Face Detection for Real time images in complex background using HOG

Dr. N. Sasikala[1], M.Pooja[2], T.Vyshnavi[3], CH.Sudha patel[4], N.Harsha vardhan[5], Sumera[6]
[1]*Associate Professor, Kamala Institute of Technology & Science, Singapur*
[2,3,4,5,6]*Student, Kamala Institute of Technology & Science, Singapur*

Abstract **- The goal of this paper is to determine whether or not there are any faces in the image and detection of eyes and upper body. While this appears to be a trivial task for human beings, it is very challenging task for computers. The difficulty associated with face detection can be attributed to many variations in scale, location, view point, illumination, occlusions, etc. In this we are going to study and understand the water shed algorithm. The face detection is performed on live acquired images without any application field in mind. Processes utilized in the system are white balance correction, skin like region segmentation, apply threshold to mask the skin region, draw bounding box to extract the face image, statistical distance facial feature extraction like mouth and eyes and face detection of a candidate. The tested system has acceptable performance to recognize faces within intended limits. System is also capable of detecting and recognizing multiple faces in live acquired images.**

I.INTRODUCTION

Face detection is concerned with finding whether or not there are any faces in a given image (usually in gray scale) and, if present, return the image location and content of each face. This is the first step of any fully automatic system that analyzes the information contained in faces (e.g., identity, gender, expression, age, race and pose). Face detection is the first step in various other applications, including face tracking, face analysis and face recognition. In the context of face analysis, face detection tells the face analysis algorithms which parts of an image (or video) to focus on when identifying age, recognizing gender, and analyzing emotions based on facial expressions. And when it comes to facial recognition, face detection is necessary for the algorithms to know which parts of an image (or video) to use to generate the faceprints that are compared with previously stored faceprints to establish whether or not there is a match. Face detection applications use algorithms that determine whether images are positive images (i.e., images with a face) or negative images (i.e., images without a face). To be able to do this accurately, the algorithms must be trained on huge datasets containing hundreds of thousands of face images and non-face images.

II.FACE DETECTION VS FACE RECOGNITION

FACE DETECTION

The definition of face detection refers to computer technology that is able to identify the presence of people's faces within digital images. In order to work, face detection applications use machine learning and formulas known as algorithms to detecting human faces within larger images. These larger images might contain numerous objects that aren't faces such as landscapes, buildings and other parts of humans (Legs, shoulders and arms). Face detection just means that a system is able to identify that there is a human face present in an image or video. Face detection has several applications, only one of which is facial recognition. Face detection can also be used to auto focus cameras. And it can be used to count how many people have entered a particular area.

FACE RECOGNITION

One of the most important applications of face detection is facial recognition. Face recognition describes a biometric technology that goes way beyond recognizing when a human face is present. It actually attempts to establish whose face it is. The process works using a computer application that captures a digital image of an individual's face (sometimes taken from a video frame) and compares it to images in a database of stored records.

III.ALGORITHM FOR FACE DETECTION

1. Reading of Real time image
2. Sizing image Adjust the size such that each image size is same.
3. Light compensation

Under different light conditions like day light, evening light, night light, by using trial and error method adaptive threshold value is chosen. Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. Thresholding methods are categorized into six groups based on the information the algorithm manipulates. We analyse various image pre-processing methods, categorize them, and express their formulas under a uniform notation.

4. Extraction of skin tone

The first step in the face detection algorithm is using skin segmentation to reject as much “non-face” of the image as possible, since the main part of the images consists of nonskin color pixels. There are two ways of segmenting the image based on skin color: converting the RGB picture to YCbCr space or to HSV space. A YCbCr space segments the image into a luminosity component and color components, whereas an HSV space divides the image into the three components of hue, saturation and color value. The main advantage of converting the image to the YCbCr domain is that influence of luminosity can be removed during our image processing. In the RGB domain, each component of the picture (red, green and blue) has a different brightness. However, in the YCbCr domain all information about the brightness is given by the Y component, since the Cb (blue) and Cr (red) components are independent from the luminosity.

5. Removal of noise

Filtering of image is an important process done in image processing. It can be done for noise removal, blur removal, edge detection etc. Linear and non-linear filters are the algorithms which are used for filtering. Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

The filter used in this project is Median filter.

6. Splitting the image

Bounding box are used to split the image. We consider mathematical geometry box may be square or rectangle to which height and width are given manually. A bounding box is an imaginary rectangle that serves as a point of reference for object detection and creates a collision box for that object.

7. Implementing in the channel

8. Identification of facial features

In this paper the descriptor used is HOG. HOG, or Histogram of Oriented Gradients, is a feature descriptor that is often used to extract features from image data. It is widely used in computer vision tasks for object detection. The HOG descriptor focuses on the structure or the shape of an object. Now you might ask, how is this different from the edge features we extract for images. In the case of edge features, we only identify if the pixel is an edge or not. HOG is able to provide the edge direction as well. This is done by extracting the gradient and orientation (or you can say magnitude and direction) of the edges. Additionally, these orientations are calculated in ‘localized’ portions. This means that the complete image is broken down into smaller regions and for each region, the gradients and orientation are calculated. Finally, the HOG would generate a Histogram for each of these regions separately. The histograms are created using the gradients and orientations of the pixel values, hence the name ‘Histogram of Oriented Gradients’.

9. Checks for eye map and mouth map

Once the skin is detected the algorithm checks for the eye map and mouth map.

10. Detects Face

If the algorithm detects eye map and mouth map then the face is detected

IV.RESULT

The algorithm is implemented in MATLAB using Image Processing Toolbox

Implementation of algorithm for complex background real-time image in day light

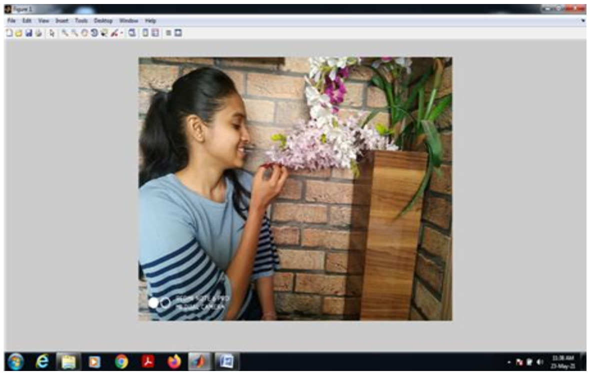

Figure:1 Reading of real time image

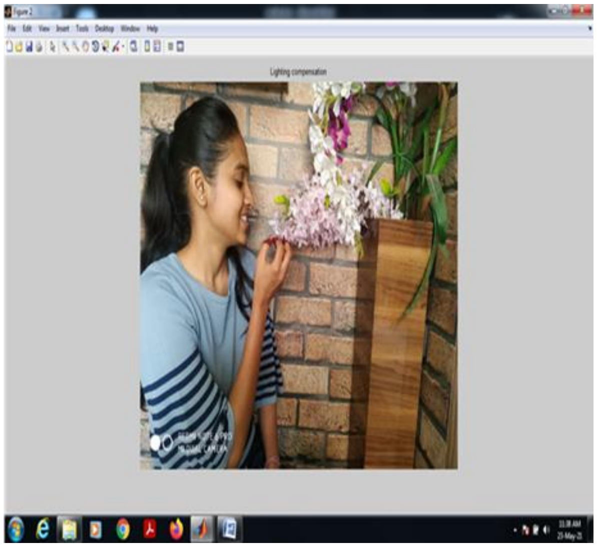

Figure:2 Light compensation

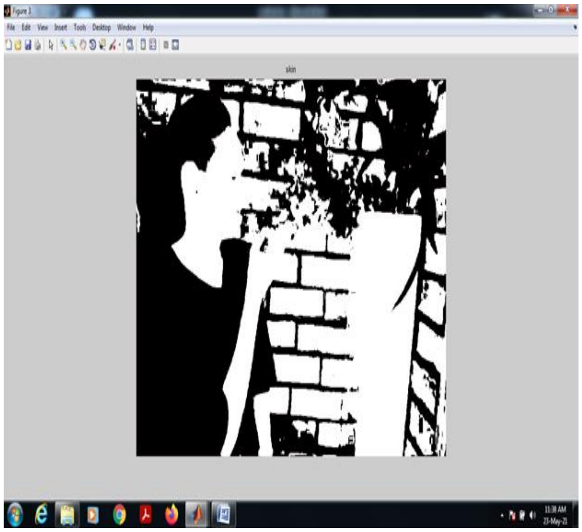

Figure:3 Extraction of skin tone

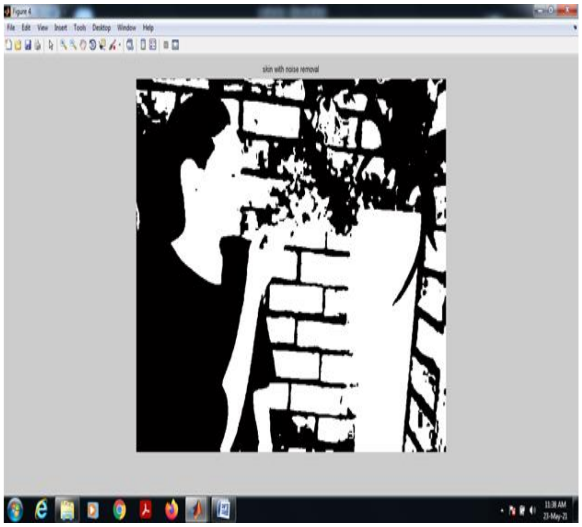

Figure:4 Removal of noise

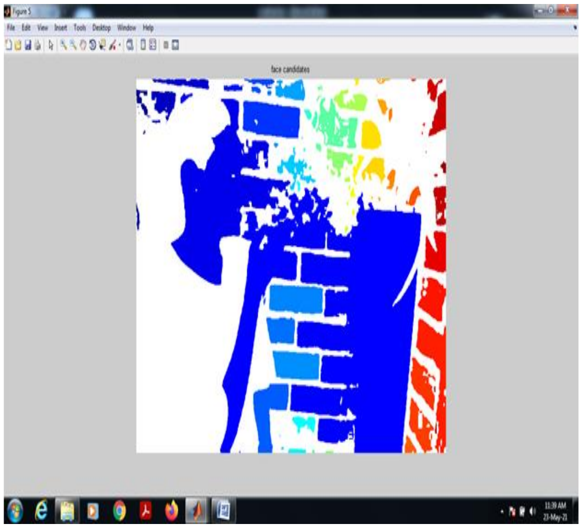

Figure:5 Face candidate

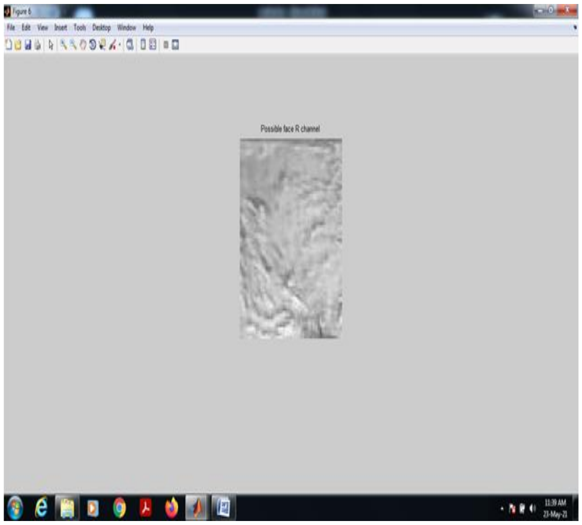

Figure:6 Possible face R channel

In the above bounding box, there is no skin found, so program doesn’t check for the presence of mouth and eyes.

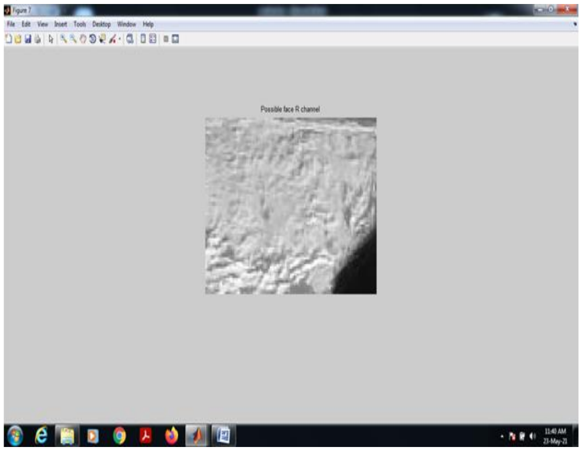

Figure:7 Possible face R channel

In the above bounding box, there is no skin found, so program doesn't check for the presence of mouth and eye.

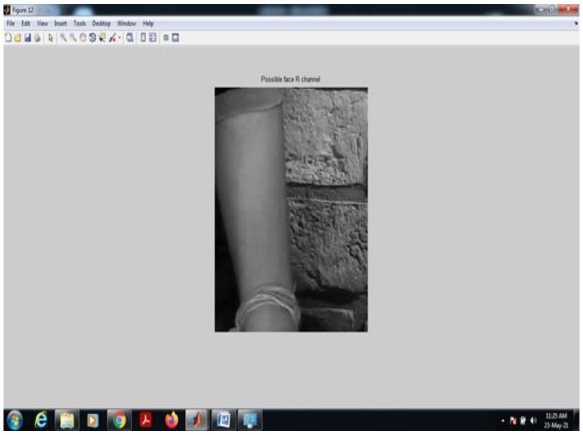

Figure:8 Possible face R channel

In the above bounding box, skin is found, so program checks for the presence of mouth and eyes, but they are not detected since in the bounding box there is no mouth or eyes.

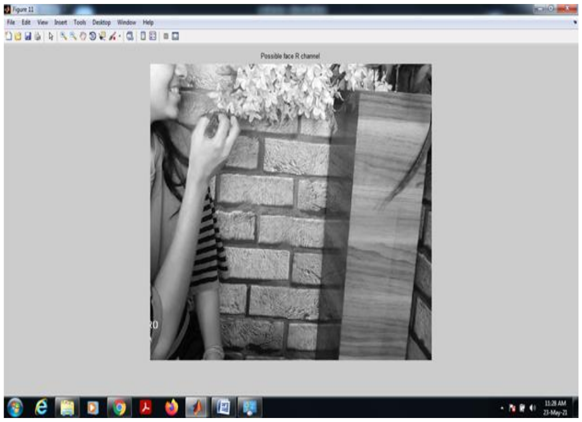

Figure:9 Possible face R channel

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes.

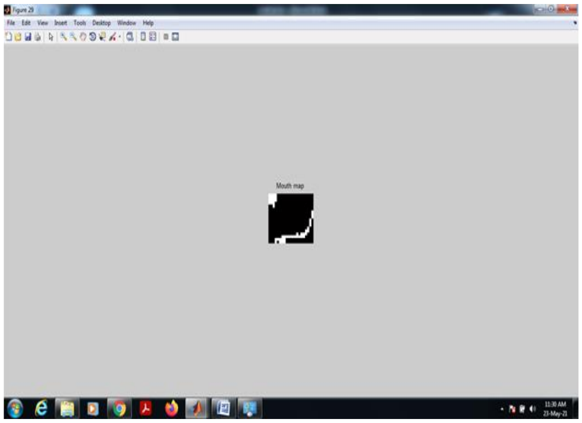

Figure:10 Mouth map

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes. But there are no eyes detected only mouth is detected.

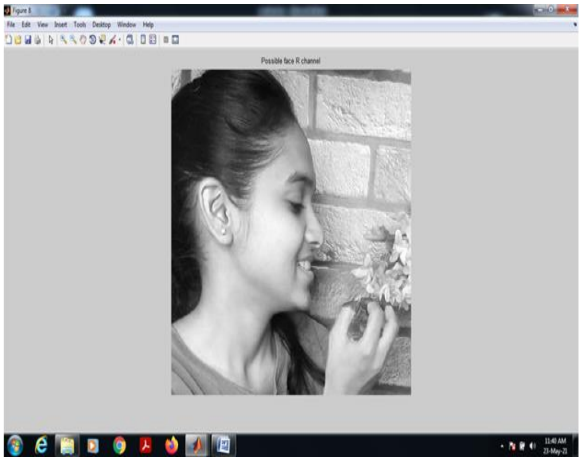

Figure:11 Possible face R channel

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes. Mouth and eyes are detected and hence face is detected.

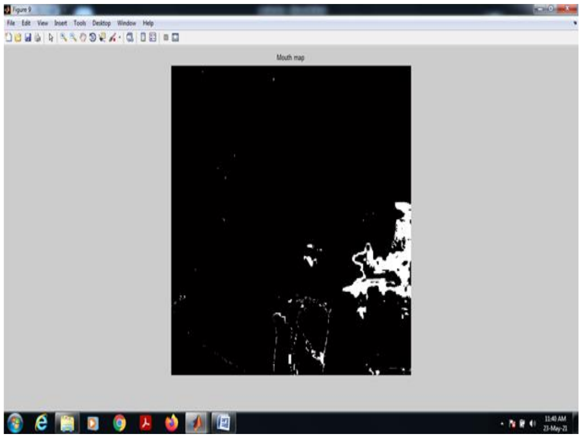

Figure:12 Mouth map

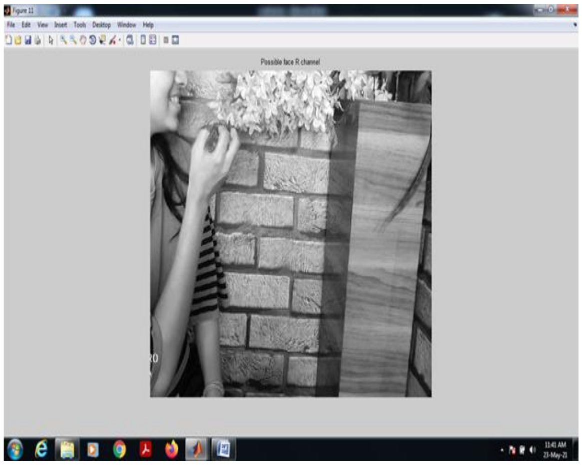

Figure:13 Eye map

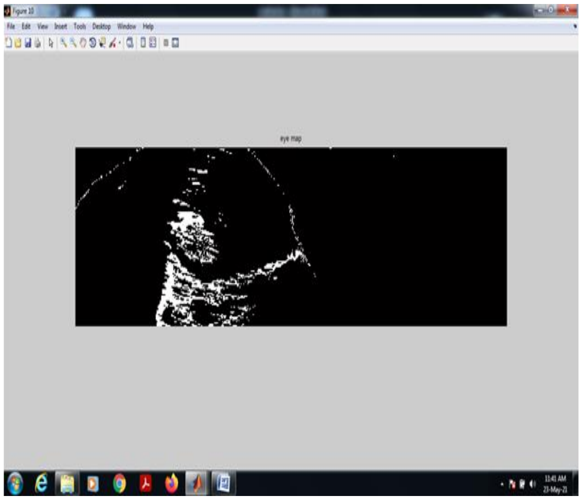

Figure:14 Possible face R channel

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes.

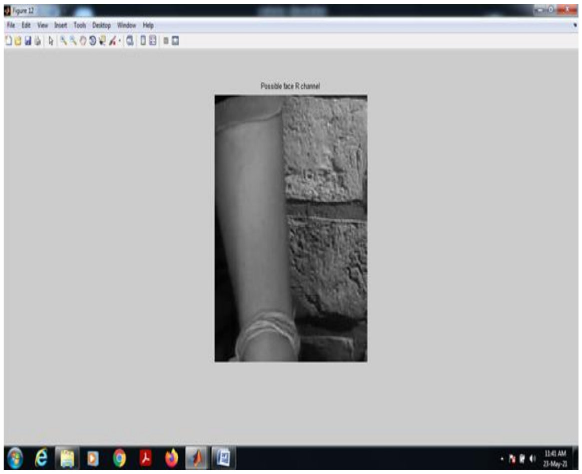

Figure:15 Possible face R channel

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes.

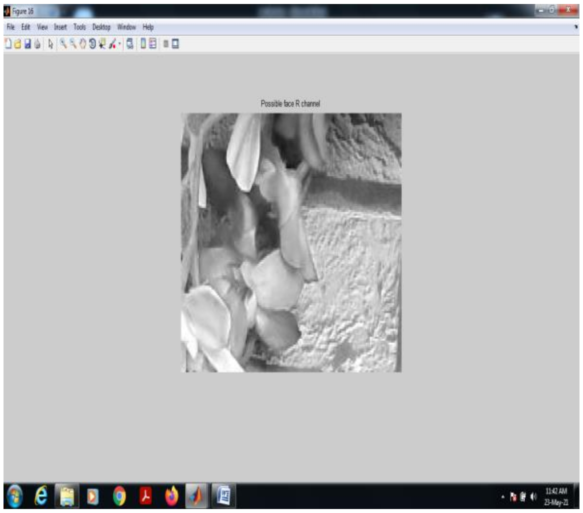

Figure:16 Possible face R channel

In the above bounding box, there is no skin found, so program doesn’t check for mouth and eyes

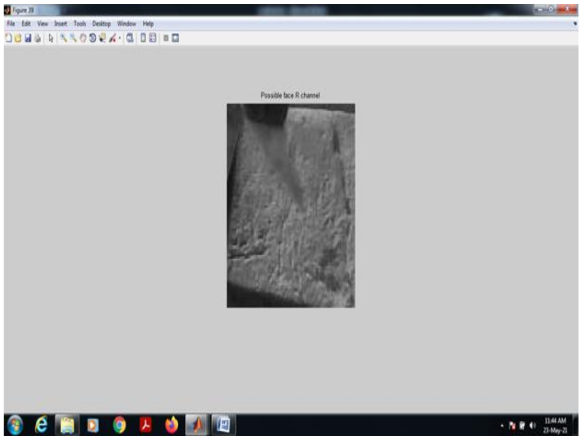

Figure:17 Possible face R channel

In the above bounding box, there is no skin found, so program doesn’t check for mouth and eyes

Implementation of algorithm for complex background real-time image in afternoon light

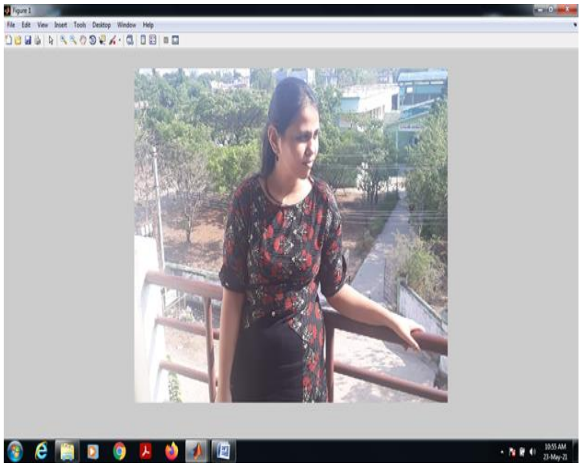

Figure:18 Reading of real time image

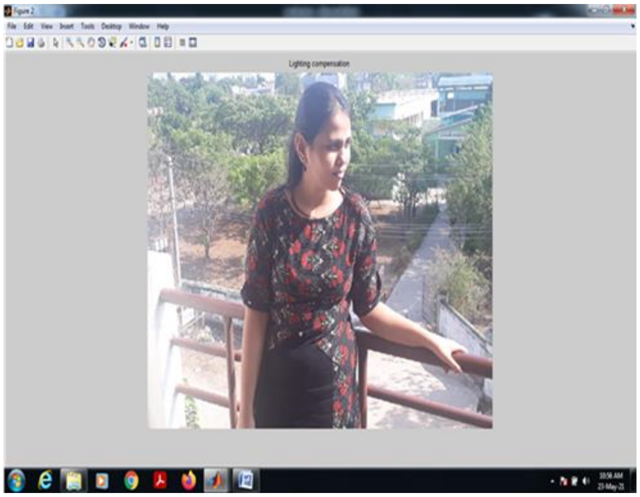

Figure:19 Light compensation

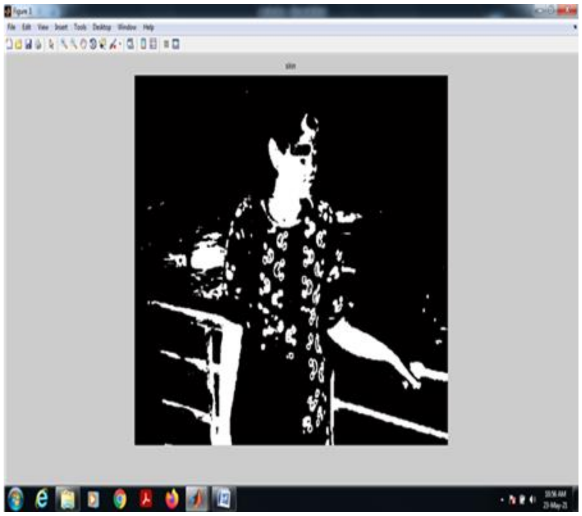

Figure:20 Extraction of skin tone

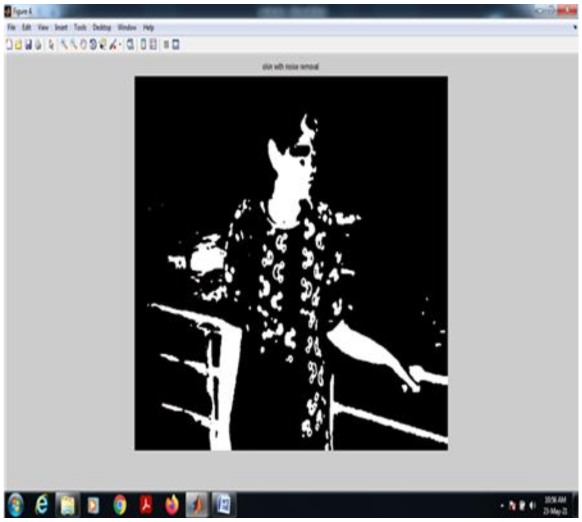

Figure:21 Removal of noise

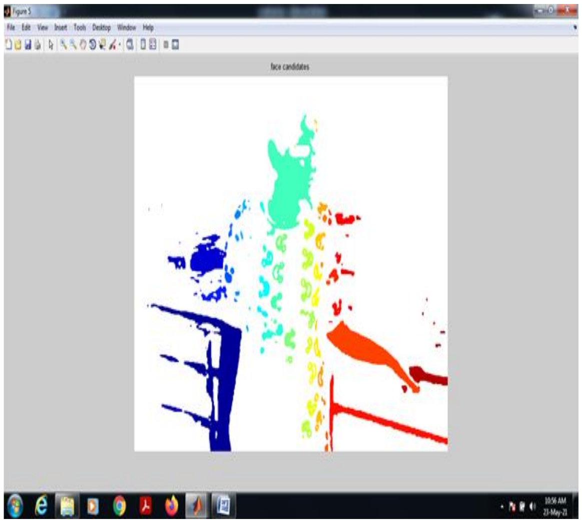

Figure:22 Face candidate

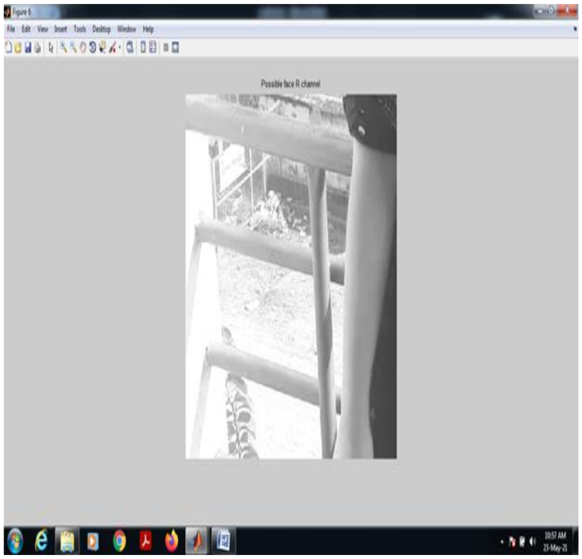

Figure:23 Possible face R channel

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes.

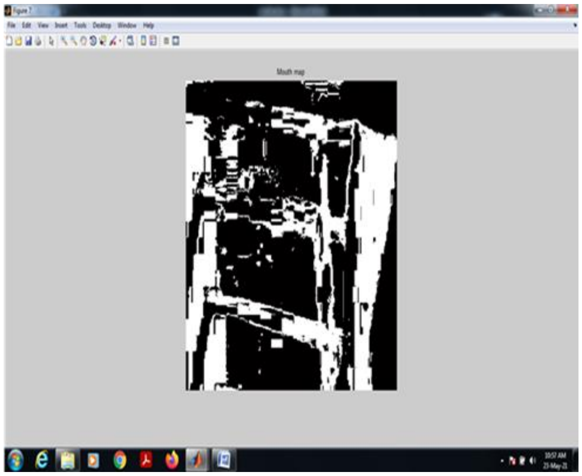

Figure:24 Mouth map

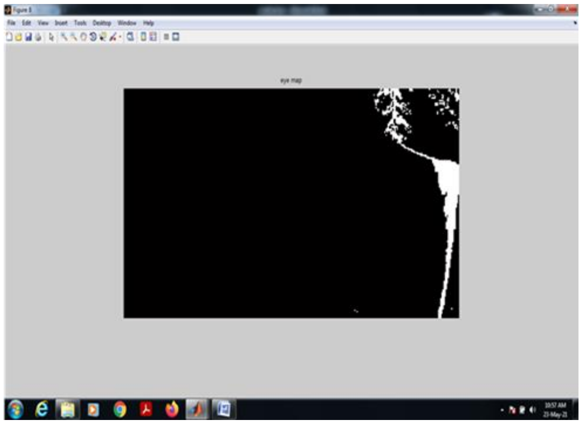

Figure:25 Eye map

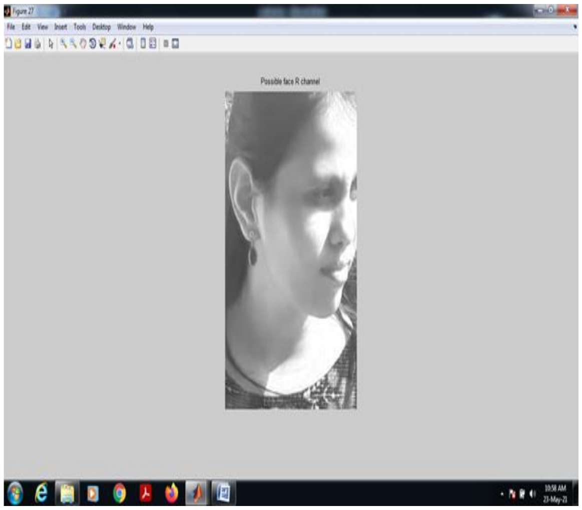

Figure:26 Possible face R channel

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes. Mouth and eyes are detected and hence face is detected.

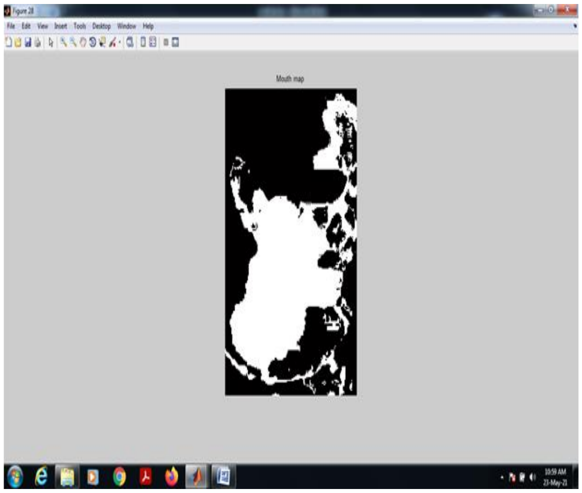

Figure:27 Mouth map

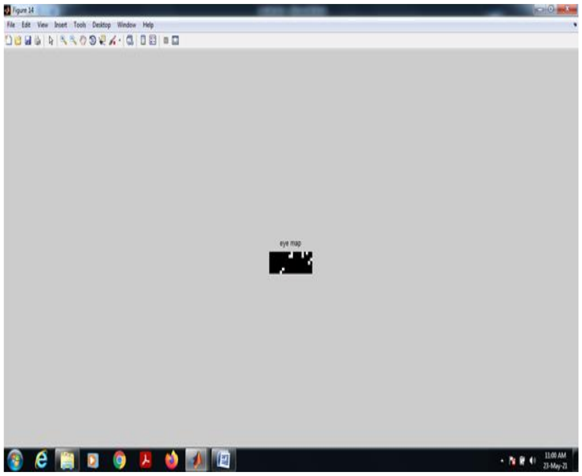

Figure:28 Eye map

Implementation of algorithm for complex background real-time image in evening light

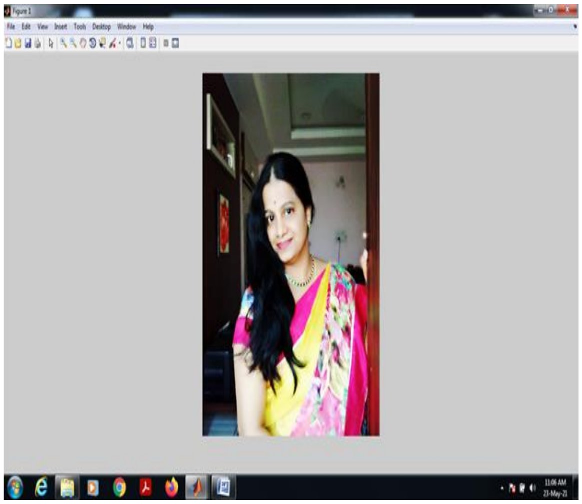

Figure:29 Reading of real time image

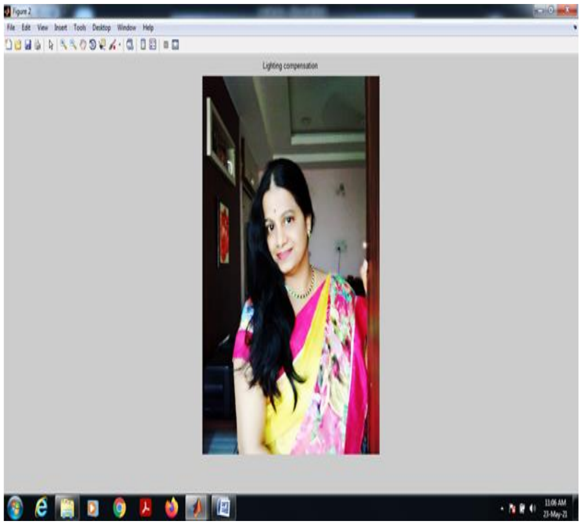

Figure:30 Light compensation

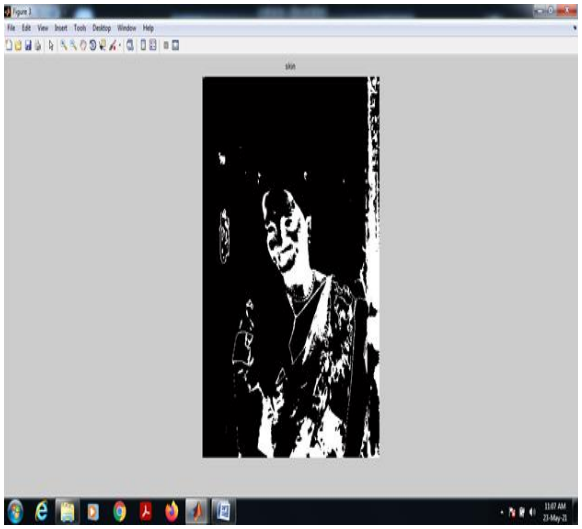

Figure:31 Extraction of skin tone

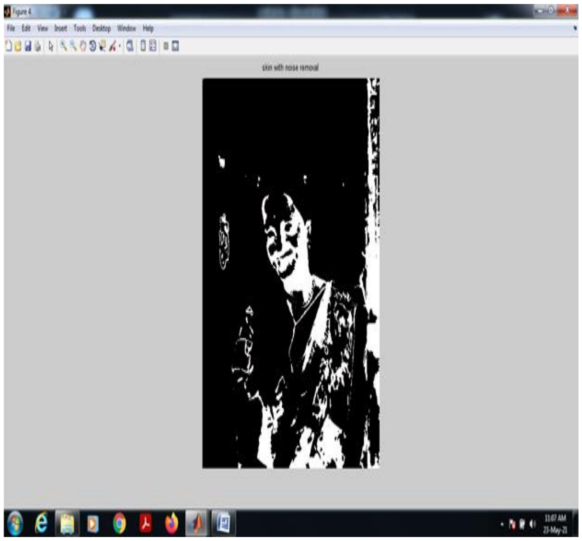

Figure:32 Removal of noise

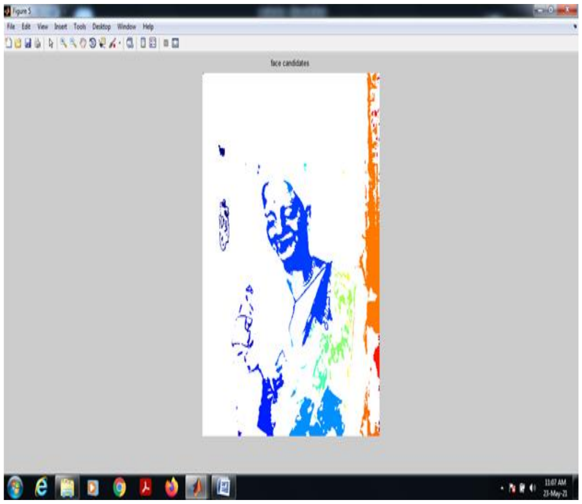

Figure:33 Face candidate

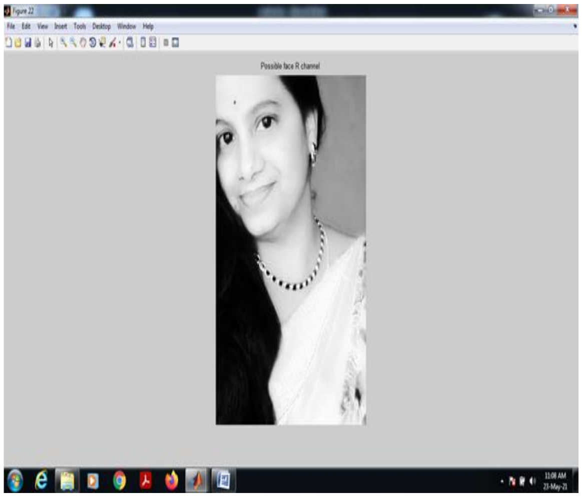

Figure:34 Possible face R channel

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes. Mouth and eyes are detected and hence face is detected.

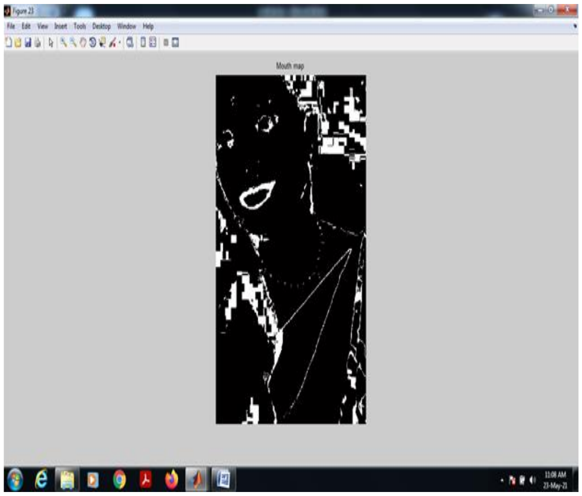

Figure:35 Mouth map

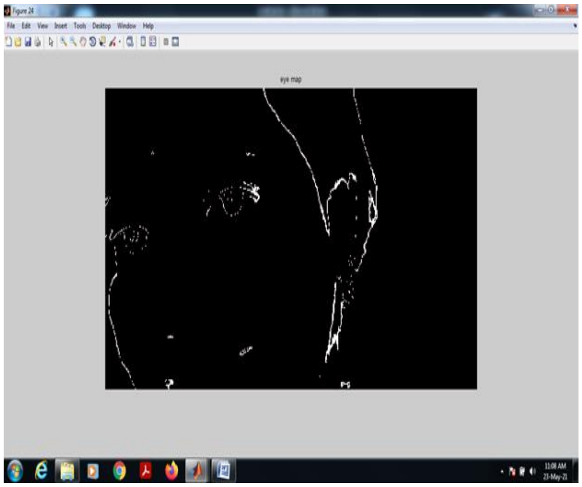

Figure:36 Eye map

Implementation of algorithm for complex background real-time image in night light

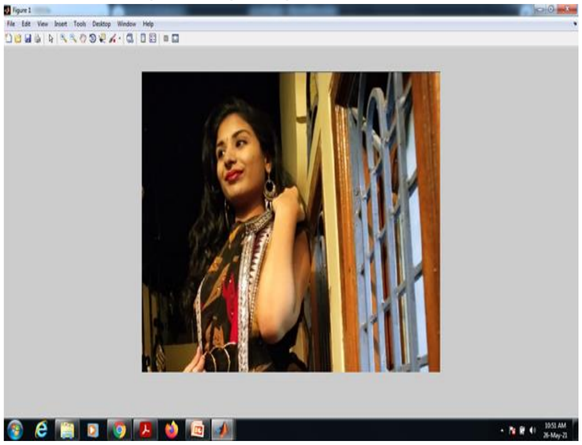

Figure:37 Reading of real time image

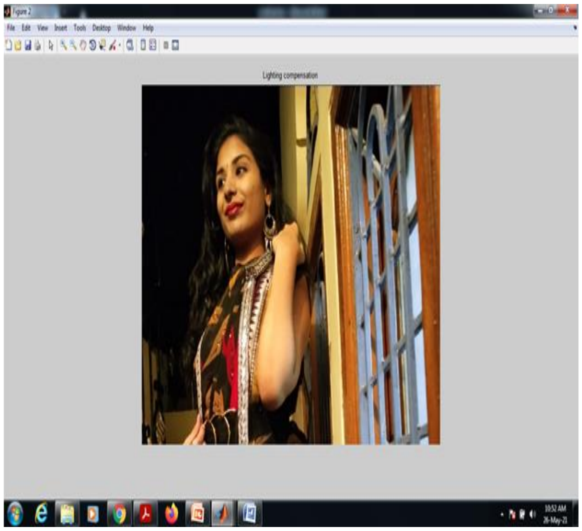

Figure:38 Light compensation

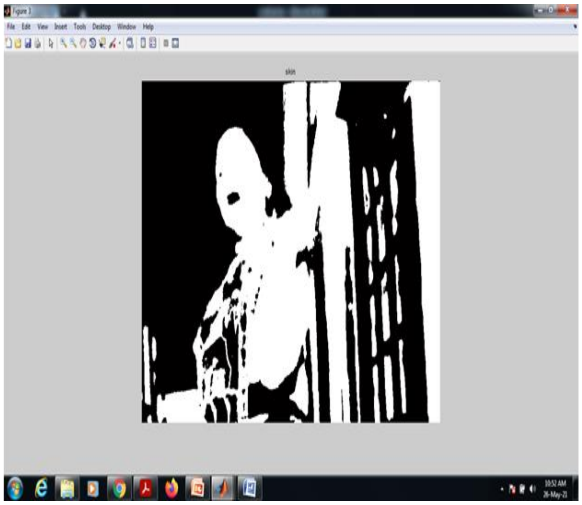

Figure:39 Extraction of skin tone

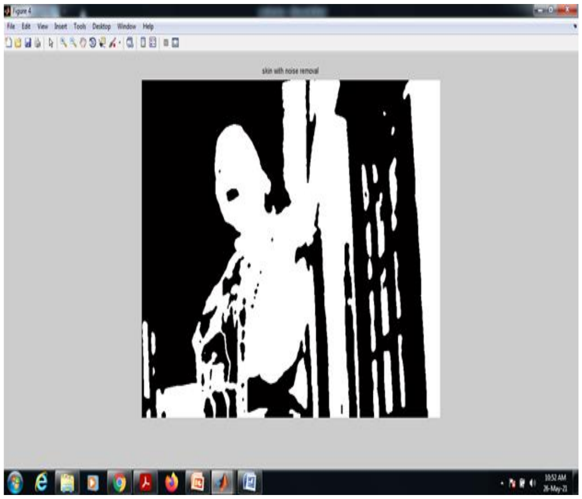

Figure:40 Removal of noise

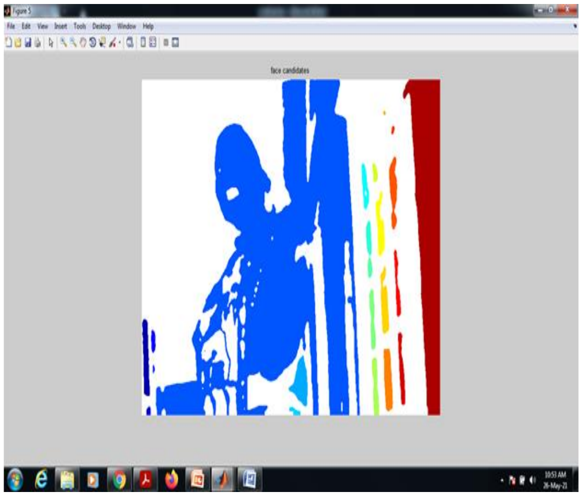

Figure:41 Face candidate

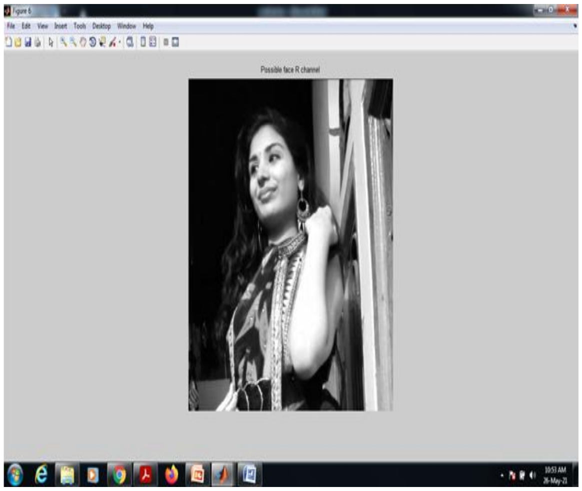

Figure:42 Possible face R channel

In the above bounding box, skin tone is detected, so algorithm checks for the presence of mouth and eyes. Mouth and eyes are detected and hence face is detected.

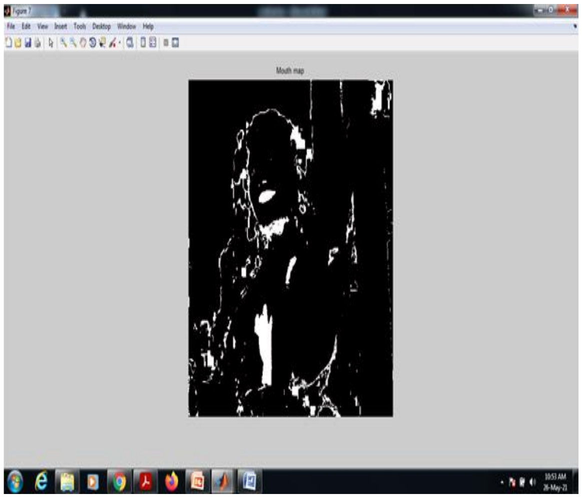

Figure:43 Mouth map

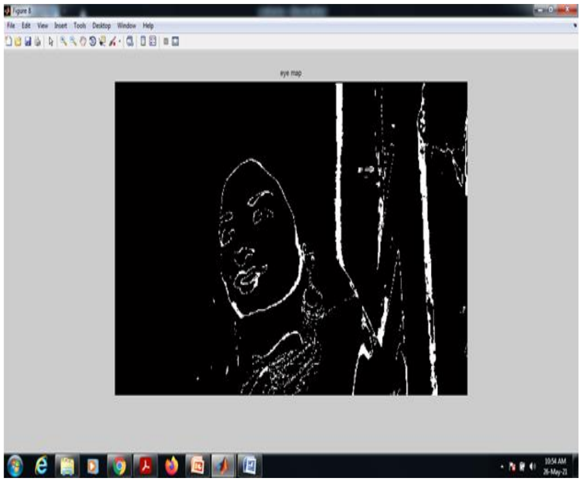

Figure:44 Eye map

VI.ACKNOWLEDGMENT

Much appreciation is extended with profound gratitude and respect to the management and principal of KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE for their constant support, timely suggestions and indispensable help which made to publish this paper. Our apologies are extended to anyone whom we may have inadvertently missed.

VI.CONCLUSION

In this paper we have presented a face detection algorithm for color images using skin-tone color model and facial features. We overcome the difficulty of detecting the low-luma and high-luma skin tones by applying a nonlinear transform to the YCbCr color space. Our method detects skin regions over the entire image, and then generates face candidates based on the spatial arrangement of these skin patches. The algorithm constructs eye/mouth/boundary maps for verifying each face candidate. Detection results for several photo collections have been presented. Our goal is to design a system that detects faces and facial features, allows users to edit detected faces, and uses the facial features as indices for retrieval from image and video databases.

REFERENCES

[1] https://www.tutorialspoint.com/dip/applications_and_usage

[2] https://www.analyticsinsight.net/biometrics-and-computer-vision-merge-to-create-limitless-possibilities

[3] https://www.nec.co.nz/market-leadership/publications-media/face-detection-vs-facial-recognition-whats-the-difference

[4] https://in.mathworks.com/products/image

[5] https://en.m.wikipedia.org/wiki/Histogram_of_oriented_gradients#:~:text=The%20histogram%20of%20oriented%20gradients,localized%20portions%20of%20an%20image

[6] https://en.m.wikipedia.org/wiki/Thresholding_(image_processing)

[7] https://searchenterpriseai.techtarget.com/definition/face-detection

[8] https://www.facefirst.com/blog/face-detection-vs-face-recognition

[9] https://en.wikipedia.org/wiki/Computer_vision

[10] https://en.wikipedia.org/wiki/Face_detection

[11] https://www.google.com/imgres?imgurl=https://images-na.ssl-images-amazon.com/images/I/51SNiDGflNL.jpg&imgrefurl=https://www.amazon.in/Digital-Image-Processing-Rafael-Gonzalez/dp/013168728X&docid=yQzvpc1beAKS8M&tbnid=s92LjwFBNmYOdM&vet=1&source=sh/x/im

[12] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Third Edition

[13] S.N. ALAM, S.S. ALAM, Understanding Matlab: A Textbook for Beginners