FINGER VEIN BASED BIOMETRIC IDENTIFICATION USING CONVOLUTIONAL NEURAL NETWORK

A PROJECT REPORT

Submitted by

SAQUIB NAWAZ (170081601035)
VARUN KUMAR K (170081601038)

Under the guidance of

Dr. G. KAVITHA, B.E., M.E., Ph.D

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

B.S. Abdur Rahman
Crescent
Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956

JULY 2021

B.S. Abdur Rahman
Crescent
Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956

BONAFIDE CERTIFICATE

Certified that this project report “**FINGER VEIN BASED BIOMETRIC IDENTIFICATION USING CONVOLUTIONAL NEURAL NETWORK”** is the bonafide work of **SAQUIB NAWAZ (170081601035)** and **VARUN KUMAR K (170081601038)** who carried out the project work under my supervision. Certified further, that to the best of our knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. G. KAVITHA

SUPERVISOR

Associate Professor

Department of IT

B.S. Abdur Rahman Crescent

Institute of Science & Technology

Vandalur, Chennai – 600 048

SIGNATURE

Dr. I. SATHIK ALI

HEAD OF THE DEPARTMENT

Associate Professor & Head

Department of IT

B.S. Abdur Rahman Crescent

Institute of Science & Technology

Vandalur, Chennai – 600 048

VIVA-VOCE EXAMINATION

The viva voce examination of this project work titled "**FINGER VEIN BASED BIOMETRIC IDENTIFICATION USING CONVOLUTIONAL NEURAL NETWORK**" submitted by **SAQUIB NAWAZ (170081601035)** and **VARUN KUMAR K (170081601038)** was held on _______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First of all, we would like to thank the **Almighty** for converting our efforts into best outcomes.

We sincerely thank **Dr. A. PEER MOHAMED, Vice Chancellor** and **Dr. A. AZAD, Registrar,** for furnishing every essential facility for doing our project. We would also like to express our sincere gratitude to **Dr. VENKATESAN SELVAM**, **Dean**, School of Computer, Information and Mathematical Sciences, for his support.

We thank **Dr. I. SATHIK ALI, Associate Professor and Head**, Department of Information Technology, for his eminent support and encouragement throughout the project. We sincerely express our gratitude to our supervisor **Dr. G. KAVITHA**, **Associate Professor/ IT**, for her total support and guidance throughout our project.

We thank all our review panel members, **Dr. P. LATCHOUMY, Associate Professor / IT, Mr. N. RAJENDRAN**, **Assistant Professor (Senior Grade) / IT** and the faculty members of the department for their time and valuable suggestions which gave us the confidence to complete our project work. We extend our heartfelt thanks to our Project Coordinator **Dr. P. LATCHOUMY, Associate Professor / IT** for her constant motivation and encouragement which got us to complete the project successfully on time.

We would also like to thank our dear parents and friends who directly or indirectly helped us in the completion of our project.

SAQUIB NAWAZ

VARUN KUMAR K

ABSTRACT

Biometric recognition technology has procured an extremely high observation due to use of traits for the purpose of reducing fraud, enhancing public safety and national security. Finger-vein biometric system is a mode of biometric recognition that employ pattern authentication techniques that is presumed on the picture of finger vein patterns. Human recognition systems are innately selective, and hence fundamentally fallible. In modern years, it has been demonstrated that the equal error rate (EER) of finger vein recognition system may reach 0.0008% under certain conditions.

In traditional finger-vein recognition system, the recognition depends on the finger-vein dark lines that is selected from the input vein images. But, the incorrect detection of finger-vein lines reduces the recognition accuracy. Convolutional Neural Network (CNN) model is implemented to obtain improved performance in terms of speed of processing of vein images with improved accuracy which can make the recognition process easy and efficient with the help of finger vein datapoints. The proposed work can fix the problem of image rotation, scale transformation and illumination change leading to an improved performance on finger vein recognition. The algorithm of FVBI (Finger Vein Biometric Identification) is used in this project to make the process much easier. The proposed model has achieved an accuracy of 95% for CNN (Convolutional Neural Networks) which has much higher accuracy than the existing model (SVM). Thus, the proposed CNN approach has 5% to 10% higher accuracy than the existing system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Networks
DL	Deep Learning
EER	Equal Error Rate
FVBI	Finger Vein Biometric Identification
GLCM	Gray Level Co-occurrence Matrix
GUIDE	Graphical User Interface Development Environment
LBP	Linear Binary Pattern
NIR	Near InfraRed
NLP	Natural Language Processing
ROI	Region of Interest
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 GENERAL

1.1.1 FINGER VEIN BIOMETRIC METHOD

In recent years, fingerprints are being widely used and it is one of the biometric methods which shows great accuracy [1]. It is because we can easily acquire the fingerprints from human body. However, it has some major drawbacks such as degradation of epidermis of the fingers, particles on finger surface etc., that resulted in inaccuracy. Hence, we need to look for different method which is more robust and accurate [2]. Now, Finger vein comes into picture. Vein authentication technology offers an assuring solution to the challenges of fingerprint biometric methods, and it has following characteristics such as uniqueness and universality. In most cases, the vein images remain unaffected even with ageing.

Finger vein recognition system is a method of biometric authentication that process the finger vein image, so that we can find the accurate information (vascular patterns and data-points) of that particular individual [3]. It uses NIR (Near InfraRed) light imaging in order to detect the finger veins. In our finger vein, the hemoglobin in the blood absorbs the NIR light from the vein authentication device, which makes the vein image to appear as dark lines of pattern. We use both index and middle fingers for vein authentication. In case, if the quality of the vein image is not good in either finger, then we can go for the thumb finger for identification and verification.

Many biometric methods such as facial recognition, iris recognition, palm vein recognition, etc, are available. But the finger vein authentication has better False Acceptance Rate (FAR) than any other biometric methods. Finger vein biometric method is one of the best and highly accurate method to reduce forgery and increase security.

APPLICATIONS OF FINGER VEIN BIOMETRIC METHOD

- Automated Teller machine.
- Vein Scanner.
- Security in end points.
- Door Access systems.
- Credit Card Authentication.

1.1.2 IMAGE PROCESSING SYSTEM

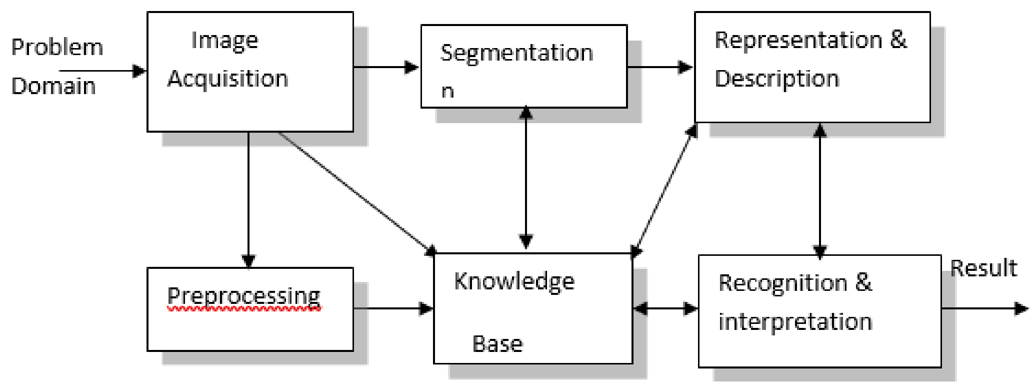


Figure 1.1 Image Processing System

From the above block diagram, the first task in the refinement is image acquisition by an imaging sensor in coincidence with a digitizer to digitize the image. In pre-processing, the input image has a fairly poor pixel level and it has extensive amount of noise. In order to reduce it, we need to enhance the quality of the image by tuning the appearance of the image, contrast and color. Segmentation partitions an image into its dividing parts or objects. The output of segmentation is usually natural pixel data, which consists of either the barrier to the sector or the pixels in the region themselves. Representation is the process of converting the raw pixel data into a form useful for substantial processing by the computer. Description is a process that helps to bring out features that can modify one class from another.

Recognition provides a label to an object based on the instruction provided by its descriptors. The knowledge about a problem domain is fused into the knowledge base. The knowledge base guides the operation of each processing module and also controls the communication between the modules. Not all modules need be necessarily present for a particular function.

In the recognition and interpretation step, the image will be checked with the database, whether it is available or not. The frame rate of the image processor is nearly about 25 frames/sec.

1.1.2 IMAGE PROCESSING TECHNIQUES

Below figure 1.2, tells about the image processing techniques as follows:

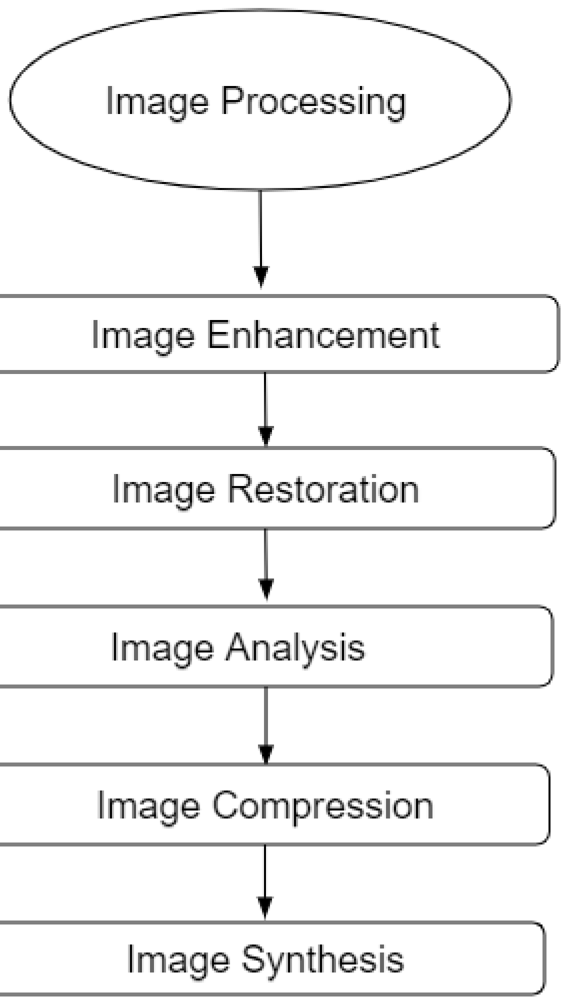


Figure 1.2 Image Processing Techniques

IMAGE ENHANCEMENT

In this operation, we are enhancing the image quality by changing the image contrast and brightness, removing noise. It just carries out the process of refinement in the image and it shows the same image but in a more comprehensible image. It will not include any extra information on the image.

IMAGE RESTORATION

We are using the image restoration to remove the noise and bring back quality image. Some images are restored with some problems like repetitive noise, camera motion, blurring, inappropriate focus and analytical distortion. Sometimes, this step is similar to enhancement but with some changes in the original image.

IMAGE ANALYSIS

In image analysis, we need to completely examine the quality and other characteristics of the image. The operations in image analysis make graphical or numerical information that is solely based on the attributes of the actual image. It usually broken down into objects and then, we can classify them. They are mainly used in industrial applications. Some of the common operations involved in image analysis includes extraction, automated and object classifications. Image analysis depends on image data.

IMAGE COMPRESSION

In image compression, it decreases the data content which is mandatory to express an image. Most of the cases, lots of images have very high redundant information. So, this can be removed by using image compression. It helps to reduce the size of the image and hence, the image is systematically stored.

IMAGE SYNTHESIS

In image synthesis, it provides images from different image data. Generally, it has the ability to generate new images which may be either impossible to acquire or non-viable.

The aim of the project is to enhance the public safety and reduce the fraud and also to increase the accuracy of authentication with the proposed Convolutional neural network (CNN) approach.

1.1.3 CONVOLUTIONAL NEURAL NETWORK

An Artificial Neural Network (ANN) is a computational model that is based on the structure and functions of the neural network of the human brain. It is an artificial network that is capable of receiving input, processing it and sending the output. An artificial neural network is built with multiple artificial neurons. Each neuron is a separate computational unit.

A kind of ANN that is used in our study is CNN (Convolutional neural network), which is mainly used in processing of data and recognition of images, so that they are helpful in designing pixel data. They are efficiently used in deep learning (DL), digital image processing and machine learning, in order to execute illustrative work, with the help of image and video recognition machine vision.

The essential processing of data needs to be reduced by designing CNN like multilayer perceptron, so that it will make the computation process much faster and easier. It has three layers namely input layer, hidden layer and output layer and other layers includes convolution layer, pooling layer (Average pooling, sum pooling, max pooling), fully connected layer, and standardization layer.

In order to make the system more reliable, effective and simple, we need to remove the shortcomings (false matching) and improve the image processing speed in CNN, which are restricted in NLP and image processing.

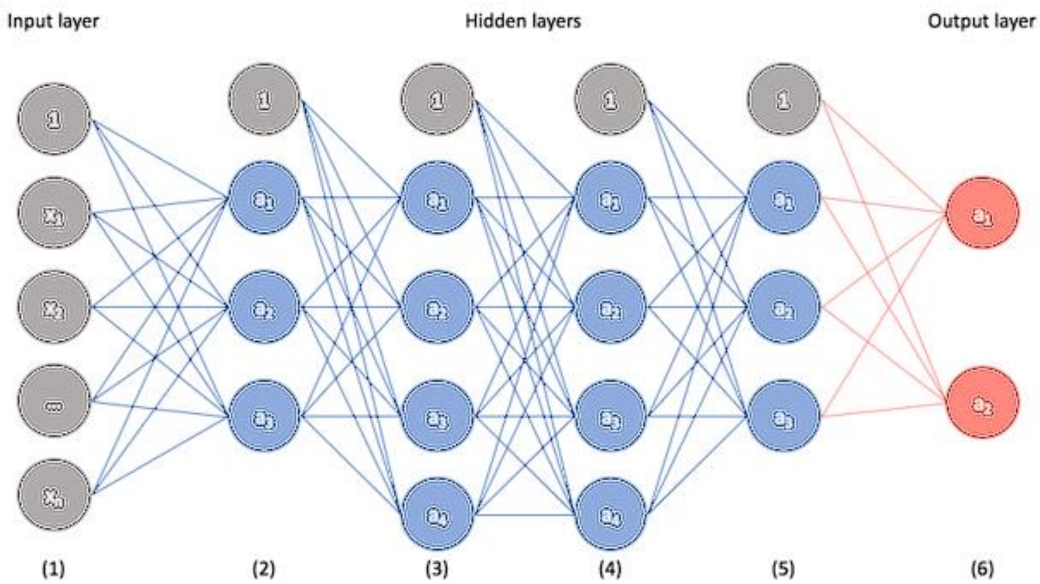


Figure 1.3 Convolutional Neural Network

1.1.5 APPLICATIONS OF FINGER VEIN BIOMETRIC METHOD

The wider range of applications in finger vein biometric method includes automated teller machine, vein scanner, security in end points, door access systems, credit card authentication. It is enumerated below.

AUTOMATED TELLER MACHINE

ATM (Automatic Teller Machine) is a banking terminal that accepts deposits and dispenses cash. ATMs are activated by inserting cash (in cases of ATM Depositing) or debit /credit card that contain the user's account number and PIN on a magnetic stripe (for cash withdrawals). It allows customers to complete basic transactions without the aid of a branch representative or teller.

VEIN SCANNER

Vein scanners use near-infrared light to reveal the patterns in a person's veins. As with irises and fingerprints, a person's veins are completely unique. A camera takes a digital picture using near-infrared light. The hemoglobin in your blood absorbs the light, so veins appear black in the picture.

SECURITY IN END POINTS

Endpoint security is the practice of securing endpoints or entry points of end-user devices such as desktops, laptops, and mobile devices from being exploited by malicious actors and campaigns. Endpoint security systems protect these endpoints on a network or in the cloud from cybersecurity threats.

DOOR ACCESS SYSTEMS

Door access control systems are digital security systems that ensure authorized access to your building. They ensure only authorized personnel are entering your building while keeping unauthorized personnel out. At the same time, door access control systems simplify entry to your building.

CREDIT CARD AUTHENTICATION

Credit card authentication is the process of confirming the validity of a customer's credit card by checking with the company that issued the card. Authentication is roughly the first half of the transaction process when using a credit card

1.2 PROPOSED SYSTEM

In this project, it is proposed to design a finger vein biometric authentication system for detecting the fraud using an enhanced version of Convolutional Neural Network (CNN). The main proposal of this version of CNN is to find the compound gap between the training and testing dataset, and to improve the separating hyperplane in the finger vein. For the unknown vein samples, it provides better classification and projection capability. The central purpose of CNN is to perform every attempt to build up the dissociation between the two different categories, in order to make the dissociation more reliable. The importance of this dissociation is that we can easily make difference in finding the training dataset and test dataset. GLCM (Grey level co-occurrence matrix) method enhances the segmented image. To indicate the presence of an object on a finger vein, we need to perform marker-controlled watershed segmentation process that intensifies the region and break down into different blocks. In an image, separation process is very hard process, and it takes too long time to complete the task.

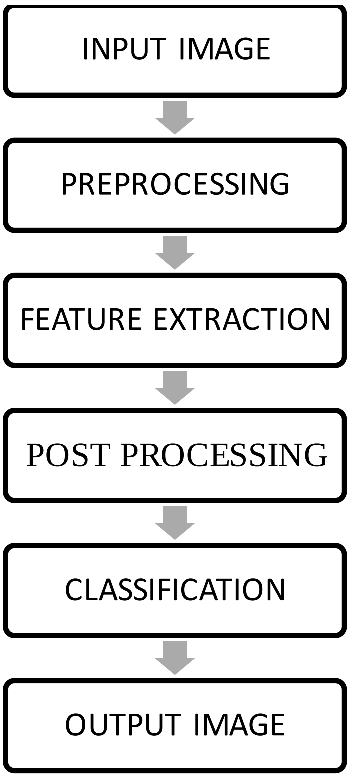


Figure 1.4 Proposed System Workflow - Flowchart

The figure 1.4 represents the work flow of proposed system, where we fed the finger vein image as an input image and then we need to do the pre-processing to enhance the quality of a finger image. Once the pre-processing step is complete, then do the feature extraction step to define a set of features or characteristics to analyze the finger vein image. In classification, proposed CNN is applied with ReLU (rectified linear unit) to find whether the input image is recognized or not. The final output is the number of recognized image from the database.

CHAPTER 2

LITERATURE OVERVIEW

The Convolutional Neural Network for Finger-Vein-based Biometric Identification is proposed by Rig Das, Emanuela Piciucco, Emanuele Maiorana, and Patrizio Campisi [1]. It focuses on developing the CNN based finger vein recognition system by using 3 publicly-available databases. The original images are pre-processed for extraction of Region of Interest and a binary mask for a specific finger region is taken. The CNN is trained with a low learning rate and hence the network learns and converges slowly. The work has proved to have around 70% to 80% of matching probability which can be further improved. The performance of the system can be further improved by choosing training samples in multiple sessions. The training accuracy of the finger vein images can further be improved by an effective strategy for image identification.

The User Identification system based on finger vein patterns using Convolutional neural network is proposed by K. S. Itqan, A. R. Syafeeza, F. G. Gong, N. Mustafa, Y. C. Wong and M. M. Ibrahim [2]. The authors focus on developing a Convolutional neural network based finger vein recognition system. The system was used to train only 50 images obtained inhouse and it was applied to retrain new incoming images. The disadvantage is that training data set is very minimal and is not obtained from a standard data repository. The work reports that pre-processing of the images are done using MATLAB by converting into gray scale images and cropped for desired size. Further, the quality of the image captured can be improved by applying efficient pre-processing techniques.

Face recognition based on Convolutional Neural Networks is proposed by Musab Coúkun, Ayúegül Uçar [3]. The authors propose an adapted Convolutional Neural Network (CNN) architecture which can perform the sum with two different normalization operations to one of the layers. Here, the classification is done using SoftMax classifier. The batch normalization can be used to improve the performance of the network. The authors employ a batch normalization process for the two layers, namely the first and final convolution layer. The work can be improved by training the network for images of different illumination levels. The training rate of the CNN network can further be improved that may result in reduced error rate.

Convolutional Neural Network-Based Finger Vein Recognition using NIR image sensors is proposed by Hyung Gil Hong, Min Beom Lee and Kang Ryoung Park [4]. They have proposed the solution for wrong observation of the finger vein dark lines beneath the skin that can bring down the accuracy level of the finger vein recognition by using CNN. The CNN is implemented to take the difference of two images for authentic matching. The work shows less recognition accuracy for testing data due to overfitting problem of CNN. Further, in this approach data augmentation can be performed only for good quality images and it is difficult to get these type images in available data sets and inhouse training sets.

Nearest Centroid Neighbor Based Sparse Representation Classification for Finger Vein Recognition is proposed by Shazeeda, Bakhtiar Affendi Rosdi [5]. It focuses on the algorithms for finger vein recognitions that solely depend on the computation of distance measurements. In this method, to improve the recognition estimation, we need to examine spatial arrangement, distribution of data and distance. It mainly comprises of two stages namely closest nearest centroid neighbors (k) and k nearest neighbors. So, we can conclude that to attain efficient finger vein recognition, we need to perform classification method as an important step.

Finger Vein recognition using local binary pattern is proposed by Bakhtiar Affendi Rosdi, Chai Wuh Shing and Shahri Azmin Suandi [6]. The work focuses on the captured pattern of finger vein which is more clear in LLBP and the grayscale values are normalized to values between 0 to 255. The work has not considered the finger alignment problem, horizontal and vertical displacement problems which are major issues in finger vein authentication methods. The proposed method has a high computation time and template size which puts in more complexity. Further, the experiments are conducted using the data bases that is not a part of any standard repository. The Equal Error Rate of the system can be further reduced to achieve high accuracy.

Some of the disadvantages that are faced in the above section is that the number of filters needed for image processing is less which makes the pre-processing step very complex to process. The noise in the finger vein image is not removed efficiently, which can downscale the image quality. The Softmax function is used as a classifier which makes the optimization much harder which is used in the above papers for finger vein biometric recognition, which will reduce the desired accuracy and it will increase the error rate.

In Near Infrared (NIR) image sensors, the spatial resolution of the finger vein image is limited which can cause ill-conditioned tomography problem and the depth sensitivity is less mostly in adults, so that the dark lines in finger vein is less visible to human eye, which makes the further process difficult to continue. The above stated disadvantages are overcome in the proposed framework.

CHAPTER 3

PROBLEM DEFINITION

3.1 EXISTING SYSTEM

With the rapid development in the area of Information Technology, the identity confirmation and safety of information or data is a basic problem. Thus, Finger Vein Biometric Recognition (FVBR) has procured an extremely high observation as it delivers a very high robustness, reliability and security for the personal identification. Since it is a better biometric method compared to earlier methods, it has grasped more observation from the researchers. A survey was performed on finger vein biometrics recognition taken from enduring research work that is done on the preprocessing, extraction of region of interest, feature extraction, classification stage and the performance parameter of the biometric recognition system which is Equal Error Rate (EER).

3.1.1 EXISTING SYSTEM METHODOLOGY

In these days, the identity confirmation and safety of information or data is a basic problem. Predominantly, we were using the SVM (Support Vector Machine), in order to find the accuracy of the finger vein using the finger vein biometric authentication system. In Machine learning and pattern classification, SVM have procured excellence by understanding the linear and non-linear separation area on the input image. With the direct SVM algorithm, we can reconfigure a set with the adjacent pairs of points from different classes. Once the algorithm discovers a breaking point in the dataset, it will attach itself to the candidate set. It is effectual in peak dimensional spaces.

3.2 ISSUES IN THE EXISTING SYSTEM

- It is less accurate.
- The segmentation has shortcomings. It is because when the finger vein patterns is broken down into smaller segments, it may cause external fragmentation.

- Feature extraction is not accurate. In Support vector machine, the features are not properly used in finger vein identification, which has led to inaccuracies.
- Computation load in SVM is very high as the existing system could not able to handle large datasets.

3.3 PROBLEM IDENTIFICATION

- Analysing the finger vein images of different characteristics like high security level, variance can reduce the accuracy of biometric authentication and identification.
- Each and every finger vein image must be unique and accurate, if it is not clearly viewed or accurate, then we cannot process with less quality images. So, we need high quality finger vein images to be taken as input images.
- Since the accuracy level is less, the efficiency will reduce notoriously.

3.4 ADVANTAGES OF PROPOSED SYSTEM

- The proposed model needs to train large amount of data. It will take a very long time and hence, it is essential that the system on which it is trained has very high speed and low complexity, which make it very well suited to operate on real scenarios.
- Computation load needed for Convolutional Neural network is much reduced, combined with very simple classifiers.
- It has the ability to learn and extract complex image features.
- The accuracy of the proposed model increases with larger input dataset.
- In the proposed model, the features are properly extracted by CNN with convolution layer and max pooling layer, so that it will reduce the time consumption. ReLU (Rectified Linear Unit) function is used to overcome the problem of vanishing gradient which allows the proposed model to perform faster and better.

CHAPTER 4

DEVELOPMENT PROCESS AND DOCUMENTATION

4.1 REQUIREMENTS ANALYSIS AND SPECIFICATIONS

4.1.1 HARDWARE REQUIREMENTS

Processor	: Intel i5 6^{th} Gen or equivalent.
Memory	: 8 GB DDR4 RAM.
System Type	: CUDA Supported GPU from Nvidia or Tesla.
Hard Disk Drive	: 50 GB

4.1.2 SOFTWARE REQUIREMENTS

4.1.2.1 GENERAL

MATLAB (matrix laboratory) is a computing environment of numbers and a programming language of the fourth generation. Produced via Math Works, MATLAB allows matrix manipulation, function and data plotting, algorithm implementation , user interface development, and interfacing with programs written in other languages, including C , C++ , Java, and Fortran.

Although MATLAB is primarily intended for numerical computing, the MuPAD symbolic engine is used in an optional toolbox, allowing access to symbolic computing capacities. An additional bundle, Simulink, introduces dynamic and embedded systems with graphical multidomain simulation and Model-based architecture.

MATLAB had approximately a million users spanning industry and academia in 2004. Users of MATLAB come from diverse backgrounds in engineering, science and economics. MATLAB is widely used both in academics, research and industrial enterprises.

In control engineering, MATLAB was first adopted by researchers and practitioners, but soon spread to many other domains. It is now also used in education, particularly teaching linear algebra and numerical analysis, and is popular among image processing scientists. The MATLAB application is based around the language MATLAB.

4.1.2.2 FEATURES OF MATLAB

- High-level Scientific Programming terminology.
- Production environment for the code, scripts, and data management.
- Interactive methods to discover, develop and problem-solve iteratively.
- Mathematical functions for linear algebra, statistics, study of Fourier, filtering, optimization and the integration of numbers.
- Functions in 2-D and 3-D graphics for data Visualization.
- Tools designed to create custom graphical user interfaces.
- Algorithm that is based on MATLAB works with exterior programs and languages, such as C++, JAVA, and Fortran.

Some of the MATLAB tools that are helpful in implementing algorithm proficiently that includes:

MATLAB Editor

This editor provides basic features like debugging and editing.

Code Analyzer

Code analyzer is used to check the code for any issues or errors and recommend some improvements to refine the efficiency.

MATLAB Profiler

To execute every line of code, we need to store the time. This is the main purpose of MATLAB profiler.

Directory Reports

It helps us to scan all the folders in the directory and it can make the report of the output and coverage of code.

Designing Graphical User Interfaces

Here, with the help of an interactive tool GUIDE (Graphical User Interface Development Environment), we can design, edit and layout UI. It lets you to include push and pull buttons, sliders, radio buttons and some MATLAB plots. By MATLAB functions, we can create GUI technically.

CHAPTER 5

SYSTEM DESIGN

5.1 ARCHITECTURE DESIGN

The proposed system focuses on developing an efficient and accurate system for finger vein biometric identification system. This system will help to reduce forgery and it enhances the public safety.

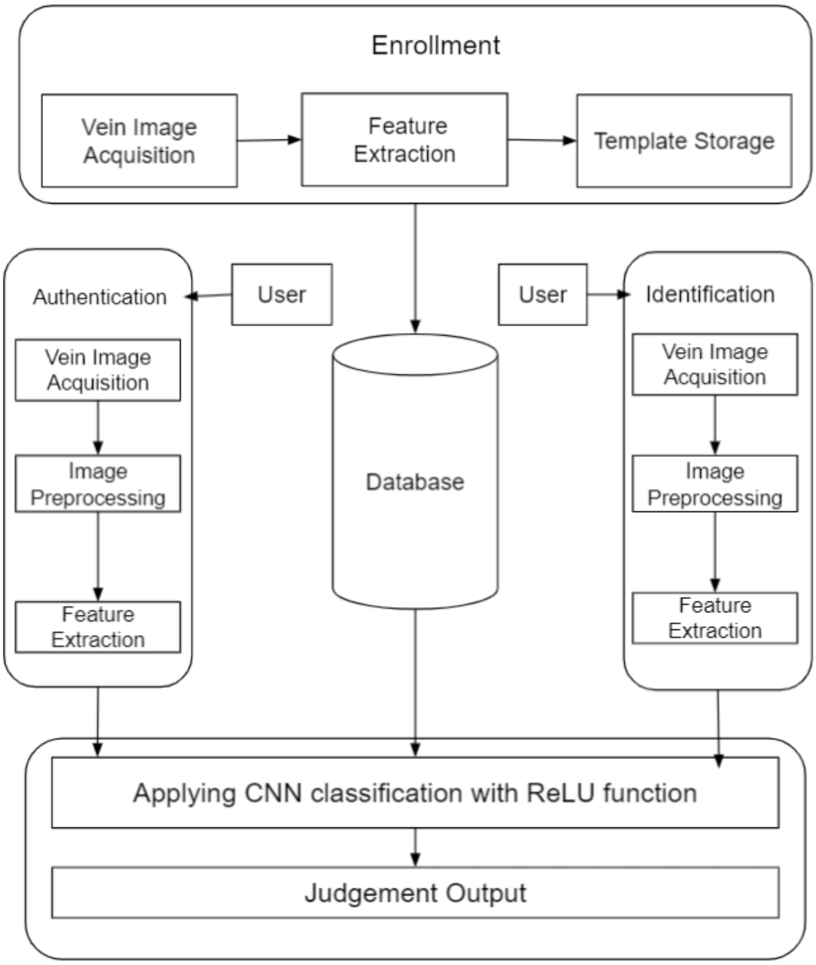


Figure 5.1 Overall Architecture Design

Figure 5.1 depicts the overall architecture design of finger vein biometric identification system. The process initiates with a vein image acquisition where the finger is placed in an acquisition device through the NIR (near infra-red) source. Then, the input image is sent to the feature extraction block to extract all the necessary features of the image. Once done, it will be dispatched in the template storage where the images are stored for certain period of time. We took 3 publicly available database for our project namely HKPU, FV-USM, UTFVP. At first, the input image is being preprocessed. In pre-processing, the quality of the input image is enhanced with the help of CLAHE (Contrast Limited Adaptive Histogram Equalization) which lowers the level of noise amplification, so that we can see a clear view of the finger vein image.

In feature extraction, the finger vein image is being analyzed to get the texture information. The features that are used in feature extraction for analytical data includes mean, variance, standard deviation, energy, entropy, kurtosis, skewness, IDM. The four important features compared for the finger vein recognition namely mean, variance, homogeneity, entropy. The texture information are used to know the statistical data of the finger vein image. Once the features are extracted, the input image will check for the matching recognition by applying CNN with ReLU function. The main purpose of ReLU is that it overcomes the problem of vanishing gradient.

ReLU helps to perform the CNN progress much faster and better. In classification, the input image will be checked with the database whether the input image matches with the database. If the input image matches with the database, then we get the output as recognized. If it does not match with the database, then we get the output as not recognized. So, this is the overall architecture design of Finger Vein Biometric identification system.

5.1.1 FLOWCHART

Figure 5.2 depicts the sequential steps of the finger vein biometric identification system. First of all, a finger vein image is taken as an input image. Then, we need to preprocess the image for enhancing the quality of the input image by removing the noise from the input image. Now, the input image is passed to the next step which is extraction of ROI (Region of Interest). In this step, segmentation takes place, where the input image is taken as binary segment and gray segment. Now, the gray segment will be used for further processing, as we need to do feature extraction. With the binary segment, we cannot do feature extraction, because it does not contain lots of information about the texture features.

In feature extraction, the features are being taken from the finger vein input image for statistical purposes. Here, the texture information such as mean, standard deviation, variance, kurtosis, correlation and contrast, are being taken from the input image as statistical information.

The other processes like matching recognition and post processing followed by classification are done by applying CNN (Convolutional neural networks) with activation function such as ReLU (Rectified Linear unit). Once the input image is recognized, we get the output as recognized image.

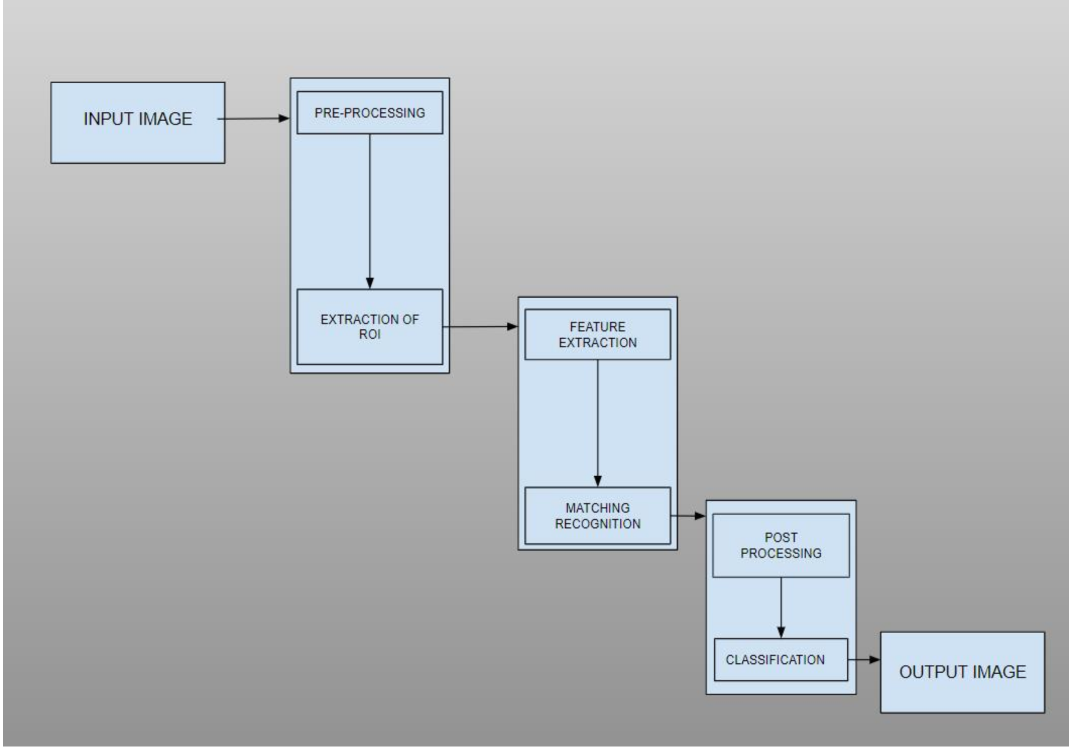


Figure 5.2 Flowchart Diagram

5.2 ALGORITHM of FVBI (FINGER VEIN BIOMETRIC IDENTIFICATION):

STEP 1: A finger vein image is taken as an input image that is selected for loading in the GUIDE (Graphical user interface development environment)

STEP 2: The loaded image is selected to preprocess for enhancing the finger vein image quality.

STEP 3: With the enhanced finger vein image, ROI (region of interest) is extracted so that segmentation is done.

STEP 4: Features are extracted from the segmented finger vein image for the texture analysis.

STEP 5: Build a proposed CNN recognition model

STEP 6: With step 5, we can check whether the input finger vein image matches with the records in the database.

5.3 DETAILED DESIGN

5.3.1 MODULE -1 → PRE-PROCESSING

- This is the first step in the finger vein biometric method where, the input finger vein image is taken from publicly available database.
- Then, the vein image needs to be pre-processed in order to improve the quality of the image.
- In the input image, the amount of noise is very high and dark lines are not clearly visible to human eye.
- By removing noise and increasing the clarity of the image, we need to perform pre-processing.
- Here, the image quality is improved and we can see the enhancement in the image with the dark lines that are clearly visible in the finger vein image.
- Once, the image is pre-processed, it can be sent for ROI extraction for segmenting the image.

ALGORITHM OF PRE-PROCESSING

```
function pushbutton8_Callback(hObject,handles)

I = handles.ImgDataI;

if size(I,3)==3
gray = rgb2gray(I);
else
    gray=I;
end
Ehanc=imadjust(gray);
 axes(handles.axes1)
  imshow(Ehanc);title('Enhanced Image');
 helpdlg(' Image is Preprocessed ');

  handles.ImgData = Ehanc;

  guidata(hObject,handles);
```

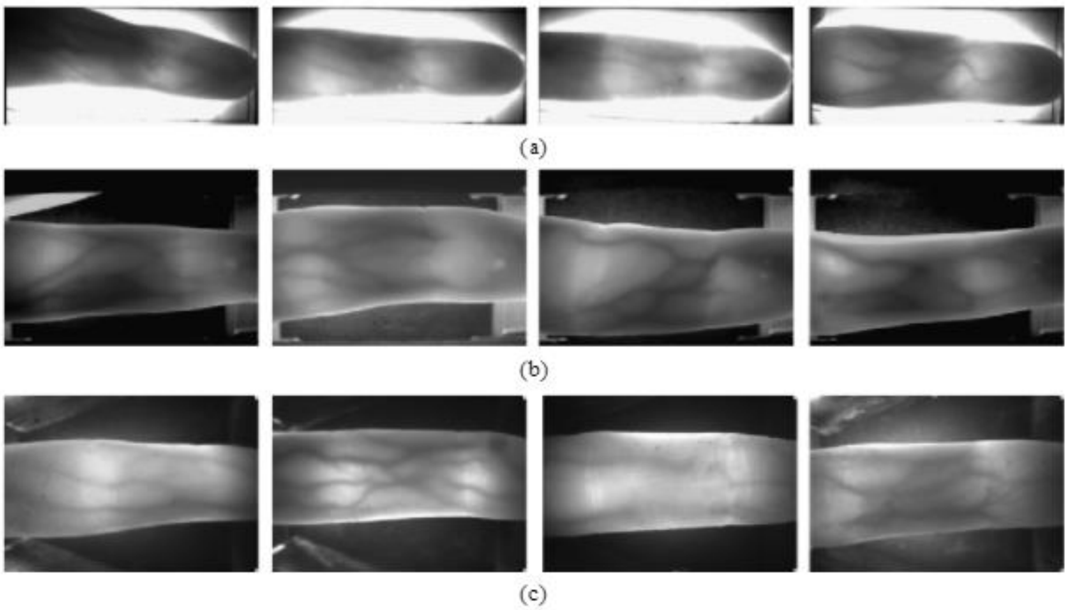


Figure 5.3 Input images from three different publicly available databases

FINGER REGION ALIGNMENT

When an image of a finger vein is taken from an acquired device, it looks more darker than the background part. It is because the infrared light is being absorbed by the finger. In order to localize the finger vein region from captured images, we need to perform segmentation. It is performed by detecting the boundary in between the finger vein region and the background part. Once the segmentation is done, we can see a clear view of the finger vein image.

Figure 5.4 Sample of Finger Vein Image

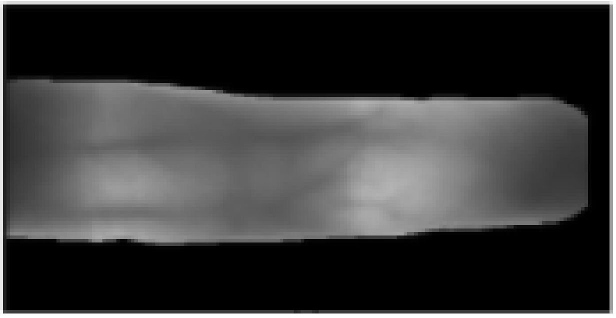


Figure 5.5 Finger Vein Image after Segmentation process

5.3.2 MODULE – 2 → EXTRACTION OF ROI (REGION OF INTEREST)

- In the extraction of ROI, the left and right part of the finger vein is not clearly visible due to the inadequate lighting of the acquisition device.
- By using the filters, we can remove those parts where the finger vein image is not captured precisely.
- Now, the noise in the image is detached by using component labelling method. Due to bright lighting (where the light is provided from external source), the dark lines of the vein patterns cannot be perceived. To create a ROI mask, we need an average estimation of the adjacent pixels using a 5 x 20 mask.

- By using the co-occurrence matrix, we can represent it in the normal tissues of finger vein. Monitoring of the finger vein images must be assigned to the regions in normal tissue.
- Once the process is completed, the binary segment and grey segment is extracted from the input finger vein image.

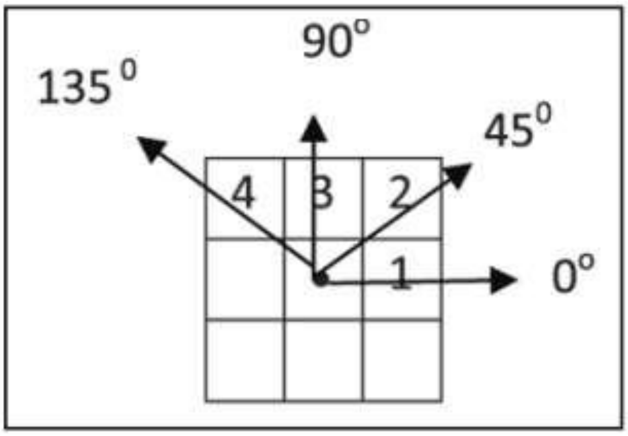


Figure 5.6 Diagrammatic Representation Of data points in Finger Vein Pattern at different angles

- The propitious co-occurrence matrix that is used in our study is GLCM (Grey level Co-occurrence matrix), which distinguishes the behaviour of the analytical property of the finger vein images.
- Some of the features includes: Contrast, correlation, entropy, homogeneity and Euclidean norm.

1. CONTRAST:

$$f_1 = \sum_i \sum_j (i-j)^2\, p(x,y)$$

2. CORRELATION:

$$f_2 = \sum_i \sum_j \frac{(i-\mu_x)(j-\mu_y)\,p(x,y)}{\sigma_x \sigma_y}$$

3. ENTROPY:

$$f_3 = \sum_i \sum_j \left(\frac{p(x,y)}{\log p(x,y)} \right)$$

4. HOMOGENEITY:

$$f_4 = \sum_i \sum_j \left(\frac{p(x,y)}{1+|i-j|} \right)$$

5. EUCLIDEAN NORM:

$$d(i,j) = d(i,j) = \sqrt{(i_1 - j_1)^2 + (i_2 - j_2)^2 + \ldots + (i_n - j_n)^2}$$
$$= \sqrt{\sum_{k=1}^{n} (i_k - j_k)^2}$$

ALGORITHM FOR ROI: (REGION OF INTEREST)

- STEP 1: Start
- STEP 2: Declare function(region,edges) returns the images with ROI extraction.
- STEP 3: Change the size of the image to 640 x 480 in order to get a clear view of the finger vein image.
- STEP 4: Perform the if-else loop conditions for processing the image.
- STEP 5: Define the image filter.
- STEP 6: End

5.3.3 MODULE – 3 → FEATURE EXTRACTION

INTENSITY FEATURES

- First and foremost important feature in feature extraction is intensity feature. This feature acts as a primary source by providing information in MRI images and thoracic Computed Tomography (CT) images.

TEXTURE FEATURES

- Texture refers to the increase in quality of the finger vein data-points by expanding the statistical property finger vein.
- Texture descriptor is one such characteristic property that tell about the roughness, contrast and arrangement of the structure.
- Several methods that are involved in the texture features extraction includes Gabor filter, linear binary pattern (LBP), GLCM and unpredictable dimension. They have integrated all the features to create a hybrid method, so that it can refine the judicious ability of texture features.

- The second order statistical property that can be used in the extraction of the texture features is GLCM. At first, we need to express an finger vein image into a grey level image and the occurrences of the data pairs in between the current and the neighbor pixel is done by GLCM at different orientation and scale.

GEOMETRIC FEATURES

- In the geometric features with the help of segmented modules, various two-dimensional features are being calculated, that includes minor axis length and major axis length on the finger vein.

- We can calculate the perimeter by multiplying perimeter module and pixel resolution. Area can be calculated by multiplying pixel spacing with the no. of module pixels that are enclosed in the module region.

GLCM

- GLCM is abbreviated as (Grey Level Co-occurrence matrix). It is one of the important feature extraction. It is an analytical process which narrates the connections of spatial of every pixel. The information of an image are gained by these statistics, that includes correlation, energy, contrast, kurtosis, variance, RMS, entropy, Standard Deviation, homogeneity, skewness and smoothness.

ALGORITHM FOR FEATURE EXTRACTION

```
function pushbutton2_Callback(hObject, handles)

img2 =handles.Lenregion;
imG=im2double(imresize(img2,[28 28]));

    imG1=(imG(:))';

img2=imresize(img2,[160 120]);

G = img2;

g = graycomatrix(G);
stats = graycoprops(g,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(G);
Standard_Deviation = std2(G);
Entropy = entropy(G);
RMS = mean2(rms(G));
Variance = mean2(var(double(G)));
a = sum(double(G(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(G(:)));
Skewness = skewness(double(G(:)));

m = size(G,1);
n = size(G,2);
in_diff = 0;
for i = 1:m
    for j = 1:n
        temp = G(i,j)./(1+(i-j).^2);
        in_diff = in_diff+temp;
    end
```

5.3.4 MODULE – 4 → CLASSIFICATION

- In Classification, the relevant class labels are being related with the finger vein sample by applying the distance measurements.

- To overhaul the weights in the finger vein, the training data are used. In order to facilitate uncomplicated classification in the high proportion feature space, we need to apply non-linear mapping task that can convert the non-linear separating problem into linear separating problem in feature space.

- It is one of the important role in the digital image analysis.

CONVOLUTION LAYER

- The first layer in the classification process to extract the features from finger vein input image is Convolutional layer. It maintains the connection between the image pixels by learning the features of the image using matrices of input data.

- With the convolution of an image, we can execute operations such as sharpening, edge detection and blurring by applying different filters.

PADDING

- Padding is a process in which filter fails to suit in the input finger vein image. In that case, we are provided with two choices: either we need to pad the input image with zeroes, which can be called as zero-padding, so that it will be well suited.
- Else release the bit of the picture where the filter does not suit. It can be called as valid padding, that keeps the valid bit of an image.

MAX POOLING

- Max Pooling refers to the process of variation in input representation in which it extracts the largest element from the rectified feature map. After extracting the largest element, it could perform average pooling. Sum total of all the elements from the feature map is called sum pooling.

FULLY CONNECTED LAYER

- In the fully connected layer, we need to compress our matrix of the image into a vector and provide it to the output layer.

PSEUDOCODE FOR CLASSIFICATION

CALL function ClassifyCNN returns Trainfeatures,TestData

TrainFeatures returns double Trainfeatures

TestData returns double TestData

CALL Ypr for grayscale values

START LOOP

FOR i=1:size(TrainFeatures,1)

Cv(i)=corr2(TrainFeatures(i,end-12:end), Testdata(end-12:end));

END LOOP

ALGORITHM FOR CLASSIFICATION

```
net = trainNetwork(XTrain,YTrain,layers,options);
YPredOut=classifyCNN(net,TestFeatures,TrainAllSetFeatures,YTrain);
Imnew = handles.ImgDataI;
YPredOut=double(YPredOut);
if YPredOut<=15
figure;
imshow(Imnew);
title(['Query image is Recognized- Person : ' num2str(YPredOut)]);
msgbox(['Query image is Recognized- Person : ' num2str(YPredOut)]);
strOut=['Recognized- Person : ' num2str(YPredOut)];
else
figure;
imshow(Imnew);
title(['Query image is Not Recognized']);
msgbox(['Query image is Not Recognized']);
strOut=['Not Recognized'];
end

 set(handles.edit4,'string',strOut);
```

CHAPTER 6

RESULTS AND DISCUSSION

6.1 EXPERIMENTS

6.1.1 DATASET

The finger Vein Based biometric identification system is developed and evaluated using the finger vein database. This dataset composed of around 600 finger vein images from three different databases namely HKPU (Hong Kong Polytechnic University), FV-USM (University Sains Malaysia), and UTFVP (University of Twente, Netherlands). Table 6.1 gives the structure of the finger vein database.

Table 6.1 Finger Vein Image Databases

DATABASE	DETAILS OF FINGERS	IMAGE SIZE	TOTAL IMAGES
HKPU	Left hand Index & middle finger	513 x 256	200
FV-USM	Left & Right index & middle finger	640 x 480	170
UTFVP	Left & Right index & middle finger	672 x 380	320

With the proposed CNN approach, the input image is checked whether:

(i) Recognised – If the input finger vein image is matched with the database.

(ii) Not Recognised – If the input finger vein image is mismatched with the database.

6.1.2 PERFORMANCE ANALYSIS

The proposed CNN approach aims to improve the performance of the finger vein biometric identification system. From the table 6.2, we can understand that the input images are taken from three different publicly available databases. The existing SVM (Support vector machines) has low accuracy compared to CNN (Convolutional neural network) in HKPU (Hong Kong Polytechnic University). Similarly in the other two databases, it is clearly shown that the accuracy is much higher in CNN compared to SVM. CNN has better accuracy with all the three databases compared to SVM (support vector machine).

Table 6.2 Accuracy of HKPU, FV-USM, UTFVP with SVM and CNN

Databases	SVM (Accuracy in Percentage)	CNN (Accuracy in Percentage)
HKPU	86.27	92.30
FV-USM	85.57	95.62
UTFVP	89.42	94.31

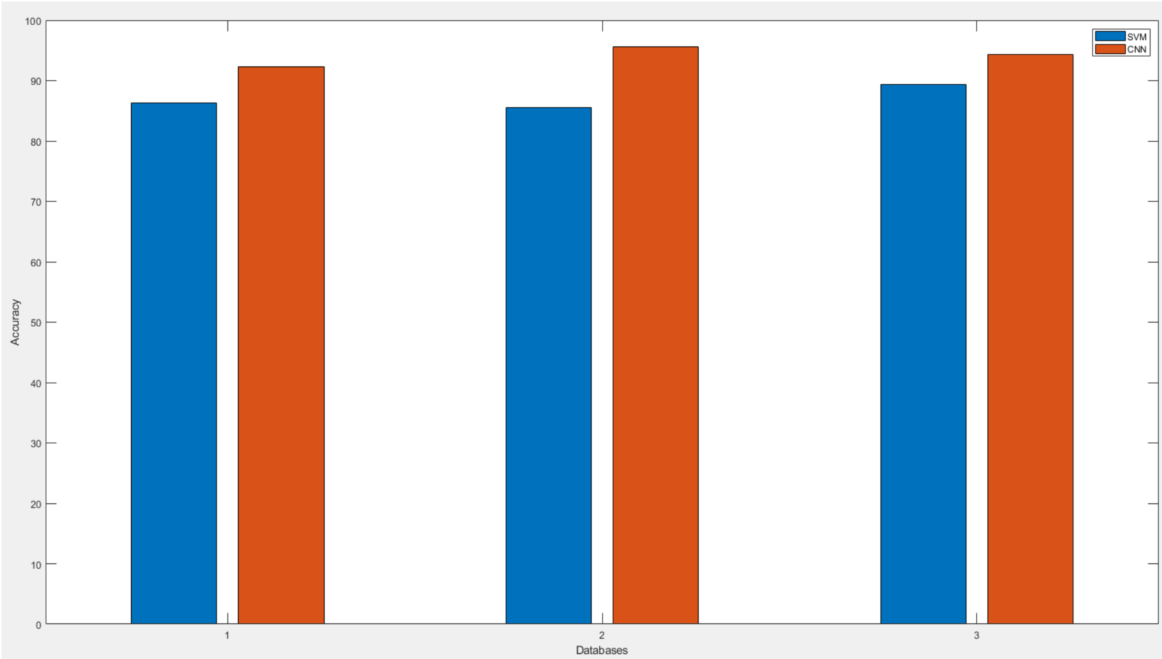


Figure 6.1 Comparison Chart of SVM and CNN in terms of Accuracy.

As it can be seen from the above figure 6.1, the comparison chart shows that the accuracy of CNN is higher compared to SVM. It has been shown that the HKPU database has an accuracy of 92.3% in CNN which is much greater than the existing SVM model. In FV-USM database, the CNN has an accuracy of 95.62% which is significantly higher than the existing SVM model. Similarly, in UT-FVP, it has been clearly depicted in the graph that CNN has better accuracy of 94.31% by fine tuning the training data compared to existing SVM model.

In Appendix A1, we have given the source code of our report. In Appendix A2 figure 1.1, it is the GUI (Graphical user interface) of finger vein biometric system. It consists of loading the image, pre-processing, extraction of ROI, feature extraction and final output with CNN recognition. When we perform each process, a dialog box will appear once the process is complete. We need to select a random input image that we have downloaded from different sources. Once the image is selected (Appendix A2 figure 1.2), we can see the input image is loaded (Appendix 1.3) in the GUI. In pre-processing (Appendix A2 figure 1.4), image enhancement is clearly visible, as we can see the clear view of dark line in finger vein. After pre-processing, extraction of ROI

(Appendix A2 figure 1.5) we can see the segmentation process with binary and grey segment (Appendix A2 figure 1.6). Once the process is performed, we can see an enhanced image (Appendix A2 figure 1.7) in the GUI with binary image next to the enhanced part. Then, features of extraction (Appendix A2 figure 1.8) is performed on the enhanced image, once the process done, we can see the statistical data of the enhanced image is shown below the GLCM features in GUI. Now, perform CNN recognition (Appendix A2 figure 1.9) to predict the output, whether it can recognize the input finger vein image or not.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

Finger vein based biometric recognition system using CNN approach is powerful to finger-vein asymmetric and shading, where, the input image is matched with the database is researched by using the vein images. We have found the result to provide much higher accuracy with the proposed approach. Under environmental conditions, the CNN was applied on the output in order to check whether the input image is recognized or not. The performance is checked based on the comparison between existing model (SVM) and proposed CNN with the help of three publicly available databases. Lots of training data are needed to effectively train the proposed CNN model. It is difficult to collect lots of finger vein images data. Hence, it is important to increase training data by appropriate data increase, that represents the uniqueness of training data. In order to lower the defect and hurdle of the finger vein based biometric method, we have made our trained CNN model with three public database that are extended to other researchers through the proposed CNN method that can be put in other biometric methods, and the performance can be evaluated.

In future work, we would like to conduct research on other biometric methods with the CNN that is proposed in our study and it will be done on a real-time scenario basis.

APPENDIX A1

SOURCE CODE

```
function varargout = FingerVeinRecognition_GUI(varargin)

gui_Singleton  = 1;
gui_State  = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',   gui_Singleton,  ...
                   'gui_OpeningFcn',  @FingerVeinRecognition_GUI_OpeningFcn,  ...
                   'gui_OutputFcn',   @FingerVeinRecognition_GUI_OutputFcn,  ...
                   'gui_LayoutFcn',   [] , ...
                   'gui_Callback',    []);
if nargin&&ischar(varargin{1})
gui_State.gui_Callback  = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}]  = gui_mainfcn(gui_State,  varargin{:});
else
gui_mainfcn(gui_State,  varargin{:});
end
function FingerVeinRecognition_GUI_OpeningFcn(hObject,  eventdata,
handles,  varargin)

handles.output  = hObject;
ss = ones(200,200);
axes(handles.axes1);
imshow(ss);
axes(handles.axes2);
imshow(ss);

guidata(hObject,  handles);
```

```
function varargout = FingerVeinRecognition_GUI_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)

[FileName,PathName] = uigetfile('*.jpg;*.png;*.bmp','Pick an Image');
if isequal(FileName,0)||isequal(PathName,0)
warndlg('User Press Cancel');
else
    P = imread([PathName,FileName]);
  axes(handles.axes1)
imshow(P);title('Finger Vein Image');
helpdlg(' Image is Selected ');

handles.ImgDataI = P;

guidata(hObject,handles);
end
function pushbutton2_Callback(hObject, eventdata, handles)

img2 =handles.Lenregion;
imG=im2double(imresize(img2,[28 28]));
imG1=(imG(:))';
img2=imresize(img2,[160 120]);
G = img2;

g = graycomatrix(G);
stats = graycoprops(g,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
```

```
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(G);
Standard_Deviation = std2(G);
Entropy = entropy(G);
RMS = mean2(rms(G));
Variance = mean2(var(double(G)));
a = sum(double(G(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(G(:)));
Skewness = skewness(double(G(:)));
% Inverse Difference Movement
m = size(G,1);
n = size(G,2);
in_diff = 0;
for i = 1:m
    for j = 1:n
        temp = G(i,j)./(1+(i-j).^2);
in_diff = in_diff+temp;
    end
end
IDM = double(in_diff);

GLCMfeat = [Contrast,Correlation,Energy,Homogeneity, Mean,
Standard_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis,
Skewness,IDM];

TestFeatures= [imG1(1:end-13),GLCMfeat];
set(handles.edit5,'string',Mean);
set(handles.edit6,'string',Standard_Deviation);
set(handles.edit7,'string',Entropy);
set(handles.edit8,'string',RMS);
```

```
set(handles.edit9,'string',Variance);
set(handles.edit10,'string',Smoothness);
set(handles.edit11,'string',Kurtosis);
set(handles.edit12,'string',Skewness);
set(handles.edit13,'string',IDM);
set(handles.edit14,'string',Contrast);
set(handles.edit15,'string',Correlation);
set(handles.edit16,'string',Energy);
set(handles.edit17,'string',Homogeneity);
helpdlg(' Features are Extracted ');

handles.FeatureGLCM= TestFeatures;
guidata(hObject,handles);

function edit1_Callback(hObject, eventdata, handles)
function edit1_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
function edit2_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
function edit4_Callback(hObject, eventdata, handles)
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit5_Callback(hObject, eventdata, handles)

function edit5_CreateFcn(hObject, eventdata, handles)
if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit6_Callback(hObject, eventdata, handles)
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit7_Callback(hObject, eventdata, handles)
function edit7_CreateFcn(hObject, eventdata, handles)
```

```
if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

function edit8_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

function edit9_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

function edit10_CreateFcn(hObject, eventdata, handles)
if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit11_Callback(hObject, eventdata, handles)

function edit11_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit12_Callback(hObject, eventdata, handles)
function edit12_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

function edit13_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

function edit14_CreateFcn(hObject, eventdata, handles)
```

```
if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit15_Callback(hObject, eventdata, handles)

function edit15_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit16_Callback(hObject, eventdata, handles)

function edit16_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit17_Callback(hObject, eventdata, handles)

function edit17_CreateFcn(hObject, eventdata, handles)
```

```
if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit19_Callback(hObject, eventdata, handles)

function edit19_CreateFcn(hObject, eventdata, handles)

if ispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function pushbutton7_Callback(hObject, eventdata, handles)

TestFeatures=handles.FeatureGLCM;
load DataBase_FingerVein
for i=1:size(TrainAllSetFeatures,1)
XTrain(:,:,:,i)=reshape(TrainAllSetFeatures(i,:),28,28,1);
YTrain(i,1)=categorical(TrainLabel(i));
end

layers = [
imageInputLayer([28 28 1])

    convolution2dLayer(3,8,'Padding','same')
```

```
batchNormalizationLayer
reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,16,'Padding','same')
batchNormalizationLayer
reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,32,'Padding','same')
batchNormalizationLayer
reluLayer

fullyConnectedLayer(16)
softmaxLayer
classificationLayer];

options = trainingOptions('sgdm', ...
    'InitialLearnRate',1e-3, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropFactor',0.1, ...
    'LearnRateDropPeriod',5, ...
    'MaxEpochs',100, ...
    'Verbose',false, ...
    'MiniBatchSize',128);

net = trainNetwork(XTrain,YTrain,layers,options);
YPredOut=classifyCNN(net,TestFeatures,TrainAllSetFeatures,YTrain);
Imnew = handles.ImgDataI;
YPredOut=double(YPredOut);
```

```
if YPredOut<=15
figure;
imshow(Imnew);
title(['Query image is Recognized- Person : ' num2str(YPredOut)]);
msgbox(['Query image is Recognized- Person : ' num2str(YPredOut)]);
strOut=['Recognized- Person : ' num2str(YPredOut)];
else
figure;
imshow(Imnew);
title(['Query image is Not Recognized']);
msgbox(['Query image is Not Recognized']);
strOut=['Not Recognized'];
end

 set(handles.edit4,'string',strOut);



function pushbutton8_Callback(hObject,  eventdata, handles)

I = handles.ImgDataI;

if size(I,3)==3
gray = rgb2gray(I);
else
   gray=I;
end
Ehanc=imadjust(gray);
 axes(handles.axes1)
imshow(Ehanc);title('Enhanced  Image');
helpdlg(' Image is Preprocessed  ');
```

```
handles.ImgData  = Ehanc;

guidata(hObject,handles);


function pushbutton9_Callback(hObject,  eventdata, handles)
I=handles.ImgData;

img = imresize(im2double(I),  0.5);

mask_height=4;
mask_width=20;
[ROI, edges] = ROI_segment(img,mask_height,mask_width);
ROI=imresize(ROI,2);

GrayROI=uint8(ROI).*I;
BB=regionprops(ROI,'BoundingBox');
GrayROI= imcrop(GrayROI,BB.BoundingBox);


 axes(handles.axes2)
imshow(ROI);title('Binary  Image');
  figure;
subplot(121);
imshow(imcrop(ROI,BB.BoundingBox));title('Binary  Segment');
subplot(122);
imshow(GrayROI);title('Gray Segment');
helpdlg('  ROI is Segmented');

handles.Lenregion  =GrayROI;
guidata(hObject,handles);
```

```
function [region, edges] = ROI_segment(img, mask_h, mask_w)

[img_h, img_w] = size(img);


if mod(img_h,2) == 0
half_img_h = img_h/2 + 1;
else
half_img_h = ceil(img_h/2);
end


mask = zeros(mask_h,mask_w);
mask(1:mask_h/2,:) = -1;
mask(mask_h/2 + 1:end,:) = 1;

img_filt = imfilter(img, mask,'replicate');



img_filt_up = img_filt(1:floor(img_h/2),:);
[~, y_up] = max(img_filt_up);


img_filt_lo = img_filt(half_img_h:end,:);
[~,y_lo] = min(img_filt_lo);


region = zeros(size(img));
```

```
function ClasOut=classifyCNN(net,TestData,TrainFeatures,YTrain)
TrainFeatures=double(TrainFeatures);
TestData=double(TestData);
YPr = classify(net,reshape(TestData,28,28,1));
for ik=1:size(TrainFeatures,1)
Cv(ik)=corr2(TrainFeatures(ik,end-12:end),TestData(end-12:end));
end

[maxOut,Out]=max(Cv);
ClasOut=YTrain(Out);
```

ROI_Segment.mat

```
function [region, edges] = ROI_segment(img, mask_h, mask_w)

[img_h, img_w] = size(img);



if mod(img_h,2) == 0
half_img_h = img_h/2 + 1;
else
half_img_h = ceil(img_h/2);
end



mask = zeros(mask_h,mask_w);
mask(1:mask_h/2,:) = -1;
mask(mask_h/2 + 1:end,:) = 1;

img_filt = imfilter(img, mask,'replicate');
```

```
img_filt_up = img_filt(1:floor(img_h/2),:);
[~, y_up] = max(img_filt_up);


img_filt_lo = img_filt(half_img_h:end,:);
[~,y_lo] = min(img_filt_lo);


region = zeros(size(img));
for i=1:img_w
    region(y_up(i):y_lo(i)+size(img_filt_lo,1),  i) = 1;
end


edges = zeros(2,img_w);
edges(1,:)  = y_up;
edges(2,:)  = round(y_lo + size(img_filt_lo,1));
```

classifyCNN.mat

```
function ClasOut=classifyCNN(net,TestData,TrainFeatures,YTrain)
TrainFeatures=double(TrainFeatures);
TestData=double(TestData);
YPr = classify(net,reshape(TestData,28,28,1));
for ik=1:size(TrainFeatures,1)
Cv(ik)=corr2(TrainFeatures(ik,end-12:end),TestData(end-12:end));
end

[maxOut,Out]=max(Cv);
ClasOut=YTrain(Out);
```

APPENDIX A2

SCREENSHOTS

Figure A 1.1 FINGER_VEIN GUI

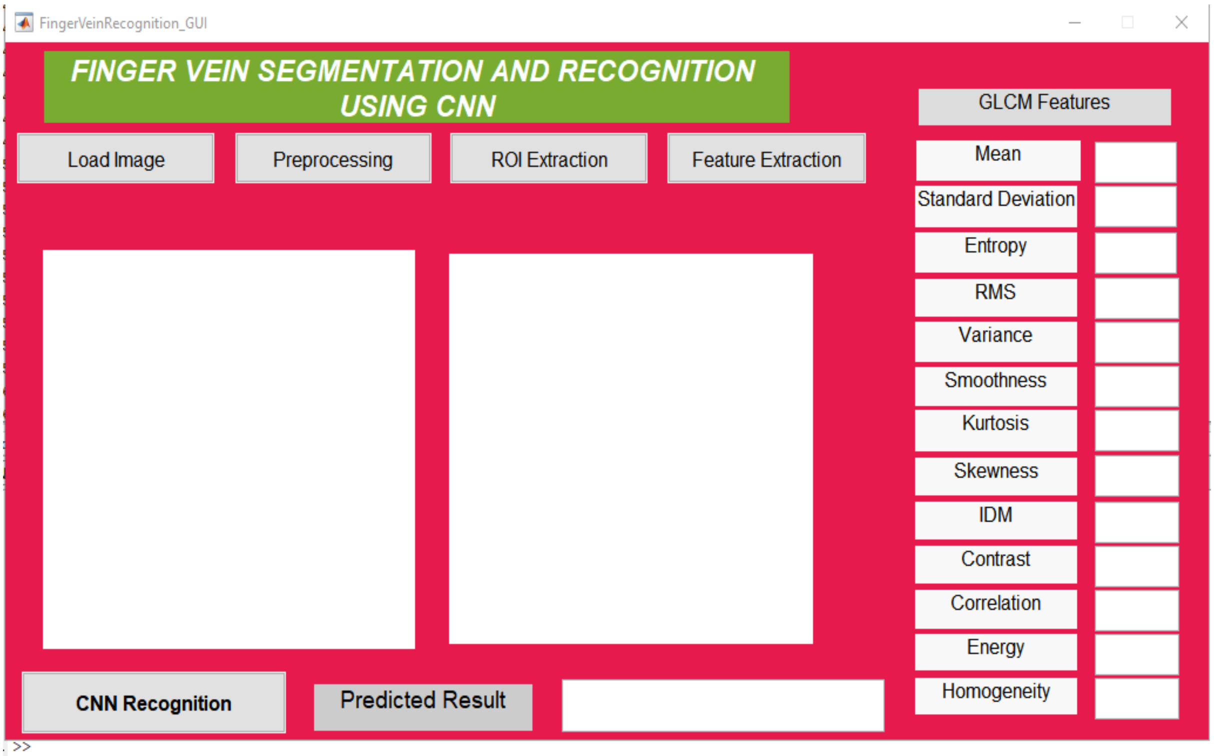


Figure A 1.2IMAGE SELECTION

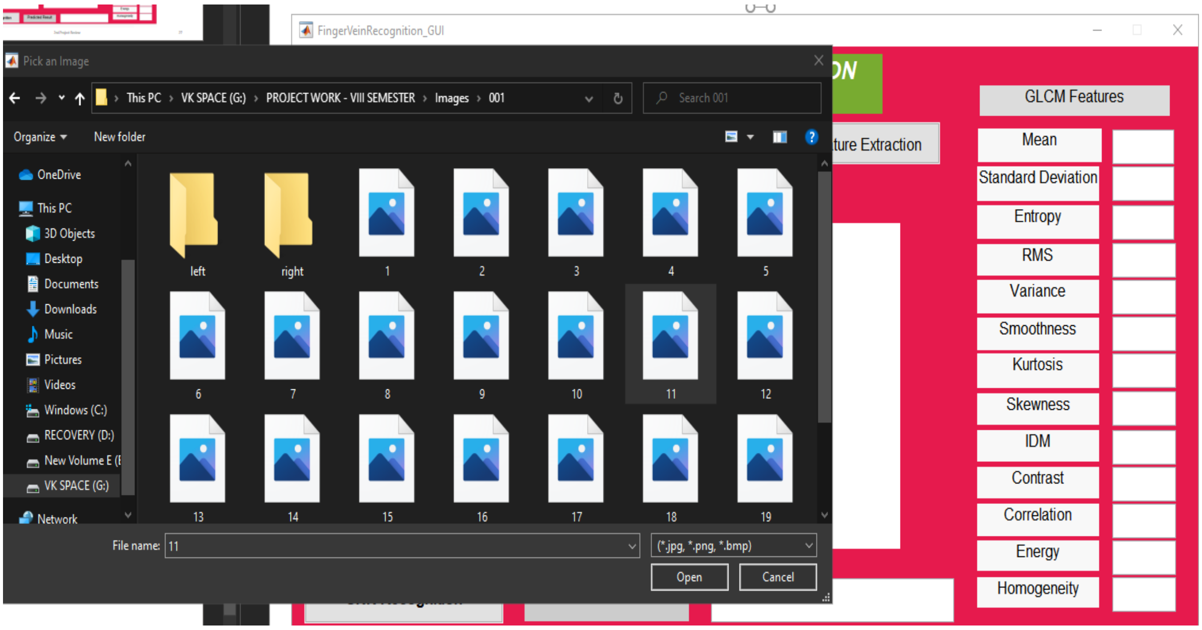


Figure A 1.3 IMAGE LOADED

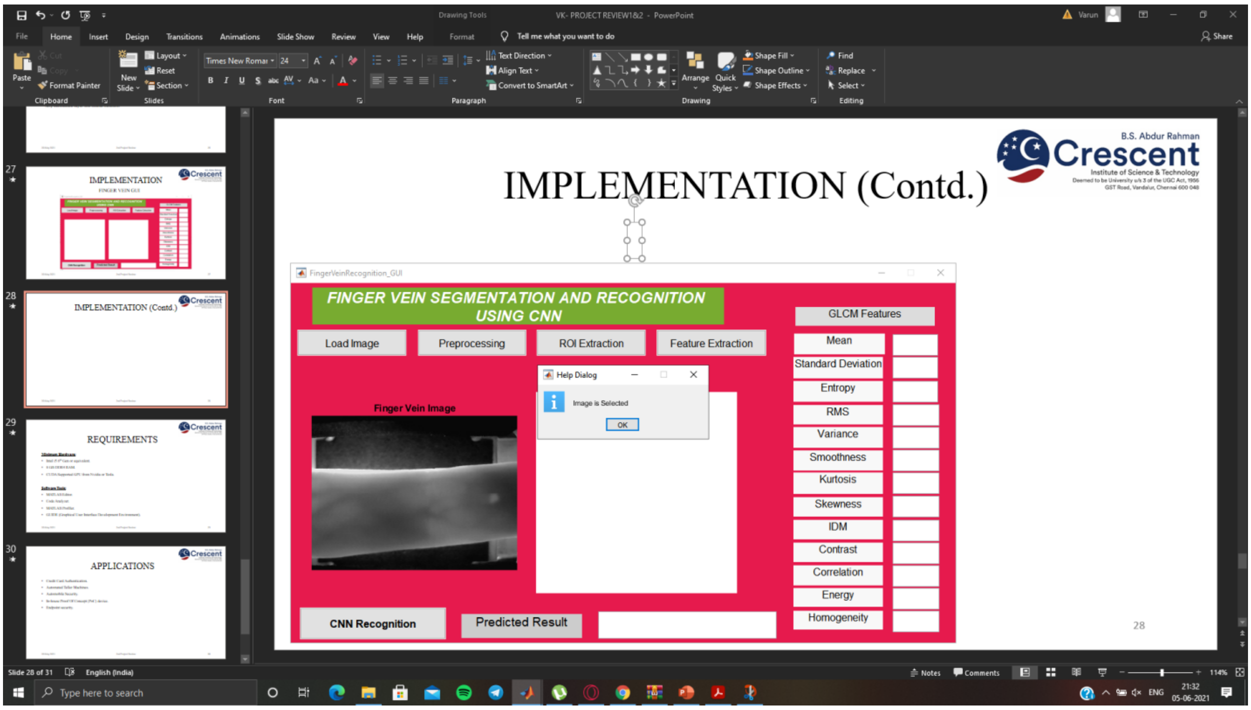


Figure A 1.4 IMAGE PREPROCESSING

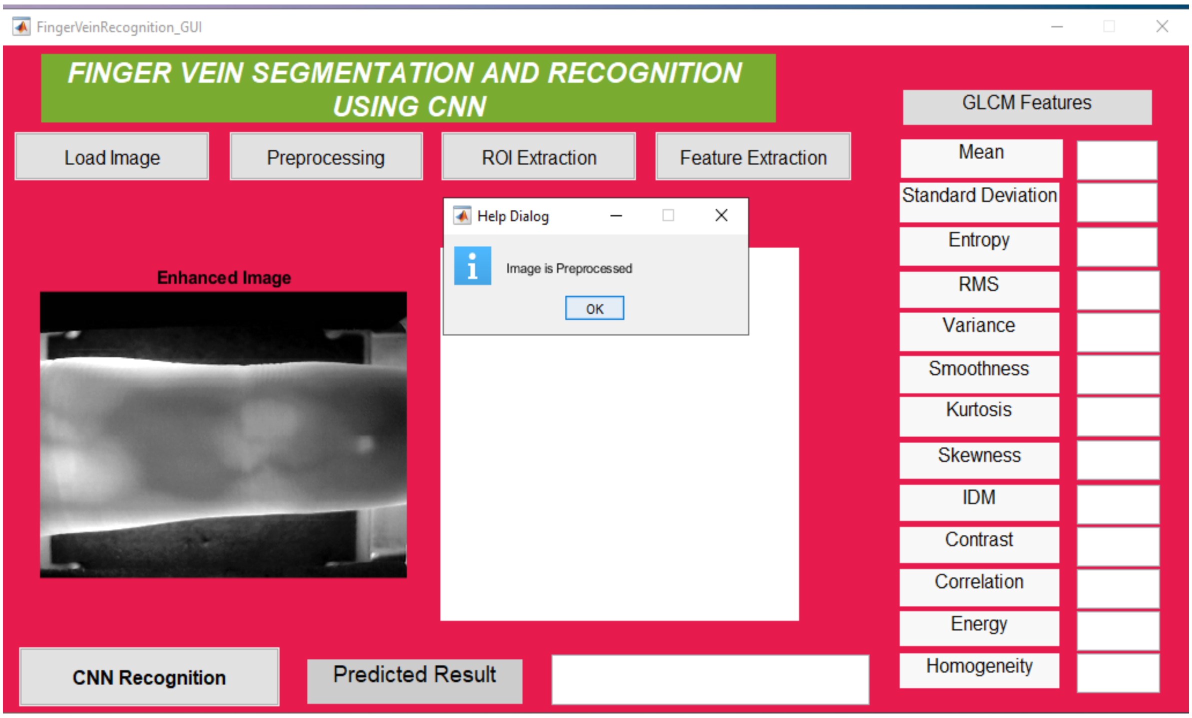


Figure A 1.5 EXTRACTION OF ROI

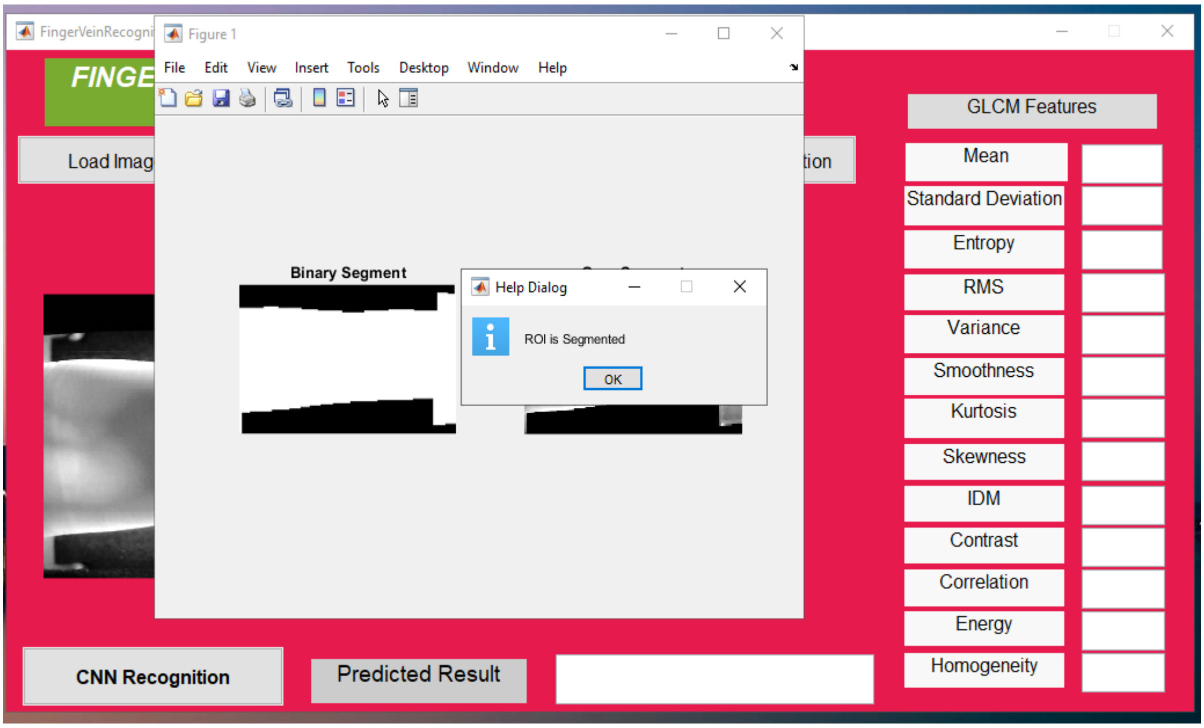


Figure A 1.6 BINARY SEGMENT & GRAY SEGMENT

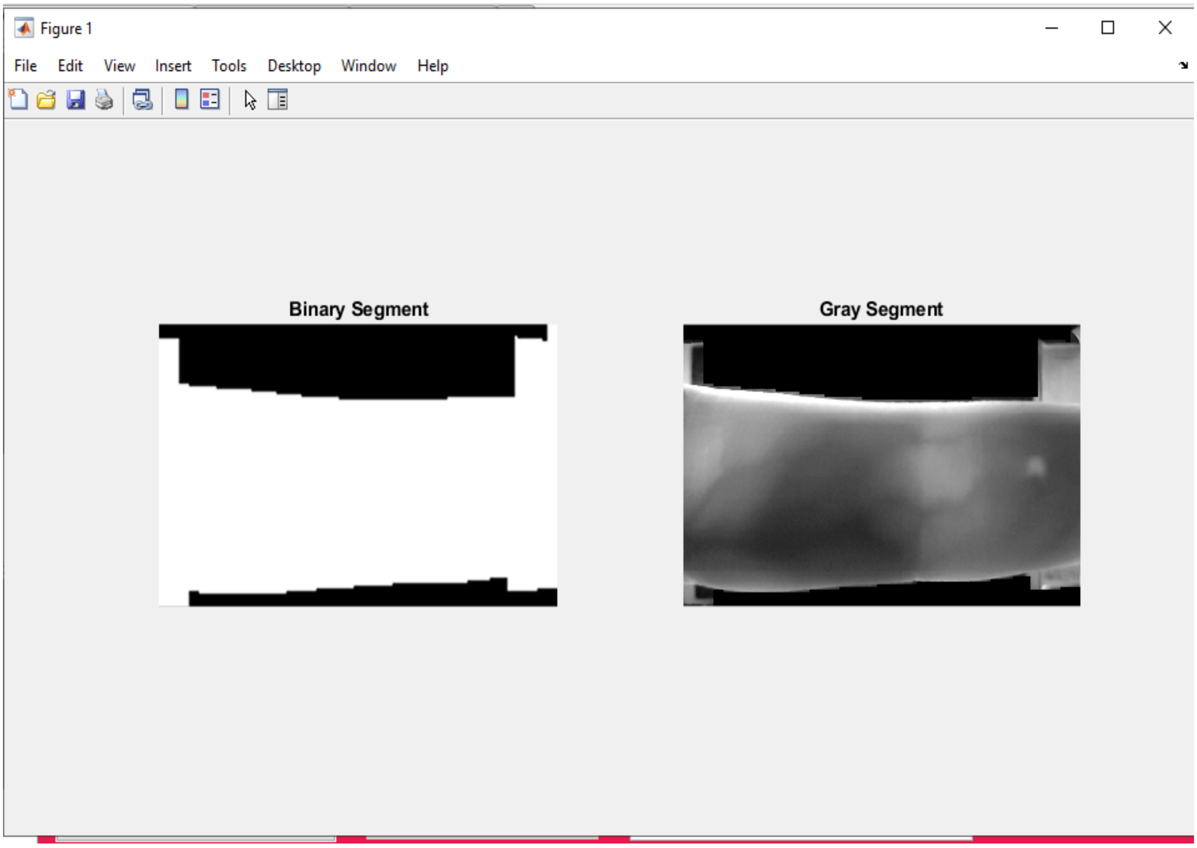


Figure A 1.7 ENHANCED IMAGE

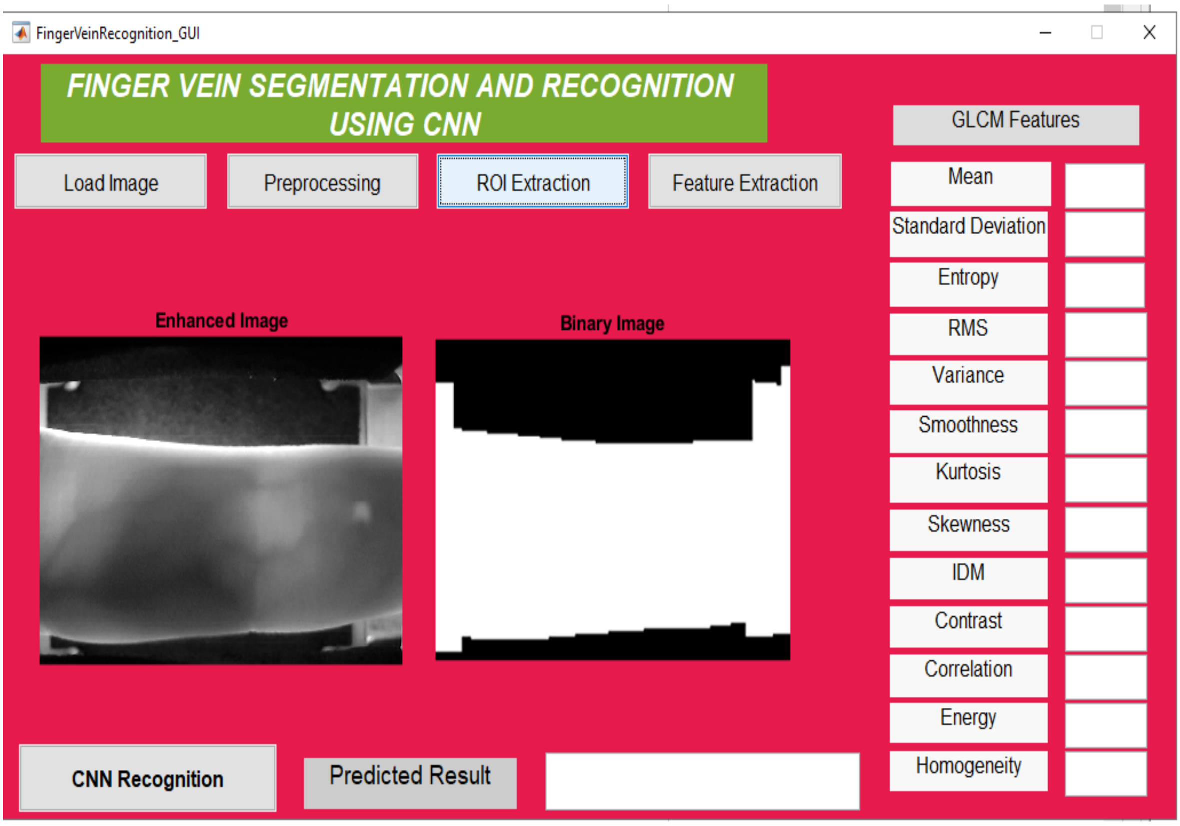
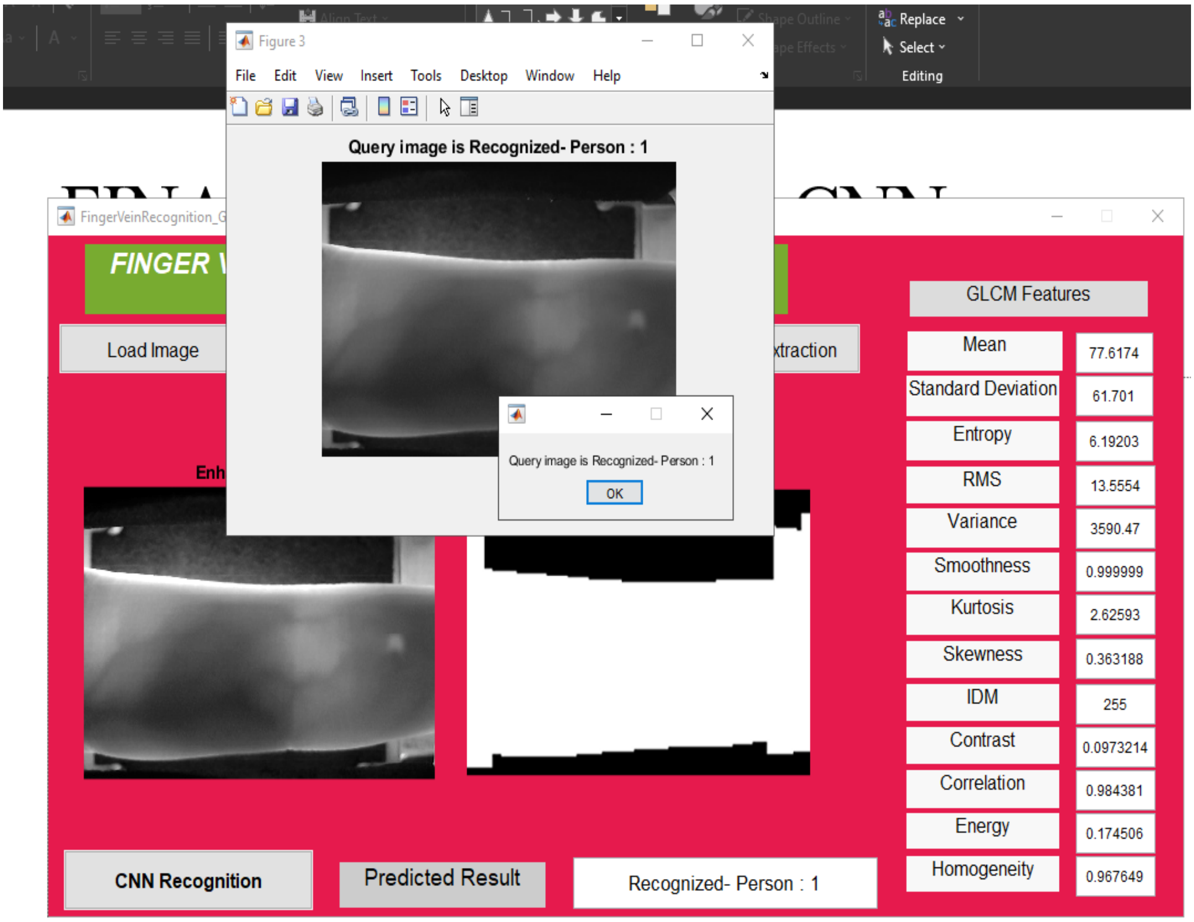


Figure A 1.8 FEATURES OF EXTRACTION

Figure A 1.9 FINAL RESULT WITH CNN RECOGNITION



REFERENCES

[1] Rig Das, Emanuela Piciucco, Emanuele Maiorana, Patrizio Campisi, “Convolutional Neural Network for Finger-Vein Based Biometric Identification”, IEEE Transactions on Information and Security, vol. 14, no.3, pp. 1-14, 2019.

[2] K. S. Itqan,A. R. Syafeeza , F. G. Gong, N. Mustafa, Y. C. Wong and M. M. Ibrahim, “User Identification based on Finger Vein Patterns Using Convolutional Neural network”, ARPN Journal of Engineering and Applied Sciences, vol. 11, no.4, pp.1-4, 2016.

[3] Musab Coúkun, Ayúegül Uçar, “Face Recognition Based on Convolutional Neural Network”, International Conference on Modern Electrical and Energy Systems (MEES*)*, pp. 376-379, 2017.

[4] Hyung Gil Hong, Min Beom Lee, Kang Ryoung, “Convolutional Neural Network-Based Finger Vein Recognition using NIR Image Sensors”, Sensors, vol.17, no.6, pp. 1-10, 2017.

[5] Shazeeda, Bakhtiar Affendi Rosd, “Nearest Centroid Neighbor Based Sparse Representation Classification for Finger Vein Recognition”, IET Biometrics, vol. 8, no.1, pp. 49-58, 2019.

[6] Bakhtiar Affendi Rosdi, Chai Wuh Shing, Shahri Azmin Suandi, “Finger Vein Recognition Using Local Line Binary Pattern”, Sensors, vol. 11, no.12, pp. 11357-11371, 2011.

[7] K. Syazana-Itqan, A. Syafeeza, N. Saad, N. A. Hamid, and W. H. B. M. Saad, “A Review of Finger-Vein Biometrics Identification Approaches”, Indian Journal of Science and Technology, vol. 9, no. 32, 2016.

[8] E. C. Lee and K. R. Park, “Image Restoration of Skin Scattering and Optical Blurring for Finger Vein Recognition”, Optics and Lasers in Engineering, vol. 49, no. 7, pp. 816-828, 2011.

[9] S. Xie, L. Fang, Z. Wang, Z. Ma, and J. Li, “Review of Personal Identification Based on Near Infrared Vein Imaging of Finger”, 2nd International Conference on Image, Vision and Computing (ICIVC)*,* pp. 206-213, 2017.

[10] Y. Yin, L. Liu, and X. Sun, "SDUMLA – HMT: A Multimodal Biometric Database", Biometric Recognition, Lecture Notes in Computer Science, vol 7098. Springer, Berlin, Heidelberg, pp.260 -268, 2011.

[11] A. Kumar and Y. Zhou, "Human Identification Using Finger Images", IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 2228-2244, 2012.

[12] B. T. Ton and R. N. Veldhuis, "A High Quality Finger Vascular Pattern Dataset Collected Using a Custom Designed Capturing Device", 6th IAPR International Conference on Biometrics, pp.1- 5, 2013.

[13] 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 Conference on Image and Signal Processing (CISP), pp. 1-10, 2013.

[14] W. Yang, X. Huang, F. Zhou, and Q. Liao, "Comparative Competitive Coding for Personal Identification by Using Finger Vein and Finger Dorsal Texture Fusion", Information sciences, vol. 268, pp. 20-32, 2014.

[15] Y. Lu, S. Yoon, and D. S. Park, "Finger Vein Identification System Using Two Cameras", Electronics Letters, vol. 50, no. 22, pp. 1591-1593, 2014.

[16] J. Hashimoto, "Finger Vein Authentication Technology and Its Future", Symposium on VLSI Circuits, pp. 5-8, 2006.

[17] R. Raghavendra, J. Surbiryala, K. B. Raja, and C. Busch, "Novel Finger Vascular Pattern Imaging Device for Robust Biometric Verification," IEEE International Conference on Imaging Systems and Techniques, pp. 148-152, 2014

[18] Y. Lu, S. J. Xie, S. Yoon, J. Yang, and D. S. Park, "Robust Finger Vein ROI Localization Based on Flexible Segmentation", *Sensors*, vol. 13, no. 11, pp. 14339-14366, 2013.

TECHNICAL BIOGRAPHY

Mr. Varun Kumar K is a final year student at B.S. Abdur Rahman Crescent Institute of Science & Technology, Tamil Nadu, India. He is currently pursuing his B. Tech Degree in Information Technology. His areas of interest include Data Science and Artificial Intelligence. Mr. Varun Kumar has completed certification course on Machine learning from Internshala.

Mr. Saquib Nawaz is a student at B.S. Abdur Rahman Crescent Institute of Science & Technology, Tamil Nadu, India. He is currently pursuing his B.Tech Degree in Information Technology. His areas of interest are Artificial Intelligence and Cloud.