Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019)
IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4

Low Cost Eye Gaze Tracker Using Web Camera

M. Sai Mounica, M. Manvita, C. Jyotsna, Amudha J.
Dept. of Computer Science & Engineering, Amrita School of Engineering, Bengaluru,
Amrita Vishwa Vidyapeetham, India
m.mounica28@gmail.com, markalamanvitareddy@gmail.com, c_jyotsna@blr.amrita.edu, j_amudha@blr.amrita.edu

***Abstract*—The traditional gaze tracking systems are of invasive, expensive or not of a standard hardware. To address this problem, research has been focused onto systems with simple hardware with gaze tracking systems. We propose a system which uses web camera and the free open source Computer Vision Library Open CV. Gaze estimation plays a major role in predicting human attention and understanding human activities. It is also used in market analysis, gaze driven interactive displays, medical research, usability research, packaging research, gaming research, psychology research, and other human-machine interfaces. This paper describes how to develop a low cost eye gaze system using a simple web camera. The system captures real time video of the person to detect the eyes in the initial frames and extract the features of eyes. Once the features are extracted, the eyes are tracked in the subsequent frames and the gaze direction is estimated using the computational intelligence techniques.**

Keywords—eye tracking; gaze estimation; human computer interaction; web camera; calibration

I. INTRODUCTION

Eyes are the best information source that serves as a replacement for human attention. Eyes are the crucial features because of its appearance and expression variety.

Gaze is the direction in which a person looks. Gaze tracking is to track the movement of the eyes to know exactly where the person is looking at. Recently, eye gaze detection techniques are used in applications like human machine interfaces, monitoring driver vigilance, and identifying website key objects and for gaze tracking for gaming design. New applications are used in technical environments, such as operating rooms in hospitals, industrial control units and airplane cockpits for eye-tracking communication interface [1],[2].

Numerous efforts to build home brew eye tracking system is because of the high cost of existing commercial systems. Proposed system builds a low cost eye gaze system and makes it affordable for the average man.

Existing Eye Gaze systems that detect and track gaze often use infrared light, web cameras or some head mounted system. Unfortunately, commercial systems remain too expensive. Cost is not the only factor behind developing this low-cost system, but to make it a technology that can be used for niche applications.

II. LITERATURE SURVEY

Developing an eye tracker includes three mechanisms. Eye detection 2) Eye tracking 3) Gaze Estimation, this chapter explains what the three modules are and briefs about the papers referred for implementing these modules.

In [3], developed an adaptive eye location approach for eye detection using generic algorithm. Experimental results show that the method has the potential to find the eye location and the feasibility of the approach, and its possible use for the location of facial landmarks and their annotation during the derivation of shape-free images. In [4], using eye template matching technique, centers of both the eyes are detected.

An eye tracking method with good speed and size and direction invariant was proposed [5]. The method not only focused in accuracy but also on real-time processing and general interface to detect and track eyes. The method is based on a single template matching using Genetic Algorithms (GA). Generally, GA is time consuming; however, the problem was resolved using Evolutionary Video Processing. The evolutionary Video Processing uses genetic information as a relation between frames to track eyes. This method reduces processing time and increases the accuracy. The effectiveness of method is verified in two simulations, the single image processing and the video processing. In the single image processing efficiency is 76.0%, on the other hand, the video processing is better, 97.9%. The average processing time per frame is 28 milliseconds.

The method also had a high tolerance to head movements. The method used has treated the three problems associated with eye tracking: Gathering information from eye and simultaneous eye tracking, the change of eye appearance due to blinking of eye, motion of head and eye movement, and generality of the system. Genetic Eye Tracking [6] was used to solve these three problems. Artificial template matching is used to support the third problem, generality. The method is divided into three parts: first is tracking the eye and acquiring the eye information, which is called Genetic Eye Tracking, the second part uses the information gained in first step to measure the eye movements and then in the third part this information is used for selecting an item on the communication screen by the eye gaze. The eye tracking accuracy is about 99.6%.

An eye detection method that matches templates of the eye detection using genetic algorithm was proposed [7]. An experimental result show that the method proposed in this paper detects eye in less amount of iteration compared to the conventional template matching's worst case complexity.

A method to generate templates based on particle swarm optimization, for frontal face localization in real time was proposed [8]. PSO algorithm was used to maximize the line integral value to generate templates, using less number of particles in order to reduce the computational time.

In deformable template first, computation is costly. The energy terms for weight factors are determined manually, secondly. Unexpected results are yielded with improper selection of the weight factors. As the accuracy is very high and the method is simple and effective we can use this model.

Blink Detection has problems with detecting blinks when eyes are quickly moving up and down and this method will not be affected by light changes and it can overcome the head movement issues, variant projection [9].

III. SYSTEM ARCHITECTURE

The proposed method uses normal web camera and computer vision library open CV. The design of the system is having a closer resemblance with the existing system. The tools used in the proposed system is distinct from the tools which is available.

To develop the low cost eye gaze system, we use a Computational Intelligence Technique called Neural Network. The system consists of a web camera, a display (communication screen) and a computer. The communication screen consists of different regions or blocks, the user gazes at.

For several decades, tracking the user's eye-gaze information has been technologically possible. However, the eye-gaze tracking systems that exist are very expensive still. These high prices on commercial system have limited the eye-tracking technology utilization. The costs associated for building a commercial eye gaze system are 1) Material Cost, 2) Research and development cost and 3) business cost. Hence the expensive commercial systems have led to building home-brew gaze trackers. Proposed method reduces the cost of eye gaze systems.

The material cost can be reduced by using simple web cameras. The software cost can be reduced by the use of OpenCV(Open Source Computer Vision and Image Processing libraries).

The general algorithm of an eye gaze system [10] is as shown in Fig. 1. The initial step is obtaining a single frame from the camera and detect the face. The next step is the detection of the eyes. Once the eyes are detected, in subsequent frames the eyes are tracked. The last step in eye gaze system is eye gazing. Eye gazing understands the gaze patterns of the user.

A. Eye detection

The first step in Eye Gaze System is to get a single frame from the video obtained from the web camera. After getting the frame, Haar Cascade Classifier is used to detect the face. Once the face is detected, the next step is detecting the eye region. Haar Cascade Classifier is again used to detect the eye region. From the detected eye region, the left eye region and right eye region are split. For further processing, only left eye is considered. This is to reduce the computational complexity from tracking two eyes at the same time using neural networks. The entire process in shown in Fig. 2.

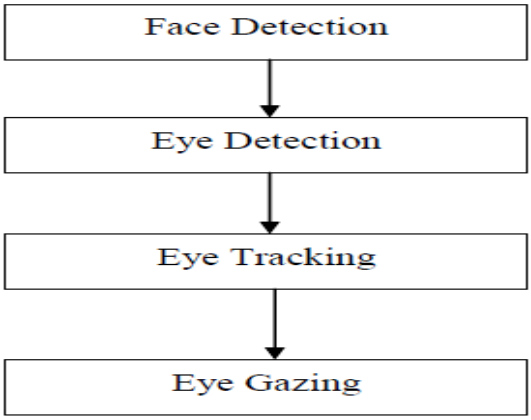

Fig. 1. General algorithm

B. Eye Tracking

Now that we got how to detect the eye image, when we consider a real time video of the user, we need to capture some number of frames. In each frame, we detect the eye images and store them in a folder. So for each frame the eye detection algorithm [11] works and tracks the eye position in each of the frames.

C. Calibration

We must first teach and get the computer familiar with what the eye looks like when the user's eyes are fixated on known locations on the screen, in order to know where the user's eyes are exactly fixating on the computer screen, this is what we do during the validation and calibration procedure. The user fixates on 6 points on the screen during a calibration session, while the eye is monitored by the eye tracker. This is calibration. After this is performed, the computer then performs validation. In this, the targets are represented to validate the information and determined whether the eye position estimated is indeed close to the targets' known position. Calibration is indicated as successful if the errors are minimal. Now the trials may start. The computer is interpolated between the calibrated landmarks on the screen during the experimental trials, to locate where the eye is fixating.

The next step is the calibration, where the subject is requested to sit in front of the communication screen and asked to gaze at the objects on the screen. The calibration is performed for six regions of the screen namely, 1) top left, 2) top middle, 3) top right, 4) bottom left, 5) bottom middle, and 6) bottom right. For each region the pixel values of different images in frames is retrieved. Then these values are given as input to the neural network. Targets are set and the neural network is trained with these values to get the weight and bias values. Six classes are defined in the target. For different samples, classes are assigned in this step.

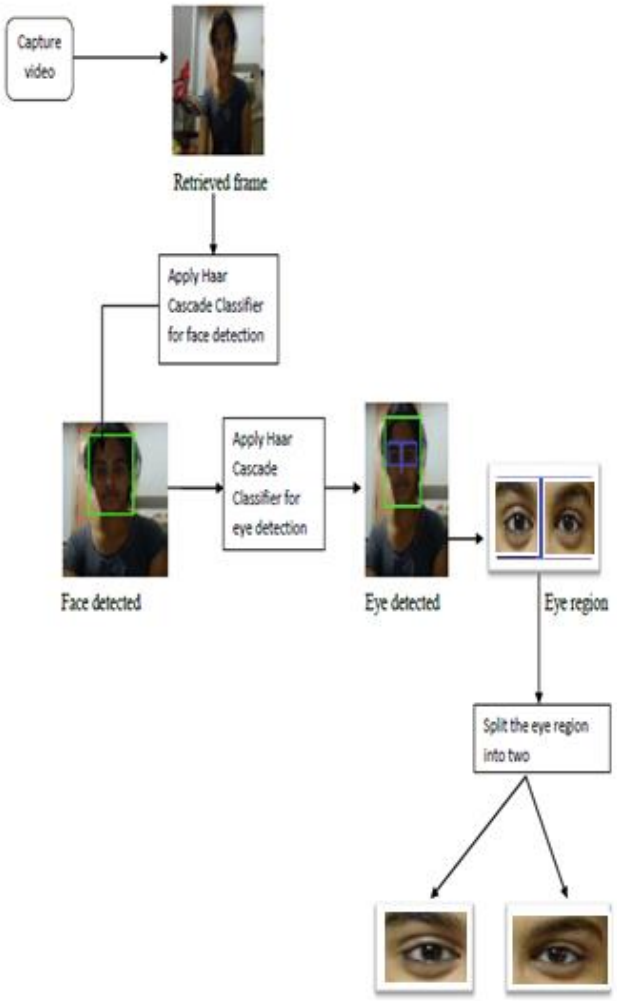

Fig. 2. Flow of Eye Detection

D. Validation

In validation, the trained neural network is given a random input and the system should be able to determine which class/region the user is looking at. If the system gives correct region, it is said that it validates accurately. The accuracy of the validation depends on various factors like the distance from the screen and the light conditions. Calibration and validation process is shown in Fig. 3.

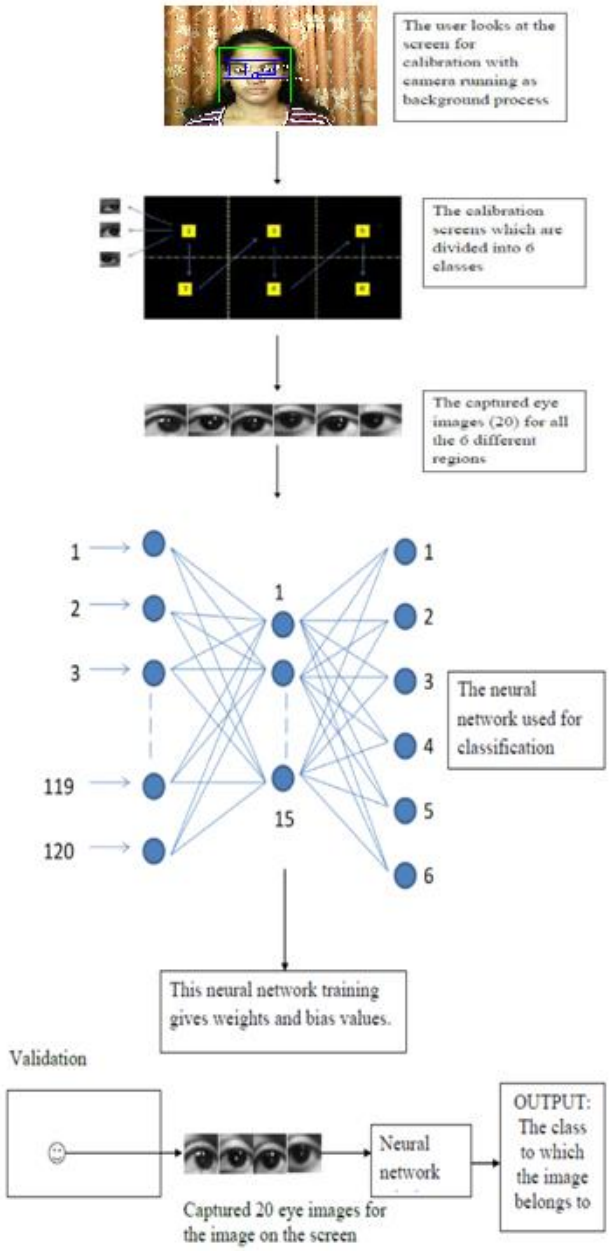

Fig. 3. Calibration and Validation

IV. IMPLEMENTATION

A. Eye Detection

The eye detection method follows the same steps as the facial detection. The eye detection is applied separately for each eye using Haar classifier. The function is used twice in one iteration of the run process. For eye detection, use the Haar detection method, set the ROI, copy only the ROI to a new

image and return that image. A screen shot of the returned image is shown in Fig. 4.

Fig. 4. Eye detection Image

However, while testing it was found that the Haar detection for the eyes was not as accurate as the face detection. First the ROI from the facial detection is divided into two separate halves of the face, one for the left and one for the right. So the facial box was halved to get two images. Then using these images, the Haar detection was done separately.

Basic steps of eye detection can be summarized as Capture video and retrieve a frame. Apply Haar Cascade Classifier for face detection and eye detection on the frame. Once eye region is detected, split the eye region into two. After separating the two eye regions, we can perform eye tracking .

B. Calibration

The subject is provided with a 6-point calibration screen that has six regions. In each screen, the object is placed at a region.
The objects are displayed in the pattern shown in the Fig. 5. The subject gaze at the object on the screen and the gazing is done for 10 seconds. For each gazing, 20 images are extracted

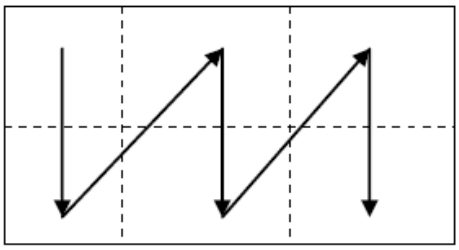

Fig. 5. Pattern of Object on screen

C. Eye Tracking

The frames should be obtained from the running video stream. Using Haar Cascade Classifier, detect the face region. Now using Haar Cascade Classifier detect the eye region. Split the eye region to get the Left and Right eye region. The eye region is matched with templates generated using Genetic Algorithm [12] to obtain highest correlation coefficient and track the eye.

D. Eye Gazing

Display the test frames with object embedded at a position
in it. Obtain the frames from the video stream. The eye regions are obtained as described in eye detection algorithm. The right eye regions are sent to the neural network to train the network. Each region is defined as a class in the nprtool. Test samples are again fed as input to the neural network to determine the object belongs to which class.

We calibrate the points and we record the eye images. The pixel values of the eye images are stored in the Matlab Workspace. The image size is 48*48. 120 samples are taken. So the matrix size will be 120*2034. The target is 6 regions to be classified. The two datasets are imported to the nprtool in the Matlab.

Nprtool solves a classification problem in pattern recognition using a feed forward pattern network (two layer). Matlab Neural Network Toolbox provides tools for designing, visualizing, implementing, and simulating neural networks was provided by a toolbox called Neural Network in Matlab..

We will follow Matlab's examples to learn to train neural networks using four graphical tools to solve problems in pattern recognition, function fitting.

Start the master GUI by typing nnstart. This enables us to access the GUIs for various tasks like Pattern recognition, Function fitting, time series analysis and Data clustering. Command line operation can also be used to use the toolbox, which we will not cover. Standard steps in designing a NN in Matlab are i) Collect the data. ii) The images are converted into. mat file and are stored as a vector to be given as input to the neural network. iii) Create the network. iv) The network is selected in the npr tool. It gives various networks like back propagation, feed forward, recurrent networks. v) Configure the network. vi) We considered the feed forward network with one layer as hidden layer with 70% as training samples and 15 as hidden number of neurons. vii) Train the network. viii) Train the network with the set input samples and target samples by clicking on the "train" button. ix) Validate the network. x) The input eye images for a random point on screen are given with same sized samples as given for training are to be given to test the network. The network will use the weights and bias values obtained from training and gives the output as the class to which the image belongs to among the six classes. xi) Use the network.

V. Result Analysis

The hardware used here is an eye tracker (web camera) which should be attached to the laptop or personal computer. The eye tracker is tested under favorable conditions. The user is asked to sit in front of the system at a distance of 30-40 centimeters and the calibration screen is presented for the user. As a user views the calibration screen, the webcam records the video of the user's eye movement. The webcam is of 8 megapixels. The image is presented before the user for 10 seconds and the user is asked to look at it while the system generates the frames corresponding to each image in the calibration screen. The eyes are then tracked using neural network and shown in Fig.

6. The testing is done to determine the class to which the eye image belongs.

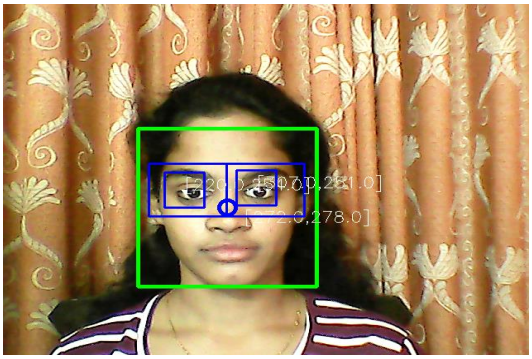

Fig. 6. Face and Eye detection

A. Distance from the screen

The distance of the user from the screen is selected to be 40cm. If the distance is too large the detecting eye fails and if it is too close, the harmful radiations from the screen makes it difficult for the user to gaze at the object. Webcam can track eye for a maximum distance of 50 to 100 cm. If it is more than 100 cm, it won't be able to track the eye.

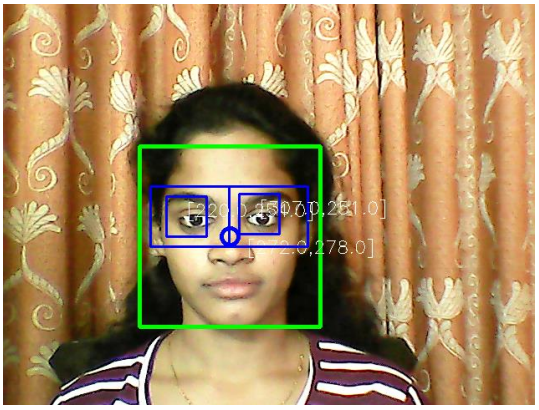

Fig. 7. Detecting eye from a Distance 50cm

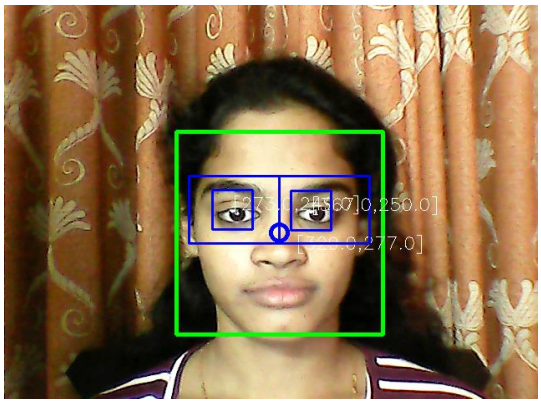

Fig. 8. Detecting eye from a Distance 30- 40 cm

Detecting eye from a distance of 50 cm and a distance of 30 - 40 cm is shown in Fig.7 and Fig.8.

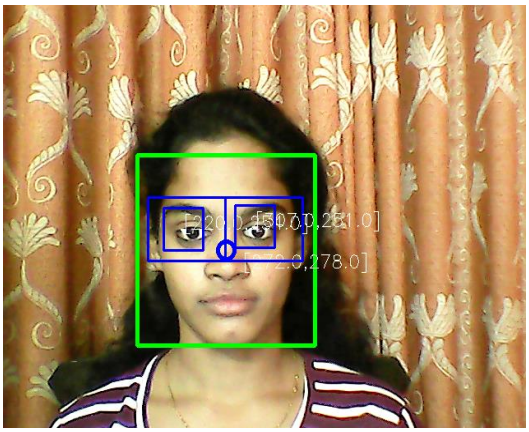

Fig. 9. Well Illuminated

B. Illumination of the environment

The environment has to be well illuminated otherwise the eye gaze system fails to detect eyes. Fig. 9 and fig.10 show he detection of eye in well illuminated and less illuminated environments.

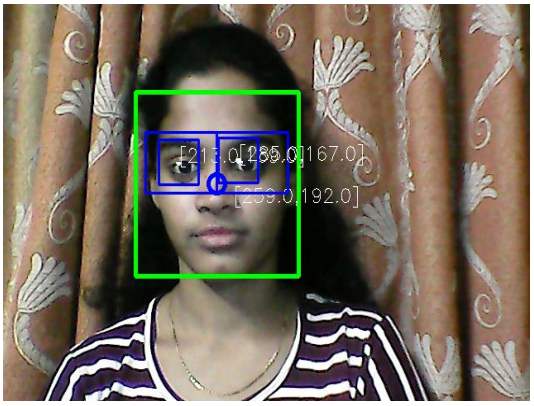

Fig. 10. Less Illuminated

C. Neural Network Analysis

The analysis is made on the given data sets. The data considered is divided into training, testing and validation methods. The training process is carried out. After calculating the error, the confusion matrix is produced, then the neural network is designed.

In the testing phase, the input and the neural network pattern [13] [14] is taken and used for analysis and the correct classification with the sufficient accuracy is obtained and checked. The following test cases are considered.

For the given input samples are given, i.e., the training percentages and number of hidden neurons, which are

considered as the parameters for building the network. Accuracy is calculated and the parameters are fixed for the maximum accurate measures. By manipulating the parameters, the results are accordingly checked and analyzed.

D. Confusion Matrix:

A confusion matrix, allows performance visualization of a supervised algorithm. It is a table layout in specific. The instances of a predicted class are represented in columns, and the instances in an actual class are represented by rows. It easy to see if the system is confusing two classes. Common confusion is mislabeling one as another. For further inspection, this matrix summarizes the results of testing the algorithm shown in Fig. 11.

Fig.11. Confusion Plot

For training:

Table I. TRAINING ACCURACY

Time for gazing (sec)	No.of samples	No.of hidden neurons	Training %	Error %	Accuracy
10	120	10	60	16.66e-0	83.3%
10	120	10	70	26.19e-0	73.8%
10	120	15	60	0	100%
10	120	15	70	0	100%

E. Validation

For testing, the same sized input and target data set is given, after which the network is tested. The confusion matrix is plotted and with the help of the matrix we can determine which samples belong to which class/region. If the matrix gives different classes for different frames, we consider the highest frequently occurring class. If the class is correctly determined, the test is successful. Confusion matrix for testing is shown in Fig. 12.

Table II. TESTING ACCURACY

Time for gazing (sec)	No.of samples	No.of hidden neurons	Training %	Error %	Accuracy
10	120	10	60	0	100%
10	120	10	70	27.77e-0	72.2%
10	120	15	60	16.66e-0	83.3%
10	120	15	70	3.1561e-0	88.9%

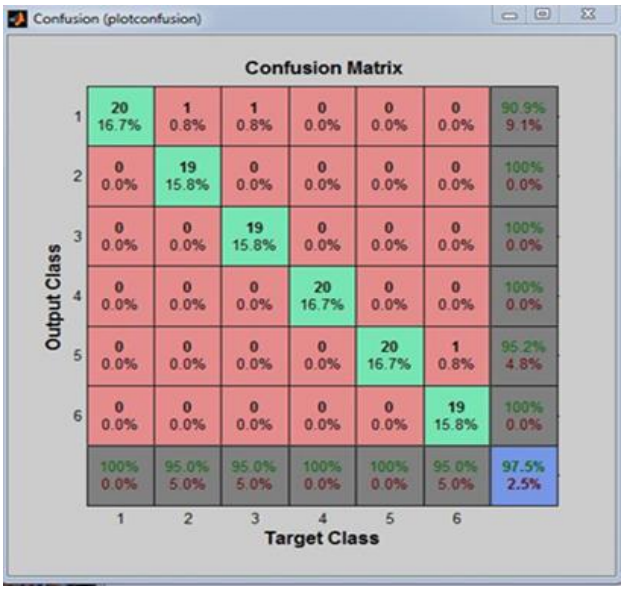

Fig. 12. Confusion matrix for Testing

VI. CONCLUSION

Gaze is an important tool and can be used in many applications such as market research, academic and scientific research and in medical field (helpful in studying neurological, vision and communication disorders) and also in gaining some insight about how people view images and animations. One of the major drawbacks of gaze estimation is the cost involving equipment such as high performance cameras followed by the burden of wearing different equipment's by user. In this report, we proposed a new gaze estimation procedure which makes use of a simple web-cam that reduces the cost considerably. The eye tracking accuracy is found to be good. Even though the gaze estimation is not entirely accurate, it is an acceptable step in the direction of reducing the cost of gaze estimation that makes it simpler to use. If the user is sitting at a distance more than 100cm, the system fails to track the eyes.

REFERENCES

[1] Venugopal, Divya, Joseph Amudha, and C. Jyotsna. "Developing an application using eye tracker." 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, 2016.

[2] Pavani, M. Lakshmi, AV Bhanu Prakash, MS Shwetha Koushik, J. Amudha, and C. Jyotsna. "Navigation Through Eye-Tracking for Human–Computer Interface." In Information and Communication Technology for Intelligent Systems, pp. 575-586. Springer, Singapore, 2019.

[3] Huang, Jeffrey, and Harry Wechsler. "Eye detection using optimal wavelet packets and radial basis functions (rbfs)." International Journal of Pattern Recognition and Artificial Intelligence 13.07 : 1009-1025,1999

[4] Dutta, Palash, and Debotosh Bhattacharjee. "Face detection using generic eye template matching." 2014 2nd International Conference on Business and Information Management (ICBIM). IEEE, 2014.

[5] Akashi, T., Wakasa, Y., Tanaka, K., Karungaru, S., & Fukumi, M, "Using genetic algorithm for eye detection and tracking in video sequence". Journal of Systemics, Cybernetics and Informatics, 5(2), 72-78, 2007

[6] Nishimura, Toshiya, et al. "Eye interface for physically impaired people by genetic eye tracking." SICE Annual Conference 2007. IEEE, 2007.

[7] Al-Mamun, Hawlader Abdullah, et al. "Eye detection in facial image by genetic algorithm driven deformable template matching." International Journal of Computer Science and Network Security 9.8 : 287-294, 2009

[8] Perez, Claudio A., et al. "Face and iris localization using templates designed by particle swarm optimization." Pattern recognition letters 31.9 : 857-868, 2010

[9] Al-Rahayfeh, Amer, and Miad Faezipour. "Eye tracking and head movement detection: A state-of-art survey." IEEE journal of translational engineering in health and medicine 1 : 2100212-2100212, 2013

[10] Santos, Rafael, et al. "Eye gaze as a human-computer interface." Procedia Technology 17 : 376-383, 2014

[11] Orman, Zeynep, Abdulkadir Battal, and Erdem Kemer. "A study on face, eye detection and gaze estimation." IJCSES2.3 : 29-46, 2011

[12] Akashi, Takuya, et al. "Using genetic algorithm for eye detection and tracking in video sequence." Journal of Systemics, Cybernetics and Informatics 5.2 : 72-78, 2007

[13] Nandakumar, Hitha, and J. Amudha. "A comparative analysis of a neural-based remote eye gaze tracker." 2014 International Conference on Embedded Systems (ICES). IEEE, 2014.

[14] Torricelli, Diego, et al. "A neural-based remote eye gaze tracker under natural head motion." Computer methods and programs in biomedicine 92.1 : 66-78, 2008