



Detecting the Location of Eyes on the Front Facial Images Using Two-Phase Method

Noopur Desai¹, Vidhi Dharia², Ishani Shah³, Ghanshyam Prajapati⁴

Students, Dept. of Computer Science Engineering, Shri S'ad Vidya Mandal Institute of Technology, Bharuch, India^{1,2,3}
Asst. Prof., Dept. of Computer Science and Engineering, Shri S'ad Vidya Mandal Institute of Technology, Bharuch, India⁴

ABSTRACT: Biometric identification has played an important role in many of real life applications. Out of these applications, eye detection is crucial whenever we are dealing with face analysis and recognition of facial features. Eye detection is somewhat difficult due to multiple variations in human faces, size of eyes and due to spectacles. This paper attempts to detect eyes in human images. The proposed approach is divided into two phases. The first phase deals with the detection of face in the image. Once the face is detected, the second phase deals with the detection of eyes in the face detected. The results of the implemented algorithm are quite accurate.

Keywords: Eye detection, EyeMap, Biometric Identification

I. INTRODUCTION

Biometrics has evolved over the years and is one of the most researched topics in the field of image processing. The study of biometrics refers to any measurement or characteristic of the human that is distinctive and differs from person to person. This includes a person's voice, DNA, finger print, eye, face, etc. Such characteristics have always been used as access control methods.

The use of eye detection as biometric identification is often desired due to its accuracy and reliability. Another advantage is the eyes are protected and remain reliably unchanged for decades as compared to fingerprints that are constantly exposed and susceptible to damage. Also the scanners used for fingerprints need to be kept clean.

As one of the salient features of the human face, eye detection plays an important role in face recognition and facial expression analysis. As compared to other facial features, eyes are relatively stable feature and thus it is important to detect eyes before the detection of other facial features. In recent years, considerable amount of research has been carried out to detect, analyse and recognize various human faces and their features. Eye detection applications are used for eye-gaze tracking, iris detection, video conferencing, auto-stereoscopic displays, face detection and recognition, etc. Apart from this, eye detection can also be used in the field of security for login purposes and also by police for criminal records.

Research in the field of biometrics and particularly in the field of facial recognition have made it computationally feasible to define and carry out eye detection mathematically, automate the process and thus scan large volume of facial images in short period of time for easy and quick retrieval of data/features.

Another important aspect of detecting eyes is that the gaze can be determined. In humans, the gaze or the eye contact plays a very important role in determining various interesting features like interest or aggressiveness of the person. Common example of this is that a person when interested in looking at a particular thing makes his iris larger. Such type of eye detection leads to social learning.

This paper uses the method of E-map for detecting the eyes. Generally eye detection is done in two phases: Identifying the face and then detection of the eye from it.

II. RELATED WORK

A lot many researchers have implemented various methods over the years for the detection of eyes. Different implementations have been carried out based on texture, depth, shape and color information. Some of these characteristics have also been combined to get more accurate results.

Bhoi et al [1] has used template based eye detection method wherein the template is correlated with different regions of the face and the region that gives maximum resemblance with the template refers to eye region.

Sidibe et al [2] has proposed a method which first detects face regions based on the skin color and then the eyes are extracted within the detected faces.

Asteriadis et al [3] has used eye localization on a face based on geometrical information. A face detector is applied first and the edge map is extracted. A vector is assigned to every pixel, pointing to the closest edge pixel. Length and slope information for these vectors is consequently used to detect and localize the eyes.

Hassaballah et al [4] has proposed a method that is based on Gray Intensity Facial Images. The method is based on two facts : entropy of eye and darkness of the iris.

Another algorithm based on Gray Intensity Face without Spectacles has been proposed by Peng et al [5]. The authors have combined two existing techniques, feature based method and template based method.

Wang et al [6] have detected eyes in facial images with unconstrained background for grey-level images.

Qing Gu [7] has used the method of segmenting the human face and detecting the eyes. The method segments the face into numbered regions like face, mouth, eye and hair regions. Then the detection system in this project is implemented using color images with complex backgrounds under various lighting conditions.

III. PROPOSED APPROACH

We have divided our approach into two phases. Phase (a): detects the face from the input image and Phase (b): detects the eyes from the face.

IV. FLOW CHART

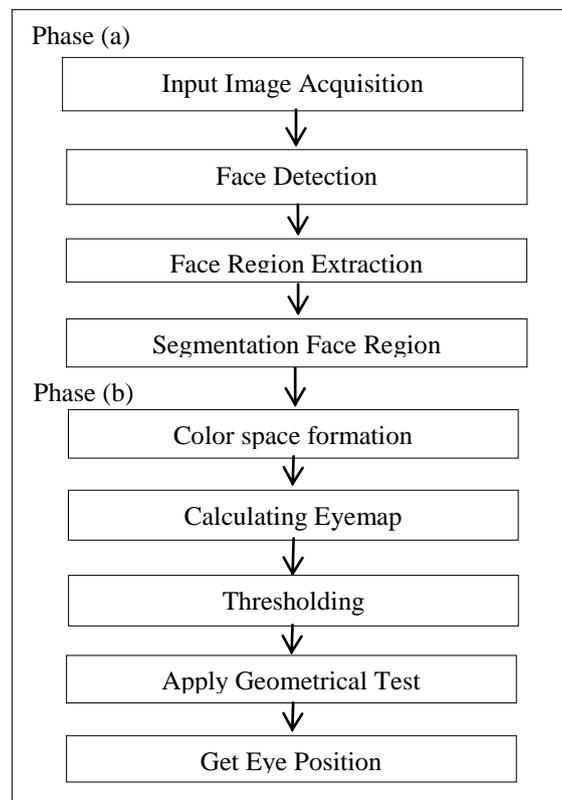


Fig. 1 Flowchart

A. Face Detection and Segmentation

Facial recognition methods have some steps in common namely: Input image acquisition, Detection of face from skin color, Extraction of face and Removal of other parts and finally Segmentation of the face region. In detection of face we have converted image to L^*a^*b form so that we can quantify visual differences. The L^*a^*b color space is derived from the CIE XYZ tristimulus values. The L^*a^*b space consists of a luminosity 'L*' or brightness layer,

chromaticity layer 'a*' indicating where color falls along the red-green axis, and chromaticity layer 'b*' indicating where the color falls along the blue-yellow axis.

For face detection, we have provided the front face of a person as an input image.

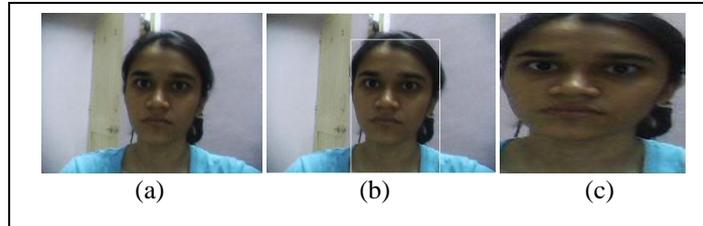


Fig.2 (a) Input Image, (b) Detected Face, (c) Segmented Face

Algorithm: Phase (a)

Input: Video Input RGB image.

Output: Detection of Segmented Face form input image.

Step 1: Convert the RGB image into another Colorspace L*a*b image.

Step 2: Find Second(a) & Third(b) Component of L*a*b Colorspace image

Step 3: Convert Gray-scale threshold to Black & White images using im2bw()

Step 4: Multiplication of a and b component.

Step 5: if (pixel value = 1)
 i.e. is a skin pixel
 else (pixel value = 0)
 i.e. is a non-skin pixel

Step 6: Find the different regions in the image by implementing connectivity analysis using 8-connected neighborhood and labeling that regions

Step 7: Finding the region having the largest area

Step 8: Extraction of face and Removal of other parts.

End of Algorithm: Phase (a)

Fig.3 An Algorithm for Phase (a)

B. Eye Detection from Detected Face

Once the face is detected, we have implemented eye detection with the help of EyeMap. For constructing EyeMap, we convert the RGB values to YCbCr color space. YCbCr is used as it is more suitable for skin colors. Eyes have relatively high response in Cb subspace while their response in Cr subspace is generally lower. We will calculate chrominance eyemap by $EMC = Cb^2 + N_Cr^2 + Cb/Cr$ [7] where Cb^2 , Cr^2 and (Cb/Cr) are normalized in range [0,255] while N_Cr is inverse of Cr (i.e. $255 - Cr$).

Luminance components have both dark and light pixels. We can thus design morphological operations to attenuate these areas. We will use erosion to emphasize dark areas in the image and dilation to emphasize light areas.

$$EMAP = 8100 * EML * EML * EMC^9 \quad [7]$$

An image is constructed using above formula for EyeMap from input cropped eye region image.

Once EyeMap is found, using Edge Detection Techniques, we mark the edges and fill the holes, we find the centroid in the image to get eye detected and mark the objects with red circle.

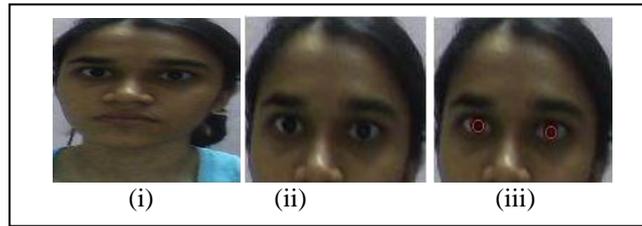


Fig. 4 (a) Segmented Face, (b) Cropped Face, (c) Detected Eye

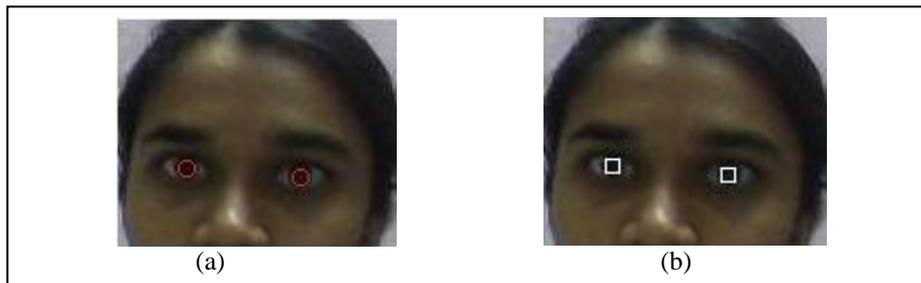


Fig. 5 (a) Detected Eyes (b) Manual Result by Mouse Click

Algorithm: Phase (b)
Input: Segmented Face image.
Output: Detected Eyes.

Step 1: Extraction Eye Portion using im2crop().

Step 2: Convert Cropped eye portion in to another Colorspace YCbCr image.

Step 3: Finding Y, Cb, Cr component of YCbCr colorsapce image.

Step 4: Calculating EMC (EyeMap for Chrominance) using formula:
 $EMC1 = Cb^2 + (Cr \text{ complement})^2 + (Cb/Cr) EMC = EMC1^9$

Step 5: Convert Y into Dilation and Erosion using imdilate () and imerode().

Step 6: Calculating EML (EyeMap for Luminance) using formula:
 $EML = 2 * \text{Dilation} - 2 * \text{Erosion}$

Step 7: Calculating EMAP (EyeMap) using formula:
 $EMAP = 8100 * EML * EML * EMC$

Step 8: Finding edge using sobel method.

Step 9: Fill holes in regionprops using imfill() function.

Step 10: Find Eye portion using Regionprops function in Centroid method.

Step 11: Plot circle using roundness metric formula:
 $metric = 4 * \pi * \text{area} / \text{perimeter}^2;$

Step 12: mark objects using threshold=0.95 with a Red circle.
 if (metric > threshold)
 centroid = stats(k).Centroid;
 plot(centroid(1),centroid(2),'ro');
 end

End of Algorithm: Phase (b)

Fig. 6 An Algorithm for Phase (b)

V. COMPARATIVE STUDY WITH MANUAL RESULTS

We detected the eyes using our algorithm and compared the results with manual plotting of eyes on the same image. Table (1) shows the exact positional coordinates of the eyes, in terms of x and y coordinates for two dimensional space, plotted using a mouse and the last column of the the table gives the error calculated from the actual obtained results as:

$$E = \text{SQRT}[(x_m - x_o)^2 + (y_m - y_o)^2]$$

here E is an error term, SQRT is square root of value given in parenthesis, (x_m, y_m) is manual extreme point while (x_o, y_o) is an extreme point using our method. As Figure shows the line chart for the comparative study of extreme points are calculated manually and by these method. While the error term is shown point-wise in figure next using bar chart. Less error term yields more accurate results. For an input image, we get better results in findings of extreme points. More experimental results are shown and described in the next section.

manual x coordinate x _m	x co-ordinate for our method x _o	manual y coordinate y _m	y co-ordinate for our method y _o	Error E
	28.5	61	57	3.97
51	57	101	101	6
70	71.5	104	103	1.12
22	26	52	55	2.65
52	53.5	53	62	8.87
164	162	191	191	2
46	67	105	117	17.2

manual x coordinate x _m	x co-ordinate for our method x _o	manual y coordinate y _m	y co-ordinate for our method y _o	Error E
78	77.5	63	63.5	0
145	145	48	99	51
176	166.5	104	108	8.62
70	67.79	59	58.82	2.2
132	133	60	64	3.87
305	304	187	187	1
124	139	109	118	12

Table 1 : Manual Results and Results by our Method for Right Eye

Table 2 : Manual Results and Results by our Method for Left Eye

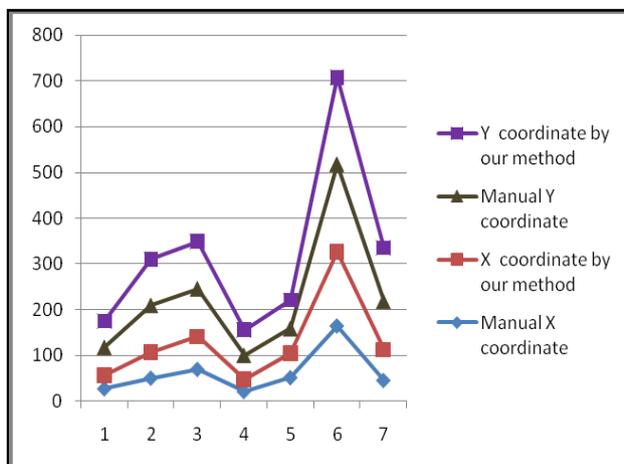


Fig. 7 Line Chart for Comparative Study of Right Eye

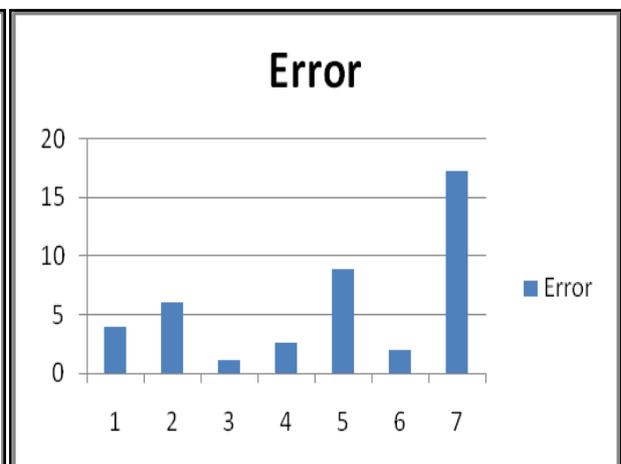


Fig. 8 An Error Term Showing Bar Chart for Right Eye

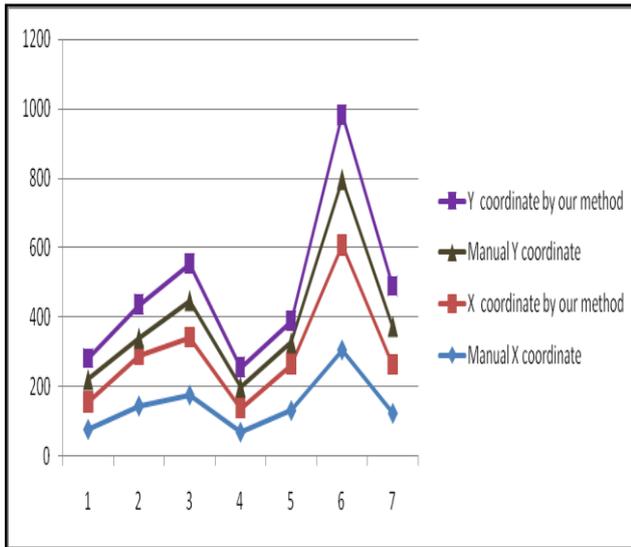


Fig. 9 Line Chart for Comparative Study of Left Eye



Fig. 10 An Error Term Showing Bar Chart for Left Eye

VI. MORE EXPERIMENTAL RESULTS WITH PERFORMANCE ANALYSIS OF THE PROPOSED METHOD

Following are some more results of our method. First five images are taken through webcam, in that images first three images are having poor lightning condition still accurate results are obtained. Next two are in better lightning condition. The remaining four are digital images and in that accurate results are not found. These results conclude our method is more efficient for real world webcam images for proper lightning condition.

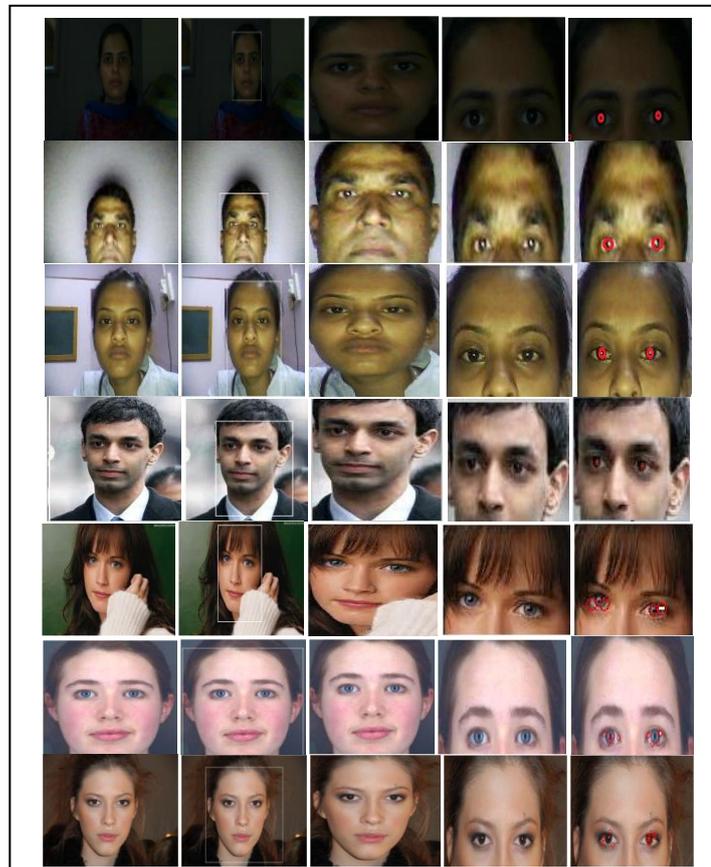


Fig. 11 Some More Results
www.ijirccce.com



VII. CONCLUSION

In this method, we have implemented eye detection using EyeMap. The method is easy to implement as it does not require any complex mathematical calculation or previous knowledge about the eye. Also it is a simple method and can be easily implemented by hardware. In order to improve the results, we need to use a database of eyes that can provide with a wide variety of eyes in different environment and in different faces. Limitation to this method lighting environment. Specific lighting environment is required to get accurate results.

REFERENCES

- [1] Nilamani Bhoi, Mihir Narayan Mohanty, 'Template matching Eye Detection in facial image', International Journal of Computer Applications (0975 – 8887), Volume 12– No.5, December 2010.
- [2] D. Sidibe, P. Montesinos, S. Janaqi, 'A simple and efficient eye detection method in color images', International Conference Image and Vision Computing, New Zealand 2006.
- [3] S. Asteriadis, N. Nikolaidis, A. Hajdu, I. Pitas, 'An Eye Detection Algorithm Using Pixel to Edge Information', Department of Informatics, Aristotle University of Thessaloniki, Box 451, 54124 Thessaloniki, Greece.
- [4] M. Hassaballah, Kenji Murakami and Shun Ido, 'Eye and Nose Fields Detection From Gray Scale Facial Images', International Association of Pattern Recognition Conference on Machine Vision Applications, June 13-15, 2011, Nara, Japan.
- [5] Kun Peng, Liming Chen, Su Ruan, Georgy Kukharev "A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles" Journal of Computer Science & Technology, October 2005.
- [6] Qiong Wang, Jingyu Yang, 'Eye Detection in Facial Images with Unconstrained Background', Journal of Pattern Recognition Research, Received JUN 2, 2006 Revised Sep. 12, 2006, Published Sep. 25, 2006.
- [7] Qing Gu, 'Finding and Segmenting Human Faces', Department of Information Technology, Uppsala, February 2008.
- [8] Antonio Haro, Myron Flickner, Irfan Essa, 'Detecting and Tracking Eyes By Using Their Physiological Properties, Dynamics and Appearance', Appears in Proceedings of CVPR 2000 pp. 163-168.
- [9] M. Hassaballah, T. Kanazawa, S. Ido, S. Ido, 'Efficient eye detection method based on grey intensity variance and independent component analysis,' IET Computer Vision, vol. 4, no. 4, pp. 261-271, 2010.
- [10] T. Kawaguchi and M. Rizon, 'Iris detection using intensity and edge information,' Pattern Recognition, vol. 36, pp. 549-562, 2003.
- [11] J. Canny, 'A Computational Approach to Edge EE Transactions on Pattern Analysis and Machine Intelligence, vol 8, no. 6, 1986.
- [12] W. min Huang and R. Mariani, 'Face detection and precise eyes location', in Proceedings International Conference on Pattern Recognition (ICPR'00), 2000.

BIOGRAPHY

Noopur Desai, Vidhi Dharja and Ishani Shah are pursuing their Bachelors of Engineering in Computer Science and Engineering from Shri S'ad Vidya Mandal Institute of Technology affiliated to Gujarat Technological University. Their research interests include Digital Image Processing and Biometric Identification.

Ghanshyam Prajapati has completed his Bachelors of Engineering and Masters of Engineering and is currently working as Assistant Professor in Shri S'ad Vidya Mandal Institute of Technology affiliated to Gujarat Technological University. His research interests include Digital Image Processing, Biometric Identification and Pattern Recognition.