International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 07 | July - 2022 Impact Factor: 7.185 ISSN: 2582-3930

FINGERPRINT ATTENDENCE SYSTEM WITH LIVENESS DETECTION

ALSSIYA K ABRAHAM , Rinsu Aravind (Guide)

Department of Computer Science and Engineering

Holy Grace Academy of Engineering

Mala, Thrissur

ABSTRACT

Over the years, the manual method of taking attendance in any higher institution of learning are often violated because they can easily manipulate the manual attendance register. The attendance of the students is essential for the progress of an institution as they help to move the institution forward. Due to the uniqueness and consistency of fingerprints, they have been used for identification over many years and is being automated due to advancement in computing capabilities.

Taking the advantage of recent technologies, we have developed an automatic fingerprint-based attendance registering system using Slim-Residual Convolutional Neural Network (Modified-CNN) and Octantal Neatest-Neighbourhood Structure (ONNS). Firstly, the CNN score feature vector is processed to discriminate between live fingerprints and fake ones.

Secondly, the ONNS method is used to find closest minutia, from each sector of octant, to the central minutia (core point of the fingerprint) and used for fingerprint matching with the fingerprints of the students maintained in the database.

Experimental results are carried on the images collected from the students to show the results of the proposed algorithms.

CHAPTER 1
INTRODUCTION

Compared to traditional identity verification, such like a key, a card, and a password, biometrics are not easy to do theft or easy loss. Among the many biometric verification methods, such as face, iris, sound, fingerprints and movement, fingerprints have become the most popular and reliable ownership. Verification methods due to its diversity, consistency and the whole universe. At the same time, the safety of fingerprint recognition systems has become increasingly important and has gradually raised public attention, as some studies have shown that fingerprint recognition systems have four mainly security threats, such as the use of counterfeit fingerprints attack finger sensors, communication modules, software modules, and data storage.

The normal attendance system is followed in education system in which the teacher says everyone's name student and tagged attendees cause time wastage study time. This becomes even more difficult especially the current situation where the number of students in the class is very high great. Managing the presence data of such a large group and it is very difficult. Something wrong with the current system an opportunity for the student to mark a false visit. Fingerprints supported devices are used in corporate environments. These devices that use computers to store and verify fingerprints. It can it has been replaced by an educational environment.

The proposed integration approach to this project could improve the resistance of the fingerprint verification system to some extent, and at the same time slightly affect the performance of real fingerprint recognition.

The integrated fingerprint system increases the Fraud Rejection Rate of the custom users to some extent, which can not only reject fraud, but also effectively prevent spoof attacks to ensure the safety of the fingerprint recognition system.

Based on the theory that living and artificial finger images are varied texture, a method of combining points is proposed between fingerprint matching school and FLD school for producing interactive features and polynomial features as a result feature vector. The integration system effectively prevents spoof attacks on the surface of the sensor of fingerprints in combination Software-based FLD has become a fingerprint recognition system, to avoid expensive costs by combining additional hardware resources. Development of an image classifier system classify the image real or fake images, deep learning and classical machine learning methods are compared for feature extraction, fully automated fingerprint segmentation using ONNs based and deep

leaning methods on a extracted dataset. The methods are tested on image collected from the students to show the results of the proposed algorithms.

1.1 FEATURE EXTRACTORS

The features that are we used in our method is ONNS and Modified CNNs. ONNs takes the unique feature of each fingerprint while place on the senser and gives to the Modified CNN, neurons for training through a score value. After that CNN start to learn by logistic regression classifier and identify the liveness.

1.1.1 Octantal Neatest-Neighbourhood Structure

Octantal Neatest-Neighbourhood Structure (ONNS), is utilized for obtaining the matching score of two fingerprint images. The purpose of matching fingerprints is to determine two fingerprints appear on the same finger with to calculate the similarity of the two images. Algorithms with fingerprints based on the minutiae are currently widely used, and certain types of minutiae are limited to both: conclusions and bifurcations. It can be defined using parameters like links, direction and type.

The algorithm creates ONNS per minute by equally dividing the minutiae-focused area into 8 categories (angle of each category is 45◦), and direction (θi) of minutia (Mi) is considered the first angle. Next, the minutia closest to the central minutia is found in each one sector.

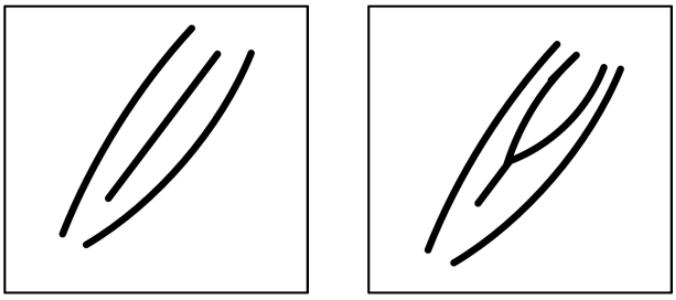

Figure 1.1: The minutiae of fingerprint image

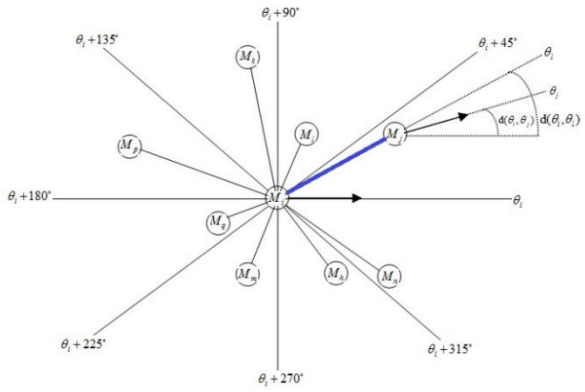

Figure 1.2: 8 sectors in ONNS

1.1.2 Modified convolutional neural network(modified-CNN)

Convolutional Neural Networks (CNNs), which have been widely used in computer vision, make outstanding performance in image classification, object detection and many other tasks, attributing to the impressive ability of extracting local features. CNN has played an important role in development of acquisition of fingerprints, which can remove or minimize dependence on domain information.

Most available methods using CNN transmit pre-trained CNN models instead of redesigning a new network structure aimed at gaining a lifetime of fingerprints. These methods use fingerprint images to fine-tune CNN models previously trained in natural photography. The inevitable difference between fingerprints and natural images makes parameters of pre-trained models in natural images no find good performance in getting fingerprints.

The proposed modified CNN framework is different from the original residual network in that only nine improved residual blocks are stacked into Slim-CNN and less convolutional kernels are employed, making less training time and improving classification performance for fingerprint spoof detection. CNN-based method, which adopts a voting strategy based on minutiae-centered multiple local patches, showing state-of the-art average classification accuracy.

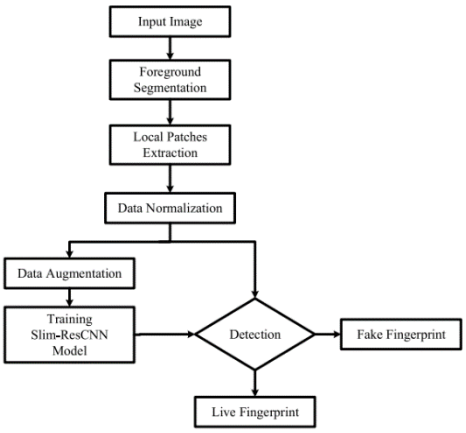

Figure 1.3: flow chart of fingerprint detection using modified CNN

1.1.3 Minutiae detection

The method we used in our project to find out the fake or real fingerprint by checking the minutiae of the input fingerprint. In real time the scanner scans the finger and take the image of the fingerprint. By using image processing mechanisms, the image gets enhanced, segmented and by ONNs it will divide into 8 sectors. By calculating the distance measure, we can characterise the two different types of minutiae: Endings and Bifurcation. The Figure 1.1 shows the two types of minutiae.

By these features, we can determine whether the input fingerprint is real or fake. There is an algorithm which used to find the core point in the fingerprint, that will be further discuss later.

2.1 CLASSIFIERS

A classification model tries to draw some conclusion from the input values given for training. It predicts the class labels or categories for the new data. In our project we are using deep learning classifier for classification. The deep learning model is Modified CNN. It is used for classify the fake and real fingerprint. That is for liveliness detection. If a manipulated fingerprint is used for authentication Modified CNN detects it as a fake fingerprint and access will be denied

CHAPTER 2
LITERATURE SURVEY

2.1 Fake Finger Detection Based on Time-Series Fingerprint Image Analysis [2]

This work introduces a new approach to detect fake fingers, based on the analysis of time-series fingerprint images. When a user puts a finger on the scanner surface, a time-series sequence of fingerprint images is captured. Five features are extracted from the image sequence. Two features represent the skin elasticity, and three features represent the physiological process of perspiration. Finally, the Support Vector Matching (SVM) is used to discriminate the finger skin from other materials such as gelatine. The experiments carried out on a dataset of real and fake fingers show that the proposed approach and features are effective in fake finger detection.

2.2 Fake Finger Detection Based on Thin-Plate Spline Distortion Model [3]

This paper introduces a novel method based on the elasticity analysis of the finger skin to discriminate fake fingers from real ones. We match the fingerprints before and after special distortion and gained their corresponding minutiae pairs as landmarks. The thin-plate spline (TPS) model is used to globally describe the finger distortion. For an input finger, we compute the bending energy vector by the TPS model and calculate the similarity of the bending energy vector to the bending energy fuzzy feature set. The similarity score is in the range [0, 1], indicating how much the current finger is similar to the real finger. The method realizes fake finger detection based on the normal steps of fingerprint processing without special hardware, so it is easily implemented and efficient. The experimental results on a database of real and fake fingers show that the performance of the method is available.

2.3 Wavelet based fingerprint-based detection[4]

This paper proposed a simple and effective approach for fingerprint liveness detection based on the wavelet analysis of the fingertip surface texture. Experimental results show that our method can successfully differentiate live finger tips from fake finger tips made of most commonly used material in fingerprint spoofing.

2.4 Fake finger detection by skin distortion analysis[5]

Attacking fingerprint-based biometric systems by presenting fake fingers at the sensor could be a serious threat for unattended applications. This work introduces a new approach for discriminating fake fingers from real ones, based on the analysis of skin distortion. The user is required to move the finger while pressing it against the scanner surface, thus deliberately exaggerating the skin distortion. Novel techniques for extracting, encoding and comparing skin distortion information are formally defined and systematically evaluated over a test set of real and fake fingers. The proposed approach is privacy friendly and does not require additional expensive hardware besides a fingerprint scanner capable of capturing and delivering frames at proper rate. The experimental results indicate the new approach to be a very promising technique for making fingerprint recognition systems more robust against fake-finger-based spoofing attempts.

2.5 Analysis of Fingerprint Pores for Vitality Detection[6]

Spoofing is an open-issue for fingerprint recognition systems. It consists in submitting an artificial fingerprint replica from a genuine user. Current sensors provide an image which is then processed as a "true" fingerprint. Recently, the so-called 3rd-level features, namely, pores, which are visible in high-definition fingerprint images, have been used for matching. In this paper, we propose to analyse pore's location for characterizing the "liveness" of fingerprints. Experimental results on a large dataset of spoofed and live fingerprints show the benefits of the proposed approach**.**

2.6 Fake-fingerprint detection using multiple static features[7]

Recently, fake fingerprints have become a serious concern for the use of fingerprint recognition systems. We introduce a novel fake fingerprint detection method that uses multiple static features. With regard to the usability of the method for field applications, we employ static features extracted from one image to determine the aliveness of fingerprints. We consider the power spectrum, histogram, directional contrast, ridge thickness, and ridge signal of each fingerprint image as representative static features. Each feature Is analysed with respect to the physiological and statistical distinctiveness of live and fake fingerprints. These features form a feature vector set and are fused at the feature level through a support vector machine classifier. For performance evaluation and comparison, a total of 7200 live images and 9000 fake images were collected

using four sensors (three optical and one capacitive). Experimental results showed that proposed method achieved approximately 1.6% equal-error rate with optical-based sensors. In the case of the capacitive sensor, there was no test error when only one image was used for a decision. Based on these results, we conclude that the proposed method is a simple yet promising fake-fingerprint inspection technique in practice.

2.7 Presentation Attack Detection Using a Tiny Fully Convolutional Network [10]

Fingerprint authentication is widely used thanks to its simple process and low cost, but it is vulnerable to fake fingerprints. Many researchers have been working on presentation attack detection to ensure the security of fingerprint systems. However, the existing studies only focus on improving the detection accuracy, and let processing time and memory requirement be out of focus. Hence, it is difficult to integrate the existing algorithms to embedded and mobile systems. This paper proposes a method to detect presentation attacks using a small fully convolutional network. The proposed network is designed using the structure of the fire module of Squeez-Net. The use of the fire module results in a network which has around 0.5 million parameters. The proposed network is trained using images of 32 × 32, 48 × 48, or 64 × 64 pixels. Since the network has no fully connected layer, it can interfere with images of any size. This advantage helps to improve the detection rate and allows the proposed algorithm to be easily integrated into fingerprint systems. The experiments show an average detection error of 1.43%, which is comparable with the state-of-the-art accuracy, while the processing time and memory requirement are much reduced.

2.8 Slim-ResCNN: A deep residual convolutional neural network for fingerprint liveness detection [11]

Fingerprint liveness detection has gradually been regarded as a primary countermeasure for protecting the fingerprint recognition systems from spoof presentation attacks. The convolutional neural networks (CNNs) have shown impressive performance and great potential in advancing the state-of-the-art of fingerprint liveness detection. However, most existing CNNs-based fingerprint liveness methods have a few shortcomings: 1) the CNN structure used on natural images does not achieve good performance on fingerprint liveness detection, which neglects the inevitable differences between natural images and fingerprint images; or 2) a relative shallow architecture (typically several layers) has not paid attention to the capability of deep network for spoof fingerprint detection. Motivated by the compelling classification accuracy and desirable convergence behaviours of the deep residual network, this paper proposes a new CNN-based fingerprint liveness detection framework to discriminate between live fingerprints and fake ones. The proposed framework is a lightweight yet powerful network structure, called Slim-ResCNN, which consists of the stack of series of improved

residual blocks. The improved residual blocks are specifically designed for fingerprint liveness detection without overfitting and less processing time. The proposed approach significantly improves the performance of fingerprint liveness detection on LivDet2013 and LivDet2015 datasets. Additionally, the Slim-ResCNN wins the first prize in the Fingerprint Liveness Detection Competition 2017, with an overall accuracy of 95.25%.

2.9 Anti-spoofing in action: joint operation with a verification system [9]

Besides the recognition task, today's biometric systems need to cope with additional problem: spoofing attacks. Up to date, academic research considers spoofing as a binary classification problem: systems are trained to discriminate between real accesses an attacks. However, spoofing counter-measures are not designated to operate stand-alone, but as a part of a recognition system they will protect. In this paper, we study techniques for decision level and score-level fusion to integrate a recognition and anti-spoofing systems, using an open-source framework that handles the ternary classification problem (clients, impostors and attacks) transparently. By doing so, we are able to report the impact of different spoofing counter-measures, fusion techniques and thresholding on the overall performance of the final recognition system. For a specific use case covering face verification, experiments show to what extent simple fusion improves the trustworthiness of the system when exposed to spoofing attacks.

2.10 Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition [8]

To ensure the actual presence of a real legitimate trait in contrast to a fake self-manufactured synthetic or reconstructed sample is a significant problem in biometric authentication, which requires the development of new and efficient protection measures. In this paper, we present a novel software-based fake detection method that can be used in multiple biometric systems to detect different types of fraudulent access attempts. The objective of the proposed system is to enhance the security of biometric recognition frameworks, by adding liveness assessment in a fast, user-friendly, and non-intrusive manner, through the use of image quality assessment. The proposed approach presents a very low degree of complexity, which makes it suitable for real-time applications, using 25 general image quality features extracted from one image (i.e., the same acquired for authentication purposes) to distinguish between legitimate and impostor samples. The

experimental results, obtained on publicly available data sets of fingerprints, iris, and 2D face, show that the proposed method is highly competitive compared with other state-of-the-art approaches and that the analysis of the general image quality of real biometric samples reveals highly valuable information that may be very efficiently used to discriminate them from fake traits.

CHAPTER 3

SYSTEM DESIGN

The dataset that are taken for the method are total 20 set where 10 are real fingerprint and 10 are fake real fingerprint. The dimension for the image which is fed as 252x324 in .png format. The acceptable other image formats are .bmp and .jpg . when the sensor captures the fingerprint image it will convert to the require format and introduced to feature extraction steps. The image processing method is taking place here for image enhancement.

The sample images of real and fake fingerprint as shown.

Figure 3.1: sample of Fake fingerprint and Real fingerprint

The input image is fed for some mechanism in order to get a clear input to extract the features.

The below are the steps:

Step 1: Scan fingerprint (test/query input)

Step 2: Fingerprint Segmentation

Step 3: Modified-CNN

a) Training Phase of Modified-CNN

b) Testing Phase of Modified-CNN

Step 4: ONNS

Step 5: Fingerprint recognized/rejected and attendance marked in the corresponding day's excel sheet if recognized.

3.1 Scan fingerprint

The optical sensor scans the input image of fingerprint and convert to jpg\ pmp \jpg format. The scanned fingerprint is a new one it will check its features and stores to the real fingerprint file which is already installed file. Else it is a registered fingerprint then checks for the following method. By these checking, the system can understand whether the input fingerprint is fake or real. These is an GUI for software testing phase also can use for any hardware failure occurs.

The below Figure shows the software interface model which created for the test.

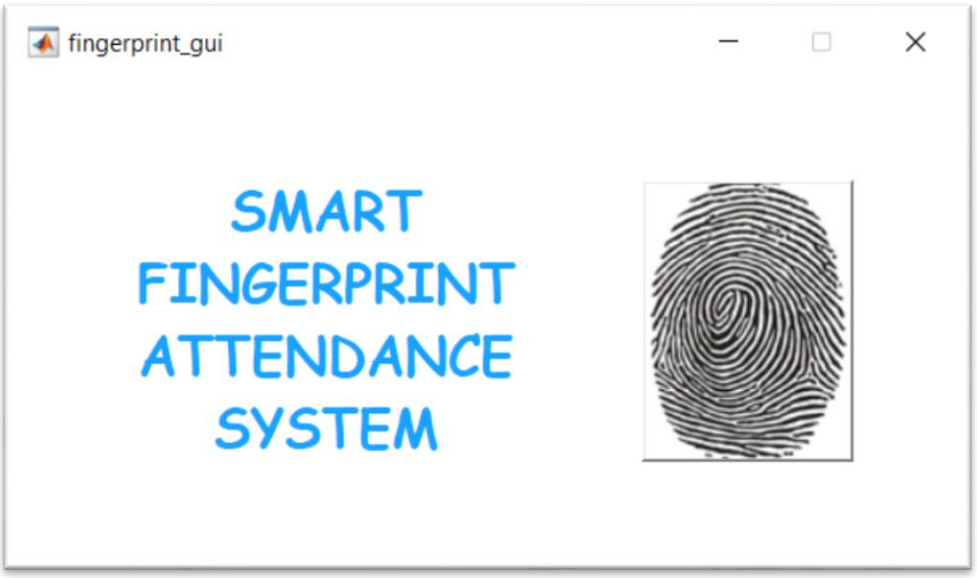

Figure 3.2: User interface software model

By clicking the fingerprint button in the window will directed to the storage file. The manual input of fingerprint can be subjected to the analysing phase. User need not to worry about the background calculation and functions happening.

3.2 Fingerprint segmentation

The input fingerprint image is taken from user interface is fed for segmentation of image in-order to define the features and also to enhance the image. As the first step the image is segmented where:

- Range filter: In the range method, the colour value of each pixel is replaced with the difference of maximum and minimum of the colour values of the pixels in a surrounding region.

- Local adaptive threshold: local adaptive thresholding is used to convert an image consisting of gray scale pixels to just black and white scale pixels. Usually a pixel value of 0 represents white and the value 255 represents black with the global thresholding to 245 representing different gray levels.
- Morphological opening and closing operation: opening and closing are dual operation used in image processing for restoring an eroded image. Opening is generally used to restore or recover the original image to the maximum possible extent. Closing is generally used to smoother the contour of the distorted image and for getting rid of the narrow breaks and long thin gulfs.

  Closing is also used for getting rid of the small holes of the obtained image. The combination of the opening and closing is generally used to clean up artifacts in the segmented image before using the image foe digital analysis.

  Here in our method, we used a disk-shaped structuring element of radius 6.
- Morphological erosion operation: it shrinks the foreground objects. Enlarge foreground holes. A larger size of the structure element, the effect of erosion increase.by this using the square-shaped structuring element.
- Boundary smoothing: the smoothing method is a complex Fourier transform domain. Fourier transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier of frequency domain, while the input image is the spatial domain equivalent.

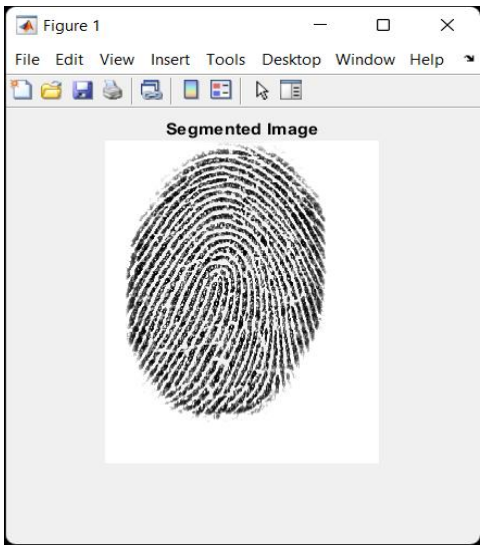

Figure 3.3 Fingerprint segmentation

3.3 Modified CNN

In this study, we propose a Slim-Res Convolutional Neural Network (modified CNN) based approach for fingerprint recognition. The modified CNN consists of Conv1, Conv2, Conv3 (Conv3_1, Conv3_2), and Conv4 (Conv4_1, Conv4_2), followed by a global average pooling (Avg_Pool) and a final classification layer.

During the training phase; Extract Features from the Modified CNN model for each image in the database and save the extracted features.

When in testing phase;

1. Select input test/query image
2. Extract Features from the fully connected layer of Modified CNN model for the test/query
   a. Conv1 is responsible for connecting the input and extract the initial features which are delivered to the followed residual block.
   b. The Conv1 is followed Conv2, Conv3 (Conv3_1, Conv3_2) and Conv4 (Conv4_1, Conv4_2).
   c. Global average pooling reduces network model parameters which prevents overfitting and reduce the network computation cost.
3. Train for classification using SoftMax layer on the *TrainFeaturesCNN* and the *spoof Labels*
4. Classify the *TestFeaturesCNN* into live fingerprints or fake ones using the trained SoftMax layer

This method provides a high quality of classification property. Therefore, this gives the expected accuracy.

Ground Name	Output Size	Block Type=B(3,3)
Conv1	112×112	$[3 \times 3/1, 32]$
Conv2	112×112	$\begin{bmatrix} 3 \times 3/1, 32 \\ 3 \times 3/1, 32 \end{bmatrix} \times 3$
Conv3_1	56×56	$\begin{bmatrix} 3 \times 3/2, 64 \\ 3 \times 3/1, 64 \end{bmatrix}$
Conv3_2	56×56	$\begin{bmatrix} 3 \times 3/1, 64 \\ 3 \times 3/1, 64 \end{bmatrix} \times 2$
Conv4_1	28×28	$\begin{bmatrix} 3 \times 3/2, 128 \\ 3 \times 3/1, 128 \end{bmatrix}$
Conv4_2	28×28	$\begin{bmatrix} 3 \times 3/1, 128 \\ 3 \times 3/1, 128 \end{bmatrix} \times 2$
Avg_Pool	1×1	$[28 \times 28]$

Classification layer in Modified CNN network

3.4 Octantal Neatest-Neighbourhood Structure (ONNS)

Using the proposed ONNS, we develop walking algorithm for detecting the core point. By the core point, the given fingerprint image is divided into eight sectors. From there select the nearest minutiae points

and analysis the real or fake character. Before entering into the CNNS process, the image must make some clear image in-order to divide with clear measures. The following steps are taken place:

3.4.1 Fingerprint enhancement

The basic idea behind enhancing the image to remove the noise, sharpen, or brighten the image. There is certain process done for enhance the image like: ridge segmentation, ridge orientation, ridge frequency by a call to 'FREQEST', ridge filter.

Ridge segmentation is used for normalize the intensity value of the image where to identify the part of the fingerprint. The region is determined by the standard deviation, so the ridge region have zero mean. The image is divided into block of size 'blksze x blksze' and standard deviation is computed for each block. There is a threshold mentioned, if the standard deviation is grater than the threshold it is considered as the part of the fingerprint. The below Figure 3.4 is an example of enhanced image during the training phase.

Ridge orientation gives the information about the orientation of the fingerprint. It is the directionality of the fingerprint in X and Y direction. By weighted summation of the data it gives an analytic solution of principal direction. It estimates the local ridge orientation at each point by finding the principal axis of variation in the image gradients. Figure 3.5 shows the ridge orientation of the fingerprint during the training phase.

Ridge frequency, the cropped region which need to enhance the image need a frequency to highlight the ridge and valley of the fingerprint. The darker shade is the ridge and the valley is lying between the ridge. The image block will rotate to vertical. Find peaks in projected grey values by performing a grayscale dilation and then finding where the dilation equals the original values. Determine the spatial frequency of the ridge by dividing the distance between the first and last peaks by the no. of peaks-1. If no peaks are detected, or the wavelength is outside the allowed bounds, the frequency image is set to 0.

Ridge filter take the valid frequency data and generate an array of the distinct frequencies. These will result to generate a table by multiplied by 100 to obtain an integer index, returns the index within the un-frequency array. After that it generate filter corresponding to these distinct frequency and orientation is incremented. Then it generates a rotated version of the filter, find indices of matrix values from radians to an index value that corresponding to round. Finally do the filtering. The Figure 3.5 and Figure 3.6 shows after result, during the training phase, of the above explained.

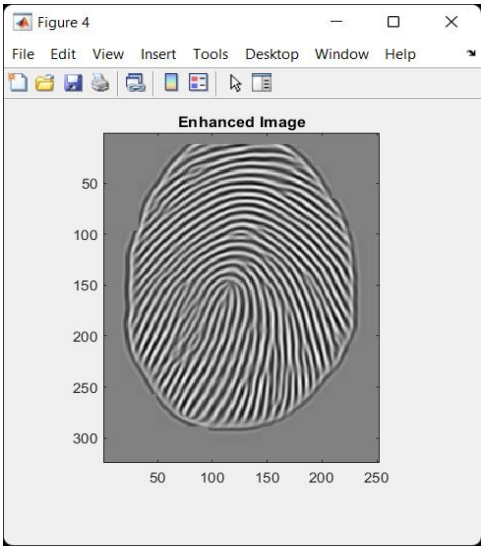

Figure 3.4: Enhanced image of fingerprint

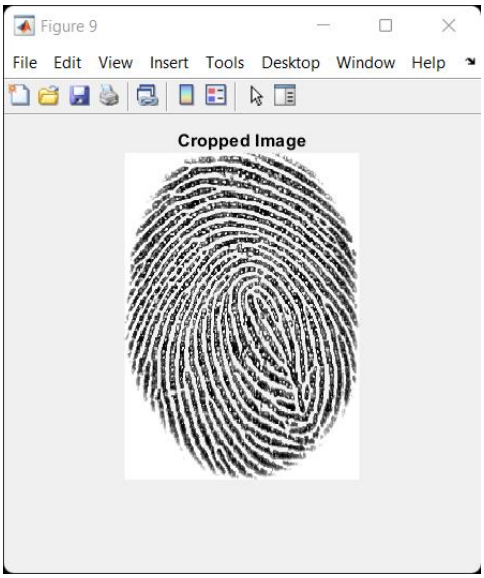

Figure 3.5: Cropped image

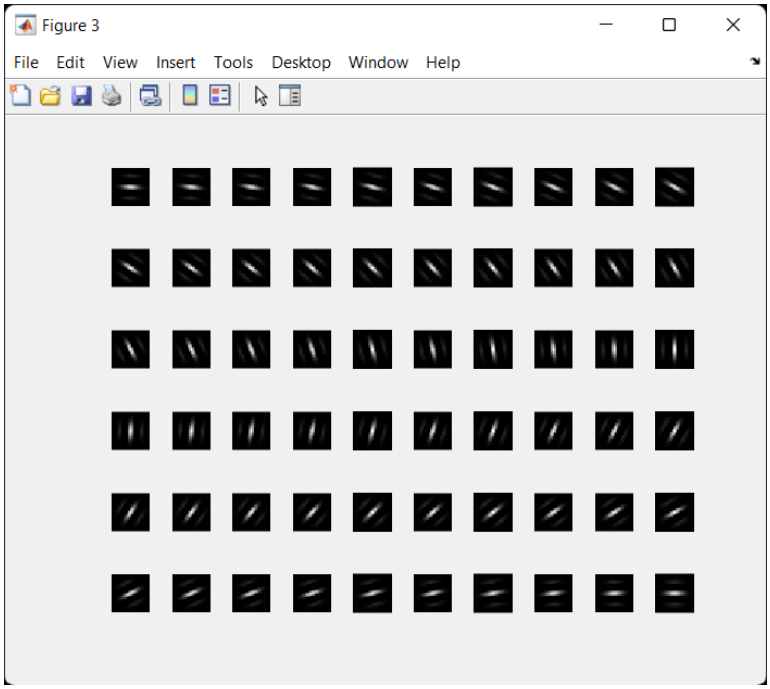

Figure 3.6: Ridge orientation

3.4.2 Binarization

Binarization is a process of converting a coloured image into a black and white image. For better classification and simplicity of analysing the image character converting 0-255 scale to 0-1 grey scale. Here the data features are converting into binary format. Somehow this is a better way to neglect the noise happen in the image.

During minutiae detection these binarized image can show up the accurate character of the ridge. Whereas it is also an advantage to compare the realness during in real-time process.

We use a binary mask to highlight the region of interest (ROI), which can easy to understand the classifier to take the region to be taken for binarization. The interested part is highlighted for binarization, which shown in white and the background I shown in black. These are done through pixel analysis of the image. The boundary of the region is taken as a polygonal shaped as shown in the Figure 3.8 below.

The below Figure 3.7 and Figure 3.8 shows the binarized image of the fingerprint and the mask used for binarization, which get output during test phase.

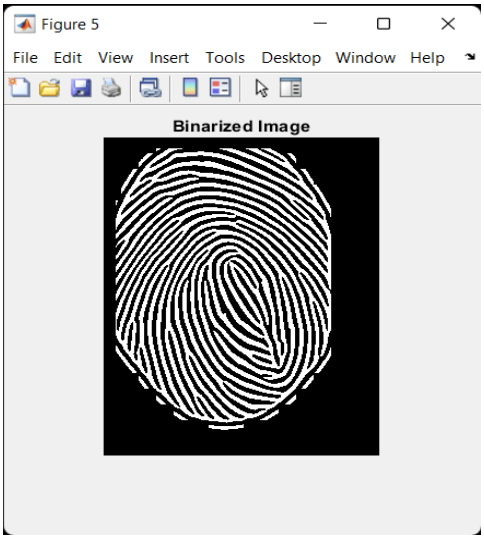

Figure 3.7: Binarized image

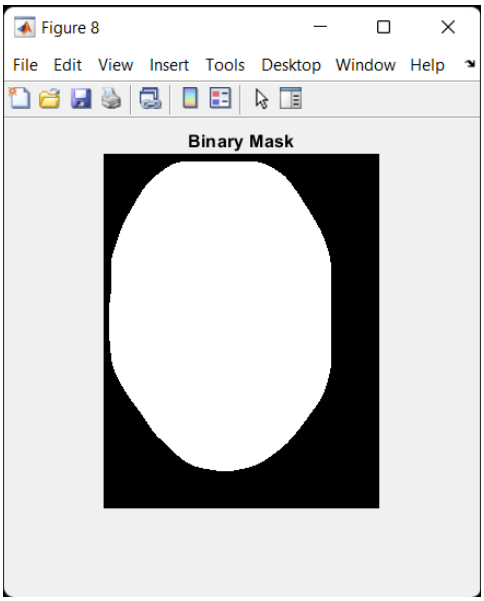

Figure 3.8: Binary mask

3.4.3 Thinning

As the image is enhanced and binarized, it can easy to thin the outline of each ridge of the fingerprint. By the method of thinning, we get a skeleton Figure of the fingerprint. This shows the perfect lining of the minutiae, where further used for minutiae detection. The darken binarized image is thinned to a skeleton bounded image, only the bounded pixels are highlighted and inner pixels are made to 1 (white). The Figure 3.6 shows a thinned Figure of the fingerprint during the test phase.

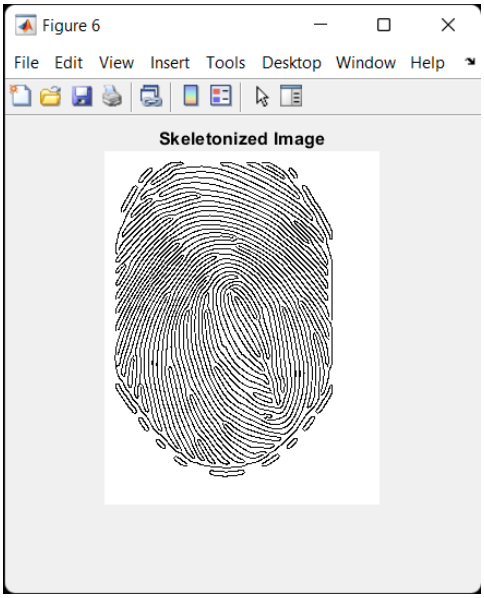

Figure 3.9: Skeleton image of fingerprint

3.4.4 Minutiae detection

The measurements of the minutiae are the main data part which determines the realness of the input fingerprint. By the detecting of the two types of characters: ridge and bifurcation, the authorization of the person. If the person is a new one, his/her fingerprint features will be stored by extracting the features mentioned above sections.

Minutiae is detected by taking the image into 2x2 block size and sum it. If the sum is equal to 2, then it is a ridge of the fingerprint. If the sum is equal to 4, then it is a bifurcation of the fingerprint. The Figure 1.1 shows the two types of fingerprint character. Figure 3.10 shows the minutiae detected parts which denoted blue as ridge and red ad bifurcation.

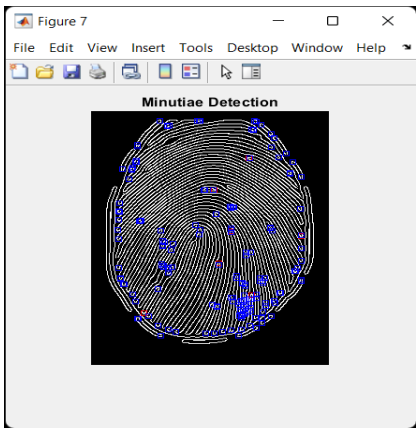

Figure 3.10: Minutiae detection

3.4.5 Removal of false minutiae

During detection time some noise can make as minutiae which led to false classification of real fingerprint data. Therefore, such part of detection is removed to reduce the false rate and make the classification right.

3.4.6 Core point detection

Before introducing to ONNS function, we need to find the core point which is reference to the minutiae for the distance calculation. So, we use an algorithm called walking algorithm, for detecting of core point.

Generally, in fingerprint there are two kind of loop property can see, delta and core. It is confusing that which part is the core point. So, by this algorithm as name suggests, the region is segmented, oriented then take a walk-through pixel by pixel.

The main function of the walking algorithm to get all singular points in a given fingerprint image. Fingerprint singular points (upper core, lower core and delta), defined as where the orientation field is discontinuous or the ridge curvature is the highest, are essential for registration and identification (specially for image-based approach instead of minutiae-based approach). Below steps indicate the detection of singular points in the algorithm:

Let $Path = \{P0, P1, \cdots, Pk\}$ denote the walking path and δk denote the least distance between the end point of *Path*. Given a threshold T, then we will stop walking once $\delta k < T$, where we say a loop occurs on the walking path. The centre point of the detected loop serves as a candidate for the upper core.

1) Sampling starting points
2) The upper left corner of the bounding box of foreground
3) The lower right corner of the bounding box of foreground
4) Sampling step length along x- and y-coordinate
5) Detect cores
6) For different rotation angle of ‘wdf’
7) Walking directional field of cores
8) Found a loop, end the walking process
9) Get a candidate, take a neighbour as a start point to confirm
10) Merge points that are too close to each other

As a result, we get a core point corresponding to the input fingerprint image. From this it will automatically starts to construct the ONNs.

3.4.7 Construction of ONNS

The construction of ONNS is make the image divide into 8 sectors with angle of 45°. Then finds the closest minutia to the central minutia is found from each sector. The below Figure 3.11 shows the division of sector of the image into 8 sectors.

In training phase, the steps are similar to the above steps. Only during the minutiae detection is different where perform from the segmented image of each image in the database (*TrainFeaturesONNS*).

In test phase, it will be change to from the segmented image of test/query input (*TestFeaturesONNS*).

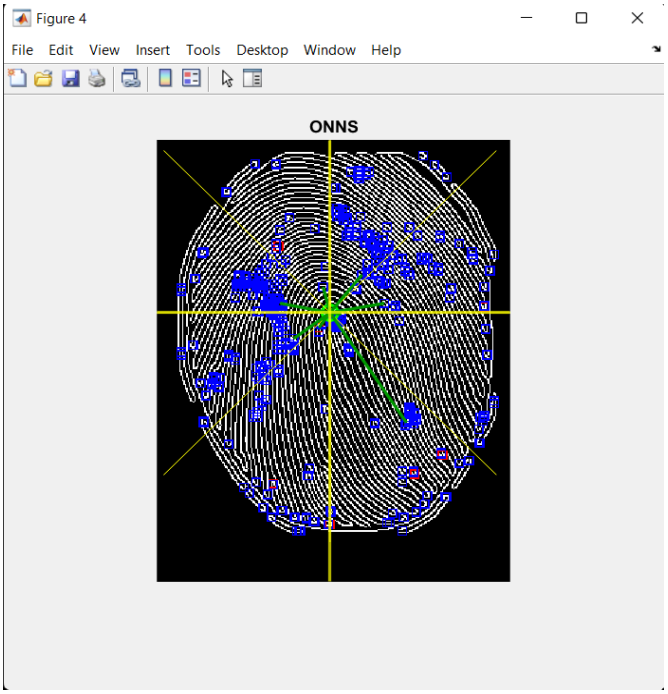

Figure 3.11: ONNS

3.4.8 Compute the similarity

Next is to Compute the similarity between the *TestFeaturesONNS* and *TrainFeaturesONNS* using the below equation:

$$\text{Distance } (p_k, q_k) = \sum_{k=1}^{n} (p_k , q_k)$$

By computing this, we get the classification of real or fake fingerprint. The overall operation of the above steps will be display during the test phase is shown below Figure 3.12.

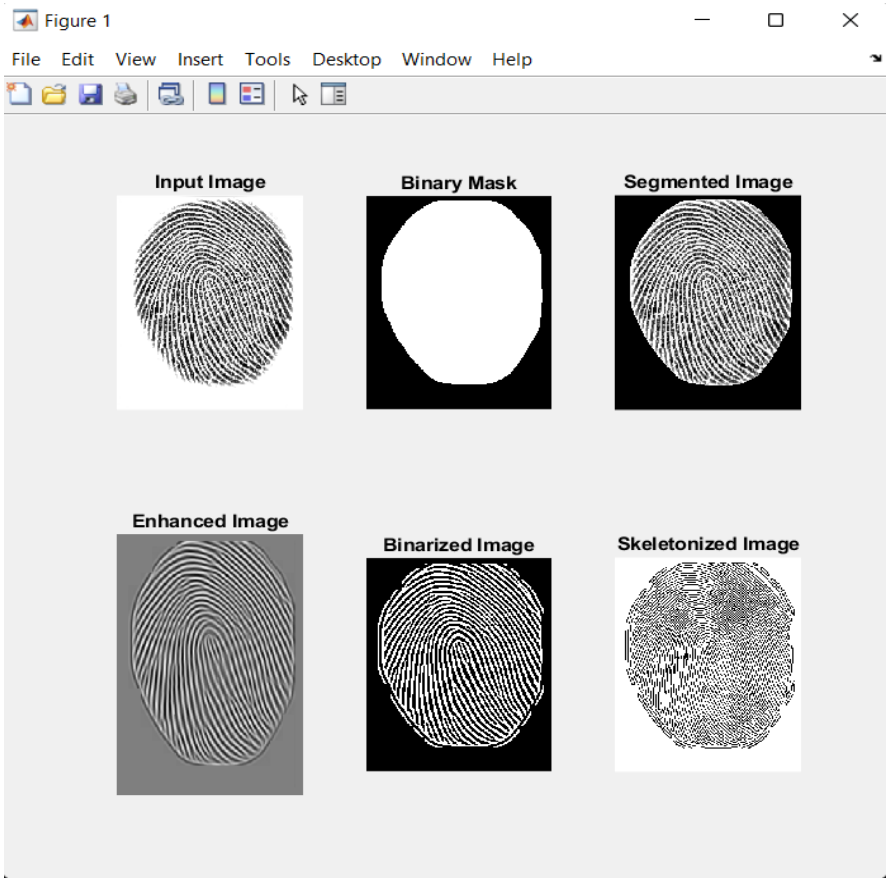

Figure 3.12: The overall output of image enhancement during test phase

The comparison of CNN can also see in window as shown below:

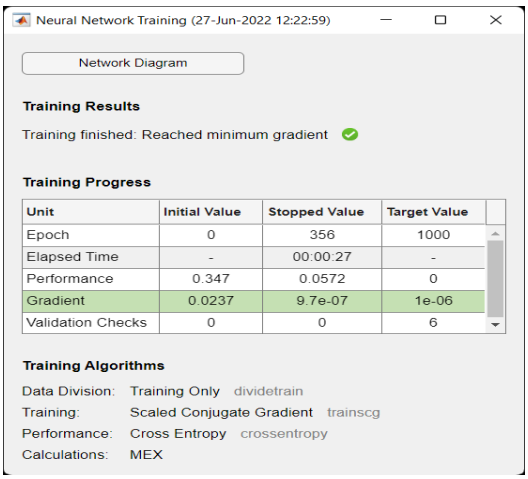

Figure 3.13: CNN cycle analysis during the test phase

3.5 Attendance marking module

We introduce a hardware module for examine the working of the code and used as a reference modal which can embedded to any hardware device or in any biometric authentication sector.

If the above phase shows a true approach, then the attendance is mark and recorded in an excel sheet. Else it pops up an error message.

Sometimes there might students get into the class lately, then it shows time out message. The below Figure shows the attendance marking in correct time entrance and in time out. The excel sheet that marks the attendance. In our project we develop our system in a general aspect so that we have to make some manual input during some operations. Here we need to update the date of excel sheet which use as the name of the sheet.

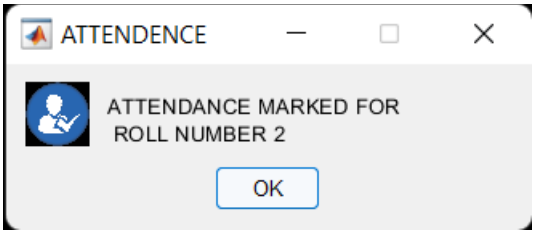

Figure 3.14: Attendance marked

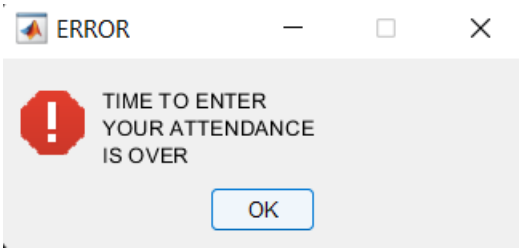

Figure 3.15: Time out section

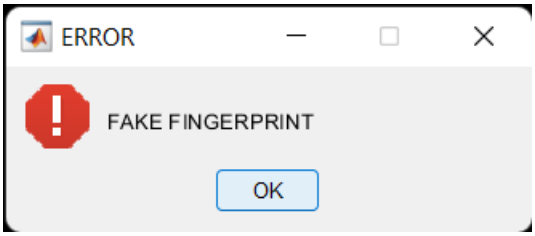

Figure 3.16: Fake fingerprint

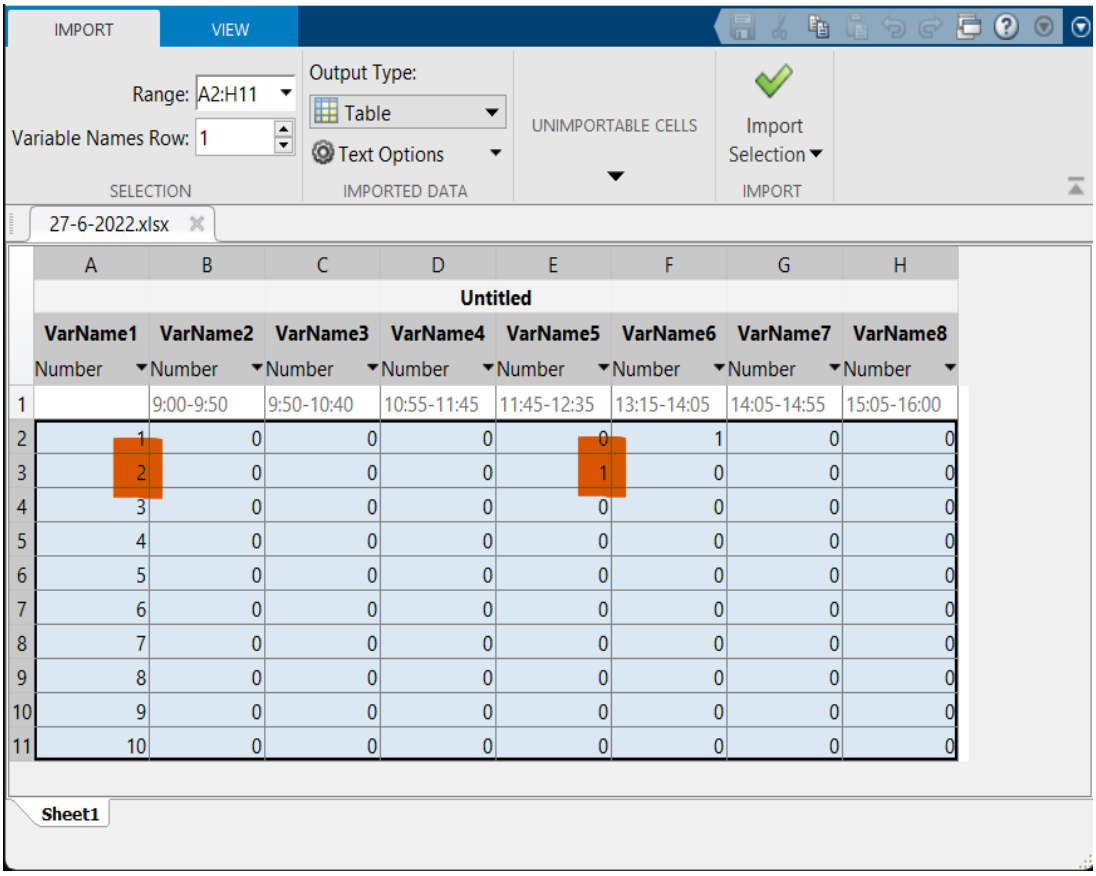

FIGURE 3.17: Marking attendance corresponding to the student

Thus, the software working of our system shows a detailed behind the interface. During the real time operation these functionalities are less overhead for the users and students cannot access easily to the operations. For future, we can add up the staff details, timetable management, student's details and also an application or a web handle for operate users manually. A large institution can afford to make use of this system and the traditional form of attendance methods can replace.

3.6 Hardware Design

Hardware components used are Arduino UNO R3,a low cost, flexible and easy to use programmable open source microcontroller board that can be integrated into a verity of electronic projects.it can control relay, LED, servos and motors as an output. Finger print sensor used here is an optical sensor, perfectly functions, small in size and ultra-low power consumption and adjustable security level. There is a RTC module has a backup battery installed. This allow the module to retain the time even when it's not being powered up by the

Arduino. This way, every time you turn on and off your module, time doesn't reset. LCD Display is a basic module used here it can display 16 characters per line and there are 2 such lines. LCD has two registers – command and data. Contrast of the display can be adjusted by adjusting the potentiometer to be connected across VEE pin. Potentiometer is used here, is a manually adjustable variable resistor with 3 terminals. The position of the wiper determines the output voltage of the potentiometer.

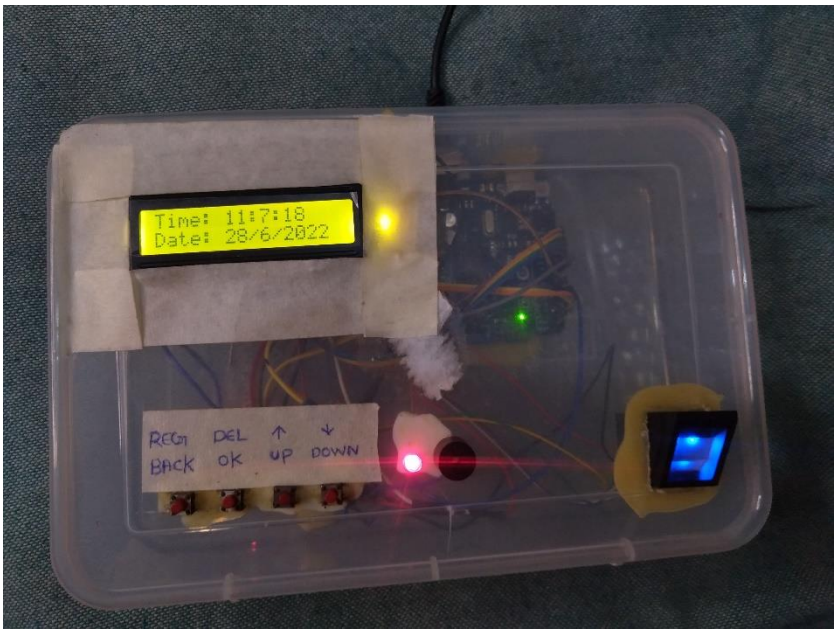

Figure 3.18: Hardware model

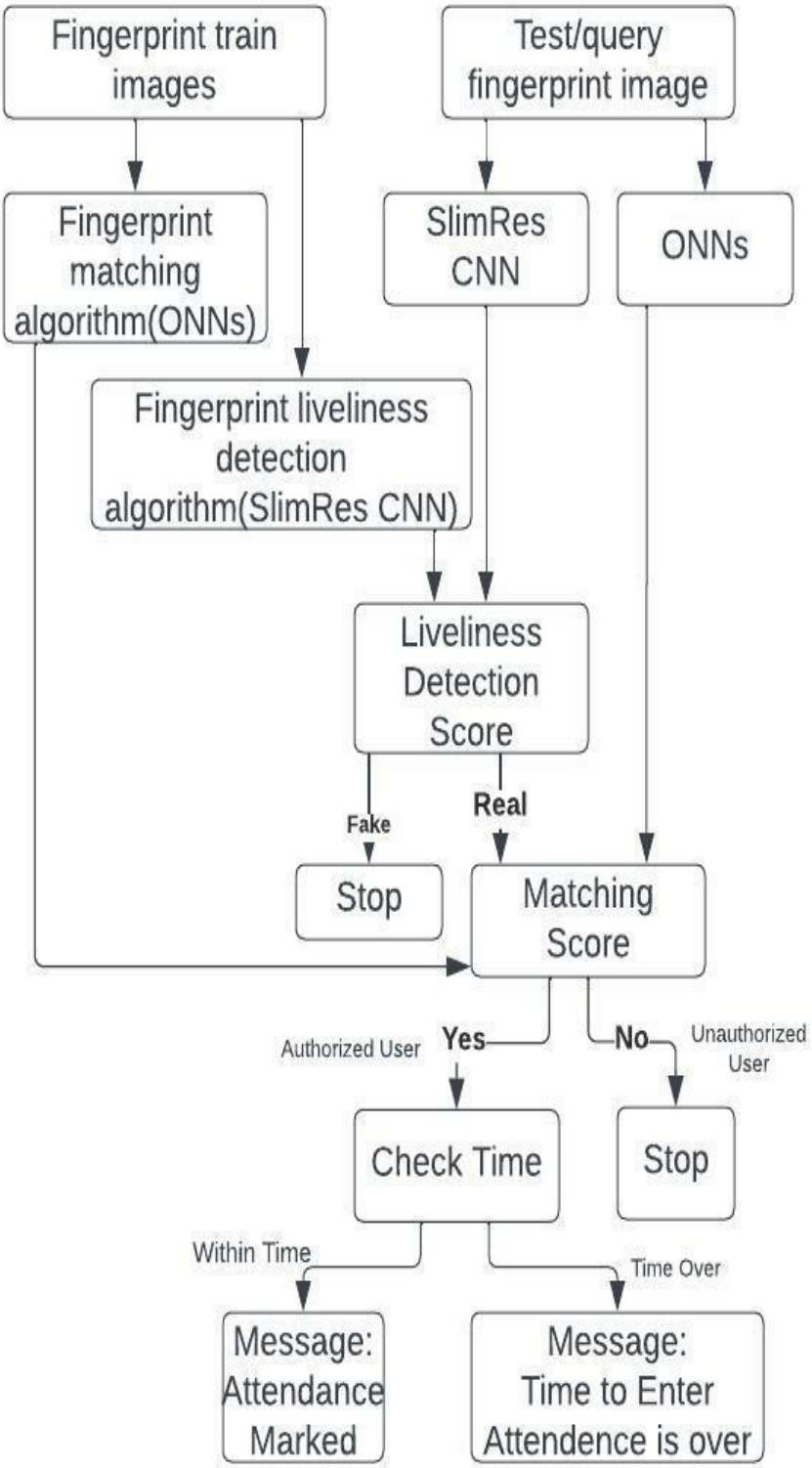

Figure 3.18: System Design

CHAPTER 4

RESULTS

In this section, we summarize our test results. We used precision, recall and F1-scores as performance measures which are derived from the values in the confusion matrix. Confusion matrix is defined in Table 4.1:

		Prediction	
		y´=0	y´=1
	y=0	True Negative	False Positive
	y=1	False Negative	True Positive

Table 4.1: Confusion Matrix

Our performance measures are given below in Eq. 4.1- 4.3 below.

$$Precision = \frac{\sum True\ Positive}{\sum True\ Positive + \sum False\ Positive} \quad \text{(Eq. 4.1)}$$

$$Recall = \frac{\sum True\ Positive}{\sum True\ Positive + \sum False\ Negative} \quad \text{(Eq. 4.2)}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \text{(Eq. 4.3)}$$

Here the two feature extraction methods used are Octantal Neatest Neighbourhood structure and modified convolutional neural network and classification methods are deep learning classifier (modified CNN).

The system achieves highest scores for precision, recall, and F1- score for real fingerprints. That is Precision=0.9091, Recall=0.9000, and F1-Score=0.9499.

Data	Precision	Recall	F1-Score
True Fingerprint	0.9091	0.9000	0.9499
Fake Fingerprint	0.432	0.563	0.491

Table 4.2: Result

Chapter 5

CONCLUSION

The fingerprint recognition system's security is aided by FLD. The study provides a score-level fusion method that combines the score of fingerprint matching with the score of fingerprint liveliness detection to produce a final integrated score for determining the identity of a person. Whether the fingerprint on the probe is a real living fingerprint despite the fact that the test set contained phoney fingerprints, the experimental results reveal that the final integrated set was constructed using materials that were not used in the training set. The system we proposed delivered impressive results.

By using a Fingerprint Sensor module to validate a true person or employee by capturing their finger input in the system in this Fingerprint Sensor Based Biometric Attendance System using Arduino. We're utilising four push buttons to register a new fingerprint, delete an existing one, and match an existing one.

REFERENCE

1) A Score-Level Fusion of Fingerprint Matching With Fingerprint Liveness Detection yongliang zhang ,chenhao gao, shengyi pan , zhiwei li , yuanyang xu , and haoze qiu Digital Object Identifier 10.1109/ACCESS.2020.3027846
2) J. Jia and L. Cai, ‘‘Fake finger detection based on time-series fingerprint image analysis,’’ in Proc. Int. Conf. Intell. Comput., 2007, pp.
3) Y. Zhang, J. Tian, X. Chen, X. Yang, and P. Shi, ‘‘Fake finger detection based on thin-plate spline distortion model,’’ in Int. Conf. Biometrics, 2007.
4) Y. S. Moon, J. S. Chen, K. C. Chan, K. So, and K. C. Woo, ‘‘Wavelet based fingerprint liveness detection,’’ Electron. Lett., vol. 41, no. 20, pp. 1112–1113, Sep. 2005.
5) A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni, ‘‘Fake finger detection by skin distortion analysis,’’ IEEE Trans. Inf. Forensics Security, vol. 1, no. 3, pp. 360–373, Sep. 2006.
6) G. L. Marcialis, F. Roli, and A. Tidu, ‘‘Analysis of fingerprint pores for vitality detection,’’ in Proc. 20th Int. Conf. Pattern Recognit., Aug. 2010.
7) H. Choi, ‘‘Fake-fingerprint detection using multiple static features,’’ Opt. Eng., vol. 48, no. 4, Apr. 2009.
8) J. Galbally, S. Marcel, and J. Fierrez, ‘‘Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition,’’ IEEE Trans. Image Process., vol. 23, no. 2, Feb. 2014.
9) I. Chingovska, A. Anjos, and S. Marcel, ‘‘Anti-spoofing in action: Joint operation with a verification system,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2013.
10) E. Park, X. Cui, T. H. B. Nguyen, and H. Kim, ‘‘Presentation attack detection using a tiny fully convolutional network,’’ IEEE Trans. Inf. Forensics Security, vol. 14, no. 11, Nov. 2019.
11) Y. Zhang, D. Shi, X. Zhan, D. Cao, K. Zhu, and Z. Li, ‘‘Slim-ResCNN: A deep residual convolutional neural network for fingerprint liveness detection,’’ IEEE Access, vol. 7, 2019.