

Embedded Vision System for Autonomous Face Recognizer Robot using Eigenface Detector-PCA Improved Face Detection for Largely Occluded Faces

¹Mittal Chavhan, ²Sunita Jadhav

¹Electronics and Telecommunication Department, Saraswati College of Engineering, Kharghar, India

²Electronics and Telecommunication Department, Saraswati College of Engineering, Kharghar, India

Abstract: The problem is to propose a method for detecting the face and recognizing the face for a robot. This is a challenging problem because faces are non-rigid and have a high degree of variability in size, shape, colour, and texture. In this report, an embedded vision system is built to locate, recognize and track the faces. Three individual sub-systems are present. The first one is face detection based on Haar-like classifier and Viola-Jones object detector. The second one is face recognition system based on Eigenface detector using principal component analysis method. The third one is face tracking based on Haar Cascade Classifier. For fast and an efficient embedded vision system for an autonomous robot, the software used is OpenCV library. The system developed has improved performance and can also detect largely occluded faces.

Keywords: Haar-like classifier; Viola-Jones object detector; Eigenface; principal component analysis; Haar Cascade Classifier.

I. Introduction

Face plays a big role in conveying identity and emotion, being the primary focus of attention in social life. Ability of human to recognize faces is remarkable. Thousands of faces learned throughout our lifetime can be recognized. Also, familiar faces can be identified at a glance even after years of separation. Computational model of face recognition can be applied to criminal identification, security systems, image and film processing and human computer interaction. This is a challenging problem because faces are non-rigid and has a high degree of variability in size, shape, colour and texture, multi dimensional. Hence it is difficult to develop a computer model of face recognition as faces are a natural class of objects suggesting broad constraints on corresponding neural activity. The approach treats the face recognition problem as an intrinsically two dimensional recognition problem. This is due to the advantage of the fact that faces are normally upright, thus, described by a small set of 2 – D characteristic view. The advantage of the approach is that it provides for the ability to learn and later recognize new faces in an unsupervised manner. Unlike Fingerprints, Iris scan and Speech recognition, face detection and recognition system [1] has two main advantages given as – It does not require aid (or consent) from the test subject and it can perform mass identification. The challenges associated with face detection are – presence or absence of structural components, face pose, facial expression, occlusion, image orientation and imaging conditions. The problem is to detect and recognize the face in real-time application using robot. In this report, we build an embedded vision system to locate, recognize and track the faces.

Thus, a fast and an efficient face detection system using Haar Classifier [2, 3] approach, a face recognition system based on Eigenface which relies on Principal Component Analysis (PCA) and a face tracking system based on Haar Cascade Classifier is implemented. We use the software Open source Computer Vision (OpenCV) [4, 5] library.

The chapter 2 describes the Hardware details of the project, chapter 3 describes the theory, chapter 4 shows the Result and chapter 5 shows the Conclusion.

II. Hardware Details Of The Project

The block diagram, thus, consists of – a laptop, a webcam, a USB to TTL converter, a microcontroller IC 89S52 (8051 family), LCD (16×2 character) display, a motor driver L293D (consisting of left motor and right motor) and a battery with two output voltages of +5v and +12v. The laptop, thus, performs the four basic functions- port controlling (USB port), camera interfacing which is done using OpenCV Cam, image processing which is face detection and face recognition and Decision making element which decides whether the detected face of a person matches with that of faces in the database, if yes the person will be tracked. The decision is, then, communicated through a serial port RS232. A USB to TTL converter is used. The basic microcontroller IC 89S52 of 8051 family is used which is also inexpensive. The microcontroller performs the three main functions

–to get command from Laptop via UART, to pass various signals to L293D motor driver depending upon the command from Laptop and to display system messages on LCD.

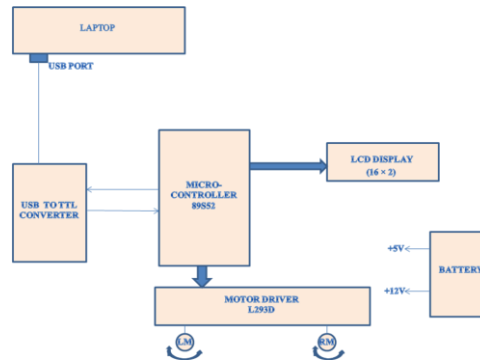


Figure 2.1. Block diagram of the project where LCD is liquid crystal display, LM is left motor, and RM is right motor:-

III. Theory

A frontal face detection system is used. We use the Viola Jones Face detection Framework [2, 3]. The Viola Jones face detection algorithm includes the following topics – Haar features, Integral Image, Adaboost and Cascading. PCA [6] stands for principal component analysis. PCA can transform a set of images i.e. multivariate dataset, being visualised as a set of coordinates in a high dimensional data space into a lower – dimensional picture, when viewed from its informative viewpoint (From information theory approach). From mathematical viewpoint, PCA is a procedure that converts a set of M face images (correlated variables) into a set of K linearly uncorrelated variables called Eigenfaces or Principal components using optimal orthogonal transformation. It includes two steps [7] – Training the recognizer or initialization operation and Recognize an unknown face.

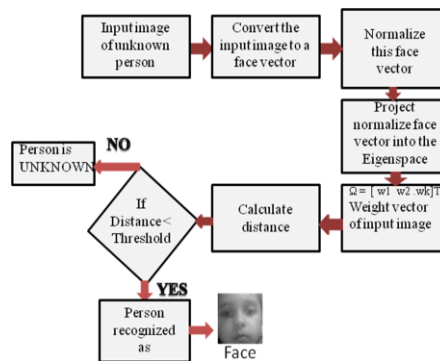
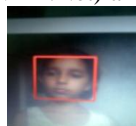


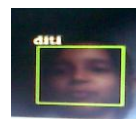
Figure 3.1. Block diagram for recognition of unknown face – Real-time face tracking [8] is implemented using Haar cascade classifier in OpenCV.

IV. Result

For the project, the Software requirements are – Windows vista operating system, Microsoft Visual Studio 2010, Emgu CV (Open CV in .Net) and C# Programming for coding.



a) Output of Face detection



b) Output of Face recognition



c) Robot of the project

Figure 4.1. Output of face detection, face recognition and robot of the project –



a) Initial left position-

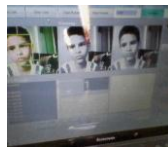


b) Movement towards right -



c) Right position of the Face Tracker –

Figure 4.2. Output of Face tracking –



a) Left position-



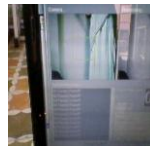
b) right -



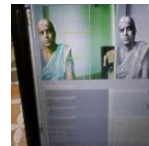
c) Rightmost position-



d) Display screen



e) no face found -



f) face not recognized -

Figure 4.3. Output of face tracking showing the display screen–

Table 4.1. Performance for different scale factor and minimum neighbour –

Minimum neighbour	Faces Detected
0	68
1	4
2	4
3	4
4	4
5	4
6	4
7	4
8	4
9	4
10	4
11	4
12	4
13	3
14	2
15	2
16	2
17	2
18	2
19	1
20	1

Scale factor	Faces Detected (%)
1.1	100
1.2	50
1.3	25
1.4	0



Figure 4.4. Database of face detection

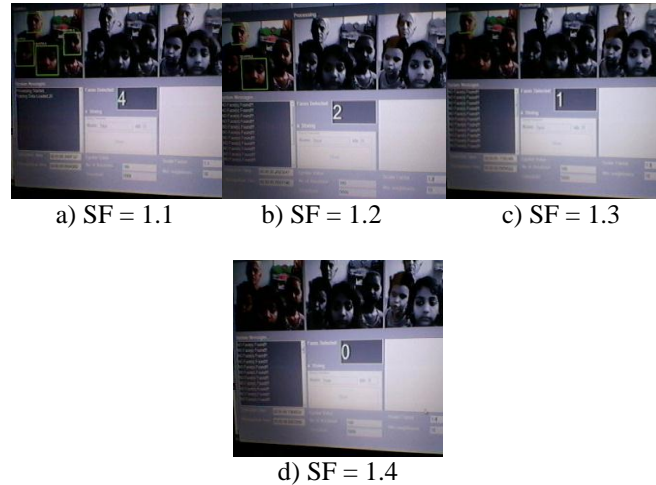


Figure 4.5. Output of face detection for different scale factor –

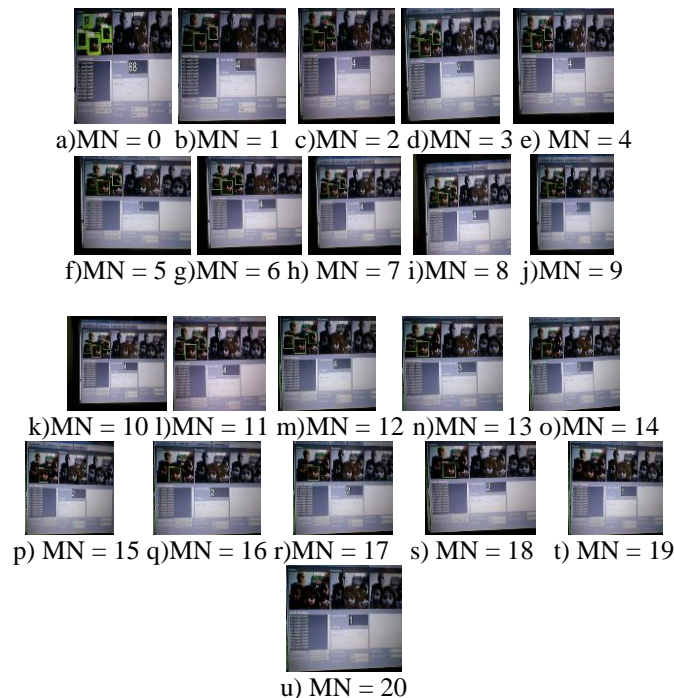


Figure 4.6. Output of face detection for different minimum neighbour –

For threshold = 5000 (default value), scale factor = 1.1, minimum neighbour = 10, number of faces per person = 2, the performance for different number of person or registered user was as shown below and for number of person = 10, the performance for different number of faces per person or registered user was as shown below.

Table 4.2. Performance for different no. of person and for different no. of faces per person -

Number of person or registered user	Recognition time (ms)	Recognition accuracy (%)
2	0.1722	100
4	0.1952	100
6	0.3173	98
8	0.3707	97
10	0.4441	96

Number of faces per person	Recognition time (s)
2	0.0004441
4	0.0013887
6	0.0014630
8	0.0016964
10	0.0020786

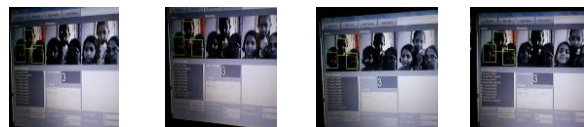
For scale factor = 1.1, minimum neighbour = 10, number of faces per person = 10, number of person or registered user = 10, the performance for different threshold values was as shown below.

Table 4.3. Performance for different threshold -

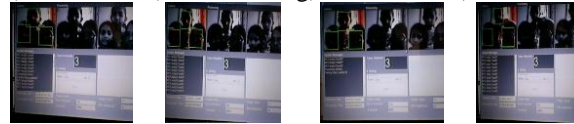
Threshold	Faces Recognized (%)
0	0
1000	0
2000	0
3000	100
4000	100
5000	100
6000	100
7000	100
8000	100
9000	100
10000	100



a)TH = 0 b)TH = 1000 c)TH = 2000 d)TH = 3000



e)TH = 4000 f)TH = 5000 g)TH = 5500 h)TH = 6000



i)TH = 7000 j)TH = 8000 k)TH = 9000 l)TH = 10000

Figure 4.7. Output of face recognition for different threshold -

For xml file haarcascade_frontalface_alt2 (eigenface)



For xml file haarcascade_eye.xml (eigeneye)



Comparison between
haarcascade_frontalface_alt2.xml (eigenface) and haarcascade_eye.xml (eigeneye) files
Eigenface
eigeneye



Thus, from above, it is seen that a fast and accurate face detection and recognition system can be developed by choosing proper parameters of the system, such as the scale factor, minimum neighbour and the threshold. It is observed that as scale factor increases from 1.1 to 1.4, faces detected decreases from 100% to 0% when number of person is 4 with 4 faces per person. Hence, scale factor is chosen as 1.1. Also, when minimum neighbour is zero, 68 faces are detected. As minimum neighbour increases from 1 to 20, faces detected decreases from 4 to 1. Hence, minimum neighbour is chosen as 10. For the threshold- faces recognized is zero, when threshold increases from 0 to 2000 and faces recognized is 100%, when threshold increases from 3000 to 10000. Hence, threshold is chosen as 5000 (heuristically). As number of persons increase, recognition time increases while recognition accuracy decreases. As number of faces per person increase, recognition time increases. An embedded vision system for autonomous face recognizer robot which can detect, recognize and track frontal, upright faces is, thus, developed. The extreme conditions is that if threshold is very low (approximately zero), it gives no recognition, whereas if threshold is greater, it gives less good match. The project works only for frontal, upright view of faces (considering 2D image space) and not for large variations in pose, facial expressions, image orientation, structural components and occlusions. Also, it is observed that by changing the xml file from frontalface to eye, largely occluded faces can also be detected.

V. Conclusion

Face recognition technology can be adopted for many applications (from crowd surveillance to improved human-computer interaction) for both verification and identification where other biometric tools have been difficult or not accessible. In this report, an embedded vision system that can locate, recognize and track the face of individuals is being developed. The face detection system using Haar classifier approach and face recognition method using the eigenface based on principal component analysis (PCA) approach is implemented. The report also shows the face tracking using the Haar cascade classifier. The 80S52 microcontroller and the software's C#.net and OpenCV library are used. A 2-D recognition using grey-scale images, thus, improves performance. It is seen that a fast and accurate face detection and recognition system can be developed by

choosing proper parameters of the system, such as the scale factor chosen, minimum neighbour and the threshold. As number of persons increase, recognition time increases while recognition accuracy decreases. As number of faces per person increase, recognition time increases. Also, largely occluded faces can be detected by changing xml file from frontalface to eye. Thus, a fast, reasonably simple and an efficient method is developed for use in constrained environments, such as office or home. The face recognition module offers possibly the most potential for future development. The eigenfeatures technique uses a combination of both global facial and localized feature information to estimate face identity. Hence, it can be used for further research.

References

- [1] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 34-58, January 2002.
- [2] P. Viola and M. Jones, "Robust real-time face detection," International journal of computer vision, vol. 57, pp. 137-154, 2004.
- [3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proc. IEEE CVPR, 2001, pp. 1-511-I-518 vol.1.
- [4] OpenCV Wiki: Face Recognition available at <http://opencv.willowgarage.com/wiki/>.
- [5] G. Bradski and A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library: O'Reilly Media, 2008.
- [6] Principal component analysis-wikipedia available at mhtml: file:///zone/k/Principal component analysis-wikipedia, the free encyclopedia.mht.
- [7] Mahvish Nasir, EmguCV with C#, Few Tutorials A Beginner's Guide from Concept to Code, 2012.
- [8] Joseph Howse, "Tracking Faces with Haar Cascades", Open Source, May 2013.