International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 2
ISSN: 2088-5334

A Mobile Palmprint Authentication System Using a Modified MNT Algorithm, Circular Local Binary Pattern, and CNN (mobileNet)

Jide Kehinde Adeniyi [a,*], Tinuke Omolewa Oladele [b], Ayodele Adebiyi [a], Marion Adebiyi [a], Tunde Taiwo Adeniyi [b]

[a] *Department of Computer Science, Landmark University, Omu-aran, Nigeria*
[b] *Department of Computer Science, University of Ilorin, Ilorin, Nigeria*
Corresponding author: [*] adeniyi.jide@lmu.edu.ng

***Abstract*—A few approaches have been proposed for hand segmentation in palmprint recognition. Skin-color information does not process sufficient information for discrimination in complex backgrounds and variable illumination. The use of guides has also been proposed, which restricts hand placement during capturing. Contour tracing algorithms have also been proposed in the literature. This worked in an even background scenario with no objects or patterns around the hand. In the case of uneven background with objects present, the traditional contour tracing algorithm cannot accurately segment the hand from the background. Hence, this paper proposes a modified Moore Neighbor Tracing (MNT) algorithm for hand detection and key-point extraction in complex backgrounds. The hand image is converted to grey, and the edges in the hand image are detected. The modified algorithm then transverses selected edges and returns the peak and valleys of each finger. This is then used to crop the palm. The modified algorithm improves the accuracy of hand detection in complex backgrounds with an F-Score of 0.8657. A mobile palmprint biometric system was also presented using Circular Local Binary Pattern (CLBP) and Convolutional Neural Network (CNN). The system showed an accuracy of 98.3% for hands captured with the mobile device and the CASIA online database. An accuracy of 99.0% was also recorded for GPDS and PolyU online databases.**

***Keywords*— Hand segmentation; improved contour tracing algorithm; feature extraction; complex background, convolutional neural network.**

Manuscript received 25 Aug. 2021; revised 12 Sep. 2022; accepted 22 Feb. 2023. Date of publication 30 Apr. 2023.

I. INTRODUCTION

Mobile devices these days have become an essential part of our lives. This is due to the wide range of services available to them [1]. These services include SMS, E-mail, internet facility, banking transactions, contact management [2]. These devices could be used to access various information about the user's digital life [3]. Hence, the need to secure these devices against unauthorized access. Available security measures on mobile devices include passwords and pattern recognition [4]. While these methods served security purposes, human limitations such as forgetfulness have plagued these methods. Biometrics was then introduced. Biometric verification uses an individual's physical or behavioral trait for recognition [5], [6]. Face recognition is among the biometric recognition methods introduced, but challenges such as pose variation, easy access to face image (to trick the system), and illumination have limited its use [6]–[8].

Palmprint recognition as a biometric trait uses an individual's palm for recognition. It is a favorable authentication method on mobile devices because of its favorable traits like high acceptance, easy acquisition, discriminating features, difficulty in leakage, and so on [9], [10]. The fact that a mobile phone can be authenticated without the need to touch the phone could make authenticating the use of a phone by a friend easier. In a typical palmprint recognition system, there are four stages. These stages are acquisition, preprocessing, feature extraction, and matching. The acquisition stage is for capturing the hand, and the preprocessing stage prepares the image captured for feature extraction. Features are extracted at the features extraction stage, and matching is performed at the matching stage [11]–[15].

Palmprint can either be contact based or contactless (based on its acquisition). Contact-based devices that require contact with hands are used to acquire the hand. These devices include desktop scanners, Charge-coupled device (CCD)

scanners, and so on [16]–[18]. In contact-based palmprint, hand segmentation is a thresholding problem [14] [19] (with the background color different from the hand color). Although contact-based palmprints achieve high accuracy, infections and lack of flexibility in acquisition affect them [9].

Contactless palmprint recognition acquires hand images without contact with the acquisition device. The device is usually a camera [20], [21]. This acquisition mode makes its acceptance high as user's hand is not restricted during acquisition. However, with this liberty comes the challenges of complex backgrounds, changing illumination, and so on [21]. The challenge is even more on mobile platforms due to limitations in processing power [21]. In a contactless system, an important preprocessing step is the region of interest extraction. This step extracts the palm. Certain hand key points must be detected [22]. The goal of hand detection in palmprint is to extract these key points. These points are the valley between the pinkie finger, the ring finger, the leading finger, and the middle finger. These points are used to crop out the palm [19], [23]. This paper proposes an algorithm for hand detection in a complex background using a Modified Moore Neighbour Tracing Algorithm, and it also presents a mobile palmprint recognition system using Circular Local Binary Pattern and Convolutional Neural Network. Several approaches have been proposed to do this. For cases where the hand was acquired with a contact-based device or a camera setup was used for acquisition, thresholding and binarization are usually employed for hand detection.

A. Hand Segmentation Review

An example is the system that Hussein et al. proposed [11]. The hand was captured using a CCD device, and hand detection was performed by thresholding and converting the image to its binary form. Verma and Chandran [24] used a document scanner for acquisition in their system. This made the background color different from the scanned hand, making segmentation easy.

A similar approach is the use of a camera setup for hand acquisition [13], [18], [22]. In this case, the hand is captured in a controlled environment with an even background that is separated from the hand color. In most cases, the camera is placed on a stand at a distance from the background that the hand is placed [13]. Hand segmentation in this scenario is usually performed by using the Otsu thresholding method to segment the hand from the background and using convex hull or pixel scanning to obtain the key-points of the hand [18].

Another notable approach is skin color thresholding. This technique uses the skin's color to segment the hand from the background [25]. In this case, the hand is captured with a complex background, and an attempt is made to detect the hand in the image using previous data on skin color. This usually entails using a different color model from RGB color model. Barra et al. [25] used the HSV color model; the mean hue was used for determining skin color. Bailadoar et al. [8] made use of flood segmentation, and they used the edge image and the image's binary equivalent for hand detection. Skin color thresholding and region growth were also proposed by Saeed et al. [7]. Their system used the center of mass to obtain the skin threshold and expanded it to other pixels using the threshold for discrimination. Gao et al. [9] used grey information and shape information for coloring likelihood degree as an improvement to the traditional active shape modeling.

Wu and Leng [21] proposed the use of a guide for hand capture. This technique is also another method used for hand acquisition in complex backgrounds. This method is usually used on a mobile platform to restrict hand placement while selecting predefined key-points on the guide. This method reduces the effort required by detecting the key-point from the guide. Wu and Leng [21] used a guide in the form of a hand for their system. Double-Line-Single-Point (DLSP) approach was proposed by Leng et al. [10]. DLSP uses two parallel lines as a guide for hand capture. These guides are usually drawn on the screen of the capturing device to allow for easy capturing.

B. Palmprint Review

A number of mobile and non-mobile palmprint systems have been proposed. A few from the current studies were examined. Poonia et al. [26] proposed a palmprint template that stores the minutiae points' geometric information. Then the Delaunay triangulation using internal angle matching is used for matching the templates, and the extracted geometric information acts as features used for matching. The method was tested, and it reported an accuracy of 95.4%.

Xu et al. [27] examined a contactless palmprint image identification and verification system. Image alignment was applied together with a deep network (spatial transformation network) to increase the recognition accuracy of the system. A residual network was used in their system for classification. The system accuracy recorded 94.73 for the CASIA database and 98.50 GPDS database.

Patil and Pawar [28] presented a Haar wavelet, DCT, and Fast ICA blend for palmprint recognition. Haar was used to decomposing the palmprint, DCT was applied, and Fast ICA was used for feature extraction. Euclidean distance was used for matching. The result on polyU palmprint database produces an accuracy of 98.5%.

Zhou et al. [29] tried to extract translation and rotation invariant features from palmprint for recognition. They built an orientation edge detector to show the response of the edges in each direction. The detector combines a phase congruency-based edge detector and a bipolar filter. After this step, a local spatial frequency detector obtains the horizontal shift map. The edge and frequency detectors repeat the process, producing the feature map. The system showed an accuracy of 95.58 and 94.91 on PolyU and CASIA databases, respectively.

II. Material and Method

This section examines the proposed methodology. The stages in the proposed methodology include: hand capture, edge detection, and edge tracing using a modified Moore Neighbour Tracing algorithm, features extraction using CLBP, and classification using Convolutional Neural Network (based on Android MobileNet). The acquisition stage of the system captures the palm. The edge detection stage aims to detect as many edges as possible in the captured hand image. The modified Moore Neighbour algorithm scans the edges for the actual hand edge in the image and detects the key-points of the hand. The key-points detected are used to crop out the

palm. CLBP is applied to the palm for feature extraction, and MobileNet is used for classification.

A. Hand Acquisition

The hand was captured in real-time using a mobile device. A Tecno CX Air with 3GB of RAM and an internal storage of 16GB. It has 4 CPUs with a frequency 1.25GHz. The hands were captured using the camera at the back of the device. The hands were captured in an unconstrained environment in real-time with no pegs or guides. An example of a hand being captured is shown in Fig. 1. Fig. 2 shows examples of a hand captured.

Fig. 1 Image during capture using a mobile phone

Fig. 2 Images captured with the mobile phone

B. Edge Detection

The captured image is converted to greyscale in Fig. 3. The edges of the greyscale image were obtained. The Canny edge detection method was used to obtain these edges. This step aims to detect as many edges as possible in the image. The algorithm is as follows [30], [31].

1) Image Filtering: The first step in canny edge detection is filtering. This is done so as to remove noise from the original image. The Gaussian filtering method is used for filtering, and this is because a simple mask can be used to compute the Gaussian filter. Convolution is performed on the mask and the original image. The Canny Edge Detection algorithm usually uses a two-dimensional function, and it is shown in equation (1) [32]–[34]:

$$G(s,h;\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{s^2+h^2}{2\sigma^2}\right)} \quad (1)$$

where s is the horizontal distance, h is the vertical distance, and \sigma is the standard deviation.

2) The gradient of the Image: The magnitude and direction of the gradient of the image are computed in the second step. The Sobel operator is used for this. The Sobel operator uses a 3X3 mask to estimate the gradient in the two possible directions (x and y). The masks are shown in Fig. 2 [32]:

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

Gx

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

Gy

Fig. 3 Masks for Sobel operator

To obtain the edge strength (magnitude), (2) is applied.

$$|G| = |Gx| + |Gy| \quad (2)$$

After obtaining the magnitude, the edge direction is calculated using (3).

$$theta = invtan(Gy/Gx) \quad (3)$$

3) Non-Maximum Suppression: After obtaining the direction of the edges, maximum suppression is applied to rid the image of edges of wrong edges. This step performs the thinning of the edges [35].

4) Hysteresis: Hysteresis is applied to eliminate disjointed edges caused by operator output fluctuating above and below the threshold. A double threshold is applied in hysteresis such that pixel value above the high threshold is seen as genuine edges, and pixel values below the low threshold are rejected. Values between the high and low threshold are accepted as a genuine edge if it is connected to an actual edge; otherwise, it is rejected [36]. Fig. 4 and Fig. 5 show sample images captured and the edges detected from the images.

Fig. 4 Images captured with the mobile phone

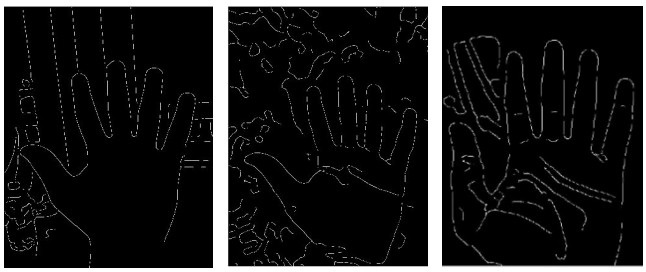

Fig. 5 Images captured with the mobile phone

C. Hand Detection

A segment with a width of one pixel and a length of one or more pixels is termed a contour [37]. To separate an object from its background, a contour is usually used. Contour tracing is a method that is used to extract an object's boundary from its background in digital images [38]. Depending on whether an object shares an edge (or an edge and a vertex) with a white pixel, a boundary could either be four pixels or

eight pixels [39], [40]. Mandee et al. [37] classified the contour tracing algorithm into three; the vertex-following algorithm, the run-data-following algorithm, and pixel following algorithm.

1) Vertex following method: traces the boundary using the vertices of the contour pixels. Though similar to the pixel following method, it saves the corner point of the contour pixel.

2) Run-data-following method uses run data pairs obtained from scanning left to the right.

3) Pixel following method follows a defined pattern to trace the contour pixels and stores the coordinates in the memory. Several methods, such as Simple Boundary Follower (SBF), modified SBF, and Moore-Neighbour Tracing (MNT) algorithms, are pixel-following boundary tracing types [38]. The MNT algorithm is shown below:

Input: A square tessellation, T, containing a connected component P of black cells.

Output: A sequence $B(b1, b2, b3, \dots, bk)$ of boundary pixel a.

Let p denote the current boundary pixel.

Let c denote the current pixel under consideration, i.e., c is in $M(p)$

Begin:

- Set B to be empty.
- From bottom to top and left to right, scan the cells of T until a black pixel, s, of P is found.
- Insert s in B.
- Set the current boundary point p to s, i.e., p=s.
- Backtrack i.e., move to the pixel from which s was entered.
- Set c to be the next clockwise pixel in $M(p)$.
- While c not equal to s do
  - If c is black
    - Insert c in B
    - Set p=c
    - Backtrack (move the current pixel c to the pixel from which p was entered)
  - else
    - Advance the current pixel c to the next clockwise pixel in $M(p)$
- end while

end

D. Modified Moore Neighbour Tracing Algorithm (MMNT)

The algorithm presented in this paper is also a pixel-following method that uses prior knowledge of the shape of the hand to predict the direction of the boundary to follow. In hand-based biometrics, the hand key-points are divided into peaks and valleys. The peak points are the pixel at the tip of the fingers, and the valley point is the pixel at the base of each finger [21], [41]. Typical hand geometry biometric system uses these points for taking the length of the fingers (which is a geometric feature that is extracted from the hand) [37], [42]. The valleys between the leading and middle fingers and the valley between the ring and pinkie fingers are used in palmprint biometric systems for Region of Interest (ROI) extraction.

The challenge of hand detection in a complex background is knowing which edge is the hand edge and which edge is the background edge. The pixel-following algorithm uses the previous knowledge of the shape of the hand to determine the direction to follow when an edge image is being scanned. At any edge junction, the algorithm follows the pattern of the hand in selecting the right edge to follow. Unlike other pixel-following algorithms that start at any pixel [43], this algorithm starts a few pixels from the base of the hand image because there are edges in the image that are not the hand edges. Edge termination is also not based on traversing a boundary pixel twice but rather on the detection of the nine key-points of the hand. The MMNT algorithm is shown below:

Input: An array of binary pixels of a hand image (P), with $P = \{0, 1\}$. Where 0 is background pixel, and 1 is an edge pixel.

Let records be a 9 X 2 array, i.e., an array of 9 rows and 2 columns.

Let $size\_row$ and $size\_col$ be the dimension of the input array, respectively.

Let a, b, c, d, e, f, g, h be the neighboring pixels of a particular pixel in consideration, i.e., $P(x, y)$, where x and y represent the pixel's row and column respectively.

Let c denote the current pixel, i.e., $c = P(x, y)$.

Start:

1. Set $size\_row$ to be $size\_row - 3$
2. Set m to be empty
3. Set Boolean $prev$ to be false
4. Set Boolean $prev1$ to be false
5. For $counter\_col = 2 : size\_col$
   a. Set $current\_col$ to $counter\_col$
   b. While $record$ is not completely filled
      i. Obtain the neighboring pixels $a, b, c, d, e, , f, g, h$
      ii. If m is empty i.e. preparing for the first peak
         If a is set and $prev$ is $false$ i.e. $a == 1$
            Increment $conter\_col$ by 1
            Set m to 2
            Set $prev$ to 0
         Elseif any of b, c, d is set
            Adjust $size\_row$ and (or) $size\_col$ to point to new c
            Set $prev$ to false
         Elseif e is set i.e. e is 1
            Adjust $size\_row$ and (or) $size\_col$ to point to new c
            Set $prev$ to true
         Else

Empty $record$
Set m to 0
Set $prev$ to false
Set $counter\_col$ to $current\_col$
Reset $size\_row$ to $input\ array\ row\ size - 3$
Break out of while loop

Endif

iii. Elseif m is set to 1 i.e. looking for finger valley

If any of a, h, g, f is set i.e. is equal to 1

Adjust $size\_row$ and (or) $size\_col$ to point to new c

Elseif b is set

Add $size\_row$ and $size\_col$ to $record$
Adjust $size\_row$ and $size\_col$ to point to new c
Set m as 2 i.e. to look for finger peak next.

Else

Empty $record$
Set m to 0
Set $prev$ to false
Set $counter\_col$ to $current\_col$
Reset $size\_row$ to $input\ array\ row\ size - 3$
Break out of while loop

endif

iv. Elseif m is set to 2 i.e. looking for finger peak

If h is set and $prev1$ is $false$ i.e. $a == 1$

Add $size\_row$ and $size\_col$ to $records$
Adjust $size\_row$ and $size\_col$ to point to new c

Elseif any of a, b, c is set

Adjust $size\_row$ and (or) $size\_col$ to point to new c
Set $prev1$ to false

Elseif d is set i.e. d is 1

Adjust $size\_row$ and $size\_col$ to point to new c
Set $prev1$ to true

Else

Empty $record$
Set m to 0
Set $prev$ to false
Set $counter\_col$ to $current\_col$
Reset $size\_row$ to $input\ array\ row\ size - 3$
Break out of while loop

Endif

v. Endif

c. End while loop

6. End for loop

Output: An array ($records$) of 9 rows and 2 columns. With each row corresponding to the key-points of the fingers in the image and the columns corresponding to the coordinates.

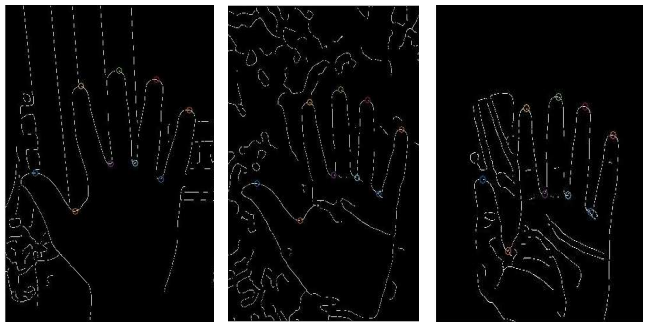

Fig. 6 Key-points detected after applying the MMNT

E. Region of Interest Extraction

After the hand's key-points are detected, the region of interest is extracted. The region of interest in palmprint is the palm of the hand. The key-points used for extracting the palm of the hand is the valley point between the leading finger and the middle finger (record (4, :)) and the valley point between the ring finger and the pinkie finger (record (8, :)). Fig. 7 shows the identified keypoint, and Fig. 8 shows the cropped palm image.

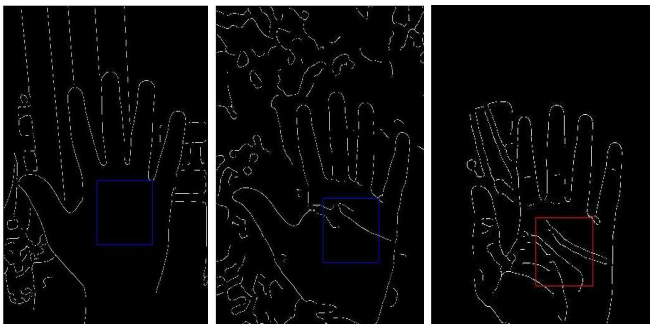

Fig. 7 Rectangular location of the palm using the key-points

Fig. 8 Cropped palm image

F. Features Extraction

The Circular Local Binary Pattern (CLBP) was used to extract features from the hand's palm. The CLBP was selected because of the need to reduce the effect of changes in intensity on the palm image. This is necessary because the mobile phone will capture images with different light intensities at different places. Given a pixel (x_a, y_a) and its circular neighbor P and R, sample point at location (x_b, y_b) can be obtained using (4).

$$(x_b, y_b) = (x_a + Rcos(2\pi b/P),\ y_a - Rcos(2\pi b/P)) \quad (4)$$

Where sampling points are denoted with $g_P = I(x_b, y_b)$ with $k \in \{0, 1, \ldots, P-1\}$.
This CLBP is then obtained with the traditional Local Binary pattern expression, as shown in equation (5).

$$CLBP_{P,R}(x_b, y_b) = \sum_{b=0}^{P-1} S(g_b - g_a)2^b \quad (5)$$

Where $g_c$and g_i are the grey value of the central pixel and its neighbors, respectively. The function for thresholding the pixel intensity $S(u)$ is given in (6).

$$S(u) = \begin{cases} 1, & u \geq 0 \\ 0, & < 0 \end{cases} \quad (6)$$

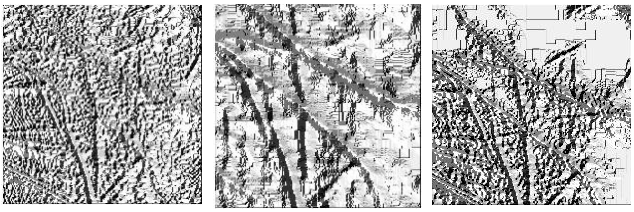

Fig. 9 CLBP output images

G. Classification

MobileNet is a popular and efficient lightweight CNN model available on the mobile platform for classification. Google developed it for various uses ranging from object detection to image classification. It uses depth-wise separable filters to deepen the network through convolution, and it reduces parameters and computations [44]. The architecture of the mobileNet model used is shown in Table 1.

TABLE I
ARCHITECTURE OF THE SYSTEM

Type/Stride	Filter shape	Input size
Conv / s2	3 x 3 x 1 x 32	224 x 224 x 1
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
5x		
Conv / s1	1 x 1 x 512 x 512	14 x 14 x 512
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
FC / s1	1024 x 25	1 x 1 x 1024
SoftMax / s1	classifier	1 x 1 x 25

The Android Neural API (NNAPI) was used to implement the network. The NNAPI consists of Machine Learning libraries, frameworks, and tools that help train models off-device and deploy the trained network on an Android mobile platform [44] [45].

III. Result and Discussion

The system evaluation was carried out using the method's performance, and the method's computational time was also stated. To evaluate the method, images captured in real time were used. The environmental condition was not constrained and varied at different times of the day and with different backgrounds. A comparison was also made with the global thresholding and skin color methods.

The Otsu threshold method was used for the global thresholding method because it is a popular thresholding method [11], [18]. Its speed is also notable under a constrained environment. For skin thresholding, active shape modeling (ASM) was used for comparison, and this made use of grey information and shape information for segmentation [8], [9]. The algorithms were implemented on an Android device using OpenCV (C++), NNAPI. The Android device was a Tecno Camon 12 with Android 9 OS, 8 MediaTek Helio P22 processor cores, and 4 GB of RAM.

A. Evaluation criteria

The system was tested in two stages. The first stage aimed to test the hand detection algorithm (MMNT) performance in complex backgrounds. The evaluation criteria used for testing the hand detection include the F-score and the average time. The second stage of testing was performed on the biometric system as a whole, and the system's performance was tested in terms of the system's accuracy.

1) Hand detection evaluation: The F-score metric was selected to evaluate the hand detection algorithm presented. F-score is the harmonic mean of precision P and recall R. The precision P is also known as the confidence or positive predictive value, and the recall R is otherwise called the sensitivity or true positive rate. Precision (P) and recall (R) are both expressed mathematically, as shown in (7) and (8). The higher the F-score, the better the method. F-score is given in (9).

$$P = \frac{TP}{TP+FP} \quad (7)$$

$$R = \frac{TP}{TP+FN} \quad (8)$$

$$F = 2\frac{PR}{P+R} \quad (9)$$

where TP, FP and FN are true positive, false positive and false negative respectively. True positive are those pixels that are labelled as the key-point that match the actual points. False positive are those pixels labelled as key-points but are not the actual key-points of the hand and false negative are the pixels not selected as the key-points but are actually key-points of

the hand in the image. A higher F-score corresponds to better segmentation.

B. Result

The results obtained from testing the system are stated below. The stated results are two. The first is the f-score and average time of the hand detection algorithm. The second result states the FAR, FRR, and accuracy of the system.

1) Hand detection result and comparison: 430 hand images were captured in real-time, and an attempt was made to segment the captured hands from their background. The performance of the modified Moore-Neighbour Tracing algorithm and other hand segmentation methods is shown in Table 2 and Table 3.

TABLE II
COMPARISON OF F-SCORE AND AVERAGE TIME FOR PROPOSED METHOD AND OTHER METHODS USING 640 BY 480 PIXELS

Segmentation method	Image size(pixel)	F-Score	Average time
Modified MNT	640 x 480	0.8657	0.9763
Skin-colour	640 x 480	0.7014	0.9685
Thresholding	640 x 480	0.4632	0.2987
Graph cuts	640 x 480	0.6854	12.5634

TABLE III
COMPARISON OF F-SCORE AND AVERAGE TIME FOR PROPOSED METHOD AND OTHER METHODS USING 3120 BY 4160 PIXELS

Segmentation method	Image size(pixel)	F-Score	Average time (s)
Modified MNT	3120 x 4160	0.9104	2.2456
Skin-colour	3120 x 4160	0.8218	2.1123
Thresholding	3120 x 4160	0.5102	0.9054
Graph cuts	3120 x 4160	0.6992	14.8743

Capturing of images was performed at different times of the day with the camera flash used for capturing at night (since this is a standard trait available on most mobile devices). The greater the F-score, the better the performance of the method. As observed from the image, the image's resolution difference had little impact on the method's performance. However, this added quite a bit to the processing time of the algorithm. The performance of the f-score showed a better hand segmentation for the modified MNT algorithm when compared with the other methods. The table also showed that an image with higher resolution would improve the hand's segmentation, though this will be at the cost of time and more processing resources.

2) Palmprint system result and comparison: To test the mobile palmprint recognition system, the mobile device's back camera was used to capture the hand of the enrolled users. There were 30 classes, 300 hands were captured, and the training and validation were 70: 30. The processing was performed on the device, and the system recorded a validation accuracy of 98.33%. The result was compared with the googleNet, AlexNet, and SqueezeNet models. Table 4 shows a comparison of the model in terms of accuracy.

TABLE IIV
COMPARISON OF ARCHITECTURE MODELS' PERFORMANCE

Model	Accuracy (%)
mobileNet	98.33
googleNet	97.33
SqueezeNet	96.33
alexNet	93.33

3) Comparison of system with other similar systems: To create an even platform for the comparison of palmprint systems, the popular online database was used to test the system. However, it should be noted that these databases have their hand images captured on an even background; hence, palm extraction does not involve a detailed method. Three databases were used for testing the system: the CASIA palmprint image database, the GPDS palm database and the PolyU palmprint database.

The CASIA database consists of 5502 palmprint images that were acquired from 312 people. Each individual has eight palmprint images (at least). Each palmprint image has a distinct position, scale, and posture because no pegs were used during capture. The background of the palm is uniform, and the illumination is distributed evenly [46]. The GPDS database has 1000 images from 100 subjects. 10 images of each recipient's right hand were captured. The placement of the hand is guided by the mask on the screen [47]. PolyU palmprint database is a palmprint database collected from 250 volunteers. It includes 195 males and 55 females. The database has 6000 palm images from 500 different palms [48]. Table 5,6,7, and 8 show the confusion matrix for the respective palm database. Table 9 shows a comparison of the accuracy of the system with other relevant systems. The accuracy in Table 9 was computed using equation (10).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

TABLE V
CONFUSION MATRIX FOR CASIA DATABASE

No = 5000	Predicted No	Predicted Yes	
Actual No	2474	26	2500
Actual Yes	74	2426	2500
	2548	2452	5000

TABLE VI
CONFUSION MATRIX FOR GPDS DATABASE

No = 1000	Predicted No	Predicted Yes	
Actual No	498	2	500
Actual Yes	8	492	500
	506	494	1000

TABLE VII
CONFUSION MATRIX FOR POLYU DATABASE

No = 5000	Predicted No	Predicted Yes	
Actual No	2488	12	2500
Actual Yes	38	2462	2500
	2526	2474	5000

TABLE VIII
CONFUSION MATRIX FOR OUR CUSTOM DATABASE

No = 1000	Predicted No	Predicted Yes	
Actual No	394	6	400
Actual Yes	11	589	600
	405	595	1000

TABLE IIIX
COMPARISON OF THE PERFORMANCE WITH SIMILAR SYSTEMS

Paper	Accuracy of Palmprint Databases (%)			
	CASIA	GPDS	PolyU	Custom Dataset
[49]	-	-	-	95.4
[27]	94.73	98.50	-	-
[28]	-	-	98.5	
[29]	94.91	-	95.88	-
Our Method	98.0	99.0	99.0	98.3

IV. CONCLUSION

This research work presented a palmprint recognition system for mobile devices (Android). A hand segmentation method for hand detection in complex background and uneven lighting scenarios was examined. The proposed method achieved a good performance in hand segmentation and biometric authentication in a completely unconstrained hand-capturing environment. To evaluate the hand segmentation method, hand images where captured in real-time using a mobile device at different times of the day and with different backgrounds. The system's performance is compared with other hand segmentation methods in Table 1, and the result showed an improved performance.

Also, the method's time cost is less than the other methods. The performance of the hand detection algorithm showed its ability to detect the key-points of hand images in complex background. This makes it suitable for hand-based (unimodal and multimodal) biometric systems on mobile and non-mobile platforms. Hence, this segmentation method suits hand-based biometrics on mobile and non-mobile platforms. The palmprint authentication system also produced a validation accuracy of 98.3%, with 30 users enrolled in the system. The best accuracy of 99.0% was obtained for the publicly available GPDS and PolyU palmprint database. This is an encouraging result as typical enrolment on mobile platforms will usually be two (both palms of the mobile device's owner).

ACKNOWLEDGMENT

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] D. Izergin and M. Eremeev, "Risk assessment model of compromising personal data on mobile devices," *E3S Web Conf.*, vol. 270, 2021, doi: 10.1051/e3sconf/202127001013.

[2] S. Sharma, R. Kumar, and C. R. Krishna, "A survey on analysis and detection of Android ransomware," *Concurr. Comput. Pract. Exp.*, doi: 10.1002/cpe.6272.

[3] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *IEEE 4th Int. Conf. Biometrics Theory, Appl. Syst. BTAS 2010*, 2010, pp. 1–7.

[4] L. Long, "Biometrics : The Future of Mobile Phones," End User Comput. Serv.," *End User Comput. Serv.*, pp. 1–5, 2013.

[5] R. Ryu, S. Yeom, S.-H. Kim, and D. Herbert, "Continuous Multimodal Biometric Authentication Schemes: A Systematic Review," *IEEE Access*, vol. 9, pp. 34541–34557, 2021, doi: 10.1109/ACCESS.2021.3061589.

[6] A. Giełczyk, M. Choras, and R. Kozik, "Lightweight verification schema for image-based palmprint biometric systems," *Mob. Inf. Syst.*, vol. 2019, 2019, doi: 10.1155/2019/2325891.

[7] U. Saeed, K. Masood, and H. Dawood, "Illumination normalization techniques for makeup-invariant face recognition," *Comput. Electr. Eng.*, vol. 89, 2021, doi: 10.1016/j.compeleceng.2020.106921.

[8] G. Bailadoar, B. Ríos-Sánchez, R. Sánchez-Reillo, H. Ishikawa, and C. Sánchez-Ávila, "Flooding-based segmentation for contactless hand biometrics oriented to mobile devices," *IET Biometrics*, vol. 7, no. 5, pp. 431–438, 2018, doi: 10.1049/iet-bmt.2017.0166.

[9] F. Gao, K. Cao, L. Leng, and Y. Yuan, "Mobile Palmprint Segmentation Based on Improved Active Shape Model," *J. Multimed. Inf. Syst.*, vol. 5, no. 4, pp. 221–228, 2018.

[10] L. Leng, F. Gao, Q. Chen, and C. Kim, "Palmprint recognition system on mobile devices with double-line-single-point assistance," *Pers. Ubiquitous Comput.*, vol. 22, no. 1, pp. 93–104, 2018, doi: 10.1007/s00779-017-1105-2.

[11] N. M. S. Hussein, S. M. Hammed, B. Ergen, and P. G. Student, "Biometric Identification System based on Hand Geometry," *Int. J. Innov. Res. Sci.*, vol. 6, no. 3, pp. 3159–3166, 2017, doi: 10.15680/IJIRSET.2017.0603005.

[12] M. Akmal-Jahan, B. Jasmine, and V. Chandran, "Minutiae-Triangle-Graph : An invariant feature representation for fingerprints and palmprints in hand biometrics," *Springer*, 2020.

[13] G. Jaswal, A. Kaul, and R. Nath, *Multimodal Biometric Recognition System Using Hand Shape , Palm Print , and Hand Geometry*, vol. II. Springer Singapore, 2019.

[14] W. M. Matkowski, T. Chai, and A. W. K. Kong, "Palmprint Recognition in Uncontrolled and Uncooperative Environment," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1601–1615, 2020, doi: 10.1109/TIFS.2019.2945183.

[15] A. Ungureanu, S. Salahuddin, and A. Member, "Toward Unconstrained Palmprint Recognition on Consumer Devices : A Literature Review," *IEEE Access*, vol. 8, pp. 86130–86148, 2020, doi: 10.1109/ACCESS.2020.2992219.

[16] M. M. H. Ali, V. H. Mahale, P. L. Yannawar, and A. T. Gaikwad, "A Review: Palmprint Recognition Process and Techniques," *Int. J. Appl. Eng. Res.*, vol. 13, no. 10, pp. 7499–7507, 2018.

[17] C. Naveena, R. Shreyas, and K. Chethan, "Texture Features in Palmprint Recognition System," *Int. J. Nat. Comput. Res.*, vol. 10, pp. 41–57, doi: 10.4018/IJNCR.2021010104.

[18] M. Aguado-Martínez, J. Hernández-Palancar, K. Castillo-Rosado, and E. Al., "Document scanners for minutiae-based palmprint recognition: a feasibility study," *Pattern Anal. Appl.*, vol. 24, pp. 459–472, 2021, doi: 10.1007/s10044-020-00923-3.

[19] M. P.Dale, M. A. Joshi, and H. J. Galiyawala, "A Single Sensor Hand Geometry and Palm Texture Fusion for Person Identification," *Int. J. Comput. Appl.*, vol. 42, no. 7, pp. 11–16, 2012, doi: 10.5120/5703-7726.

[20] S. Aoyama, K. Ito, and T. Aoki, "A Contactless Palmprint Recognition Algorithm for Mobile Phones," *Int. Work. Adv. Image Technol.*, pp. 409–413, 2013.

[21] T. Wu and L. Leng, "Video Palmprint Recognition System Based on Modified Double-line-single-point Assisted Placement," *J. Multimed. Inf. Syst.*, vol. 8, no. 1, pp. 20–30, 2021.

[22] T. O. Oladele, K. Adeniyi, and T. O. Aro, "Framework for User Authentication at a Distance for Mobile Phones Using Contactless Hand-based Multimodal Biometric System," *J. Comput. Sci. Control Syst.*, vol. 12, no. 1, pp. 24–27, 2019.

[23] L. Fei, G. Lu, W. Jia, S. Teng, and D. Zhang, "Feature Extraction Methods for Palmprint Recognition: A Survey and Evaluation," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 2, pp. 346–363, 2019, doi: 10.1109/TSMC.2018.2795609.

[24] S. Verma and S. Chandran, "Contactless Palmprint Verification System using 2-D Gabor Filter and Principal Component Analysis," *Int. Arab J. Inf. Technol.*, vol. 16, no. 1, 2019.

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

[26] P. Poonia, P. K. Ajmera, and V. Shende, "ScienceDirect ScienceDirect Palmprint Recognition using Robust Template Matching Palmprint Recognition using Robust Template Matching," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 727–736, 2020, doi: 10.1016/j.procs.2020.03.338.

[27] N. Xu, Q. Zhu, X. Xu, and D. Zhang, "An effective recognition approach for contactless palmprint," *Vis. Comput.*, 2020, doi: 10.1007/s00371-020-01962-x.

[28] J. P. Patil and C. S. Pawar, "Palmprint based Pattern Recognition Using Fast ICA," *IEE Xplore*, no. ICICCS, pp. 566–569, 2020.

[29] X. Zhou, K. Zhou, and L. Shen, "Rotation and Translation Invariant Palmprint Recognition With Biologically Inspired Transform," *IEEE Access*, vol. 8, pp. 80097–80119, 2020, doi: 10.1109/ACCESS.2020.2990736.

[30] P. Kavipriya, M. R. Ebenezar-Jebarani, T. Vino, and G. Jegan, "Ear biometric for personal identification using canny edge detection algorithm and contour tracking method," 2021, doi: 10.1016/j.matpr.2021.03.351.

[31] S. Kumar, A. K. Upadhyay, P. Dubey, and S. Varshney, "Comparative analysis for Edge Detection Techniques," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021, pp. 675–681, doi: 10.1109/ICCCIS51004.2021.9397225.

[32] S. D. Lokmanwar and A. S. Bhalchandra, "Contour detection based on gaussian filter," doi: 10.1109/iceca.2019.8822189.

[33] D. N. Lohare, R. R. Manza, and N. Tiwari, "Comparative Study of Prewitt and Canny Edge Detector Using Image Processing Techniques," 2021, doi: 10.1007/978-981-15-6014-9_86.

[34] A. Kumar and S. S. Sodhi, "Comparative Analysis of Gaussian Filter, Median Filter and Denoise Autoenocoder," in *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2020, pp. 45–51, doi: 10.23919/INDIACom49435.2020.9083712.

[35] S. Vijayarani and A. Sakila, "Face Recognition based Student Attendance System," *Int. J. Res. Publ. Rev.*, vol. 2, no. 4, pp. 289–299, 2020.

[36] A. S. Ahmed, "Comparative Study Among Sobel, Prewitt and Canny Edge Detection Operators used in Image Processing," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 19, 2018.

[37] T. H. Mandee, M. I. Ahmad, and M. N. M. Isa, "Palmprint Region of Interest Cropping Based on Moore-Neighbor Tracing Algorithm," *Sens. Imaging*, vol. 19, p. 15, 2018, doi: 10.1007/s11220-018-0199-6.

[38] R. Priyadharsini and T. S. Sharmila, "Object Detection In Underwater Acoustic Images Using Edge Based Segmentation Method," *Procedia Comput. Sci.*, vol. 165, pp. 759–765, 2019, doi: 10.1016/j.procs.2020.01.015.

[39] I. Ullah, M. S. Azmi, M. I. Desa, and Y. M. Alomari, "Segmentation of Touching Arabic Characters in Handwritten Documents by Overlapping Set Theory and Contour Tracing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, 2019.

[40] S. S. Mansouri, M. Castaño, C. Kanellakis, and G. Nikolakopoulos, "Autonomous MAV Navigation in Underground Mines Using Darkness Contours Detection," in *In: Tzovaras D., Giakoumis D., Vincze M., Argyros A. (eds) Computer Vision Systems. ICVS 2019. Lecture Notes in Computer Science*, 2019, vol. 11754, doi: 10.1007/978-3-030-34995-0_16.

[41] S. Sadhukhan, N. Upadhyay, and P. Chakraborty, "Breast Cancer Diagnosis Using Image Processing and Machine Learning," 2020, doi: 10.1007/978-981-13-7403-6_12.

[42] T. Matić, I. Aleksi, Ž. Hocenski, and D. Kraus, "Real-time biscuit tile image segmentation method based on edge detection," *ISA Trans.*, vol. 76, pp. 246–254, 2018.

[43] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A Novel Image Classification Approach via Dense-MobileNet Models," *Mob. Inf. Syst.*, 2020, doi: https://doi.org/10.1155/2020/7602384.

[44] P. Liu, X. Li, H. Cui, S. Li, and Y. Yuan, "Hand Gesture Recognition Based on Single-Shot Multibox Detector Deep Learning," vol. 2019, pp. 25–28, 2019.

[45] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications Andrew," *arXiv*, 2017.

[46] "CASIA Palmprint Database." [Online]. Available: http://www.cbsr.ia.ac.cn/english/Palmprint%20Databases.asp. [Accessed: 22-Feb-2023].

[47] "GPDS Hand Database." [Online]. Available: . https://gpds.ulpgc.es/downloadnew/download.htm. [Accessed: 22-Feb-2023].

[48] "PolyU Multispectral Palmprint Database." [Online]. Available: . http://www4.comp.polyu.edu.hk/~csajaykr/database.php. [Accessed: 22-Feb-2023].

[49] P. Poonia, P. K. Ajmera, and V. Shende, "Palmprint Recognition using Robust Template Matching Palmprint Recognition using Robust Template Matching," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 727–736, 2020, doi: 10.1016/j.procs.2020.03.338.