International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

FACE RECOGNITION IN ANDROID

K.M. Sanghavi[1], Agrawal Mohini[2] ,Bafna Khushbu[3] ,Chopda Pooja[4] ,Dungarwal Payal[5]

[1]*Computer Department, S.N.J.B's KBJ COE, Chandwad,* kainjan@gmail.com
[2]*Computer Department, S.N.J.B's KBJ COE, Chandwad,* agrawal.mohini65@gmail.com
[3]*Computer Department, S.N.J.B's KBJ COE, Chandwad,* bafna.khushbu108@ gmail.com
[4]*Computer Department, S.N.J.B's KBJ COE, Chandwad,* chopdapooja21893@gmail.com
[5]*Computer Department, S.N.J.B's KBJ COE, Chandwad,* payal.dungarwal@gmail.com

Abstract— Human Identification can be done through various methods and one of the methods to identify human is Face recognition and it is a biometric technology Face recognition is adopted as the latest technology such as android devices. The face recognition technology is implemented by using eigenface technique in android device. These system can be used to archive for the recognition of human identity which will provide security.

Keywords- Face detection, Face Recognition, PCA.

I. INTRODUCTION

As we know mobile phones are becoming powerful, security of the data stored in mobile phones like email addresses, sensitive documents, etc., becomes very important. Password protection to address security use these in there current phones. The face recognition scheme is much more secure and flexible as it provides distinctive print to gain access and also the user need not remember passwords. The main aim or goal of this project is to implement a face recognition application on the DROID phone, which could be used to unlock the phone or authorize a transaction when the registered user is recognized.

For face detection and Eigen face algorithms for face recognition we used color segmentation combined with template matching. Eigenface face recognition approaches can be classified in an appearance-based method, because the main aim of eigenface face recognition uses information from the raw pixel image which is used for training and classification of image identity. Projecting an image of a face that can be seen as a vector is the main aim. On same lighting conditions the digital images of human faces are taken, then normalized into a grayscale image. On the same resolution the Image is processed and then used as the vector dimension where the components are derived from the pixel value of the image.

Human face recognition procedure consists of two phase. The first phase is where the face detection process takes place very rapidly in humans except in certain circumstances where the object is located at a far distance. The second phase is recognition stage which is recognizing the face as individuals face. The face recognition stages as it was then imitated and developed as a model for facial image recognition is one of the biometric technologies is widely studied and developed by the experts. It is because in general, the face image can provide the specific information related to personal identification .Thus the face image has high variation as the input. These variations are caused by two factors. The first factor is variation on own face and second factor is the variation caused the object transformation of face into face image. By face recognition system the variations of the face image must be able to be resolved.

@IJMTER-2015, All rights Reserved

II. LITERATURE SURVEY

Detection is not required for face recognition. Because of the already present face images in the database in some cases. There are many challenges associated with face detection of faces captured in uncontrolled environments. are pose variance, feature occlusion, facial expression, imaging conditions, etc. are some of the factor Haar-cascades and scanning images using an increasing window, skin color segmentation, template matching and morphological processing algorithms are some common approaches used for face detection.

2.1 Face detection methods:

2.1.1 Haar-cascades algorithm

P. Viola and M.J. Jones, proposed "Robust Real-Time Face Detection" and its effectiveness is due to the robust real-time processing capabilities.

2.1.2 Skin color segmentation

Mattias Junered, 2010, "Face Recognition in Mobile Devices" proposed normalized RGB, HIS and YCRCB color space transformations are commonly used in isolating skin color region.

2.1.3 Morphological Image Processing

M. Turk and A. Pentland,proposed "Eigenfaces for recognition" this process helps to regroup the skin pixels by eliminating the non-skin visible pixels.

2.1.4 Template Matching

M. Ballantyne, R. S. Boyer, and L. Hines. Woody Bledsoe proposed this approach for finding an object represented by the template in a given image.

2.2 Face recognition methods:

2.2.1 Projection Methods

M. Turk and A. Pentland ,proposed "Eigenfaces for recognition" in these Projection methods help you to do this by introducing discrimination for better classification and reduction in dimensions.

2.2.2 Statistical Methods

Moghddam and Pentland proposed probabilistic Eigenfaces based on PCA these methods provide a best way to remove redundant information in the input data.

2.2.3 Graph Matching Methods

L. Wiskott, J. M. Fellous, C. von der Malsburg, proposed "Face recognition by elastic bunch graph matching", in these transformation of the input images is done using these models. Graph building is the most common way to link features to nodes and evaluate the transformations to mimic face rotations.

2.2.4 Neural Network Methods

T. Kohonen. Self-organization and associative memory. Springer-Verlag, Berlin, 1989 used a neuron network to recognize aligned and normalized faces. Self-organizing maps (SOM) and Convolutional Neural Network (CNN) were used by Lawrence et al. for face recognition.

III. FEATURE EXTRACTION

An image is captured from the mobile phone and feature extraction takes place the image is stored in

@IJMTER-2015, All rights Reserved

the database then the image is matched from the database then the feature are extracted by using eigen face method . The figure describe the execution of the system..

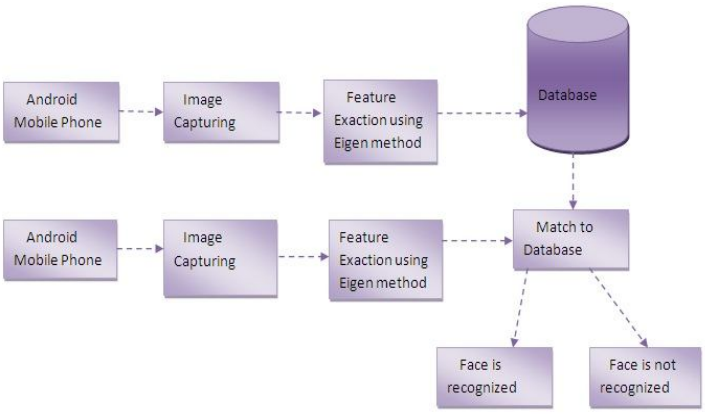

Fig. 1 System Architecture.

IV. APPROXIMATE FACE DETECTION ALGORITHM

Detection (Trainy Image[1…m],Y)

1. Begin.
2. If(NULL ! cameraBitmap)
   Initialize
   Width=cameraBitmap.getwidth ();
   Height=cameraBitmap.getheight();
   FaceDetector d=new FaceDetector(width,height,trainyImage);
   Faces=new FaceDetector.Face[trainyImage];
3. if(Facefound>0)then
   From i=0 to Facefound
   calculate
   Face[i].getmidpoint(midpoint);
   eyeDistance=Faces[i].eyeDistance();
   confidence=Faces[i].confidence()
   then draw rectangle.
   End if
4. End if
5. End.

In face detection algorithm trainy images are captured …the width and the height is initialized so that the face will fit in that. Then we define the object of faceDetector by initializing the parameters as width, height, trainy image.one more object is created of face in which the images are there.

Check the condition if face is found from the stored place then we have to calculate the mid point of the face. From that midpoint we have to calculate the eye distance and the confident value of the face. Then we have to draw the rectangle .then the face will be matched. And the message will be displayed that the face has being successfully detected.

V. APPROXIMATE FACE RECOGNITION ALGORITHM

After successfully executing a face detection algorithm, an appropriately sized and scaled image of a

@IJMTER-2015, All rights Reserved

face could be used for face recognition. The initial approach to solving the face recognition problem involved a common computer vision algorithm known as Eigen face. This algorithm can be easily implemented in OpenCV. The algorithm basically works by comparing the unknown face to a database of faces. After computing a general "difference" between the database faces and the unknown face, the closest matching face is chosen. Determining the difference between to different images can be a complicated task; however, with the use of Principal Components Analysis (PCA), the task can be simplified. Rather than comparing every single pixel of an image to that of another, all points and data can be projected onto a subspace.

By using Principle Components Analysis, data can be reduced into a single point in multidimensional space. Therefore, the distance between two points in a PCA subspace can be measured to determine the overall difference of an image.

The eigen faces approach for face recognition is summarized as follows:

1. Collect a set of characteristic face images of the known individuals. This set should include a number of images for each person, with some variation in expression and in the lighting (say four images of ten people, so *M*=40).
2. Calculate the (40 x 40) matrix *L*, find its eigenvectors and eigenvalues, and choose the *M'* eigenvectors with the highest associated eigen values (let *M'*=10 in this example).
3. Combine the normalized training set of images according to Eq. (6) to produce the (*M'*=10) eigenfaces $\mu_k, k = 1,....., M'$.
4. For each known individual, calculate the class vector Ω_k by averaging the eigenface pattern vectors Ω [from Eq. (8)] calculated from the original (four) images of the individual. Choose a threshold θ_ε that defines the maximum allowable distance from any face class, and a threshold θ that defines the maximum allowable distance from face space [according to Eq. (9)].
5. For each new face image to be identified, calculate its pattern vector Ω, the distance ε_k to each known class, and the distance ε to face space. If the minimum distance $\varepsilon_k < \theta_\varepsilon$ and the distance $\varepsilon < \theta$, classify the input face as the individual associated with class vector Ω_k. If the minimum distance $\varepsilon_k > \theta_\varepsilon$ but $\varepsilon < \theta$, then the image may be classified as "unknown", and optionally used to begin a new face class.
6. If the new image is classified as a known individual, this image may be added to the original set of familiar face images, and the eigenfaces may be recalculated (steps 1-4). This gives the opportunity to modify the face space as the system encounters more instances of known faces.

VI. EMPIRICAL RESULTS

We collected 200 persons (2 images of per person) 1 for learning and 1 for testing from database. We selected the centers of eyes and lips in order to avoid recognition error related to incorrectly detected faces. We rotated images to make the line connecting eye centers horizontal, resized the image to make the distance between the centers of eye equal to 26 pixel, calculated centers of face using the centers of eye and lips, cropped 64x64 central part of the face, performed histogram equalization on the cropped part of the image.

In all the experiments we use the same templates and change only the recognition method and the number (percent) of used features. The experiment is obtained from the terms FMR(False Match Rate) and FNMR(False Non Match Rate).FMR is the error calculation when system reject the face which is stored in the database. And FNMR is the error which system is receiving or identifying the face who not registered. The results are shown in the below table.

@IJMTER-2015, All rights Reserved

Table No. 1. Table of Experimental Result.

Threshold	FMR(%)	FNMR(%)	Accuracy(%)
30	0,28	9,88	89,24
31	0,36	9,72	89,92
32	0,48	9,16	90,36
33	0,96	8,44	90,6
34	1,08	8,16	90,76

COMPARISION AND GRAPHICAL RESULT

The performance chart is shown below which shows the accuracy rate of the pervious system and the implemented system. The red line indicates the older system whose accuracy rate was less and the implemented system has increased the accuracy rate so it is beneficial to use the system.
The graphical representation is shown how the image is captured then the face is detected and matched with the image stored in the database and then the verification is done. The Eigen value is calculated and according to that the recognition is done the basic requirement is android phone through which the image is captured and face is detected and recognized.

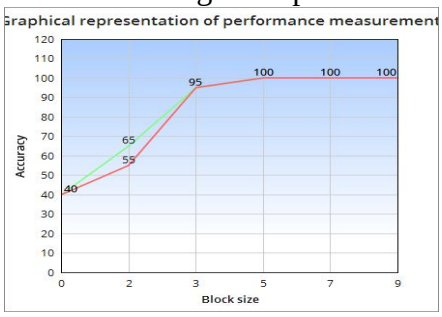

Fig. 2 Performance Chart.

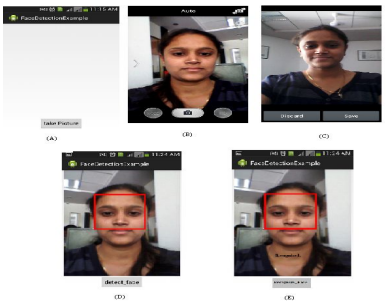

Fig. 3 GUI Result.

CONCLUSION

We discussed the implementation of a face recognition system using PCA .The focus was thus to implement a face recognition system good enough to be used for access control. We demonstrate the face recognition system in android device using eigenface. The system can be used as the base for the development of android applications such as android mobile security application and as an archive for the recognition of human identity.

REFERENCES

[1] M. King, B. Zhu, and S. Tang, "Optimal path planning," *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.
[2] H. Simpson, *Dumb Robots*, 3rd ed., Springfield: UOS Press, 2004, pp.6-9.
[3] M. King and B. Zhu, "Gaming strategies," in Path Planning to the West, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.
[4] B. Simpson, et al, "Title of paper goes here if known," unpublished.
[5] J.-G. Lu, "Title of paper with only the first word capitalized," *J. Name Stand. Abbrev.*, in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Translated J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [*Digest 9th Annual Conf. Magnetics Japan*, p. 301, 1982].
[7] M. Young, *The Technical Writer's Handbook*, Mill Valley, CA: University Science, 1989.

@IJMTER-2015, All rights Reserved