Hand geometry recognition: an approach for closed and separated fingers

Adeniyi Jide Kehinde[1], Oladele Tinuke Omolewa[2], Akande Oluwatobi Noah[1], Adeniyi Tunde Taiwo[1]
[1]Department of Computer Science, Landmark University, Omu-Aran, Nigeria
[2]Department of Computer Science, University of Ilorin, Ilorin, Nigeria

Article Info

Article history:

Received Jan 12, 2021
Revised Jun 9, 2022
Accepted Jul 4, 2022

Keywords:

Biometrics
Euclidean distance
Hand geometry
Manhattan distance
Support vector machine

ABSTRACT

Hand geometry has been a biometric trait that has attracted attention from several researchers. This stems from the fact that it is less intrusive and could be captured without contact with the acquisition device. Its application ranges from forensic examination to basic authentication use. However, restrictions in hand placement have proven to be one of its challenges. Users are either instructed to keep their fingers separate or closed during capture. Hence, this paper presents an approach to hand geometry using finger measurements that considers both closed and separate fingers. The system starts by cropping out the finger section of the hand and then resizing the cropped fingers. 20 distances were extracted from each finger in both separate and closed finger images. A comparison was made between Manhattan distance and Euclidean distance for features extraction. The support vector machine (SVM) was used for classification. The result showed a better result for Euclidean distance with a false acceptance ratio (FAR) of 0.6 and a false rejection ratio (FRR) of 1.2.

Corresponding Author:

Adeniyi Jide Kehinde
Department of Computer Science, Landmark University
Omu-Aran, Nigeria
Email: adeniyi.jide@lmu.edu.ng

1. INTRODUCTION

Biometrics as a means of identification has been an area of interest in recent years [1]. This can be attributed to its reliability and dependability when compared with the traditional methods of identification. Traditional methods of identification which include password, cards, tokens and so on are prone to being forgotten and easily getting lost [1], [2]. Biometrics is the use of a person's behavioral or physiological trait for identification [3], [4]. Biometrics can be classified as being physiological or behavioral. Physiological uses what an individual is for identification while behavioral uses what an individual does for identification [5], [6]. Physiological biometric traits include hand geometry, palmprint, and face. Behavioral traits include signature, speech and so on.

Hand geometry is the use of measurements taken from the human hand for recognition [6]. It is a physiological trait and several researchers have examined this biometric trait for identification [4], [7], [8]. Some of the advantages of this biometrics include easy of capture, easy integration with other biometric traits such as palmprint [9]. However, challenges to the use of this biometrics include restriction of hand pose during capture [8]. In most literatures, the approach is to specify that users are to keep their fingers separate during capture [1], [4], [7]. In some systems however, users are advised to keep their hands together during capturing [10], [11].

Among these studies is the work of [8]. They proposed a hand geometry identification system for mobile devices. In their system, the mobile camera is used to capture the hand of the user. The users are instructed to separate their fingers during capture. The placement of the hand is in such a way to ensure skin color in five areas of the camera. This helps in hand color segmentation. Skin thresholding is used for the different color models. Morphological operations were used to remove noise, holes and blobs. 55 distances were measure from the hand image. For discrimination, Euclidean distance, bagged trees, k-nearest neighbor (KNN), latent Dirichlet allocation (LDA), subspace discriminant (SSD), subspace KNN (SKNN), and weighted KNN (WKNN) were used. Their method achieved an equal error rate (EER) of 0.9 at its best.

Agbinya [7] also presented a model for biometric security systems using human palm geometry. In their system, hand acquisition was performed with the aid of a document scanner. The hands were captured with the fingers separated. After this, it was fed into a segmentation software. The hue, saturation and value (HSV) were used for hand segmentation. The left inner geometry, left outer geometry and right palm outer geometry were used as feature for the system. The length and angles of the fingers were extracted.

A user identification system that made use of wavelet feature of human geometry graph was presented by [4]. The system made use of GPDS150 hand database. The database is made of hand images whose fingers are separate from each other. The system starts with grey scale conversion, then performs filtering (morphological filters) and binarization (using Otsu thresholding). Canny edge detection was used for hand segmentation. The wrist portion was cropped out of the image. To locate 12 nodes of the hand, the contour of the hand was traced. By locating twelve landmarks (nodes) after tracing the contour of the hand using eight connected component technique, the hand image was depicted as a weighted undirected complete connected graph. The Euclidean distance determines the weight of a pair of nodes in terms of the root of square differences among the coordinates. A multiclass support vector machine (SVM) classifier was employed to evaluate the efficiency of user recognition using biometric feature of hand geometry.

Charfi *et al.* [12] provided a personal recognition system focused on local attributes, using hand modality. The hand images were obtained from the Bogazici University hand database and the fingers are not joined together. K-mean clustering was used for hand segmentation and the key-points of the hand were located. Ten finger lengths were measured and five measurements were taken from the center of the palm. Hand shapes were also extracted and fusion was performed with weighted sum rule using the EER for weight allocation. The scale invariant feature transform (SIFT) descriptor was used for features extraction. The Euclidean distance was used for matching. The results obtained showed a EER of 2.25, false acceptance ratio (FAR) of 2.46, false rejection ratio (FRR) of 2.1 and an RR=97.82.

In [13], the authors proposed a Bimodal biometric framework for hand shape and palmprint recognition based on sparse SIFT representation. The Bosphorus hand database and the Indian Institute of Technology of Delhi (IITD) hand database were used for testing the system. The hand database had the hand images captured with fingers separated from each other. K-means clustering algorithm and by morphological operators were used for segmentation. SIFT was selected for fusion of both the hand geometry and the palmprint. SVM is used for classification. Cascading feature level fusion was used. A CIR of 96.15 was obtained.

Khaliluzzaman *et al.* [11] presented a hand geometry-based personal verification system in a closed hands scenario. Digital camera and mobile phones were used for capturing hand images. During capturing of the hand image, there was no gaps or overlapping. The background of the captured image also has to be even. The system they proposed started with preprocessing. The preprocessing steps include skin color conversion from red, green, blue (RGB) to YCbCr, binarization and use of morphological operation to fill in the gaps, boundary extraction of right hand and region of interest extraction. The Euclidean distance was used for matching. A FAR of 2% and FRR of 1% was recorded.

Comparative study of algorithms used for learning in biometric recognition using hand geometry was presented by [11]. A system was built using a Negatoscope, a wooden box and a digital single-lens reflex (DSLR) camera to obtain the images. Binarization and contour segmentation was performed for hand and finger segmentation. SVM, BayesNet and smoothened receptor (SMO) was used for classification. The comparative study was performed between the different classifiers used for classification. An accuracy of 99.85% was recorded as the best result for SVM. This paper presents a finger geometry biometric system that identifies individual using hand geometry, irrespective of whether their fingers are closed or separate.

2. RESEARCH METHOD

The block diagram shown in Figure 1 depicts the major steps taken in identifying an individual's hand using the measurements taken from the hand. The steps start with the acquisition of the hand image using a mobile device. After the acquisition, preprocessing steps follow. The preprocessing steps include grey conversion, edge detection, keypoint detection, size normalization, and normalized key-point detection.

After preprocessing, features are extracted, and classification is performed. The steps above are further examined below.

2.1. Grey conversion

The input image to this biometric system is in the RGB color model and were manually acquired using a mobile device camera. In an RGB color model, each pixel is described in terms of its red, green and blue intensity. To convert the manually captured image into its grey form, the red, green and blue color value has to be mapped into a single value. The weighted average of the red, green and blue intensity of each pixel is computed as the grey scale intensity using (1):

$$I = 0.21X + 0.72Y + 0.07Z \tag{1}$$

where X, Y and Z are the intensity values of a pixel's red, green and blue color. Because of the sensitivity of humans to green, the weight assigned to green is higher than red and blue intensities. Some output image is shown in Figure 2.

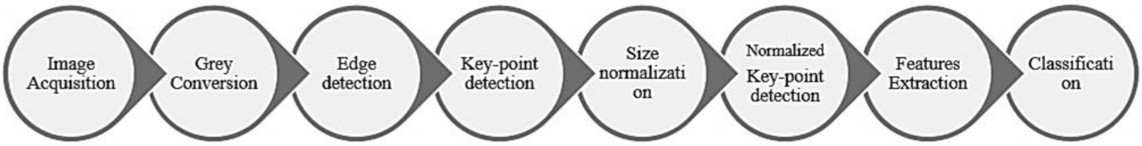

Figure 1. Image showing the major steps of the system

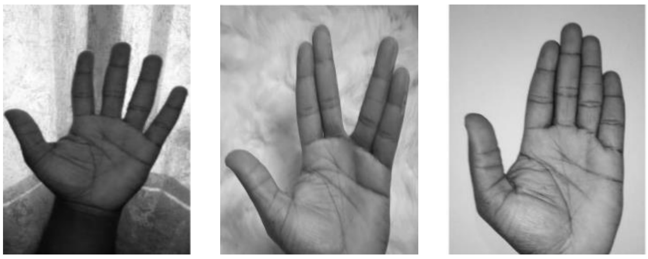

Figure 2. Greyscale image of hand

2.2. Edge detection

Edges in image are areas in images where there are notable changes in pixel intensity. They are used in images to filter out parts of an image. Canny edge algorithm is a common edge detection algorithm and it is credited as an optimal edge detector [14], [15]. The three criteria that Canny edge detection satisfies include low error rate, great localization and single response. Low error rate satisfies the condition that no edge should be missed and non-edge pixels should be ignored. Great localization implies the difference between the edges located by the indicator and the actual edges should be low. Single response simply means each edge should have a single response. Canny edge algorithm follows a series of steps to detect edges in images. The steps are [16], [17].

2.2.1. Filtering

The first step in canny edge detection algorithm is filtering. Filtering an image removes noise from the image. Noises are unwanted parts of an image that is cause by several things including environmental factor, technological factor, noises introduced during transfer of file and so on. The most commonly used filtering method in canny edge algorithm is the Gaussian filter. In Gaussian filter, a Gaussian kernel function is used for filtering. The kernel function is expressed in (2) [12].

$$G(a, b, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(a^2+b^2)/2\sigma^2} \tag{2}$$

σ is the scale factor of space. The smaller the σ, the lesser the smoothing and rich edge of the image. a and b are the coordinates in x and y direction respectively.

2.2.2. Finding gradient

After removing noise from the image, the algorithm finds the edge strength by calculating the image's gradient. To obtain the gradient of the image, the Sobel operator is used. The Sobel operator performs a spatial gradient measurement of the image using a 2-D mask. The Sobel operator uses a pair of 3x3 convolution masks to convolve the image. The two masks are shown in Figure 3. The G_a estimate the x-direction gradient, and G_b estimates the y-direction gradient.

-1	0	+1
-2	0	+2
-1	0	+1

G_a

+1	+2	+1
0	0	0
-1	-2	-1

G_b

Figure 3. The Sobel operator masks

$|G|$ in (3) is the magnitude (or edge strength) of the image's gradient. The direction of the edge is also obtained in this step and it is given in (3) [18].

$$|G| = |G_a| + |G_b| \tag{3}$$

$$\theta = \text{atan2}(G_y, G_x) \tag{4}$$

The phase angle in computed using (4). In (4), θ will give an error if G_a is zero. Hence, to alleviate this, whenever G_a is zero, θ will be set to either 90° or 0°. When $G_a = 0$, θ is set to 0° if $G_y = 0$. Otherwise, θ is set to 90° [18].

2.2.3. Traceable edge direction

After obtaining the edge direction (θ), it is converted to a traceable direction on the image. To do this, if a pixel m is surrounded by other pixels x as shown in Figure 4. Then, there are only four possible directions in the neighboring pixel. The possible directions are 0° (in the horizontal direction), 45° (in the positive diagonal), 90° (in the vertical direction) and 135° (in the negative diagonal). So, the edge direction (θ) can be approximated to one of these directions (the closest). Figure 5 shows these divisions and the regions to summarize.

v	v	v	v	v
v	v	v	v	v
v	v	m	v	v
v	v	v	v	v
v	v	v	v	v

Figure 4. A pixel m surrounded by other pixels v

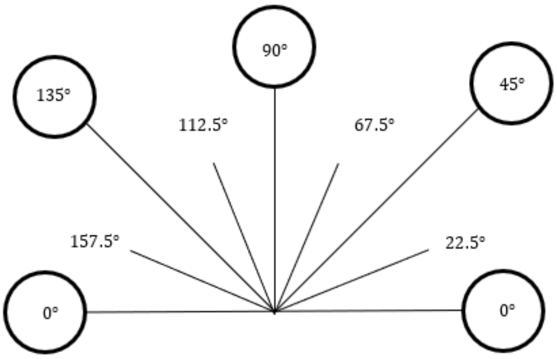

Figure 5. Edge direction and their traceable direction

From Figure 5, edge directions that fall between 0° − 22.5° and 157.5° − 180° are set to 0°. Edge directions that fall between 22.5° – 67.5° are set to 45°. Edge directions between 67.5° − 112.5° are set to 90°. Edge directions that fall between 112.5° – 157.5° are set to 135° [19].

2.2.4. Non-maximum suppression

After the edge directions have been converted to their traceable version, the pixels that are not considered edges are set to zero (suppressed). After non-maximum suppression is applied, the output is a thinned image. A thinned image is a compact representation of the pixels of an edge image where each edge is only one pixel in thickness [20].

2.2.5. Hysteresis

The final step of Canny edge algorithm involves the use of hysteresis to eliminate breaking up of edge contours. This breaking up of edge contours is known as streaking. Streaking is as a result of the fluctuation of the output of the suppression stage below and above the threshold. If one threshold is used, as a result of the noise there would be cases where the edge goes below the threshold. Likewise, it could also go above the threshold value thereby creating dashed line appearance. To eliminate this, hysteresis uses a double threshold approach. The two thresholds are S1 and S2, where S1 is the threshold that is high and S2 is the threshold that is low. High and low threshold here means the value assigned to the low threshold S2 must be lower than the high threshold (S1). All pixels in the image with a value higher than S1 is taken as an edge pixel. Any pixel value that is less than S2 is taken as a non-edge pixel and thus suppressed. Pixel values that fall between S1 and S2 are taken to be an edge if they are connected to a pixel taken to be an edge pixel. Otherwise, they are assumed to be a false edge pixel and thus suppressed. Figure 6 shows the edge image obtained from some hands [17].

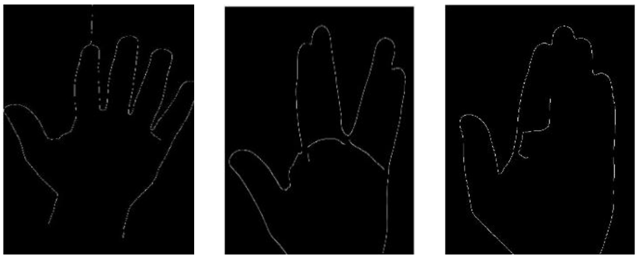

Figure 6. Edge image of hand

2.3. Key-point detection

To detect the key-points of the hand, the edge image obtained after applying the Canny edge detection algorithm was traced. The tracing algorithm follows the edge pixel and locates the peak and valley of each finger. The algorithm is:

```
Input: an edge image
Key-point [9][2]
P=0;
peak=false
valley=true
Locate first edge pixel from the base of the image, starting from the left to the right
while (p<9) {
        Locate the next connected pixel (r, c)
        If (peak==false) {
                Keypoint (p,:)={row, col}
                Peak=true
                p++
                valley=false
        }
        If (valley==false) {
                Keypoint (p,:)={row, col}
                Peak=false
                p++
                valley=true
        }
}
Output: A 9 × 2 matrix with each row corresponding to a keypoint
```

After the first key-points were detected as shown in Figure 7 (with circle at the peak and valley of the fingers), they were used to crop out the finger parts of the hand. The cropped-out finger image was resized to 640×480 to normalize the size. Finally, new key-points were extracted from the finger image. This new key-point was passed to the feature extraction stage. This is show in Figure 8.

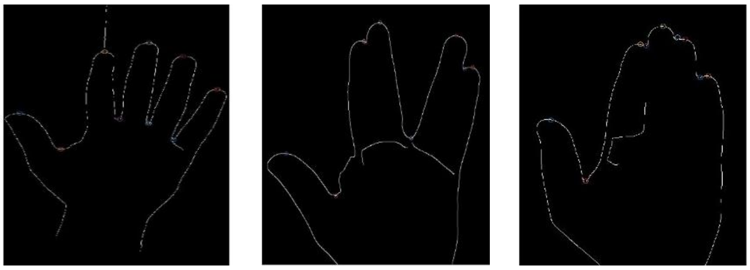

Figure 7. Keypoint detection of hand edge image

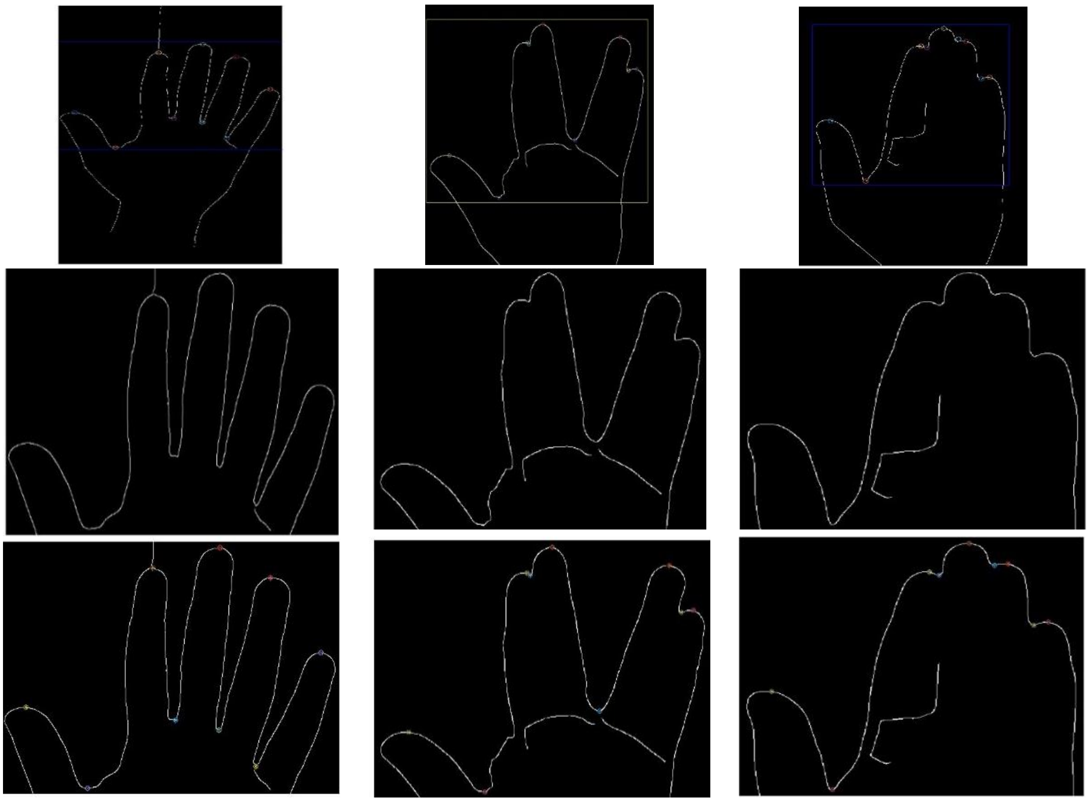

Figure 8. Finger cropping, hand resizing and second keypoint detection

2.4. Keypoint detection

In the proposed system, the Manhattan distance and Euclidean distance were used to measure distances on the hand. These distances are taken as features of the hand. The Manhattan distance and Euclidean distance between two points j and p are as expressed in (5) [21] and (6) [22]. 20 features were extracted from the hand image. Taking the key-points for the hand as a to i, then the distances taken are $f(x_1)$ to $f(x_{20})$.

$$Manhattan(o,p) = |j_1 - j_2| + |p_1 - p_2| \tag{5}$$

$$Euclidean(o,p) = \sqrt{(j_1 - p_1)^2 + (j_2 - p_2)^2} \quad (6)$$

$a = peak\ of\ thumb$
$b = valley\ between\ the\ thumb\ and\ leading\ finger$
$c = peak\ of\ leading\ finger$
$d = valley\ between\ the\ leading\ finger\ and\ the\ middle\ finger$
$e = peak\ of\ the\ middle\ finger$
$f = valley\ between\ the\ middle\ finger\ and\ the\ ring\ finger$
$g = peak\ of\ the\ ring\ finger$
$h = valley\ between\ the\ ring\ finger\ and\ the\ pinky\ finger$
$i = peak\ of\ the\ pinky\ finger$
$f(x_1) = distance\ from\ a\ to\ b$
$f(x_2) = distance\ from\ b\ to\ c$
$f(x_3) = distance\ from\ c\ to\ d$
$f(x_4) = distance\ from\ d\ to\ e$
$f(x_5) = distance\ from\ e\ to\ f$
$f(x_6) = distance\ from\ f\ to\ g$
$f(x_7) = distance\ from\ g\ to\ h$
$f(x_8) = distance\ from\ h\ to\ i$
$f(x_9) = distance\ from\ a\ to\ d$
$f(x_{10}) = distance\ from\ b\ to\ e$
$f(x_{11}) = distance\ from\ d\ to\ g$
$f(x_{12}) = distance\ from\ f\ to\ i$
$f(x_{13}) = distance\ from\ c\ to\ f$
$f(x_{14}) = distance\ from\ e\ to\ h$
$f(x_{15}) = distance\ from\ a\ to\ f$
$f(x_{16}) = distance\ from\ c\ to\ h$
$f(x_{17}) = distance\ from\ b\ to\ g$
$f(x_{18}) = distance\ from\ d\ to\ i$
$f(x_{19}) = distance\ from\ a\ to\ h$
$f(x_{20}) = distance\ from\ b\ to\ i$

The distances $f(x_1)$ to $f(x_{20})$ were taken for opened fingers and closed fingers. Both closed and open finger hand images were collected 4 times to train the classifier. Four false hands were also used for training the classifier. This led to a total training set of 8 for each hand. Similar features were also extracted with Euclidean distance. The distances are shown in Figure 9.

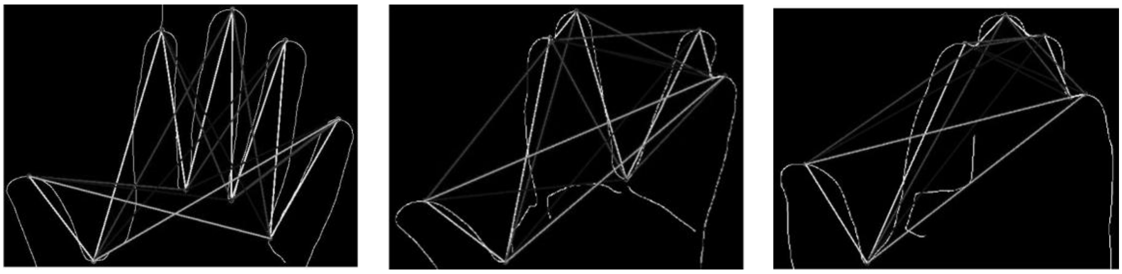

Figure 9. Distances measured from the fingers of the hand

2.5. Matching

For matching, the SVM was used as the classifier. SVM has been a tool used greatly in pattern recognition and classification [5], [9], [23]. It is a binary classification method by supervised learning. SVM has a relatively high accuracy for binary classification [24]. To use this classifier, there is need to select a good kernel and adjust the parameters of the functions to obtain the best accuracy [6], [25]. For a data that is linearly separable, we can obtain an hyperplane f(x)=0 that separate the data as shown in (7) [26].

$$f(x) = \sum_{i=1}^{n} w_i\, x_i + b = 0 \quad (7)$$

w is a vector with n-dimensions, b is a scalar number. The poison of the separation hyperplane is determined by w and b. i is either

$w.x_i + b \geq 1$ for x_i in one class or

$w.x_i + b \leq -1$ for x_i in the other class

3. RESULTS AND DISCUSSION

A total of 1,000 hand images was used to train and test the system. The hands were captured in a non-constrained manner. Users were at liberty to join some or all the fingers. The background was made as even as possible during capture. The hands were capture from 20 individuals, each with a minimum of 20 hand images. 12 out of the volunteers were male and the rest female. The hand images had 500 genuine and 500 impostors. 50% (500 images) of the captured images was used for training the system. The remaining 50% (500 images) was used for testing. The system was tested on a DELL Inspiron n4110 laptop with a RAM of 4 GB and a dual core i3 processor with a frequency of 2.20 GHz each. After training a hand with SVM, the optimized objective function obtained from extracting features with Manhattan distance is depicted using Figure 10. The function evaluation is also shown in Figure 11.

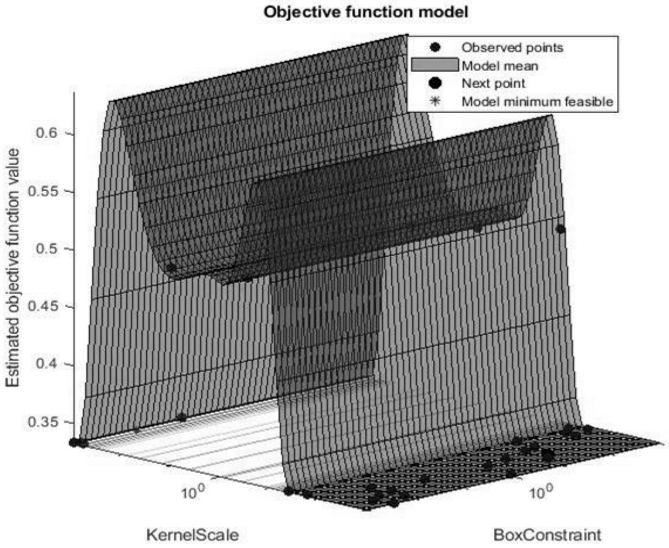

Figure 10. Manhattan objective function model

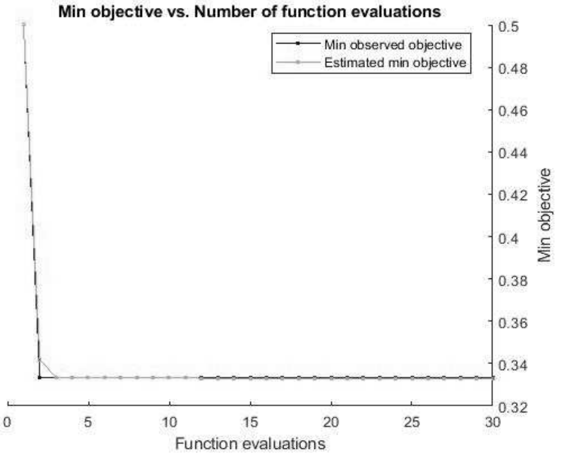

Figure 11. Manhattan function evaluation

The objective function model is shown in Figure 12. Figure 13 shows a *MaxObjectiveEvaluations* of 30 was reached and the total function evaluation was 30. The total time elapsed was about 226.5119 s. The observed objective function value was 0.33333 and the estimated objective function value was 0.33276. A similar *MaxObjectiveEvaluation* and total function was noted for Euclidean based features extraction. The time was however 99.8098 s. The observed objective function value and estimated objective function value were both 0.33333. The graphs are depicted in Figures 12 and 13.

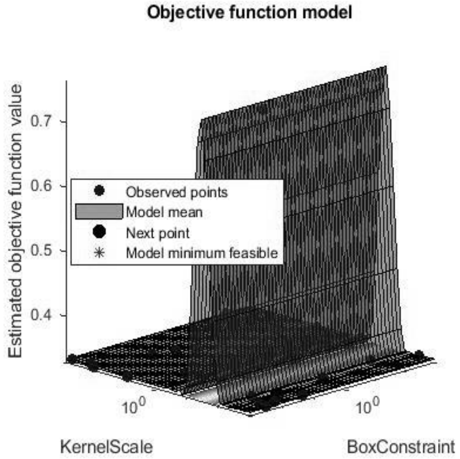

Figure 12. Euclidean objective function model

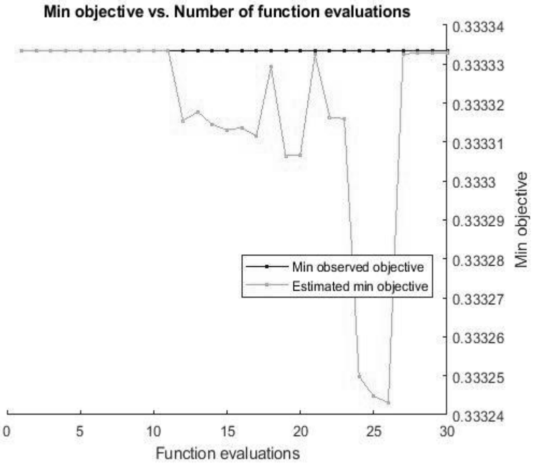

Figure 13. Euclidean function evaluation

The system was tested with manually acquired images of closed and open finger. The confusion matrix of both distance metrics is depicted in the Tables 1 and 2. The false acceptance ratio (FAR) and false rejection ratio (FRR) were obtained using (8) and (9). For the features extracted with Manhattan distance, a FAR and FRR of 0.8 and 1.6 were obtained respectively. The Euclidean distance produced a FAR of 0.6 and an FRR of 1.2 It was observed that while the results were relatively close, for features extraction, the Euclidean distance performed slightly better that the Manhattan distance. However, from the confusion matrix in Tables 1 and 2, the false positive (FP) and false negative (FN) of Euclidean distance was less than that of Manhattan distance by 1.

$$FAR(FPR) = \frac{FP}{FP+TN} \quad (8)$$

$$FRR(FNR) = \frac{FN}{FN+TP} \quad (9)$$

Where FP is the false positive, TN is the true negative, FN is the false negative, TP is the true positive, FPR is the false positive rate and FNR is the false negative rate.

Table 1. Confusion matrix for Manhattan distance

No=500	Predicted no	Predicted yes	
Actual no	248	2	250
Actual yes	4	246	250
Total	252	248	500

Table 2. Confusion matrix for Euclidean distance

No=500	Predicted no	Predicted yes	
Actual no	249	1	250
Actual yes	3	247	250
Total	252	248	500

3.1. Comparison of result

Comparing the FAR and FRR of Euclidean distance and Manhattan distance shows that the result of Euclidean distance is less than that of Manhattan distance. This can be traced to the FP and FN of both distances. The FP and FN obtained for Euclidean distance were less than that of Manhattan distance by 1. Examining further, it was observed that this can be traced to the distance measured by each. Euclidean distance was obtained in two decimal places while Manhattan produced an integer. This was observed to be the difference between the instance where FP and FN increased in Manhattan distance. Furthermore, Table 3 shows a comparison of the result obtained from this system and those from other similar systems.

Table 3. Comparison of result with similar systems

Paper	Restriction	FAR (%)	FRR (%)
[4]	Separate fingers	0.34	1.38
[6]	Separate fingers	0.69	2.08
[12]	Separate fingers	2.46	2.1
[11]	Closed fingers	2	1
Our method	**Closed and separate fingers**	**0.8**	**1.6**
		0.6	**1.2**

4. CONCLUSION

In this paper, we presented an approach to the use of measurements taken from the human hand for recognition. This technique solves the open finger/closed finger constraint on hand geometry. Experimental results show a good accuracy. The contribution of this paper includes the presentation of a method to solve close fingers/open fingers challenge of hand geometry. It compares Manhattan distance and Euclidean distance as a feature extraction method in hand geometry. It was observed that the FAR and FRR for Euclidean distance was a little lower than Manhattan distance. The training time required for Euclidean distance is also considerably lower than that of Manhattan distance for the same amount of training dataset. The paper also presented a method for normalizing measurements taken from the fingers irrespective of the distance of capture. This further increased the accuracy of the distances measured on the image as the images were normalized.

ACKNOWLEDGEMENTS

Authors appreciate Landmark University Centre for Research and Development, Landmark University, Omu-Aran, Nigeria for fully sponsoring the publication of this research article.

REFERENCES

[1] A. K. Sharma, A. Raghuwanshi, and V. K. Sharma, "Biometric system-a review," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 6, no. 5, pp. 4616–4619, 2015.

[2] Z. Wang, J. Yang, and Y. Zhu, "Review of ear biometrics," *Archives of Computational Methods in Engineering*, vol. 28, no. 1, pp. 149–180, Jan. 2021, doi: 10.1007/s11831-019-09376-2.

[3] A. S. Anwar, K. K. A. Ghany, and H. Elmahdy, "Human ear recognition using geometrical features extraction," *Procedia Computer Science*, vol. 65, pp. 529–537, 2015, doi: 10.1016/j.procs.2015.09.126.

[4] S. Angadi and S. Hatture, "Hand geometry based user identification using minimal edge connected hand image graph," *IET Computer Vision*, vol. 12, no. 5, pp. 744–752, Aug. 2018, doi: 10.1049/iet-cvi.2017.0053.

[5] B. P. Rao, G. S. Rao, and G. V. Kumari, "Flower pollination-based K-means algorithm for medical image compression," *International Journal of Advanced Intelligence Paradigms*, vol. 18, no. 2, 2021, doi: 10.1504/IJAIP.2021.10035157.

[6] S. A. Angadi and S. M. Hatture, "User identification using wavelet features of hand geometry graph," in *2015 SAI Intelligent Systems Conference (IntelliSys)*, Nov. 2015, pp. 828–835, doi: 10.1109/IntelliSys.2015.7361238.

[7] J. I. Agbinya, "Human palm geometry modelling for biometric security systems," in *2019 Cybersecurity and Cyberforensics Conference (CCC)*, May 2019, pp. 160–164, doi: 10.1109/CCC.2019.00011.

[8] S. Barra, M. De Marsico, M. Nappi, F. Narducci, and D. Riccio, "A hand-based biometric system in visible light for mobile environments," *Information Sciences*, vol. 479, pp. 472–485, Apr. 2019, doi: 10.1016/j.ins.2018.01.010.

[9] A. Bapat and V. Kanhangad, "Segmentation of hand from cluttered backgrounds for hand geometry biometrics," in *2017 IEEE Region 10 Symposium (TENSYMP)*, Jul. 2017, pp. 1–4, doi: 10.1109/TENCONSpring.2017.8070016.

[10] S. Sharma, S. R. Dubey, S. K. Singh, R. Saxena, and R. K. Singh, "Identity verification using shape and geometry of human hands," *Expert Systems with Applications*, vol. 42, no. 2, pp. 821–832, Feb. 2015, doi: 10.1016/j.eswa.2014.08.052.

[11] M. Khaliluzzaman, M. Mahiuddin, and M. M. Islam, "Hand geometry based person verification system," in *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, Oct. 2018, pp. 1–6, doi: 10.1109/ICISET.2018.8745620.

[12] N. Charfi, H. Trichili, A. M. Alimi, and B. Solaiman, "Personal recognition system using hand modality based on local features," in *2015 11th International Conference on Information Assurance and Security (IAS)*, Dec. 2015, pp. 13–18, doi: 10.1109/ISIAS.2015.7492764.

[13] G. Jaswal, A. Kaul, and R. Nath, "Multimodal biometric authentication system using hand shape, palm print, and hand geometry," 2019, pp. 557–570, doi: 10.1007/978-981-13-1135-2_42.

[14] Y. Pititeeraphab and C. Pintavirooj, "Identity verification using geometry of human hands," in *2018 11th Biomedical Engineering International Conference (BMEiCON)*, Nov. 2018, pp. 1–4, doi: 10.1109/BMEiCON.2018.8609986.

[15] C.-S. Chen and Y. Jeng, "Improving GPR imaging of the buried water utility infrastructure by integrating the multidimensional nonlinear data decomposition technique into the edge detection," *Water*, vol. 13, no. 21, Nov. 2021, doi: 10.3390/w13213148.

[16] S. P. Kumar, T. V Narendra, and N. A. Vinay, "Short hand recognition using canny edge detector," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, no. 5, pp. 902–907, May 2017, doi: 10.23956/ijarcsse/V7I5/0200.

[17] S. A. Ahmer and P. Lohi, "Human identity recognition using ear lobes geometric features," *International Journal Of Engineering And Computer Science*, Jun. 2016, doi: 10.18535/ijecs/v5i6.02.

[18] S. Bhardwaj and A. Mittal, "A survey on various edge detector techniques," *Procedia Technology*, vol. 4, pp. 220–226, 2012, doi: 10.1016/j.protcy.2012.05.033.

[19] S. M. Albarzinji, “An efficient approach for improving canny edge detection algorithm,” *International Journal of Advances in Engineering and Technology*, vol. 7, no. 1, pp. 59–65, 2014.
[20] S. I. Azeezullah and D. A. Huynh, “Canny edge detection report,” *Image Process*, pp. 1–14, 2013.
[21] A. K. Singh, A. K. Agrawal, and C. B. Pal, “Hand geometry verification system: A review,” in *2009 International Conference on Ultra Modern Telecommunications and Workshops*, Oct. 2009, pp. 1–7, doi: 10.1109/ICUMT.2009.5345652.
[22] P. Mulak and N. Talhar, “Analysis of distance measures using k-nearest neighbor algorithm on KDD dataset,” *International Journal of Science and Research (IJSR)*, vol. 4, no. 7, pp. 2101–2104, 2013.
[23] M. V. P. Do Nascimento, L. V. Batista, and N. L. Cavalcanti, “Comparative study of learning algorithms for recognition by hand geometry,” in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2014, pp. 423–428, doi: 10.1109/SMC.2014.6973944.
[24] Y. Song, Z. Cai, and Z. L. Zhang, “Multi-touch authentication using hand geometry and behavioral information,” *Proceedings-IEEE Symposium on Security and Privacy*, pp. 357–372, 2017, doi: 10.1109/SP.2017.54.
[25] D. Rezgui and Z. Lachiri, “ECG biometric recognition using SVM-based approach,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 11, pp. 94–100, Jun. 2016, doi: 10.1002/tee.22241.
[26] S. Sangeetha and N. Radha, “A new framework for IRIS and fingerprint recognition using SVM classification and extreme learning machine based on score level fusion,” in *2013 7th International Conference on Intelligent Systems and Control (ISCO)*, Jan. 2013, pp. 183–188, doi: 10.1109/ISCO.2013.6481145.

BIOGRAPHIES OF AUTHORS

Adeniyi Jide Kehinde received his bachelor degree in Computer Science from Adamawa State University, Adamawa, Nigeria. He holds an MTech in Computer Science from Federal University of Technology, Akure, Nigeria. He obtained his doctoral degree in the Department of Computer Science, University of Ilorin, Ilorin, Nigeria. His interest includes various areas such as biometrics, computer vision, security, artificial intelligence and machine learning. He can be contacted at email: adeniyi.jide@lmu.edu.ng.

Oladele Tinuke Omolewa is a Senior lecturer in the Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria. She obtained her B.Sc. in Computer Science from the University of Benin, Benin City, Edo State. She bagged her M.Sc in Mathematics (Computer Science Option) from the University of Ilorin and a Ph.D. in Computer Science also from the University of Ilorin. She is currently an Associate Professor at the University of Ilorin. She is known across board for her innovative research work in Bioinformatics and Computational Biology. Her re-search interest includes Artificial Intelligence, Data Mining, and Cyber Security. He can be contacted at email: Oladele.to@unilorin.edu.ng.

Akande Oluwatobi Noah received his bachelor degree in Computer Science from Ladoke Akintola University of Technology, Ogbomosho, Nigeria. He holds an MTech in Computer Science from Ladoke Akintola University of Technology, Ogbomoso, Nigeria. He obtained his doctoral degree in the Department of Computer Science, University of Ilorin, Ilorin, Nigeria. His interest includes various topics such as biometrics, computer vision, security, artificial intelligence and machine learning. He can be contacted at email: akande.noah@lmu.edu.ng.

Adeniyi Tunde Taiwo received his bachelor degree in Computer Science from Adamawa State University, Adamawa, Nigeria. He holds an MTech in Computer Science from Federal University of Technology, Akure, Nigeria. He is currently working towards a doctoral degree in the Department of Computer Science, University of Ilorin, Ilorin, Nigeria. His interest includes various areas such as biometrics, computer vision, security, artificial intelligence and machine learning. He can be contacted at email: adekitos2@gmail.com.