SciEn
Journal of Engineering Advancements Vol. 03(04) 2022, pp 131-143 https://doi.org/10.38032/jea.2022.04.001

A GUI-Based Peg-Free Hand Geometry Recognition for Biometric Access Control using Artificial Neural Network

Kazeem B. Adedeji [a,*] *and Oluwatimilehi A. Esan* [b]

[a]Department of Electrical and Electronics Engineering, The Federal University of Technology, Akure, Ondo State, Nigeria
[b]Department of Computer Engineering, The Federal University of Technology, Akure, Ondo State, Nigeria

Received: August 12, 2022, Revised: October 27, 2022, Accepted: November 09, 2022, Available Online: December 01, 2022

ABSTRACT

Hand geometry has been a widely used biometric authentication because it is generally believed that the human hand has sufficient anatomical features which could be used for personal identification. Many hand geometry systems use pegs, which guide hand placement on the scanner. The system prompts the user to position the hand on the scanner several times and only captures when the current position is satisfied. In such a system, measurements are not very precise and this reduces accuracy during feature extraction. The system also has a higher false acceptance rate. This paper presents a peg-free hand geometry recognition system that does not depend on the orientation of the hand. Several features from test hand images are extracted and stored in the database, which are used to train an artificial neural network (ANN). To facilitate easy usage of the hand geometry verification system (peg-free), a GUI was developed using MATLAB software. The developed system was validated and the overall result shows that the system can be used for biometric verification using hand geometry where the orientation and placement of the hand are not a necessity. The results show that the developed system performed better with a relatively low false acceptance rate and false rejection rate of 0.01% and 0.02% respectively. The system also has a lower mean square error of 8.84×10^{-5}.

Keywords: Access Control, Artificial Neural Network, Biometric, Hand Geometry, Security.

1 Introduction

The fight against insurgence and combating cyber security has been the focus of nearly all countries nowadays. One way to combat security issues is to have the biometrics of every individual accessing a resource, which could be physical or cyber. Thus, personal identification for access control becomes increasingly vital in the quench of security. In the past and recent years, biometric-based identification systems have been used to provide an automated means of recognizing individuals based on their physiological and behavioural features [1]. A plethora of human anatomical features have been used for biometric systems. As presented in Figure 1, some of these include the use of fingerprints, face, iris, retina, voice, handwritten, face thermogram, hand geometry recognition, palm vein, and DNA matching [2]-[15]. In facial recognition systems (Fig. 1(a)), the position of some parts of the face is analysed to find a match [4]-[7]. Unfortunately, this system has some limitations. One of the limitations is that its major focus is the face (hairline down) and, consequently, the user may have to look straight at the camera before recognition could be possible. In the fingerprint recognition system (Fig. 1(b)), an optical sensor is utilized to capture line patterns on the finger, which are then analyzed. In the practical sense, some line patterns on the finger surface are so similar that it makes classification a difficult task. Thus, a high false acceptance rate is a common feature of this system. The use of iris recognition (Fig. 1(c)) for biometrics has also been reported [8], [9]. When an iris scan is performed, some features within the iris are captured, and they are then transformed into a bar code. This system has been an excellent security technique when used for biometric identification. However, a major problem with this system is the resistance from the users. The user needs to be in a good position for the scanner to read the iris. In most cases, it irritates the user. Another technology is the palm vein recognition system [11]–[13]. As shown in Fig. 1(d), it involves identifying some vein patterns on the palm. The speed at which the scanning is performed is relatively faster and more accurate. The system also provides some level of user convenience. However, it is costly. Therefore, the system is seldom used, although it could be used in applications where the security demand is vital. In hand geometry systems [16]-[19] (Fig. 1(e)), a fusion of the hand and palm features were used for personal identification.

Amongst the biometric recognition systems, the hand geometry system is frequently used due to the fact that the verification process is straightforward [20]–[26]. In this system, some features, which include shape, palm size, length, and width of the fingers, are extracted from the human hand. It has been reported that the hand features of humans are distinctive [1], [19] and after a certain age, a person's hand does not alter considerably. Unlike other biometric systems, during the dry season when some individuals experience dry skin, the accuracy of the verification process is not negatively affected [19]. Likewise, other biometric traits require exclusive scanners for data collection. However, hand images may be obtained using a low-cost webcam or digital camera. These, amongst other features, make the hand geometry-based system a widely accepted biometrics system for access control. Many hand geometry systems use pegs, which guide the hand's placement on the scanner. The system prompts the user to position their hand on the scanner several times and only takes an image of the hand when their current position is satisfied. In such a system, measurements are not very precise and this reduces accuracy during feature extraction.

*Corresponding Author Email Address: kezman0474@yahoo.com

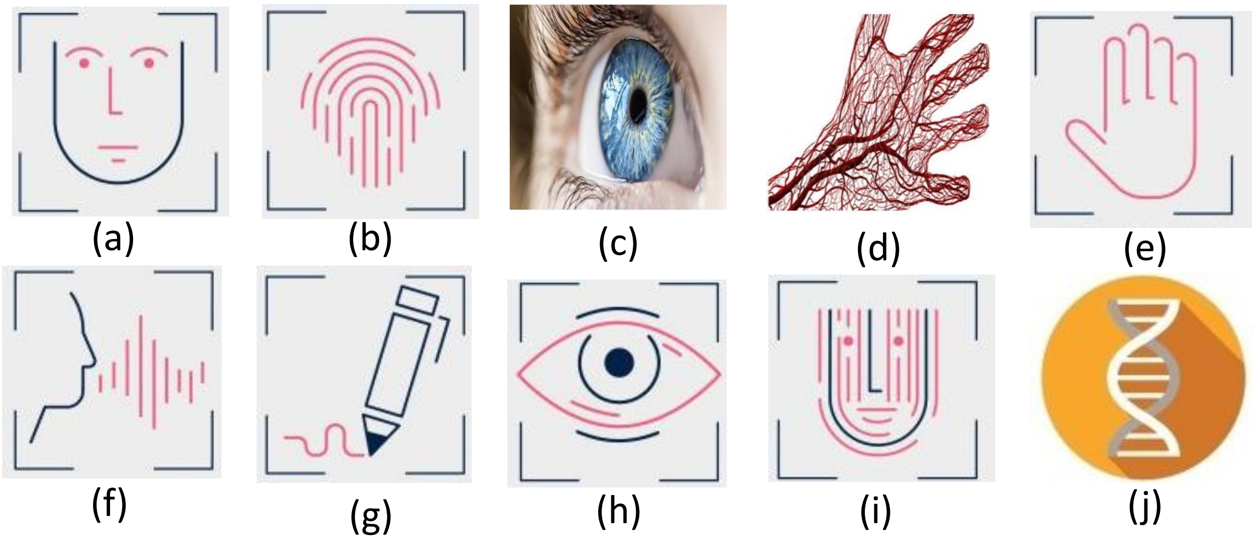


Fig. 1 Human traits used for biometric recognition (a) face recognition (b) fingerprint (c) iris recognition (d) palm vein recognition (e) hand geometry (f) voice recognition (g) hand written (h) retina recognition (i) face thermogram (j) DNA matching.

Thus, a system that is independent of the hand orientation is vital which is the focus of this study. Several research studies in this domain have been presented [19]-[35] with different levels of success in terms of false acceptance and rejection rate. This paper aims to improve the existing system and present a much deeper analysis of the biometric identification system. The paper is organized as follows: Section 2 presents a review of some related studies and the main contribution of this paper. Section 3 describes the system architecture and the ANN implementation. In section 4, the results of the implementation of the peg-free system are discussed, while section 5 concludes the paper and presents future work.

2 Review of Related Studies and Contributions

Several review studies on hand geometry recognition systems are available in the published literature with different levels of success. In [19], an innovative peg-free hand-geometry-based user identification system was proposed. The system uses spectral properties of a minimal edge-connected graph representation of the hand image. In this study, a multiclass support vector machine (MSVM) is employed to identify the claimed user. The applicability of the proposed system was demonstrated on two databases, namely the GPDS150 hand database and the VTU-BEC-DB multimodal database. The system achieves relatively better identification with a false rejection and false acceptance rate of 2.05% and 0.69%, respectively. Boreki and Zimmer [27] present a control access system based on hand geometry, a hardware key, and a vital sign detector. The features of the hand images are extracted using the analysis of the curvature profile of the image. This makes the proposed system invariant to the rotation and translation of the hand. The system achieves a relatively low false acceptance rate (0.8%) when demonstrated on a database having more than 360 hand images. Adan et al. [28] presented a hand biometric system for verification and recognition using a Natural Reference System (NRS). In this work, the hand's features are obtained through the polar representation of the hand's contour, which allows the study to achieve a minimum image processing and low computational cost. The system achieves a good false acceptance rate and false rejection rate of 0.45% and 3.04%, respectively when demonstrated on 5640 images that belong to 470 users. In Villegas et al. [29], wavelet features from the nearest neighbour were used to achieve biometric hand geometry identification. In this study, an input image of a hand was obtained using a scanner; the image was pre-processed and transformed to the wavelet domain. The stage of classification is performed using a simple nearest neighbour algorithm with Euclidean distances. The applicability of the system was verified using a database having 120 hand images. The system achieves a recognition rate of 70.2 with a false rejection rate of 10.4%.

Several other studies in this domain have been observed to use different classifiers for biometric identification using hand features. In [30], a hand geometry identification system was presented using a geometric classifier with a good false acceptance rate of less than 1%. In [31], a peg-free hand geometry measurement was utilized for human identification. A false rejection rate of 1% was obtained in this study. The authors [32] utilize neural networks for biometric hand recognition. The study uses 2800 hand images to train and verify the proposed system with a relatively low false rejection rate of 0.148%. In [33], a fusion of 2D and 3D hand geometries was used for biometric verification. The system uses a database consisting of 120 hand images and achieves a relatively high false rejection rate and the false acceptance rate of 10.4% and 11.4%, respectively. Polat and Yildirim [34] use a general regression neural network for hand geometry identification. A database consisting of 140 hand images was used to train and validate the system. The proposed system achieves a high false acceptance and false rejection rate of 15%. This could be due to the fact that the features of the hand images were not extracted before the identification stage; thus, a high false acceptance rate is expected. The work proposed by [35] presents a hand geometry user identification algorithm that utilizes data acquired by a standard office scanner. During verification, a false acceptance rate and a false rejection rate equal to 0.0% and 1.19% were obtained. However, during identification, a different result was reported.

At this stage, the results showed that the false acceptance rate was superb (0.0%) while the false rejection rate was 1.19%. Iula [36] utilizes a 3D ultrasound hand geometry for biometric recognition. In this study, several 2D images are first extracted at increasing under-skin depths, and from each of them, up to 26 distances among key points of the hand are defined and computed to achieve a 2D template. A 3D template is then created by combining two or more 2D templates in various ways. The performance of the proposed system was verified using a homemade database. The results presented revealed that good recognition accuracy was achieved. This study was improved in [37] with an identification rate of 100%.

Based on our examination of the existing literature studies in this domain, it is observed that a different level of success has been achieved. Aside from the study in [35] that achieved a superb performance in terms of the false rejection rate (though during only the identification stage and not the entire system), we observe that improving the false acceptance and rejection rate for the entire system is vital. This paper aims to improve the existing system, presents a much deeper analysis of the biometric identification system, and motivates the research scholar to become personally involved in an effort to improve biometric recognition through hand geometry identification.

.

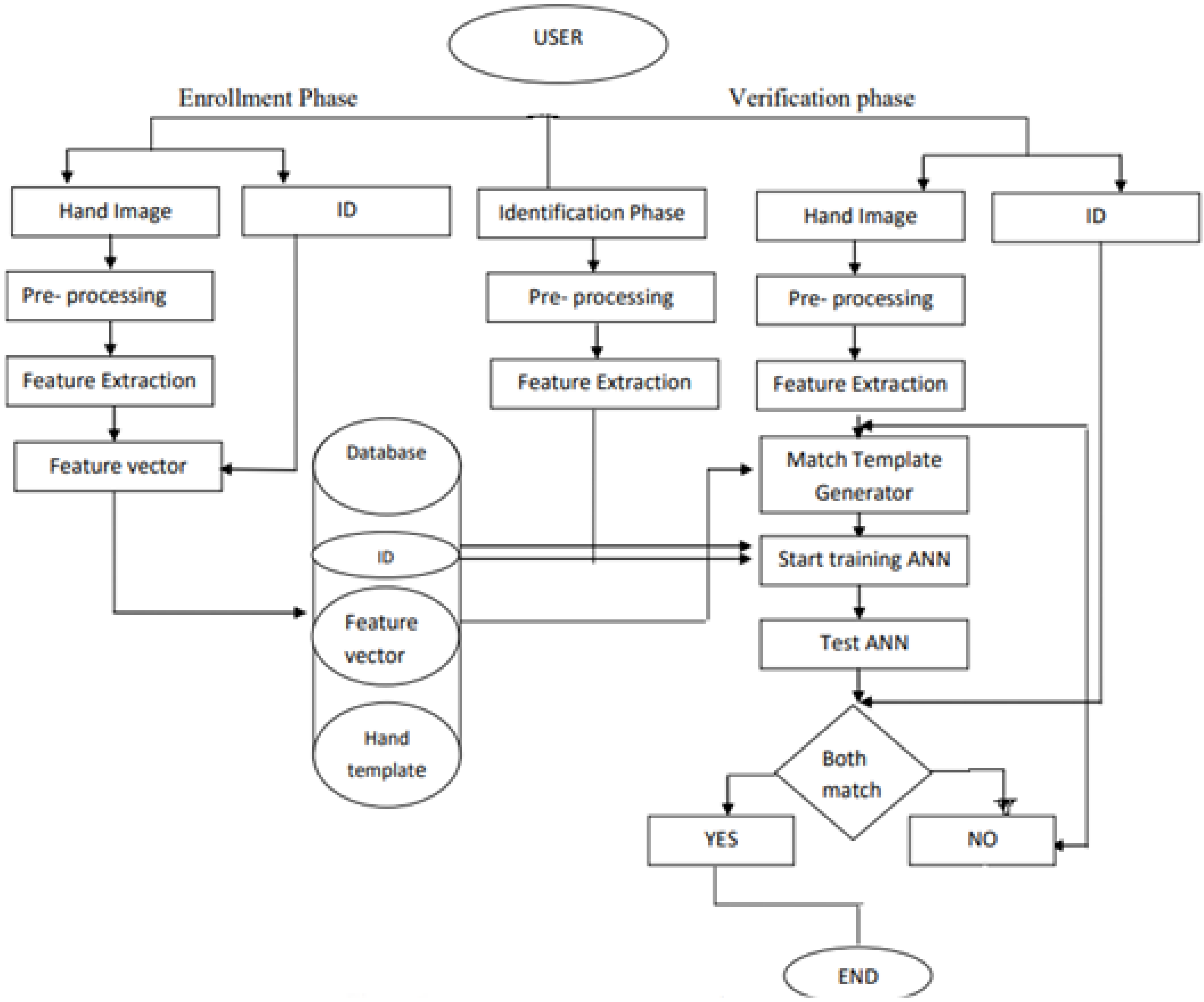


Fig. 2 The architecture of the developed hand geometry system.

3 System Architecture

The architecture of the developed hand geometry is shown in Fig. 2. The system was implemented to control the access of undergraduate students to a lecture hall in a tertiary institution. It consists of enrolment, identification, and verification. In the enrolment stage, data is collected using a CCD camera and then delivered to a feature extractor, which converts the data into numerical features. The feature extraction is done through Harris corner detection, which detects corner points in the image. It first finds the difference in intensities for the displacement of features over a window size. The output of the feature extractor was added to the database as a saved template marked with a user identity.

At the identification stage, the images are marked with an ID, which is classified into classes and also to a formulated timetable as shown in Table 1.

Table 1: Rule-based access control for computer engineering undergraduate students in a tertiary institution

ID	Rule Basis	Policy Rules for Access	Result/Output
1-50	300 level Students	Monday’s:8am-12pm Tuesday’s:12pm-4pm Wednesday’s:8am-12pm	Grant access if ID 1-50 match with the trained features
51-100	400 level Students	Wednesday’s:12pm-4pm Thursday’s:8am-12pm Friday’s:8am-12pm	Grant access if ID 51-100 match with the trained features
101-150	500 level Students	Modays’s:12pm-4pm Tuesday’s:8am-12pm Friday’s:1pm-4pm	Grant access if ID 101-150 match with the trained features

A matcher gets the stored templates relating specifically to claimed usernames during the verification step. The similarity between an input feature and the recovered template is determined using a distance measure. The system approves if the distance is less than a predetermined threshold; otherwise, it refuses. The operations involved in each of these stages are discussed in the next section of this paper.

3.1 Enrolment Phase

3.1.1 Image Acquisition

Images are obtained from the College of Engineering Pune Palm Print database, which consists of the right-hand images acquired through the use of a digital camera, each with 5 pixels. A colour or grayscale image of the hand palm is used as the input. For image processing, the images are stored in the Joint Photographic Experts Group (JPEG) format in the database. Fig. 3 shows samples of the hand images.

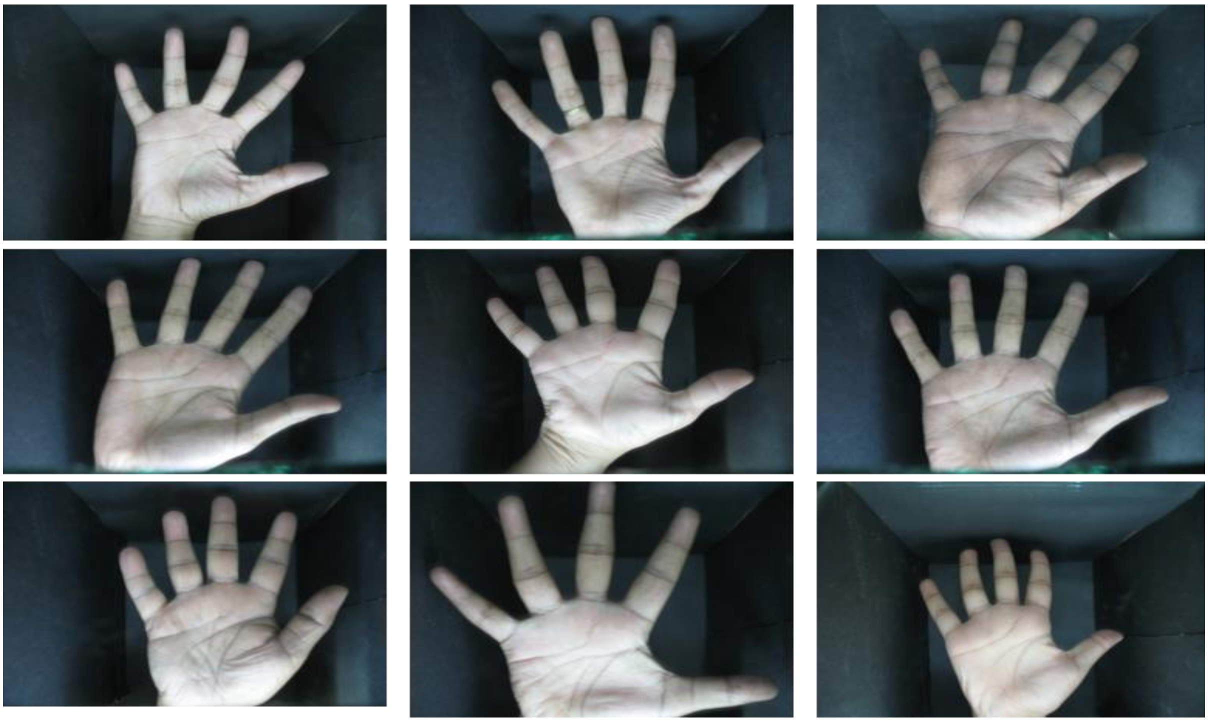


Fig. 3 Samples of the captured hand images.

3.1.2 Pre-processing

(A) Grayscale image conversion

It is observed that the original image is coloured, and must be converted to grey-scale images using the r2bgray function. This function cancels the huge saturation information in the image while keeping the luminance information. Each pixel in a grayscale image has a single sample value. After converting the image to grayscale, a threshold is determined and the image is transformed into a binary image. In a binary image, all pixels in the input image with intensity above 1 have a value of 1 (white), and 0 (black) otherwise. Thus, at a certain threshold T, the transformation of an input image A into a binary image B can be represented as

$$B_{ij} = \begin{cases} 1 for \quad A_{ij} > T \\ 0 otherwise \end{cases}, \qquad (1)$$

where i denotes images 1,2,3,..,149, j is the last image, B_{ij} =1, for the object or background pixels, and B_{ij} =0, for the background images. The process of image binarization divides pixel values into two groups: background pixels and foreground or object pixels. During the feature extraction process, image binarization aids in the extraction of minute details [17]. Thus, when input image (i, j) ≥ threshold, the output image (i, j) =1, however, if input image (i, j) < threshold, the output image (i, j) = 0.

(B) Noise removal

Noise in the image must be removed since it can cause a disparity between the actual palm and the captured image. Another reason for noise removal is that edge detection in noisy images is challenging due to the high-frequency content of both the noise and the edges. Each image has two major constituents, as shown in Eq. (2) [16].

$$L'(x,y) = L(x,y) + N(x,y) \qquad (2)$$

where L' denotes the noisy image, L denotes the actual image, N is the noise component, and x, y denote pixel coordinates. In MATLAB, Wiener2filter is used to eliminate the noise. Therefore, measuring the noise level in an image becomes vital. This can be accomplished using the signal-to-noise ratio (SNR). The higher the SNR value, the better the image quality [16]. It is possible to evaluate the SNR in an image using Eq. (3).

$$SNR(L,L') = 10log\left(\sigma^2\,(L)/MSE\,(L,L')\right) \qquad (3)$$

where σ^2 is the image variance, MSE is the mean square error. Also,

$$MSE(L,L') = \frac{1}{k}\sum_{k} d^2\left(L(x,y)\middle|\quad\middle|, L'(x,y)\right)L' \qquad (4)$$

In Eq. (4), k is the pixels' number in the image and d is the distance between the signal- the only image and the noisy image.

(C) Edge detection

One of the most significant aspects of picture pre-processing is edge detection. Only the edges of the image are used to extract geometric elements of the hand. Various methods of detecting edges include the Sobel method, Prewitt method, Roberts method, and the Canny method. In this study, Sobel edge detection was used for edge detection. Each pixel position in the image is used to compute the image's gradient.

3.2 Feature Extraction

3.2.1 Features of the Hand

Each individual hand has several features that can be used for recognition. As shown in Fig. 4, some of these features include finger length (FL), finger width (FW), finger sub-length (FSL), palm width (PW), palm length (PL), thumb length (TL), and thumb width (TW). Hand length and its contour (HL and HCL), and hand area (HA), are other features of the entire hand considered. The HL, FL, FW, and PW are frequently used traits for hand recognition and are regarded to be invariant with time and age [1]. The distance between finger lines in a finger is known as the finger sub-length (see Fig. 4).

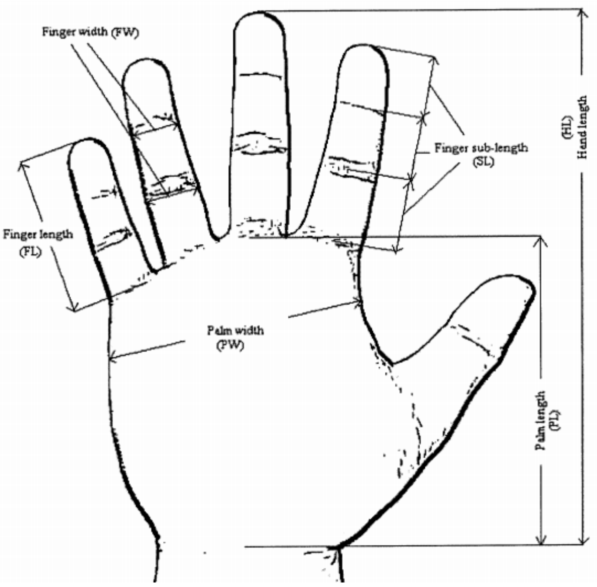


Fig. 4 Hand geometry showing some features of the hand.

3.2.2 Detection of Landmark Points

The following steps was carried out to identify the landmark points in the hand images:

1. Detection of fingertip and gap between finger tips: The point at the top of the fingertip is regarded as the fingertip point, while the lowest point of the gap between two fingers is known as the gap between finger points. The black and white image line was scanned from top to bottom. When the scan line crosses the four fingers (often five fingers) as depicted in Fig. 5(a), the scanning process stops. Then, the four cutting spots in that line are determined as shown in Fig. 5(b). By sliding the beginning points up or down along the edge, the lower gap points and the four fingertip points were discovered (see Fig. 5(c)).
2. Detection of the points of "side-points" and "palm-width-reference": The three lower gap points from Fig. 5(c), was used to estimate side points. This is done by drawing a straight line (see Fig. 5(d)) connecting the left two gap locations for the left "side-point". The intercept point between the line and the left edge can thus be designated as the "left-side" point while the intercept locations between the line and the left edge is designated as the "right-side" point. The two "palm-width-reference" points were obtained by dragging the two "side-points" down a little over the edge (around 20 pixels).
3. Detection of the four "finger-end" points: The four "finger-end" points can be calculated using the three "gap-between finger" and the two "side-point" by taking the mid-point of each pair of the gap and side points as shown in Fig. 5(e)).
4. Detection of the wrist point: As shown in Fig. 5(f), the wrist is located by scanning the image from bottom to top, taking note of the white pixel width for each row. When the inside white pixel width changes rapidly (by more than 20 pixels), the scanning process stops, and the wrist is located. The wrist point is located in the middle of the white pixel line where quick changes occur (Fig. 5 (g)).
5. Detection of two "finger-line" points of each of the four fingers: The lines on the finger are referred to as "finger-lines." Because all finger-lines are horizontally oriented, Fig. 5(h) is a complete line image, while Fig. 5 (i) only shows the horizontal lines. All continuous lines no longer than 10 pixels long were searched. The black and white image is then used as a mask to remove all lines that are not within the four fingers. A mid-point is calculated for each of the remaining lines. The finger line point was obtained by averaging localized mid-points, as shown in Fig. 5(j), and then Fig. 5(k) was obtained.
6. Detection of "finger-width" reference points: For each finger, there are four "finger-width" reference points. They are placed at 1/3 and 2/3 of the finger length, which is defined as the distance between the "tip" and "end" points of the finger. The two right "finger-width" reference points were identified for each finger by using the "gap-between-finger" or "finger-tip" points (Fig. 6(a)). The two left "finger-width" reference points were then calculated by locating places on the left side with the smallest distance to the two right "finger-width" reference points as shown in Fig. 6(b).

After these operations, all the major control points on the hand features were obtained as illustrated in Fig. 6(c).

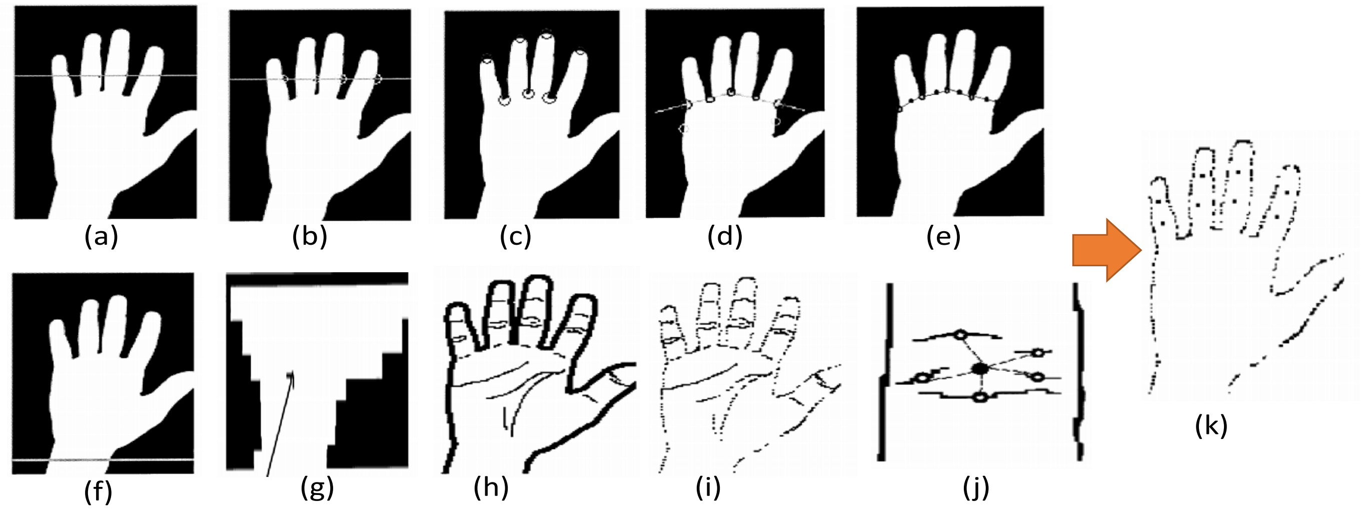


Fig. 5 Detection of landmark points (a) scan line cut the 4 fingers (b) intercepting points (c) detected fingertip and gap points (d) two side palm width reference points are predicted by using 3 gap points (e) two side palm width reference points are predicted by using 3 gap points (f) scan from bottom (g) rapid change detected (h) original detected line (i) horizontal line image (j) average of all localized points (k) all "finger-line" points.

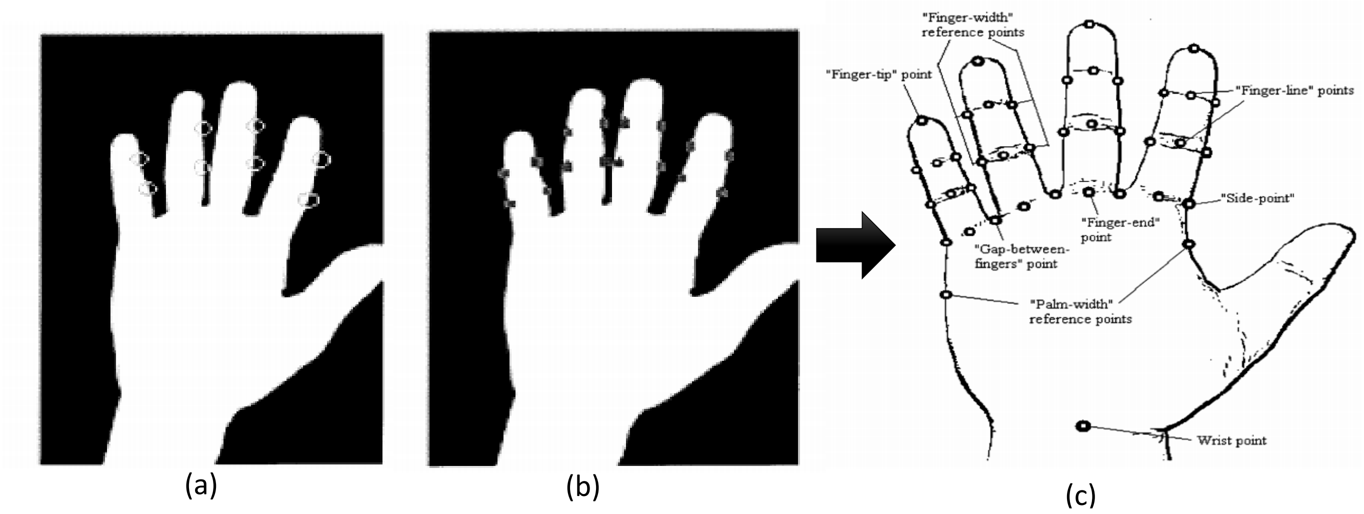


Fig. 6 Detection of "finger-width" reference points (a) right side finger width reference (b) all finger width reference (c) every control point discovered.

3.3 ANN Training and Testing

The training, validation, and testing of the data are achieved through the ANN. Fig. 7 shows the flow chart of the ANN training and testing. A back propagation neural network (BPNN) is utilized at this stage to identify features that have been removed and included in a database. The BPNN employed in this method has three layers: one input layer with 50 nodes representing 50 features, one hidden layer with 10 nodes representing one output; and one output layer with 10 nodes representing one output, as shown in Fig. 8 displays the ANN topology used for network modelling and simulation using the parameters listed in Table 2. The topology displays an overview of the neural network stimulation, the algorithm employed, the progress, and many plot options for analyzing the network's results. The data produced from the hand feature extraction measurement makes up the training data set. In the neural fitting network, the inputs used were the distances calculated for feature extraction. 70% of the parameters were used for training, while 15% each were used for both validation and testing.

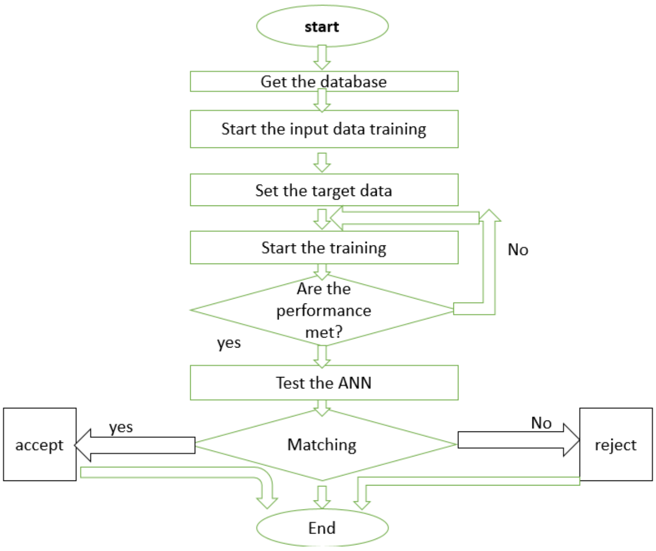
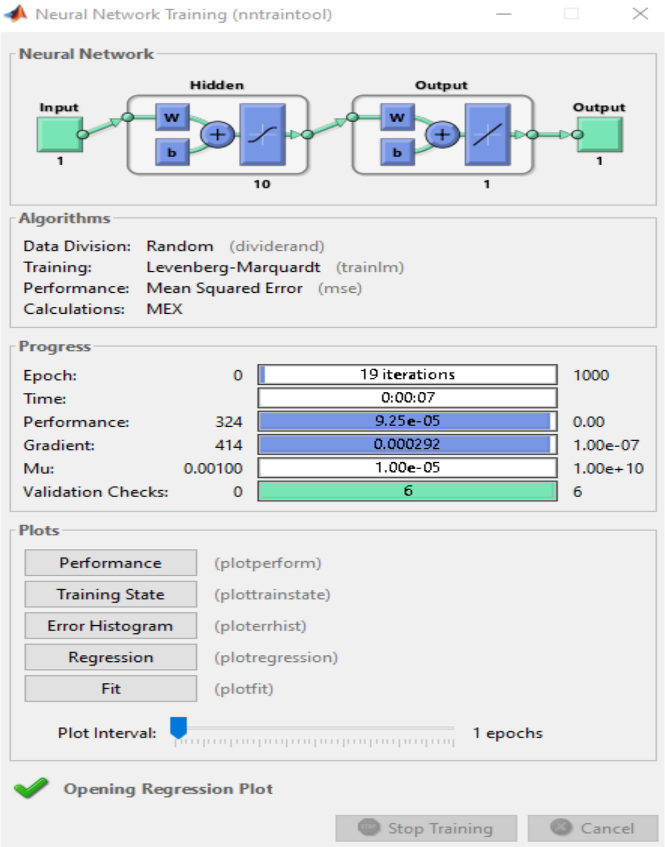


Fig. 7 Flow chat for the ANN training.



```
Algorithms
Data Division:  Random               (dividerand)
Training:       Levenberg-Marquardt  (trainlm)
Performance:    Mean Squared Error   (mse)
Calculations:   MEX

Progress
Epoch:                 0     19 iterations   1000
Time:                        0:00:07
Performance:         324     9.25e-05        0.00
Gradient:            414     0.000292        1.00e-07
Mu:              0.00100     1.00e-05        1.00e+10
Validation Checks:     0     6               6

Plots
Performance      (plotperform)
Training State   (plottrainstate)
Error Histogram  (ploterrhist)
Regression       (plotregression)
Fit              (plotfit)

Plot Interval:   1 epochs

Opening Regression Plot
```

Fig. 8 The neural network architecture.

A Levenberg-Marquardt (LM) backpropagation algorithm was used during ANN training with trainlm, which is a MATLAB function that updates weight and bias values according to LM optimization.

Table 2 Parameter used for training the dataset.

Parameters	Description
Training	LM (trainlm)
Network	Feedforward BPN
Learning Function	Learngdm
Hidden Layer	01
No of neurons	19
Max. epoch	1000

3.3.1 Back-propagation Training Stage

The following steps was used during the BPNN training stage;

1. Initialization stage: All network weights and threshold levels are set to random values that are distributed uniformly within a limited range$\frac{-25}{f_i}, +\frac{25}{f_i}$, where f_i is the total number of inputs of neuron i in the network.
2. Activation stage: Apply inputs to the BPNN to activate it. If $x_1(p)$, $x_2(p)$,…..,$x_n(p)$ denotes the input parameters and the desired outputs is represented by $y_{d.1}(p)$, $y_{d.2}(p)$,…..,$y_{d.n}(p)$;
   a. Actual outputs of the neurons in the hidden layer is estimated using

$$y_j(p) = sigmoid\left[\sum_{i=1}^{n} x_i(p) \times w_{ij}(p) - \theta_j\right] \qquad (5)$$

where n is the number of inputs of neurons j in the hidden layer, sigmoid is the activation function.

   b. The neurons' real outputs in the hidden layer is obtained using;

$$y_k(p) = sigmoid\left[\sum_{j=1}^{n} x_{jk}(p) \times w_{jk}(p) - \theta_k\right] \qquad (6)$$

3. Weight training stage: Weight in the BPNN is updated. For the output layer;
   a. The error gradient for the neurons in this layer is obtained using

$$\partial_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p) \qquad (7)$$

where $e_k(p) = y_{d.k}(p) - y_k(p)$.

   b. Evaluate the weight corrections as

$$\Delta w_{jk}(p) = \alpha . y_j(p) . \partial_k(p), \qquad (8)$$

   c. Weights at the output is estimated using

$$\Delta w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p). \qquad (9)$$

For hidden layers;

   a. The error gradient for the neurons in this layer is obtained as;

$$\partial_j(p) = y_k(p) \times [1 - y_j(p)] \times \sum_{k=1}^{i} \partial_k(p) w_{jk}(p), \qquad (10)$$

   b. The weight correction and its update at the hidden neurons are estimated using the following equations

$$\Delta w_{ij}(p) = \alpha . x_i(p) . \partial_j(p) \text{ and} \qquad (11)$$

$$\Delta w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p). \qquad (12)$$

4. Iteration stage: Perform iterations and repeat the process until the error threshold is met.

3.4 Matching

At this stage, the features extracted in the preceding part are compared to the features of that individual already in the database. To determine if the claimant is the claimed person or not, distance functions are used. The absolute distance D_a function is employed to match the feature vector [13], [38]. This is estimated as

$$D_a = \sum_{i}^{d} |y_i - f_i| \qquad (13)$$

where $f_i = h(f_1, f_2 \ldots\ldots f_d)$ is a d-dimensional Eigen vector of a registered user in the database, and $y_i = h(y_1, y_2 \ldots\ldots y_d)$ is a d-dimensional Eigen vector an unknown or the hand that is being verified (claimant). As a result, the distance between the claimant and the register user is equal to the difference between the claimer feature vector y_i and the database feature vector f_i. The system compares the result to a predetermined threshold and classifies the hand being verified after computing the distance. The hand being verified is accepted if and only if the estimated distance is less than the threshold. However, if the distance is above the threshold, the hand image is rejected.

The development and analysis were done using MATLAB software. A graphical user interface (GUI) was also developed to facilitate easy usage of the peg-free hand geometry system. The system is validated and its performance is analyzed.

3.5 Biometric Performance Analysis

The system performance was carried out using the following terms;

3.5.1 False Rejection Rate (FFR)

The FFR occurs when a valid user is denied access because the system cannot match the current biometric data to that in the database templates. The FRR was estimated using the equation [18].

$$FRR = \frac{N_{FR}}{N_{att}} \tag{14}$$

where NFR is the number of false rejection and N_{att} denotes the total number of attempts.

3.5.2 False Acceptance Rate (FAR)

When the system detects that the biometric data of an impostor matches the template of a legitimate user, the impostor is accepted and granted access. FAR was calculated using

$$FAR = \frac{N_{FA}}{N_{att}} \tag{15}$$

where NFA is the number of false attempts.

3.5.3 Equal Error Rate (EER)

The EER is a measure of the point where FRR equals the FAR. It should be noted that FRR and FAR are not continuous functions, and in practice, a crossover point may not exist [38]. The ERR is a measure of a device's accuracy; the lower the ERR, the better the system. Also, the performance of the ANN used for training and classification is evaluated using the mean square error, error histogram, and regression analysis.

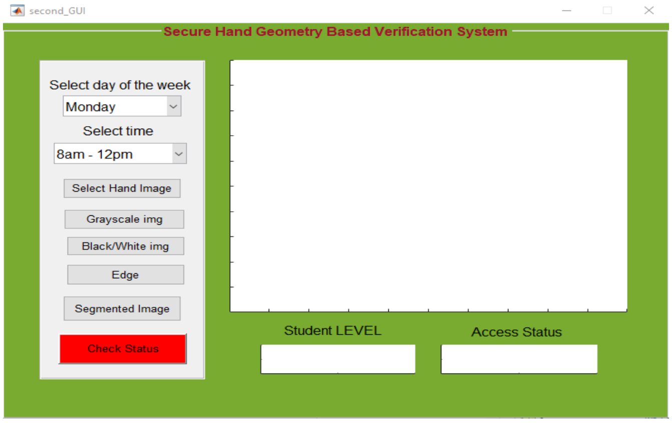
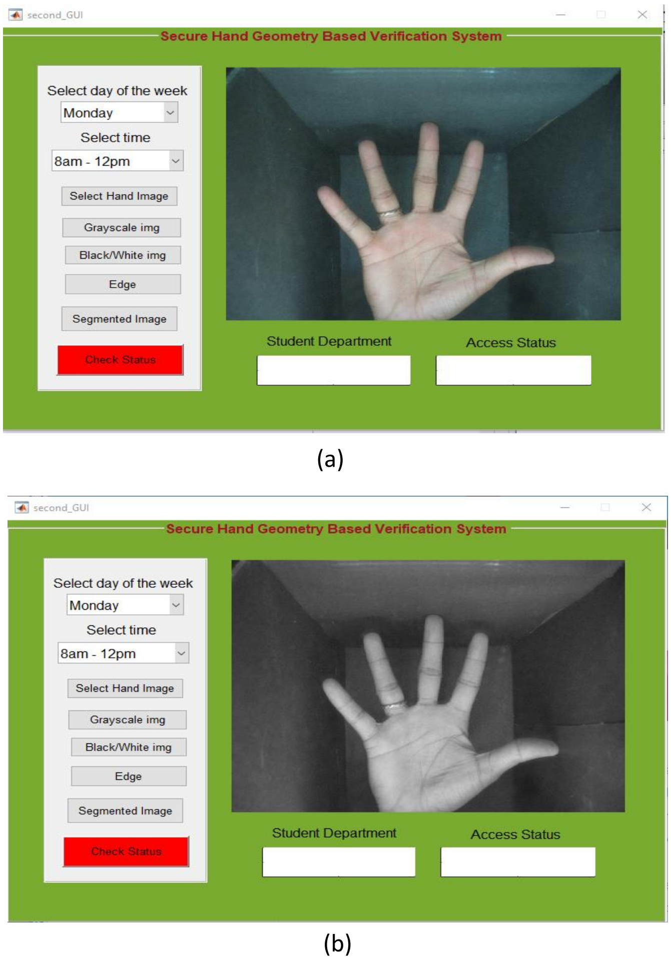


Fig. 9 The developed GUI showing several pre-processing features and access status.

4 Results and Discussions

4.1 The Developed Graphical User Interface and Implementation

As mentioned earlier, a GUI was developed to facilitate usage of the hand geometry system for access control. Also, the system was developed for use and implemented to permit or deny students' entry to a lecture hall in a tertiary institution. The developed GUI is shown in Fig. 9. As may be observed, the interface consists of two toggle buttons which are used to choose the day and time for lectures, respectively, five buttons (select hand image, grayscale image, black/white image, edge detection and segmentation) which may be used to show different pre-processing steps and a check status button which indicates the decision of the system whether to grant access or deny entry to the hall. The results produced while activating each of the buttons in Fig. 9 are illustrated in Fig. 10 to Fig. 12.



(a)

(b)

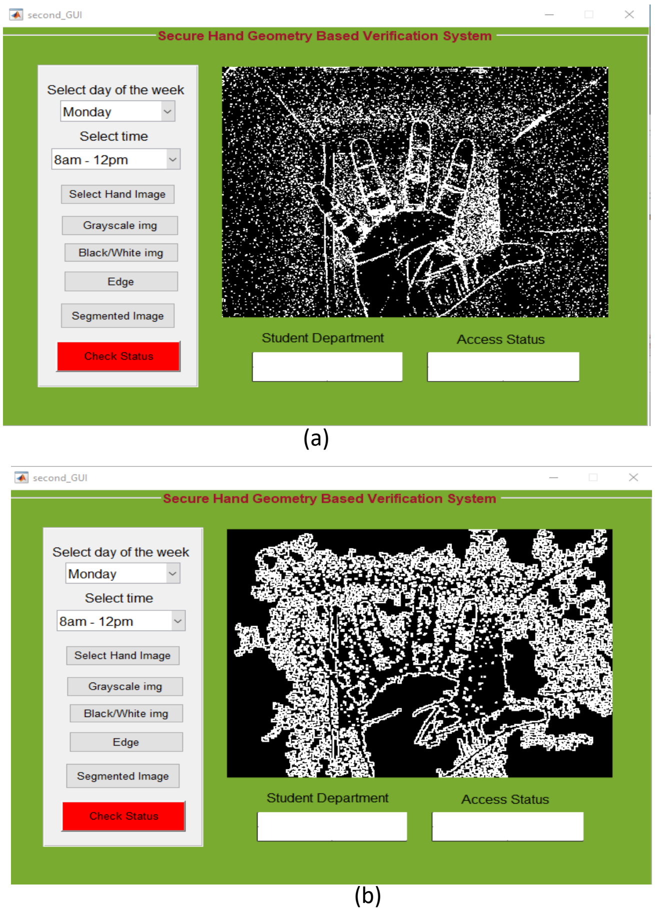
Fig. 10 The GUI showing (a) selecting the hand image (b) conversion to grayscale image.

In Fig. 10(b), the coloured image was converted to a grayscale image. The grayscale image is then converted to black and white, and the results are shown in Fig. 11(a). The next step is edge detection. In order to obtain the geometric features of the hand, the image should contain edges. This is done by using the Sobel edge detection function in the MATLAB image processing toolbox.

The result obtained after the edge detection process is shown in Fig. 11(b). The next step is the segmentation of the image, where the image is partitioned into various subgroups of pixels, which can reduce the complexity of the image and make the image easier to analyze. Fig. 12 shows the result of the image after segmentation has been performed.

4.2 Feature Validation Results

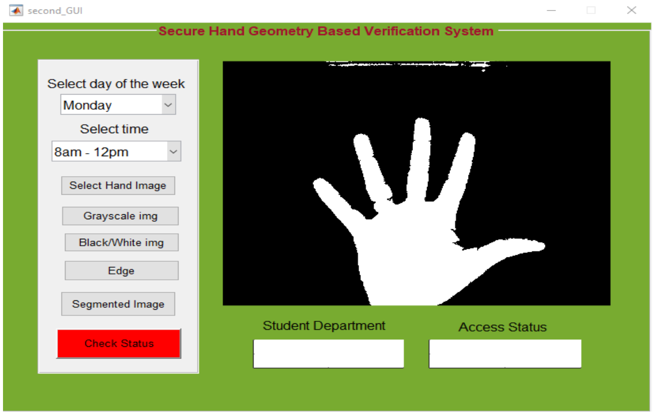
After training the dataset, the system checks and validates the hand image chosen according to the specification of the system. In this case, the system checks with respect to the class and also the timetable displayed in Table 1.



Select day of the week: Monday

Select time: 8am - 12pm

Fig. 11 The GUI showing (a) black and white conversion of the hand image (b) the image after edge detection.



Select day of the week: Monday

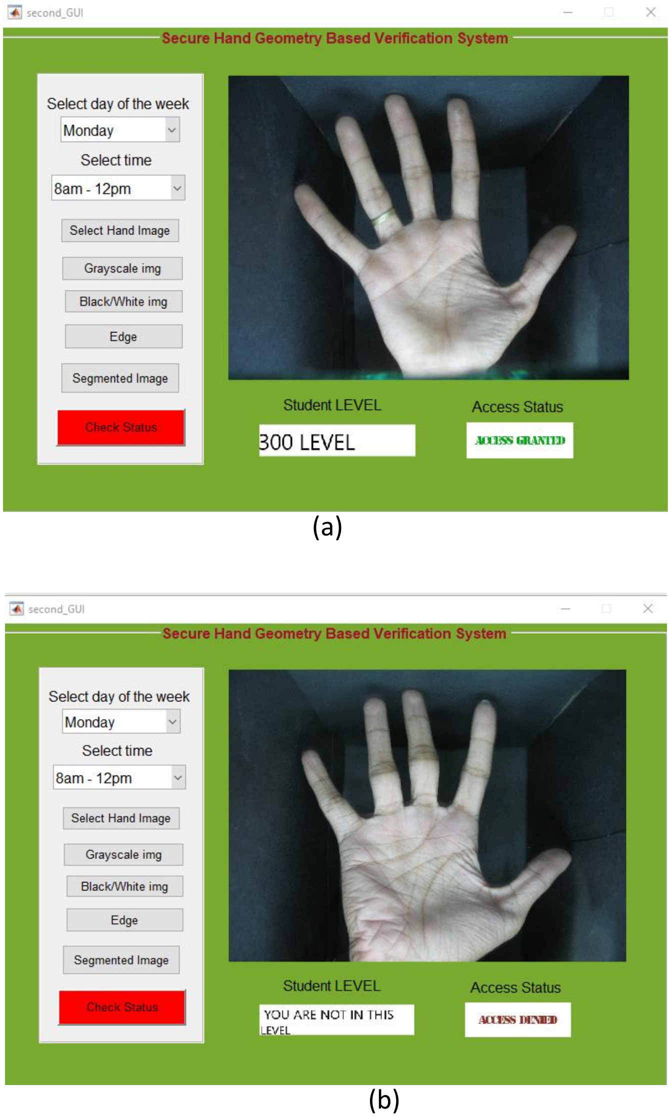
Select time: 8am - 12pm

Fig. 12 The GUI showing results of the hand image after segmentation.

When an image is chosen and the system wants to validate the image, it tells if the image is accepted or declined according to the class and timetable. For example, Fig. 13(a) depicts the result of an image validation to gain access to a 300-level class for the time period 8am-12pm, whereas Fig. 13(b) depicts the result of a declined image to gain access to the same class and time because the hand image validation to the specification failed.

4.3 ANN Performance Assessment Results

Fig. 14 shows the variation of the MSE with epoch during training, validation, and testing for the LM algorithm.



(a) Select day of the week: Monday; Select time: 8am - 12pm; Student LEVEL: 300 LEVEL; Access Status: ACCESS GRANTED

(b) Select day of the week: Monday; Select time: 8am - 12pm; Student LEVEL: YOU ARE NOT IN THIS LEVEL; Access Status: ACCESS DENIED

Fig. 13 GUI of the developed hand geometry system showing the access status and student level for (a) access granted (b) access denied.

As the number of epochs increases, the error starts to reduce. The error decreases sharply up to the last 6 epochs, after which it reduces gradually. If the epoch is increased beyond best performance, the validation graph tends to get more errors, which leads to overfitting of the data. During the training, the MSE is 1.25×10^{-4} at 10 iterations, with the best validation performance being 8.8383×10^{-5} at 13 epochs. The validation error was lowest at the 13th epoch, thus the training ended. It is important to state that the training is terminated after six consecutive upswings in validation error in the default configuration; thus, the best performance is selected from the epoch with the lowest validation error. After the training, the MSE becomes 9.25×10^{-5} at 19 epochs, with the best validation performance being 8.8383×10^{-5} at 13 epochs. Because the validation error was lowest at the 13th epoch, training was halted and weights and biases were used for future modelling.

The error histogram plot for the ANN is also presented as shown in Fig. 15. As may be observed, the amount of vertical bars on the graph is referred to as bins. Here, the overall error range is broken down into 20 smaller bins.

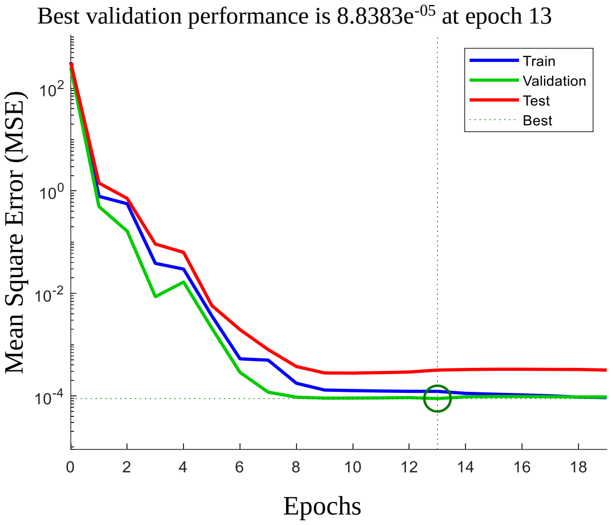


Fig. 14 Plot of the MSE against the Epochs.

The number of samples from the dataset that fall into each bin is represented on the y-axis, and the zero error line corresponds to the zero error value on the x-axis. The image dataset has an error range of around 12 samples when tested and an error range of 9.187×10^{-3} around 12 samples during validation, as shown in this graph.

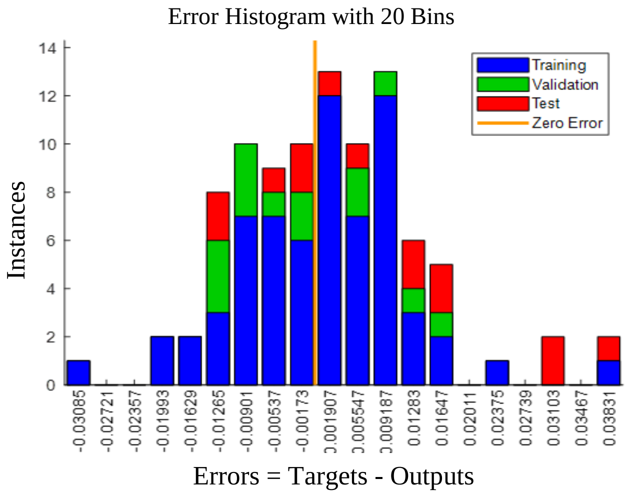


Fig. 15 The error histogram plot.

Fig. 16 shows the regression plot for the ANN. As may be observed, during training, the regression value amounts to about 0.9999. For validation, the regression value is 1, and 0.99998 for testing. When the three processes (training, testing, and validation) were considered, the regression value amount to 0.99999. This regression value indicates that the algorithm is 99.99% accurate in fitting the data.

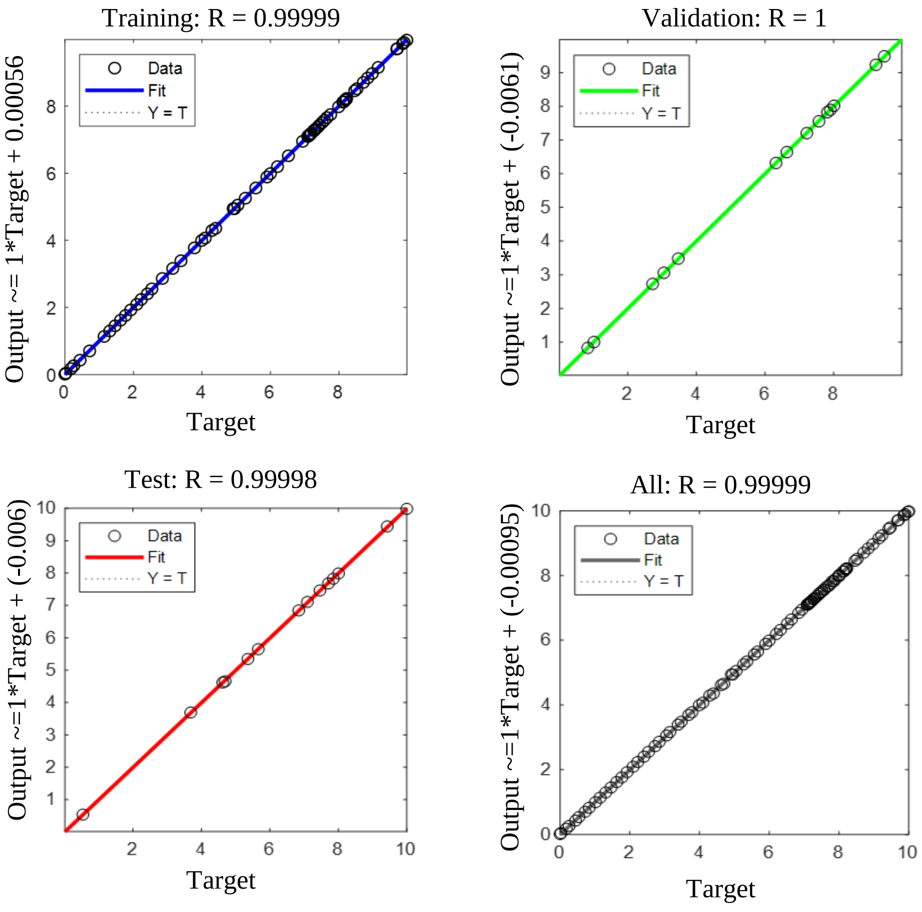


Fig. 16 Regression plot.

4.4 FAR and FRR Results

The FAR and FRR are estimated at different thresholds of the image and the results are illustrated in Fig. 17. It is observed that the FRR decreases drastically as the threshold increases up to a value of 2.5, after which it reduces slightly. Whereas the FAR increases slightly as the threshold increases.

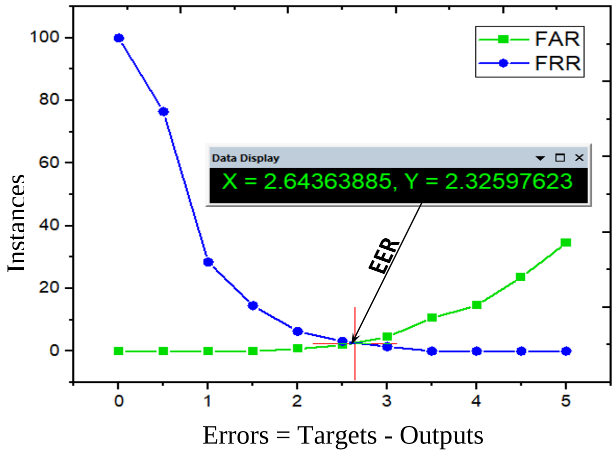


Fig. 17 Variation of the FAR and FRR with the threshold.

The system has an EER of 2.3. The arbitrary threshold proved to be good after testing with the images. There is a total of one false acceptance in every 100 tests, giving the arbitrary criterion a FAR of 0.01. In addition, 2 rejections were discovered in 100 tests, giving the FRR a value of 0.02. As may be observed in Fig. 17, it is pertinent to know that at lower threshold values, the system performs better with a relatively high FRR and a relatively low FAR.

4.5 Compared Performance Analysis

The performance of the hand geometry recognition system with related studies in the literature is presented in Table 3. The comparison was based on the size of the database, the FRR performance, FAR, EER, MSE, and regression. The results shown in Table 3 revealed that the hand geometry system presented in this paper performed better than most of the studies reported in Table 3. The FRR and FAR obtained in the proposed system are lower than those in Table 3. Aside from the studies in [35] that have a FAR of 0.0%, the proposed system has better performance than the other studies reported in Table 3. Also, the proposed system has a good regression percentage of 99.99% and a very low mean square error of 8.84×10^{-5}. Although these analyses (the regression and MSE) were investigated in other related studies presented in Table 3, In terms of the size of the database, aside from the authors in [32] that use a large database, most of the studies presented in Table 3 utilize a relatively small-to-medium database, which may not be enough to guarantee a superb performance. The database used for the current study is considered relatively large when compared to other studies in Table 3

. It is obvious from that the literature lacks a deeper analysis of the hand geometry recognition system, a gap that was filled in the current paper.

Table 3 Comparison with other similar studies

References	Technique used	Database size	FRR	FAR	MSE	Regression
[19]	Multiclass SVM	144	2.05%	0.69	na	na
[27]	Euclidean distance classifier	360	na	0.8%	na	na
[28]	Time average	470	3.4%	0.45%	na	na
[29]	Wavelet features (Nearest neighbour)	120	10.4%	11.4%	na	na
[31]	na	na	1%	0.48%	na	na
[32]	Neural net	2800	0.148%	0.357%	na	na
[30]	Geometric classifiers	714	3% (verification) 6% (identification)	<1%	na	na
[33]	Fusion of 3D and 2D hand geometries	120	10.4%	11.4%	na	na
[34]	Regression Neural Network	140	15%	15%	na	na
[35]	na	60	1.19%	0.0%	na	na
[36]	Fusion of 3D and 2D hand geometries	na	na	na	na	na
Current study	ANN	1344	0.02%	0.01%	8.84×10^{-5}	99.99%
na: Not available						

5 Conclusion and Future Work

In this paper, a peg-free hand-geometry recognition system is presented. To facilitate easy usage of the hand geometry recognition system (peg-free), a GUI was developed using MATLAB software. The developed system was validated and the overall result shows that the system can be used for biometric verification using hand geometry where the orientation and placement of the hand is not a necessity. Also, the use of an artificial neural network for the peg-free hand geometric recognition system yielded high-performance and high-accuracy results. The system has a very low mean square error and a regression value close to 1. The results obtained also show that the system has a relatively low false acceptance rate and a relatively high false rejection rate. Although the orientation and placement of the hand does not affect the performance of the system, it is noteworthy to mention that the best result can be obtained when the hand placement does not exceed 30 degrees. Therefore, further research studies can be conducted to get an efficient result when the placement of the hand is higher than 30 degrees.

References

[1] Dutagaci, H., Sankur, B. and Yörük, E., 2008. Comparative analysis of global hand appearance-based person recognition. *Journal of Electronic Imaging, 17*(1), p.011018.

[2] Yachongka, V., Yagi, H. and Oohama, Y., 2021. Biometric identification systems with noisy enrollment for gaussian sources and channels. *Entropy, 23*(8), p.1049.

[3] Wang, M., Hu, J. and Abbass, H.A., 2020. BrainPrint: EEG biometric identification based on analyzing brain connectivity graphs. *Pattern Recognition, 105*, p.107381.

[4] Kortli, Y., Jridi, M., Al Falou, A. and Atri, M., 2020. Face recognition systems: A survey. *Sensors, 20*(2), p.342.

[5] Sepas-Moghaddam, A., Pereira, F.M. and Correia, P.L., 2020. Face recognition: a novel multi-level taxonomy based survey. *IET Biometrics, 9*(2), pp.58-67.

[6] del Rio, J.S., Moctezuma, D., Conde, C., de Diego, I.M. and Cabello, E., 2016. Automated border control e-gates and facial recognition systems. *Computers & Security, 62*, pp.49-72.

[7] De Marsico, M., Nappi, M., Riccio, D. and Tortora, G., 2013. Entropy-based template analysis in face biometric identification systems. *Signal, Image and Video Processing, 7*(3), pp.493-505.

[8] Nguyen, K., Fookes, C., Jillela, R., Sridharan, S. and Ross, A., 2017. Long range iris recognition: A survey. *Pattern Recognition, 72*, pp.123-143.

[9] Salve, S.S. and Narote, S.P. 2016. Iris recognition using SVM and ANN. In Proceedings of the *IEEE International Conference on Wireless Communications, Signal Processing and Networking*, 23-25 March, Chennai, India. pp. 474-478.

[10] Hezil, H., Djemili, R. and Bourouba, H., 2018. Signature recognition using binary features and KNN. *International Journal of Biometrics, 10*(1), pp.1-15.

[11] Piciucco, E., Maiorana, E. and Campisi, P., 2018. Palm vein recognition using a high dynamic range approach. *IET Biometrics, 7*(5), pp.439-446.

[12] Abed, M.H., 2017. Wrist and Palm Vein pattern recognition using Gabor filter. *Journal of AL-Qadisiyah for Computer Science and Mathematics, 9*(1), pp.49-60.

[13] Funada, J.I., Ohta, N., Mizoguchi, M., Temma, T., Nakanishi, K., Murai, A., Sugiuchi, T., Wakabayashi, T. and Yamada, Y., 1998, August. Feature extraction method for palmprint considering elimination of creases. In Proceedings. *Fourteenth International Conference on Pattern Recognition* (Cat. No. 98EX170) (Vol. 2, pp. 1849-1854). IEEE.

[14] Sancen-Plaza, A., Contreras-Medina, L.M., Barranco-Gutiérrez, A.I., Villaseñor-Mora, C., Martínez-Nolasco, J.J. and Padilla-Medina, J.A., 2020. Facial recognition for drunk people using thermal imaging. *Mathematical Problems in Engineering, 2020*, Article ID 1024173.

[15] Abozaid, A., Haggag, A., Kasban, H. and Eltokhy, M., 2019. Multimodal biometric scheme for human authentication technique based on voice and face recognition fusion. *Multimedia Tools and Applications, 78*(12), pp.16345-16361.

[16] Yoruk, E., Konukoglu, E., Sankur, B. and Darbon, J., 2006. Shape-based hand recognition. *IEEE Transactions on Image Processing, 15*(7), pp.1803-1815.

[17] Mostayed, A., Kabir, M.E., Khan, S.Z. and Mazumder, M.M.G., 2009, December. Biometric authentication from low resolution hand images using radon transform. In 2009 *12th International Conference on Computers and Information Technology* (pp. 587-592). IEEE.

[18] Hussein, N.M.S., Hammed, S.M., and Ergen, B. 2017. Biometric identification system based on hand geometry. *International Journal of Innovative Research in Science, 6*(3), pp. 3159–3166.

[19] Angadi, S. and Hatture, S., 2018. Hand geometry based user identification using minimal edge connected hand image graph. *IET Computer Vision, 12*(5), pp.744-752.

[20] Taher, M.M. and George, L.E., 2022a. A digital signature system based on hand geometry-Survey. *Wasit Journal of Computer and Mathematic Science, 1*(1), pp. 1-14.

[21] Taher, M.M. and George, L.E., 2022. A digital signature system based on hand geometry. *Journal of Algebraic Statistics, 13*(3), pp.4538-4556.

[22] Mohammed, H.H., Baker, S.A. and Nori, A.S., 2021, February. Biometric identity Authentication System Using Hand Geometry Measurements. In *Journal of Physics: Conference Series* (Vol. 1804, No. 1, p. 012144). IOP Publishing.

[23] Dvořák, M., Drahanský, M. and Abdulla, W.H., 2021. On the fly biometric identification system using hand-geometry. *IET Biometrics, 10*(3), pp.315-325.

[24] Ghanbari, S., Ashtyani, Z.P. and Masouleh, M.T., 2022, May. User Identification Based on Hand Geometrical Biometrics Using Media-Pipe. In 2022 30th *International Conference on Electrical Engineering (ICEE)* (pp. 373-378). IEEE.

[25] Fang, L., Liang, N., Kang, W., Wang, Z. and Feng, D.D., 2020. Real-time hand posture recognition using hand geometric features and fisher vector. *Signal Processing: Image Communication, 82*, p.115729.

[26] Masood, D. and Qi, J., 2022. 3D Localization of Hand Acupoints Using Hand Geometry and Landmark Points Based on RGB-D CNN Fusion. *Annals of Biomedical Engineering*, pp.1-13.

[27] Boreki, G. and Zimmer, A., 2005, October. Hand geometry: a new approach for feature extraction. In *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)* (pp. 149-154). IEEE.

[28] Adán, M., Adán, A., Vázquez, A.S. and Torres, R., 2008. Biometric verification/identification based on hands natural layout. *Image and Vision Computing, 26*(4), pp.451-465.

[29] Villegas, O.O.V., Domínguez, H.D.J.O., Sánchez, V.G.C., Maynez, L.O. and Orozco, H.M., 2006. Biometric human identification of hand geometry features using discrete wavelet transform. *Discrete Wavelet Transforms–Biomedical Applications*, pp.251-266.

[30] Bulatov, Y., Jambawalikar, S., Kumar, P. and Sethia, S., 2004, July. Hand recognition using geometric classifiers. In *International Conference on Biometric Authentication* (pp. 753-759). Springer, Berlin, Heidelberg.

[31] Dhole, S.A. and Patil, V.H., 2012. Person identification using peg free hand geometry measurement. *International Journal of Engineering Science and Technology, 4*(6), pp.2943-2949.

[32] Martinez, F., Orrite, C. and Herrero, E., 2005, June. Biometric hand recognition using neural networks. In *International Work-Conference on Artificial Neural Networks (pp. 1164-1171)*. Springer, Berlin, Heidelberg.

[33] Kanhangad, V., Kumar, A. and Zhang, D., 2009, June. Combining 2D and 3D hand geometry features for biometric verification. In 2009 IEEE Computer Society *Conference on Computer Vision and Pattern Recognition Workshops* (pp. 39-44). IEEE.

[34] Polat, Ö. and Yıldırım, T., 2008. Hand geometry identification without feature extraction by general regression neural network. *Expert systems with* Applications, 34(2), pp.845-849.

[35] Klonowski, M., Plata, M. and Syga, P., 2018. User authorization based on hand geometry without special equipment. *Pattern Recognition, 73*, pp.189-201.

[36] Iula, A., 2021. Biometric recognition through 3D ultrasound hand geometry. *Ultrasonics, 111*, p.106326.

[37] Iula, A. and Micucci, M., 2022. Multimodal Biometric Recognition Based on 3D Ultrasound Palmprint-Hand Geometry Fusion. *IEEE Access, 10*, pp.7914-7925.

[38] Bača, M., Grd, P. and Fotak, T., 2012. Basic principles and trends in hand geometry and hand shape biometrics. *New Trends and Developments in Biometrics*, pp.77-99.

[39] Damousis, I.G. and Argyropoulos, S., 2012. Four machine learning algorithms for biometrics fusion: A comparative study. *Applied Computational Intelligence and Soft Computing*, 2012.