

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/356761255>

Design and Implementation of a Face Recognition System Based on API mobile vision and Normalized Features of Still Images

Article in *Procedia Computer Science* · December 2021

DOI: 10.1016/j.procs.2021.10.057

CITATIONS

4

READS

133

2 authors, including:



Kamel H Rahouma

Minia University

81 PUBLICATIONS 160 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Machine learning and prediction Techniques for Biomedical signal (Prediction behavior technique) [View project](#)



using RF antenna to kill fat tissues [View project](#)

18th International Learning & Technology Conference 2021

Design and Implementation of a Face Recognition System Based on API mobile vision and Normalized Features of Still Images

Kamel Hussein Rahouma ^a, Amal Zarif Mahfouz ^b*a : Department of Electrical Engineering Faculty of Engineering, Minia University, Egypt Vice Dean of the Faculty of Computers, Nahad University in Beni Suef, Egypt.**b : Department of Electrical Engineering Higher Institute of Engineering and Technology, New Minia, Egypt .*

Abstract

Biometric identification is the automated technique of measuring the biological data. The term biometrics is commonly used today to recognize a person by analyzing his/her physical characteristics and comparing these characteristics to a database such as fingerprint recognition, retinal scanning, face recognition etc. Face recognition is one of the biometric techniques that are used for identifying and verifying the identity of a person. This paper presents the design and implementation of a face recognition system for Android mobile phone platform. The design methodology includes two main steps. The first step is the extraction of the image's features and the second one is the recognition according to the classification of patterns. This system includes face detection using face detector available in the Android called object detection (Google's API mobile vision), features extraction (nose detection, mouth detection, eyes detection and cheek detection) using the same algorithm object detection API. From the feature extraction, we have 30 geometrical measurements of each person. The recognition step is done by calculating the Pearson Correlation Coefficients between the test image's geometrical measurements and measurements stored in the training database. The system will accept a facial image of the person as authenticated and allow to access the mobile applications if the percentage of correlation is greater than a chosen threshold otherwise it rejects the image. The recognition rate of the proposed approach has achieved an accuracy more than 95% of other approaches.

© 2021 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the 18th International Learning & Technology Conference 2021

Keywords: Biometrics, Face recognition, Face detection, Correlation Coefficient, Recognition rate, Mobile API vision, Normalized Features.

E-mail address: kamel_rahouma@yahoo.com ^a*E-mail address:* amal_nany@yahoo.com ^b

1. Introduction

Biometrics as a terminology refers to the combination of the Greece words (Bio) and (Metrics) which means “life measurements” [1]. It becomes highly important in computer security world [2]. The goal of a biometric is either automatic identification or verification of people identities. It means that the input data (images, speech, or videos) is given to the system and it compares this input to a database [3]. A biometric may be physiological or behavioral. Physiological biometric detects the physical features of a person and uses resources such as (retina, iris scans, fingerprints, and face recognition). Behavioral biometric is based on a user’s behavior and includes the analysis of information like the shape and flow of one’s handwriting, timing of keystrokes [4]. Face Recognition is used to identify the identity of a person from his/her input image. Face recognition adapts for application to mobile devices because of their flexibility. The main problems of face recognition on mobile devices are because of the constraints on processing power, limited storage of mobile devices [5], limited network bandwidth, privacy, and security issues [6]. Face recognition on mobile devices can be used for authentication of the user, distinguishing between a person and another person to identify social networking sites. It is widely used in the security [7-9] and marketing [10].

There are many traditional security applications such as username applications (based on identity) and password applications (credentials) [11-13]. Mobile devices with the traditional applications are easily stolen as password can be hacked or detected. However, face recognition applications are more efficient than traditional applications because they are more secure and more flexible and the user doesn't need to remember the passwords [14]. Face recognition applications are the most suitable biometrics since those devices have cameras. The objective of this paper is to present a face recognition system that applies on mobile devices. The proposed solution uses three steps.

The first step is the face detection using the API Google's Mobile vision. The process of face detection is done to decide which pixels in the image give a part of the face and which do not. It is based on achieving a high face detection rate in the smallest amount of time. The second step is the process of feature extraction to identify the main landmarks of each person. These landmarks are such as eye, mouth, nose and cheek. From the extracted features, we normalized the distance between landmarks to obtain 30 measurements for each person. These normalized distances are stored in an SQL database. The last step is the recognition step. The normalized distances are compared with the normalized distances of each face present in the database using Pearson Correlation.

The rest of this paper is organized as follows: Section 2 is a literature review which presents the previous related work in face recognition. Section 3 presents how we create the database to use in the recognition step and it explains a real illustrative example in each step. Section 4 illustrates the correlation recognition method used in the system. Section 5 explains the overall system and introduces its pseudo code as well as the complete flow chart of the face recognition system on Mobile devices. Section 6 presents some experimental results of the used algorithm for our face recognition system. Section 7 discusses the results and compares them with the other approaches of face recognition systems on mobile devices. We sum up in conclusions and introduce some of the future work points in section 8. Finally, a list of the used references is given at the end of the paper.

2. A literature Review

There are many different recognition methods and we here summarize some of them. Authors in [6] proposed a face recognition system on mobile phones based on the input image, by using Viola-Jones algorithm, applying a set of preprocessing for ROI, and then using principal component analysis (PCA). The goal of the system is to save time and give an easier access for searching pictures. Authors in [15] presented a new application for face recognition on mobile devices based on Bridge Approach (BA) to enhance the speed up the processing and it allows using the system in any location where an internet is available. Face detection and feature extraction are implemented using OpenCV library, but the face recognition is performed in WEKA library. Authors in [16] offered a design and implementation of a face recognition system for the mobile phone platform by including face and eye detection and applying a set of preprocessing for ROI. The recognition used methods are the Principal Component Analysis (PCA) that achieves an accuracy of 93.8% and the Linear Discriminant Analysis (LDA) that achieves an accuracy of 96%. Authors in [5] made deployment face recognition on mobile devices. They used the PCA algorithm for recognition. After applying the system, the accuracy was approximately 92% for a chosen threshold and the time taken to recognize the person is 0.35 sec for a small number of testing images. Authors in [14] implemented a face recognition system in MATLAB and then simulated it on DROID phones. It investigated multi algorithms for face detection (color segmentation,

template matching) and proposed two methods for recognition (Eign & Fisher face). It achieved a recognition rate of Eign face about 84.3%, fisher face 94% and the computation time on DROID was 1.58 sec.

3. Database Construction

Face recognition on Mobile device consists of two main parts. The first part is the face detection to create database. The second part is the face recognition in which we use the database to decide if the person is authorized or not. Face detection is the process of estimation of the location, scale of the face and localizing facial landmarks (e.g., eyes, nose, mouth, and cheek). Face detection be a task of distinguishing the face and non-face parts in the image. Face recognition is a task of distinguishing faces of different individuals according to the database that is stored in the system. Main steps to create database are illustrated in flow chart in Figure. 1

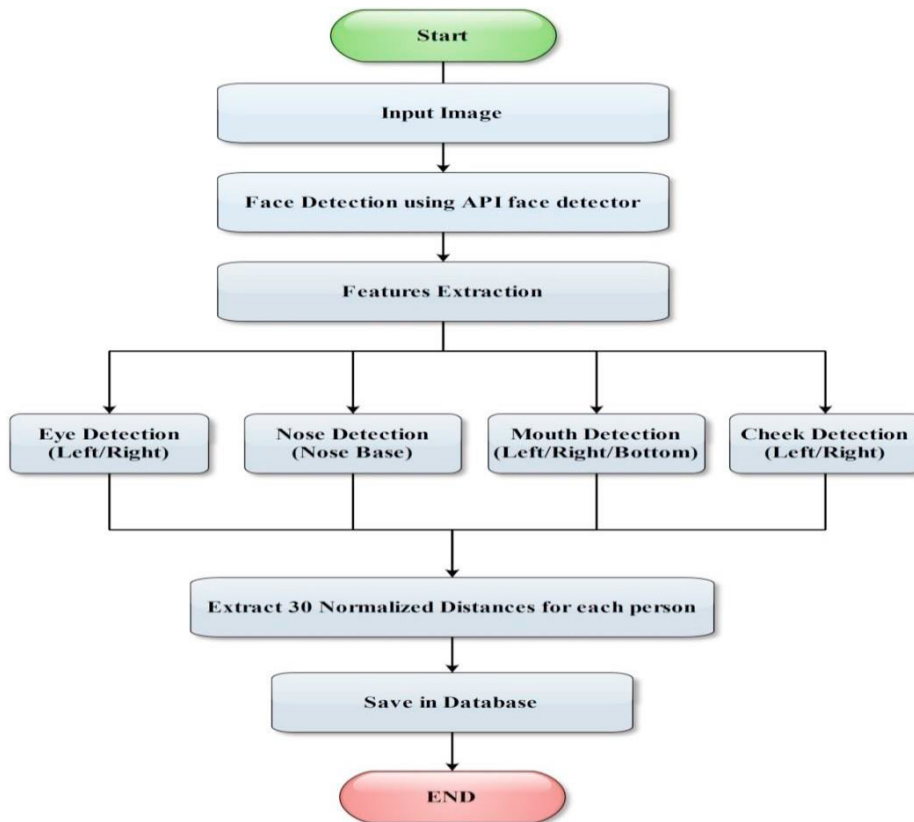


Figure 1 Database Construction

3.1 Input Image

This step aims to take an image of the person we want to detect his/her face. This input image is stored in the device in any dimension such as 1024x862 or 300x400 or 709x628

3.2 Face Detection

Face detection is the process of automatically locating human faces in visual media such as (digital images or video). A face that is detected is reported at a position with an associated size and orientation. Once a face is detected, it can be searched for landmarks such as the eyes and nose. To make face detection on mobile devices, there are two ways: the first one is to use the OpenCV library (Computer Vision). This library includes different face

detection algorithms such as Viola-Jones algorithm and other algorithms. The second way is to use the built-in face detection algorithm in the mobile device such as the object detection (API Google's Mobile vision) and feature based method (ML kit machine learning). It is easier to use the built-in algorithms, instead of using bridge method (OpenCV). The built in algorithms enable us to achieve higher rates and more accurate face detection compared to the other algorithms. We use the object detection API Google's Mobile vision face detector. The object detection API has the facial landmarks (e.g., eyes, nose, mouth, and cheek) in a photo and it draws graphics over the face to indicate the positions of the detected landmarks. Main advantages of object detection API include:

- (1) Distinguishing faces and non-faces when the process of face detection as shown in Figure.2
- (2) Detecting multiple faces in the same photo as Figure.3
- (3) Saving time because we do not need extra operations such cropping images.
- (4) Fulfilling an amazing accuracy rate because it uses the Mobile Vision API.



Figure 2: Difference between face and non-face

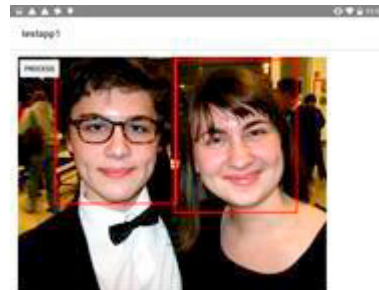


Figure.3: Multiple face detection in the same image

The operations of the API face detection are explained as follows:

- (a) Creating the face detector:

The face detector is created using the Create method of the app's main activity. It is initialized with options for detecting faces with landmarks in a photo:

```
FaceDetector detector = new FaceDetector.Builder(context).
    setTrackingEnabled(false).
    setLandmarkType(FaceDetector.ALL_LANDMARKS)
    .build();
```

- (b) Detecting faces:

The detector can be synchronized by a frame to detect faces. The result returned includes a collection of Face instances according to the dimensions of each face (x_1 , y_1 , x_2 , y_2)

```
Frame frame = new Frame.Builder().setBitmap(bitmap).build();
SparseArray<Face> faces = detector.detect(frame);
for (int i = 0; i < faces.size(); ++i) {
    Face face = faces.valueAt(i);
    float x1=face.getPosition().x;
    float y1=face.getPosition().y;
    float x2=x1+face.getWidth();
    float y2=y1+face.getHeight();
}
}
```

- (c) Querying the face detector's operational status:

A detector is an operational method used to check if the required native library is currently available or not:

```
if (!detector.isOperational()) { // ...
}
```

(d) Releasing the face detector:

After the face detection, it is necessary to release the detector instance once it is no longer needed.

```
detector.release();
```

3.3 Facial Features Extraction

This is the process of extracting facial landmarks. A landmark is a point of interest within a face such as the left eye, right eye, and nose base. Figure 4 shows some examples of facial landmarks:



Figure 4 A sample of facial landmarks

At first, system is detecting landmarks and using the landmarks as a basis of detecting the whole face. The Face API detects the whole face independently of detailed landmark information. For this reason, landmark detection is an optional step that could be done after the face is detected. Landmark detection is not done by default since it takes additional time to run. You can optionally specify that landmark detection should be done. Facial landmarks can be detected from the code:

```
for (int i = 0; i < faces.size(); ++i) {
    Face face = faces.valueAt(i);
    for (Landmark landmark : face.getLandmarks()) {
        int cx = (int) (landmark.getPosition().x * scale);
        int cy = (int) (landmark.getPosition().y * scale);
        canvas.drawCircle(cx, cy, 10, paint);
    }
}
```

It detects all landmarks with the position of (cx, cy). The detected landmarks are shown in the Figure. 5. The detected landmarks are defined in Table 1.

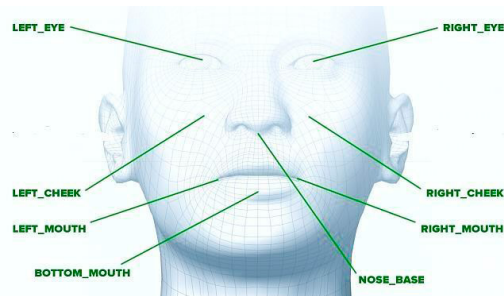


Figure 5 Facial Landmarks

The pixel of the center of the right eye is taken as (x_1, y_1) , the pixel of the center of the left eye is taken as (x_2, y_2) , the pixel of the center of the left cheek is taken as (x_3, y_3) , the pixel of the center of the right cheek is taken as (x_4, y_4) , the pixel of the bottom mouth is taken as (x_5, y_5) , the pixel of left mouth is taken as (x_6, y_6) , the pixel of the right mouth is taken as (x_7, y_7) and the pixel of the center of the nose is taken as (x_8, y_8) . We calculate the distances to obtain the 30 measurements. These measurements are in Table 1.

TABLE.1. DEFINITION OF FACIAL LANDMARKS

Facial Landmark	Definition
BOTTOM_MOUTH	The center of the subject's bottom lip.
LEFT_CHEEK	The midpoint between the subject's left mouth corner and the outer corner of the subject's left eye.
LEFT_EYE	The center of the subject's left eye cavity.
LEFT_MOUTH	The subject's left mouth corner where the lips meet.
NOSE_BASE	The midpoint between the subject's nostrils where the nose meets the face
RIGHT_CHEEK	The midpoint between the subject's right mouth corner and the outer corner of the subject's right eye.
RIGHT_EYE	The center of the subject's right eye cavity.
RIGHT_MOUTH	The subject's right mouth corner where the lips meet

3.4 Geometrical feature measurements

These measurements are computed based on the facial parts' locations as follows: Assume the pixel of the center of the right eye is taken as (x_1, y_1) , the pixel of the center of the left eye is taken as (x_2, y_2) , the pixel of the center of the left cheek is taken as (x_3, y_3) , the pixel of the center of the right cheek is taken as (x_4, y_4) , the pixel of the bottom mouth is taken as (x_5, y_5) , the pixel of left mouth is taken as (x_6, y_6) , the pixel of the right mouth is taken as (x_7, y_7) and the pixel of the center of the nose is taken as (x_8, y_8) . The normalized distance between these points is calculated by distance calculation algorithm in the formula in Eq (1) [17]:

$$\text{Distance} = \sqrt{((x_i - x_j)^2) + ((y_i - y_j)^2)} \quad (1)$$

Applying the previous formula (for $i, j = 1, 2, 3, \dots, 8$), there are 30 distances obtained as:

1. Distance between left eye and right eye (d_1)
2. Distance between right eye and right cheek (d_2)
3. Distance between right eye and nose base (d_3)
4. Distance between left eye and left cheek (d_4)
5. Distance between left eye and nose base (d_5)
6. Distance between right cheek and nose base (d_6)
7. Distance between left cheek and nose base (d_7)
8. Distance between right cheek and right mouth (d_8)
9. Distance between right cheek and bottom mouth (d_9)
10. Distance between nose base and right mouth (d_{10})
11. Distance between nose base and bottom mouth (d_{11})
12. Distance between nose base and left mouth (d_{12})
13. Distance between right mouth and bottom mouth (d_{13})
14. Distance between bottom mouth and left mouth (d_{14})
15. Distance between right mouth and left mouth (d_{15})
16. Distance between left cheek and left mouth (d_{16})
17. Distance between left cheek and bottom mouth (d_{17})
18. Distance between left cheek and right cheek (d_{18})
19. Face width (d_{19})
20. Face height (d_{20}).
21. Distance between right eye and left mouth (d_{21})

22. Distance between right eye to left cheek (d_{22})
23. Distance between left eye to right cheek (d_{23})
24. Distance between left eye to right mouth (d_{24})
25. Distance between left eye to bottom mouth (d_{25})
26. Distance between left eye to left mouth (d_{26})
27. Distance between right eye to right mouth (d_{27})
28. Distance between right cheek to left mouth (d_{28})
29. Distance between left cheek to right mouth (d_{29})
30. Distance between right eye to bottom mouth (d_{30})

There are 30 measurements that can be computed for each person. These values are stored in a (SQL database) to create a database for the authorized persons. In the recognition step, we use these measurements to know if the person is authorized or not. After obtaining previous pixels, from (x_1, y_1) to (x_8, y_8) we apply the 30 normalized distance calculation algorithms. These measurements are denoted as $D_1, D_2, D_3, D_4, D_5, \dots, D_{29}, D_{30}$.

3.5 Save the database

This database is done using android studio SQLite database. It is used as a reference to compare between persons to know if the person is an authorized user in the system or not. Database is an intermediate stage between enrollment and verification stages. In this paper, we use scfaceDB (surveillance cameras face detection Database) landmark. This database is based on extracting a number of facial landmarks using the different facial regions extractor depending on the distance between the subject and acquisition camera. The database can include up to 4160 faces which is more than enough for authorized users accessing the mobile device. The authorized person is limited. Database consists of n rows and 31 columns. The n rows are the number of authorized persons that access the system, and the 30 measurements is the geometrical feature in addition to the ID of each person. It can be updated to include as many persons as needed. Each person needs to go through the steps of processing until we reach the final geometrical distances of each person.

4. Face Recognition

This is the last step in our proposed system, and it is performed using Pearson Correlation Coefficient. Pearson correlation quantifies the strength of a linear relationship between two variables. In this paper, we use the Pearson correlation to identify the covariance between the person's features and authorized persons' features stored in the database to be able to access the mobile device. If the covariance ratio is equal to or greater than a certain threshold, the system will accept the person as an authorized one and otherwise the system will deny the access process. The MATLAB has its own built in Pearson correlation function which is used in this paper. However, the Pearson correlation function between two vectors (x, y) is given by eq. (2):

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (2)$$

Where n is number of "pairs" of data we have. n is bivariate sample $n \geq 2$. Higher values of the correlation coefficient between the two images A and B means they are highly related to each other. According to the covariance rate we decided if the person is known or not. If the covariance ratio is equal to or greater than a certain threshold, the system will accept the person as an authorized one and otherwise the system will deny the access process. It is comparing the 30 measurements of the person who wants to access the system with the (SQLite) database which contains the measurements of the authorized people. The relation between the training and testing parts is shown in Figure 6.

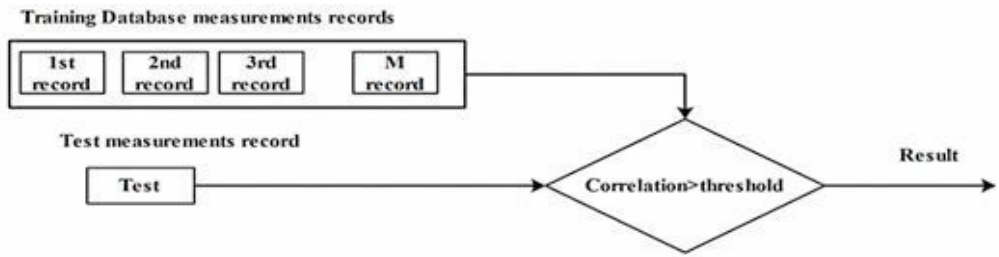


Figure.6 Training and testing parts for face recognition system [18]

5. Proposed System and its algorithm

In this section, we discuss the design of the face recognition system on Mobile device and its implementation and application in daily life as shown in Figure 7. Then, we introduce the algorithm and pseudo code of the different stages of the system such as image acquisition, training phase, testing phase, and deciding the person authority. Finally, we will give the system flow chart.

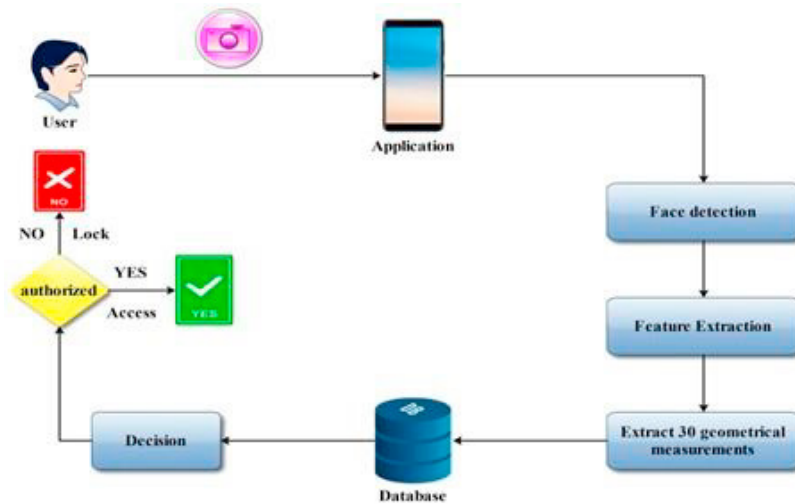


Figure.7 The proposed face recognition System

1) Image Acquisition

For the process of image acquisition: Input Image of a person. This image stored in your device.

2) The Training Phase

1. FOR each input image: $p = \{x_1, x_2, x_3, \dots, x_k\}$ the system:
 - a. Detects the face in the image.
 - b. Extracts the facial features such as (nose, mouth, eye, and cheek).
 - c. Computes the normalized distances between points.
 - d. Obtains the 30 normalized geometrical measurements.
- END FOR

Add all the normalized distances in the database to create a general database for the authorized people. When the number of authorized people is complete, the device will be ready to check up any person wants to use it by executing the testing phase.

3) The Testing Phase

FOR each input image: $p' = \{y_1, y_2, y_3, \dots, y_k\}$ the system:

- a. Detects the face in the image.
- b. Extracts the facial features such as (nose, mouth, eye, and cheek).
- c. Computes the normalized distances between two points.
- d. Obtain the 30 normalized geometrical measurements.
- e. Computes the correlation coefficients between the 30 normalized distance of the person to the 30 geometrical distances of the authorized persons to know if the person is authorized or not.

END FOR

4) The Recognition Stage

If (correlation > threshold)

- This person is authorized.
- Allow the person to access the android application and close the application automatically.

Else

- This person is not authorized.
- Do not close the application waiting for another person to try.

END IF

The last stages are given in a flow chart as in Figure 8.

6. Experimental results

In this section, we discuss and analyze some results of the face recognition system on Mobile device. These systems can differentiate between the authorized and unauthorized users. The main steps of face recognition system on mobile device are illustrated as follows:

6.1 Input Image

There is no specific resolution for the input image where the default device setup is enough. Example is shown in Figure.9

6.2 Face Detection

This detection is done by using object detection (Google's API mobile vision). The result of face detection for a real example is illustrated in Figure 10.

6.3 Feature Extraction

The second step is the features extraction. These Features are illustrated in Figure 11

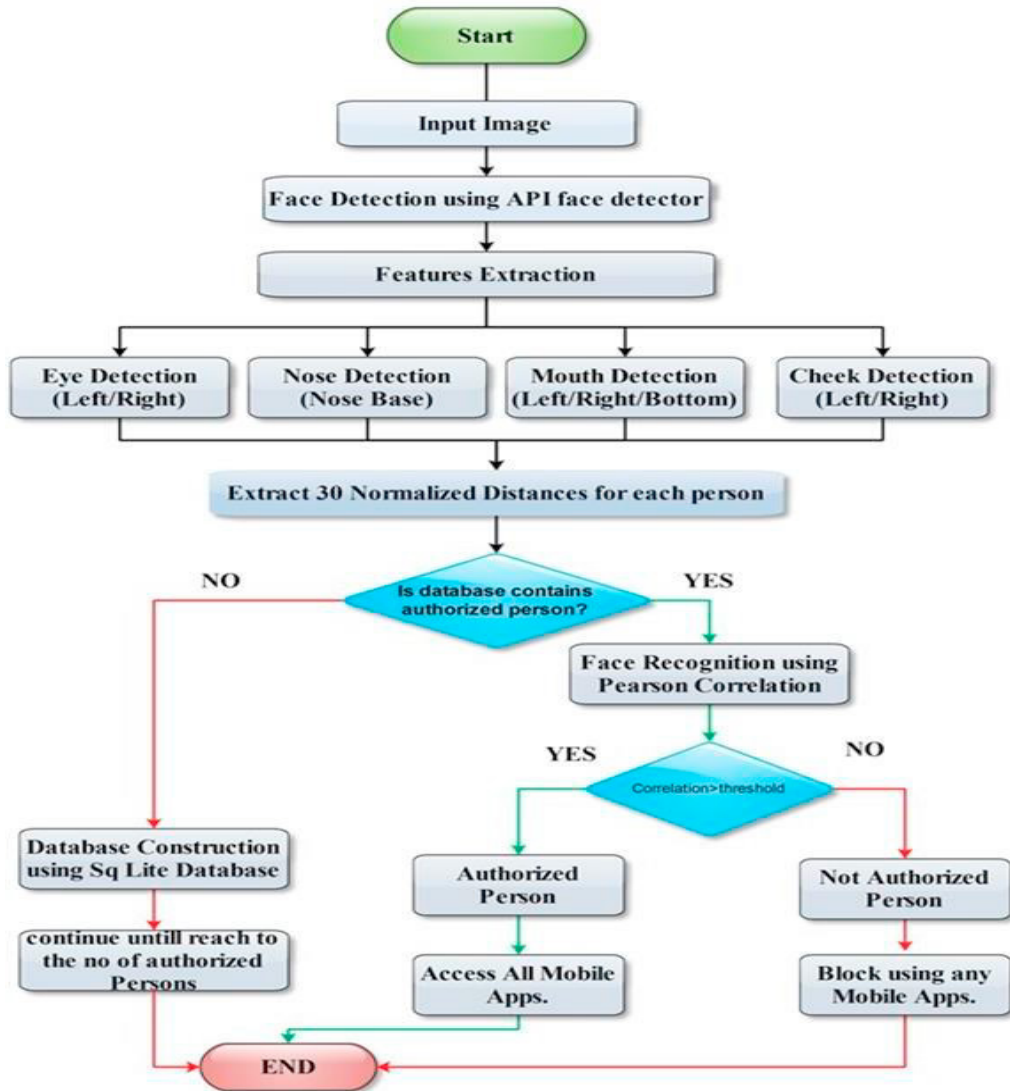


Figure.8 Flow chart of face recognition on mobile device



Figure 9 Input Image

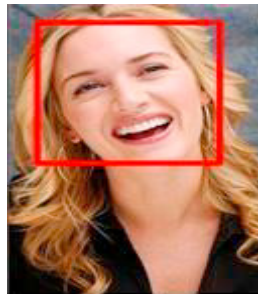


Figure 10 Face detection

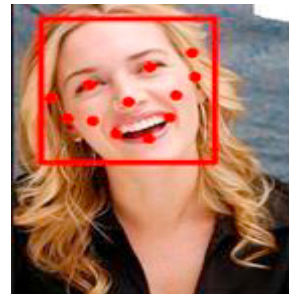


Figure.11 Features Extraction

6.4 Recognition Method

After obtaining the 30 geometrical distance of the person who wants to access the system, these distances are normalized and compared with all distances in the training database. This comparison is performed using Pearson correlation. A person can access the mobile device if the correlation coefficient is greater than or equal to a chosen threshold otherwise, the mobile will deny the access locking and blocking all the mobile applications. The performance of a face recognition system can be evaluated by computing its recognition rate. Recognition rate is calculated from the Eq (3):

$$\text{Recognition Rate} = \frac{\text{number of correctly recognized person}}{\text{total numbers of perons}} * 100 \quad (3)$$

The Recognition Rate of our system is found to be 96.5%. The recognition rate is not based any parameter such as

- i. The effects of odd face' expressions, face occlusion, illumination lighting, less resolution and skin color can be treated by the region of interest (ROI) preprocessing. However, we tested the system with and without the preprocessing stage. We found that the difference between the detection rates of the two cases is about 2.9%. However, the resulting complexity, slow speed and time consuming of the preprocessing make it preferable to ignore it.
- ii. For the complex background, we tested the system with different backgrounds. These backgrounds do not affect the recognition rate nor affect the geometrical features of each person.
- iii. For more than one face in the image, it is considered an abnormal situation because no two people will use their faces to access the mobile.
- iv. For the abnormal face images (images with twisted heads or not clear or not taken with the frontal camera), they are not accepted because the user is supposed to be in front of the frontal camera and near to the mobile to access it. Even if the user is a bit far from the mobile, the whole geometrical features will be affected in the same way. Thus, normalizing these features is not affected by being far from or near to the mobile.

7. Discussion

Our proposed system has some good issues such as follows:

1) Comparing our system and other ones in literature, our system gives higher Recognition Rates. Table 2 gives the Recognition Rates for some approaches on Mobile systems.

TABLE.2. RECOGNITION METHODS AND RECOGNITION RATE WITH ITS APPROACHES

Approaches Reference	Recognition Method	Recognition rate (%)
[15]	WEKA library	80%
[16]	PCA	93.8%
	LDA	96%
[5]	PCA	92%
[14]	Eigen Faces	84.3%
	Fisher Faces	94%
[19]	Correlation coefficient for the 1 st system	94%
	Correlation coefficient for the 2 nd system	93.33%
Our work	Pearson Correlation	96.5%

2) In the first stage, there is no need to make a ROI preprocessing because the geometrical features are not affected by such preprocessing operation. Table 3 shows the small differences between the computed features of the face before and after preprocessing. The maximum difference between the two sets is found to be 2.9%. By ignoring this difference, we save time consumed in the complex calculations and still have a high detection rate.

TABLE 3. NORMALIZED FEATURES BEFORE AND AFTER THE PREPROCESSING

Comparison	After Processing	Before Processing
Input Image		
D1	200.26232796	200.86064
D2	149.4121815649	148.03042282
D3	133.7647188162	137.767920794
D4	272.090058620	272.029410747
D5	271.974263488	275.39244725
D6	153.521985396	153.521985396

3) Applying the approach on Android Studio has a higher performance than on MATLAB. A complete comparison between face recognition implementations using MATLAB [19] and face recognition using Android studio is shown in Table 4.

4) From the previous comparison, Android studio has a higher performance than MATLAB. This may be explained because of the following reasons:

1. Mobile device is easier in taking images and storing them regardless the directions and dimensions.
2. No preprocessing is needed for the images such as (Histogram Equalization or bilateral filter).
3. Detection and feature extraction algorithms are more efficient on Android studio than on MATLAB.

More facial feature/points can be obtained on Android Studio than on MATLAB. We compared between the method used in the present paper and the one used in [19] to determine which one of them achieves a higher recognition rate. However, as far as we found from the literature, there is no research that applied the object detection using API, and Pearson correlation coefficient for the face geometrical objects normalized distances. Thus, we wanted to testify it against a known method. Table 4 gives this comparison.

TABLE 4. COMPARISON BETWEEN FACE RECOGNITION IMPLEMENTATIONS USING MATLAB AND FACE RECOGNITION USING ANDROID STUDIO

Comparison	MATLAB	Android Studio
Input Image	Has only one-dimension 300x400	Has any dimension such as 1024x862 or 709x628 and other dimensions
ROI preprocessing	Necessary to enhance the performance of the system	Not necessary as it does not affect in very small amount as illustrated above
Geometrical features	11 features	30 features
Features	Nose detect with 1 point. Mouth detect with 1 point. Eyes detect with 2 points.	Nose detect with 1 point. Mouth detect with 3 points. Eyes detect with 2 points. Cheek detect with 2 points.
Database	Excel database	SQL database
Face Detection Algorithm	Haar like features	Google's Mobile vision API
Feature Extraction Algorithm	Computer Vision	Google's Mobile vision API
Face Recognition Algorithm	There are many methods as: Correlation & Neural Network.	Pearson Correlation.
No of obtained points	4 points	8 points

8. Conclusions and Future work

In this paper, we proposed a design of face recognition system on mobile systems. This system achieves high performance when compared with other systems on mobile devices. From the results, our face recognition system achieves higher recognition rates than the other approaches. Also, face recognition on mobile devices is found to be more efficient than face detection on MATLAB for many reasons available by nature in mobile devices. In addition, the API algorithm achieves an excellent performance in detection and feature extraction compared to Haar like features. Object detection API does not need any ROI preprocessing to enhance the images.

In the future, the proposed face recognition system will be improved to recognize people from the input images regardless the direction and dimensions of the images. Also, we will add a new stage and camera for taking real live shots of people instead of using the stored fixed images.

References

- [1] Parusheva, S., "A comparative study on the application of biometric technologies for authentication in online banking", Egyptian Computer Science Journal, 39(4), 115126, 2015.
- [2] Joseph Lewis, "Biometric for secure Identity Verification: Trends and developments", University of Maryland, Bowie State University, January 2002.
- [3] Buciu, I., & Gacsadi, A., "Biometrics systems and technologies: a survey", International Journal of Computers Communications & Control, 11(3), 315-330, 2016.
- [4] Alzubaidi, A., & Kalita, J., "Authentication of smartphone users using behavioral biometrics", IEEE Communications Surveys & Tutorials, 18(3), 1998-2026, 2016.
- [5] Hassan Soliman, "Face Recognition in Mobile Devices", International Journal of Computer Applications (IJCA), pp.13-20, jul.2013.
- [6] Aesha shan, kavin shav, "Built-in Face Recognition on Smart Phone Devices", International Journal of Engineering and Technology (IRJET), Vol.4, pp.1472-1474, Jan.2017.
- [7] Dandashi, Amal and Walid Karam, "Biometrics security and experiments on face recognition algorithms", Computational Intelligence for Security and Defence Applications (CISDA), 2012 IEEE Symposium on. IEEE, 2012.
- [8] Peter, K.J., Nagarajan, G., Glory, G., Devi, V.V.S., Arguman, S., & Kannan, K. S., "Improving ATM security via face recognition", In Electronics Computer Technology (ICECT), 2011 3rd International Conference on (Vol.6, pp.373-376). IEEE.
- [9] Holat, Recep, and Selman Kulac, "ID identification by using face detection and recognition systems", Signal Processing and Communication Applications Conference (SIU), 2014 22nd. IEEE, 2014.
- [10] Nakano, Miyoko, F. Yasukata, and Minoru Fukumi, "Marketing data collection from face images using neural networks", Nonlinear Signal and Image Processing, 2005. NSIP 2005. Abstracts. IEEE-Eurasip. IEEE, 2005.
- [11] Kresimir Delac, Mislav Grgic, "A survey of biometric recognition methods", 2004.
- [12] sulochana sonkamble, 2dr.ravindra thool, 3balwant sonkamble, "Survey of biometric recognition systems and their Applications", Journal of theoretical and Applied Information Technology, 2010.
- [13] Anil K. Jain, Ajay 2010 Kumar, "Biometrics of Next Generation: An Overview".
- [14] Xing Chao, "Face Recognition in Mobile Phones", Stanford University, USA, 2010.
- [15] Angelo Galiano, "Face Recognition System on Mobile Device based on Web Service approach", International Journal of Computer Science and Information Technologies (IJCSIT), Vol.7(4), pp.2130-2135, 2016.
- [16] Hu, J., Peng, L., & Zheng, L., XFace: a face recognition system for android mobile phones", In Cyber-Physical Systems, Networks, and Applications (CPSNA), 2015 IEEE 3rd International Conference on (pp. 13-18). IEEE, Aug., 2015.
- [17] Nandini, M., Bharagavi, P., Raja, G., "Face Recognition Using Neural Networks", International Journal of Scientific and Research Publications, vol.3, PP.1-4, Mar., 2013.
- [18] Zidan, Khamis A., "Face Recognition using Neural Network", PhD Thesis, University of Technology Al-Iraqia, Oct., 2003.
- [19] Kamel Rahouma, Amal Zarif, "Face Recognition Based on Correlation and Back Propagation Neural Networks", Egyptian Computer Science Journal, Vol.43, No.3, ISSN-1110-2586, sept.2019.