, & if\ m(x_c, y_c) > g \\ 0\,, & Otherwise \end{cases}$$

Step 5. Return the output image O, which contains the detected vein patterns.

Figure 4.2 shows the sample output of WLD algorithm for three different FV images.

a.THUFV1_Train1.bmp b. THUFV1_Train2.bmp c. THUFV1_Train3.bmp

Figure 4.2 Sample Outputs of the Wide Line Detector Algorithm

4.3.3 Repeated Line Tracking Algorithm (RLT)

The RLT algorithm (Miura N. et al., 2004) is designed to track and analyze the movement along dark lines within a finger vein image. It traces the continuous vein lines by following the local intensity changes and identifying patterns that correspond to vein structures.

The process starts by detecting the movement of a point along a dark line, which is a lower intensity value associated with vein patterns compared to the surrounding tissue. The algorithm initializes a locus space that records the path of the tracking point. The tracking point moves through neighbouring pixels by excluding the points that have been tracked previously. A line evaluation function is used to choose the next pixel for tracking. This evaluates the intensity dips around the current tracking point. It also analyzes cross-sectional profiles to determine the most likely next step in the tracking process. The pixel that best fits the evaluation criteria is selected as the next tracking point.

Algorithm 4.3 Repeated Line Tracking (RLT) Algorithm

Step 1: Identify the starting point of the next movement to track its direction and line.

Step 2: Trace the path followed by the tracking point by detecting the direction of the dark line using the following steps:

Step 2.1: Initialize the locus space Tc.

Step 2.2: Locate the pixels that are neighbours to the current tracking point:

$$Neighbours = \{(x + dx, y + dy) \mid (dx, dy) \in neighbourhood\ offsets\}$$

Step 2.3: Locate a dark line that indicates the direction in which the tracking is taking place:

$$Direction = argmin\ (Intensity\ (Neighbours))$$

Step 2.4: After the locus has been registered in the locus table, the tracking pointer should be moved:

$$(x\_new, y\_new) = (x\_current + \Delta x, y\_current + \Delta y)$$

where Δx and Δy denote the changes in the x and y coordinates based on the direction

Step 2.5: Repeat Steps 2.2 to 2.4.

Step 3: The tracking count for each point within the locus space should be updated:

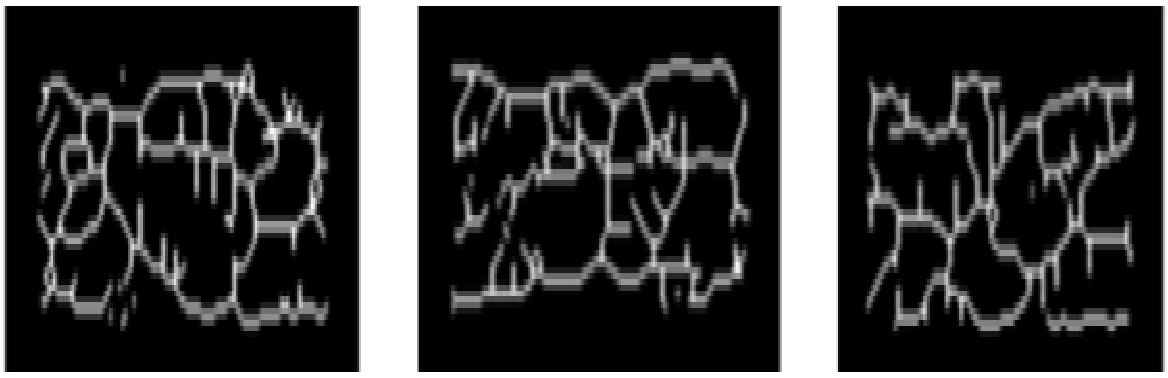
$$Tc(x, y) = Tc(x, y) + 1$$

Step 4: Repeat Steps 1 to 3, n times.

Step 5: Using the locus space, determine the vein patterns:

$$Vein\ Pattern = \{(x, y) \mid Tc(x, y) > Threshold\}$$

As the algorithm progresses, the locus table is updated with each new tracking point, reflecting the path followed and the direction taken. This process continues for about 30000 iterations, starting from random points within the image. The locus space also records the frequency of visits to each point. Finally, the algorithm identifies the vein patterns by selecting points where the tracking count exceeds a predefined threshold. The method is described in Algorithm 4.3. Figure 4.3 shows the sample outputs of the RLT method for three different images.



a.THUFV1_Train1.bmp b. THUFV1_Train2.bmp c. THUFV1_Train3.bmp

Figure 4.3 Sample Outputs of Repeated Line Tracking Algorithm

4.3.4 Hybrid Algorithm for Labelling

The hybrid algorithm combining the LMC, RLT, and WLD offers several advantages in FVR tasks by utilizing the complementary power of each method. Each of the algorithms is designed to identify vein patterns in different ways. The LMC algorithm emphasizes the central lines of veins by analyzing the curvature of intensity profiles, which detects vein centres exactly. The RLT algorithm detects continuous dark lines by tracking the path along these lines. The WLD algorithm focuses on analyzing local brightness contrasts, which helps detect veins based on intensity differences between veins and surrounding tissue.

By combining the outputs of LMC, RLT and WLD, the algorithm ensures that vein pixels are confirmed through a consensus among the three methods. If any of the algorithms detect a vein in a pixel, it increases the confidence that the pixel corresponds to a vein, while pixels that do not meet the criteria across all three methods are discarded as non-vein areas. This minimizes the likelihood of both false-positive and false-negative outcomes.

Algorithm 4.4: Hybrid Algorithm for Labelling

Step 1: Initialize Inputs

Step 1.1. Obtain the output pixel values from MC, WLD, and RLT methods.

Let MC (x, y) represent the pixel values obtained from the MC method.

Let WLD (x, y) represent the pixel values obtained from the WLD method.

Let RLT (x, y) represent the pixel values obtained from the RLT method.

Step 2: Merge Pixel Values

Step 2.1. Initialize an empty matrix for the resultant vein pattern.

Let L (x, y) be the resultant pixel value in the vein pattern.

Step 2.2. For every pixel position (x, y) in the image:

Calculate the resultant pixel value by summing the corresponding pixel values from MC, WLD, and RLT:

$$L(x, y) = MC(x, y) + WLD(x, y) + RLT(x, y)$$

Step 3: Identify Vein Pixel using the equation,

$$L(x,y) = \begin{cases} 1 & \sum_{i=1}^{n} C_i(x,y) > 1 \\ 0 & \sum_{i=1}^{n} C_i(x,y) \leq 1 \end{cases}$$

Step 4: Generate Vein Pattern by creating a binary image where vein pixels are marked based on the classification from Step 3.

The algorithm operates by initializing an image with zero values, having the same dimensions as the input images that are generated by the three algorithms. For each pixel (x, y), the summation of the pixel values from the three input images is computed. If the

sum exceeds 1, the pixel in the resultant image is set to 1, marking it as part of a vein. If the sum is 1 or less, the pixel is set to 0, indicating that it does not belong to the vein pattern. This pixel-wise fusion ensures that the final vein pattern is accurate and represents the maximum number of vein pixels while excluding irrelevant areas. By utilizing the complementary strengths of LMC, RLT and WLD, the hybrid algorithm detects more vein patterns.

The output of hybrid algorithm for the samples shown for the three algorithms LMC, RLT and WLD are given in Figure 4.4.

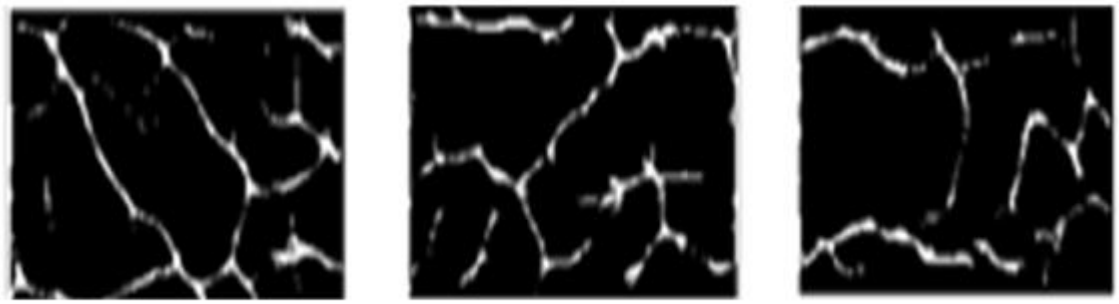


Figure 4.4 Sample Outputs of the Hybrid Algorithm

4.4 RESULTS AND DISCUSSIONS

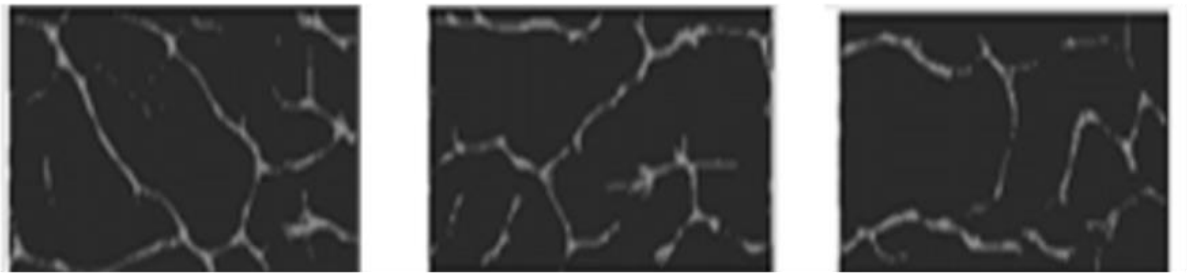
The comparison of the results produced by the LMC, RLT, WLD, and hybrid algorithms is analyzed. Visual inspections of the output images suggest a resemblance between the results of the LMC and hybrid algorithms, while the outputs of others vary. However, a more rigorous evaluation using various similarity metrics reveals significant dissimilarities between the outputs of the LMC and the hybrid algorithm. These measurements offer a numerical way to evaluate performance, proving that despite the apparent visual similarities, the underlying data and results of the LMC and hybrid algorithm differ across almost all evaluated aspects.

4.4.1 Visual Comparison of Labelled Images

The difference in the outputs of WLD, RLT and LMC can be inferred qualitatively by the visual comparison of the images. Figure 4.5 shows the outputs of the above three algorithms. The output of the hybrid algorithm is also displayed by comparing it with the other outputs.

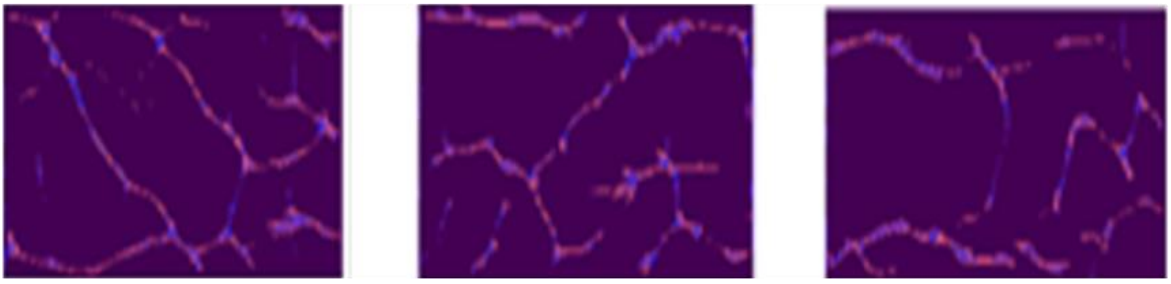
a. Output of Wide Line Detector algorithm

b. Output of Repeated Line Tracking algorithm



c. Output of Local Maximum Curvature algorithm

d. Output of Hybrid algorithm



e. Image showing the difference (blue colour) between LMC and merged output

Figure 4.5 Output of Different Algorithms for Three Sample Images

Figure 4.5a shows the result of applying the WLD algorithm to three different sample images. WLD is designed to detect wide, continuous lines, which are often indicative of vein patterns. Since veins typically appear as dark, continuous lines against lighter tissue, the WLD algorithm is effective in identifying the vein structures. However, due to its focus on wide lines, WLD may miss finer vein details or smaller vein branches.

Figure 4.5b shows the result of the Repeated Line Tracking (RLT) method. The RLT algorithm works by repeatedly tracking lines across the image. This method is better at detecting fine veins that may be missed by WLD, as it emphasizes the continuity of vein patterns. The output shows finer vein details and smoother line connections. However, RLT can sometimes be sensitive to noise, leading to the detection of irrelevant lines in some cases.

Figure 4.5c shows the result of the Local Maximum Curvature (LMC) method. LMC identifies vein patterns by detecting areas where the curvature of intensity profiles is maximized, which are characteristic of vein structures. This method is effective at finding the exact location of vein centres. High-curvature regions are visible in the output. LMC is good at detecting the central vein lines but may not capture vein structure if the veins are not perfectly aligned or exhibit proper curvature.

Figure 4.5d presents the merged output of the three methods: WLD, RLT and LMC. The outputs from these three methods are combined to create a better vein pattern image. The merged output indicates the areas where all three methods have successfully identified vein structures. The strengths of the three methods, wide lines by WLD, fine details by RLT, and precise curvature by LMC, are combined in the hybrid output. The hybrid result is a more accurate and complete representation of the vein patterns.

Even though the output of the LMC and hybrid algorithm looks similar, Figure 4.5e shows the variation between the LMC output and the merged output. The blue pixels indicate the additional vein pixels that were detected through the hybrid approach. The merged method minimizes errors such as missed veins or misclassified areas, ensuring a more accurate identification of vein patterns. This comparison stresses the effectiveness of combining multiple algorithms to enhance vein detection performance.

4.4.2 Evaluation of Similarity Metrics

The maximum curvature and merged outputs appear visually similar, but the evaluation of similarity metrics reveals that they are not identical and differ from each other. Additionally, the results from other algorithms show greater differences compared to the merged image. This suggests that the merged image contains more features, resulting from the combined effect of the three algorithms, which is likely to enhance recognition performance. The various similarity metrics considered are:

Mean Square Error (MSE) evaluates how much two images differ by computing the mean squared difference between the corresponding pixel values. It works by comparing every pixel in the reference image to its counterpart in the processed or altered image. The MSE can be calculated using the equation 4.2.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(xi - yi)^2 \tag{4.2}$$

Here, xi and yi represent the corresponding pixels in the reference image and the distorted image, respectively, and n is the total number of pixels in the image.

If the value is small, it implies that the images are more similar, and a zero value signifies that the two images are perfectly identical.

Root Mean Square Error (RMSE) is the square root of the MSE. RMSE is given by the equation 4.3.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(xi - yi)^2} \tag{4.3}$$

RMSE is easier to interpret than MSE because it directly corresponds to the pixel intensity difference between images. A small value implies higher similarity between the images.

Structural Similarity Index Measurement (SSIM) is designed to compute perceived visual resemblance between two images, considering luminance, contrast, and structural information. SSIM is calculated using the equation 4.4.

$$SSIM(x, y) = \frac{((2\,\mu_x\mu_y + C1)\ (2\,\sigma_{xy} + C2))}{((\mu_x{}^2 + \mu_y{}^2 + C1)\ (\sigma_x^2 + \sigma_y^2 + C2))} \tag{4.4}$$

where, $\mu_x\ and\ \mu_y$ represent the mean pixel values of images x and y, respectively.

σ_x^2 *and* σ_y^2 represents the variances of x and y respectively,

σ_{xy} represents the covariance shared between the images,

C1 and C2 are small constants to prevent instability.

SSIM scores range between -1 and 1, where a value of 1 means perfect similarity between the images. Unlike MSE and RMSE, SSIM captures how alterations in the image's structure influence perceived quality, aligning the assessment more closely with how humans visually interpret images.

Spectral Angle Mapper (SAM) or Spectral Similarity Measurement is a metric used in image similarity, particularly in hyperspectral imaging, where each pixel contains a spectrum of data across multiple wavelengths. SAM measures the angle between the spectral vectors of matching pixels from the two images and is represented in the equation 4.5.

$$SAM(x, y) = \cos^{-1}\left(\frac{\sum_{j=1}^{n} x_j y_j}{\sqrt{\sum_{j=1}^{n} x_j{}^2}\sqrt{\sum_{j=1}^{n} j^2}}\right) \quad (4.5)$$

where, x_j, y_j are the spectral vectors of the matching pixels from the two images, n is the number of spectral bands.

A smaller angle (SAM) indicates greater similarity between the spectra of the two images. SAM is specifically useful in remote sensing and medical imaging, where spectral information is crucial.

Visual Information Fidelity (VIF) evaluates the degree to which visual content is preserved from the original, based on natural scene statistics. It measures the extent to which information from the reference image is retained in the distorted image. VIF is calculated using the equation 4.6.

$$VIF = \frac{Information\ in\ the\ reference\ image}{Information\ in\ the\ distorted\ image} \quad (4.6)$$

VIF is calculated using complex wavelet-based models of natural images, where higher VIF values (close to 1) indicate better preservation of visual information and thus higher similarity.

MSE and RMSE offer straightforward pixel-level comparisons, while SSIM considers perceptual aspects, SAM is suited for spectral comparisons, and VIF measures the fidelity of visual information.

Table 4.1 shows the comparison of similarity metrics of the merged output and individual algorithm's outputs.

Table 4.1 Comparison between the Merged Output and the Output of Individual Algorithms

Metric	Ideal scores (No Similarity)	Algorithm		
		LMC vs Merged	WLD vs Merged	RLT vs Merged
MSE	0.0	0.0173	0.0677	0.0867
RMSE	0.0	0.1314	0.2601	0.2944
SSIM	1.0	0.3985	0.5536	0.4064
SAM	0.0	0.2101	1.5659	0 .5382
VIF	1.0	0.4705	0.0214	0.0021

From Table 4.1, on observing the LMC and merged outputs, a closer visual match between the maximum curvature and merged output can be noticed. However, it is evident from the table that they are not identical but are distinct from each other. Furthermore, the results of the other algorithms differ even more from the merged image. This indicates that the merged image encompasses a broader range of features, resulting from the combined effects of the three algorithms. Such a composite image integrates diverse aspects captured by each algorithm, thereby enhancing the overall feature set. Similarity scores between individual outputs and the merged output underscore the efficiency of the hybrid technique in obtaining unique and complementary information from each algorithm. This result validates the approach, demonstrating that the merged image benefits from the integration of multiple algorithms, leading to improved recognition capabilities.

4.4.3 Performance Evaluation of the Labelled Images using VGG16

The labelled images are utilized for training a VGG16 network to analyze the performance with non-labelled images. The process starts with the gathering and labelling of FV images to identify the vein patterns and create a training dataset. These labelled

images are resized to the VGG16 input size and then normalized. Non-labelled original images are used for testing. The VGG16 model is initialized with pre-trained weights, and it uses a binary cross-entropy loss function and RMSProp optimizer. The model's output layer is modified to align with the number of classes in the dataset. While training, the pre-processed and labelled images along with their associated labels are input to the model, along with a validation set to track the performance and avoid overfitting. For testing, the pre-processed non-labelled images are given to the trained model to obtain predictions.

a. THUFV Dataset

Figure 4.6 shows the accuracy curve of VGG16 when using labelled FV images.

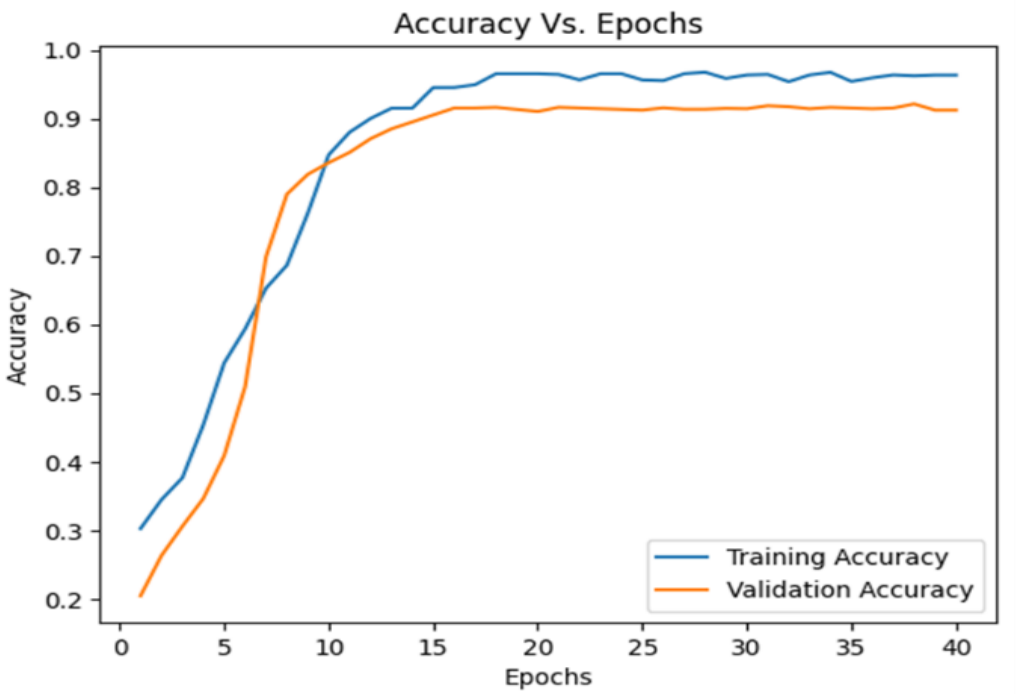


Figure 4.6 Accuracy Curve of VGG16 on Labelled THUFV Dataset

From Figure 4.6, it is evident that the training accuracy steadily improves and stabilizes around 0.96, while the validation accuracy also rises, stabilizing slightly below the training accuracy at around 0.90.

Figure 4.7 shows the loss behaviour of the VGG16 model on the labelled dataset.

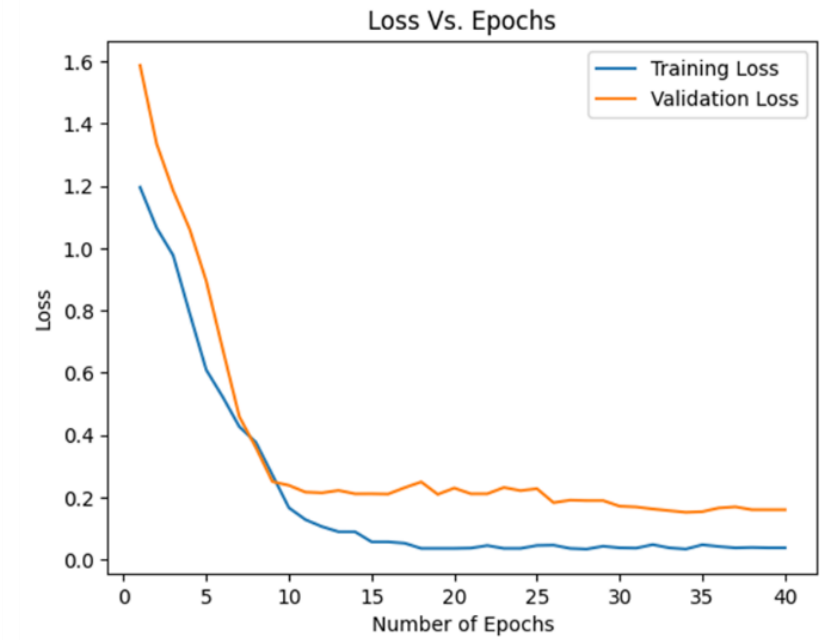


Figure 4.7 Loss Curve of VGG16 on Labelled THUFV Dataset

The training loss drops rapidly and levels off close to zero, while the validation loss also decreases significantly but stabilizes at a slightly higher value than the training loss. This trend signifies that the model is effectively minimizing loss during training and performing well on the validation set, although the gap between loss on the training and validation sets indicates overfitting has been reduced when compared to training with non-labelled images.

The performance comparison of the VGG16 model on the original dataset and the Labelled THUFV dataset (L-VGG16), shown in the Table 4.2, reveals notable improvements across all evaluation metrics.

Table 4.2 Performance Comparison of VGG16 on Original Dataset and Labelled THUFV Dataset

Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
VGG16	92.5	94.09	97.64	95.83
L - VGG16	96.34	98.29	97.87	98.08

In terms of accuracy, the model achieves a significant increase from 92.5% on the original dataset to 96.34% on the labelled dataset, marking a 4.15% improvement. This indicates that the addition of labelled data helps the model make more precise predictions. Precision also shows a marked improvement, rising from 94.09% to 98.29% with 4.46% increase. This suggests that the labelled dataset helps the model to become more accurate in identifying true vein patterns, reducing false positives. There is a slight improvement in recall as well, from 97.64% to 97.87. The F1 score rises from 95.83% to 98.08%, with an improvement of 2.35%. The improvements in different metrics show that labelling the dataset improves the performance of the model.

b. SDUMLA-HMT Dataset

The performance of the accuracy of VGG16 on the labelled SDUMLA dataset is shown in Figure 4.8. The model converged by about 25 epochs. Both training and validation accuracy improved over time, with training accuracy reaching around 95% and validation accuracy stabilizing around 90%.

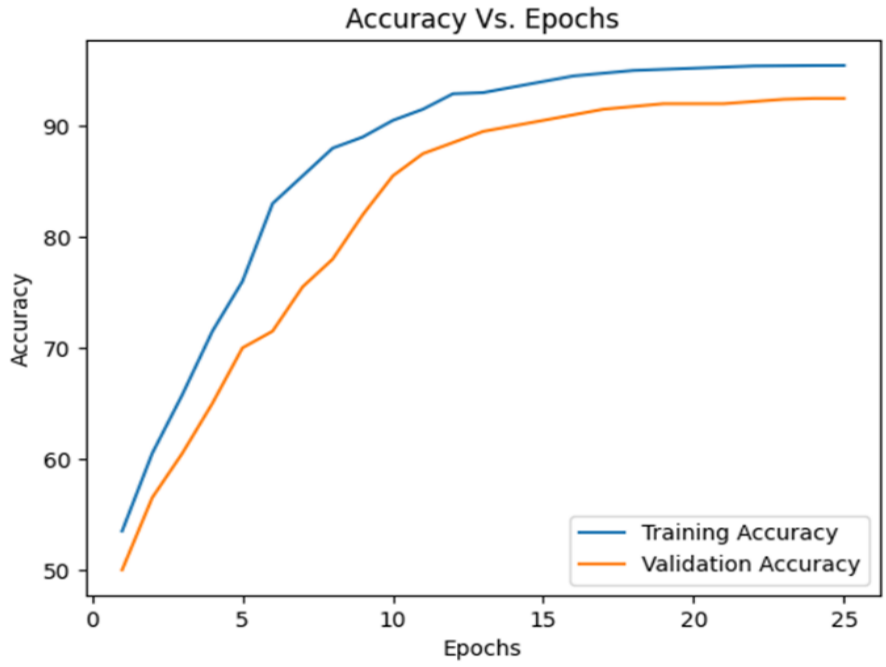


Figure 4.8 Accuracy Curve of VGG16 on Labelled SDUMLA Dataset

The Figure 4.9 shows a decreasing trend in both training and validation loss as the epochs progress.

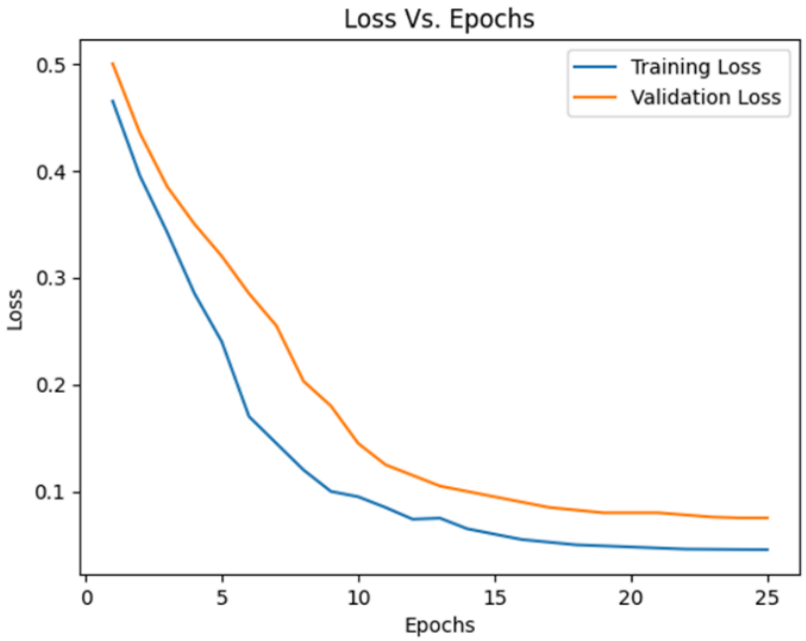


Figure 4.9 Loss Curve of VGG16 on Labelled SDUMLA Dataset

Initially, both losses drop sharply, with the training loss eventually falling below the validation loss and stabilizing at a lower value. This suggests that the model is learning effectively, aligning closely to the training data, and generalizing well to the validation data. With the use of a labelled dataset, overfitting has been reduced in this case also when compared to the model trained on the original dataset discussed in chapter 3.

The performance of VGG16 on the original dataset and the labelled SDUMLA-HMT dataset are compared in Table 4.3.

Table 4.3 Performance Comparison of VGG16 on Original Dataset and Labelled SDUMLA Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
VGG16	94.31	93.60	92.07	92.83
L-VGG16	95.45	94.78	96.76	95.76

The improvement in performance of the VGG16 model on the labelled dataset is notable across all metrics. Accuracy increases from 94.31% to 95.45%, representing an

increase of approximately 1.21%. Precision increased from 93.60% to 94.78%, which is about a 1.26% improvement. The most significant enhancement is observed in the recall, which rises from 92.07% to 96.76%, reflecting a substantial increase of around 5.10%. The F1-score also improves from 92.83% to 95.76%, with an increase of 3.16%.

The increase in recall and F1-score indicates that the addition of labelling increases the capacity of the model to recognize positive instances correctly and maintain a balance between precision and recall. This suggests that the labelling has effectively improved the model to identify true positives while ensuring that the precision and recall are balanced.

4.5 SUMMARY

The finger vein dataset is labelled using the hybrid algorithm by combining the LMC, RLT and WLD algorithm outputs. The performance of the VGG16 dataset is then evaluated using this labelled algorithm. The accuracy, precision, recall, and F1 score have been improved to 95.45%, 94.78%, 95.76% and 95.76% respectively, after labelling the dataset. Reduction in overfitting can also be noted from the accuracy and loss graphs. Based on these improvements, the hybrid algorithm is recommended for labelling finger vein images.