FACE DETECTION USING WEBCAM

NAME: G REVANTH REDDY

FACE DETECTION USING WEBCAM

ABSTRACT

Biometrics are a way to measure a person's physical characteristics to verify their identity. Face detection is used to detect the faces in the images or live video. It is a part of Object Detection. It is used to detect the faces in real time for surveillance and tracking of a person or objects. In the past few years face recognition, appreciated as one of the most promising applications in the field of image analysis. Object detection is one of the computer technologies, which connected to the image processing and computer vision and it interacts with detecting instances of an object such as human faces, building, tree, car, etc.

The primary aim of face detection is to determine whether there is any face in an image or not through a webcam. Face Detection is the first and essential step for face recognition, and it is used to detect faces in the images. Facebook is also using face detection algorithm to detect faces in the images and recognise them.

TABLE OF CONTENTS

LIST OF PLATES

1. INTRODUCTION

Biometrics is the science and technology of measuring and analyzing biological data. In Information Technology, biometrics refers to technologies that measure and analyse human body characteristics, such as DNA, fingerprints, eye retinas and irises, voice patterns, facial patterns and hand measurements, for authentication purposes. Face Detection: Finding the faces in an image/frame. Face Recognition: Recognizing the face in an image/frame.

Face detection is an important step in face recognition, which is one of the more representative and classic application in computer vision. Face is one of the physiological biometrics based on stable features. Four types of face detection are developed. They are knowledge based methods, template marching, invariant feature methods and learning based methods. In this project, learning based methods are deployed: AdaBoost Method.

The aim is to develop and propose a system to detect human faces in digital images effectively, no matters what person's ethnic, pose. Input images may be varied with face size, complex of background and illumination condition. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. It is used to detect faces in real time for surveillance and tracking of person or objects. It is widely used in cameras to identify multiple appearances in the frame. For Examples- Mobile cameras and DSLR's and Facebook is also using face detection algorithm to detect faces in the images and recognise them.

The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).

2. LITERATURE SURVEY

Face Detection is a technology to determine human face in videos and arbitrary (digital) images. The aim of face detection is detecting faces in any images or videos. If yes, it's reported the location on the images or videos.

Face detection can be regarded as a specific case of object-class detection. For the detection is to locate the face in the digital images/video stream, no matter what the pose, scale, facial expressions. In other words, face detection algorithms to handle pattern classification. It tasks to identify a given image to decides it has face or not.

Face Detection can be implemented by four methods: Knowledge based methods, Template matching, Invariant feature methods and Learning based methods. These methods will be introduced with the following:

1. **Knowledge based methods:** The models are used human knowledge to find a face patterns from the testing images. Based on the nature of human faces, algorithms scan the image from top-to-bottom and left-to-right order to find facial feature.
2. **Template marching:** The model is used several templates to find out the face class and extract facial features. Rules are pre-defined and decide whether there is face in the image.
3. **Invariant feature methods:** The model is bottom-up approaches and used to find a facial feature (eyebrows, nose), even in the presence of composition, perspective vary, so it is difficult to find a face real time using this method. Statistical models are developed to determine the faces. Facial features of human faces are: shape, texture, skin.
4. **Learning based methods:** The models are trained from a set of training set before doing detection. For the large amount of training data, it can be provided high accuracy recognition rate to resist variation, expression and pose of faces images.

Rajeswari. K, Sakthivel Punniakodhi, Shafiya Begum. R, "Face Detection and recognition using various frames of a video", International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 11, November – 2013. "This paper mainly focused on identifying a person or user from a video frame from the input video. The face detection performed by 3 different steps: 1. Skin color process. 2. Neural Network computing is performed. 3. Provide upright face detector. Mainly focused to solve problems of existing system such as illumination variation and pose variation by the use of local mapping analysis. **Local Mapping Analysis**: The neural network is used for training the images using unsupervised learning and in term it also produces a low dimensional and discredited representation of the training samples called as a map."

3. SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

Operating System	**:**	Windows 10 and below, Linux. Mac OS X, Unix
Programming Language	**:**	Python 3.6

Python

Python is high-level programming language like a Perl, Ruby etc. Which are used as a scripting language. It was conceived by Guido van Rossum in 1989. Free, Python is product of open source. People allows to use it in business or commercial without any charge.

1. Easy to read, Syntax in Python is clear and readable. Beginner can be easily to read and handle Python's coding very well.
2. Reusability, Python is easily reused modules and packages. Peoples can be developed their own library and reused it later project.
3. Object-Oriented Programming. Unlike scripting language, Python is designed to be object-oriented. OO programming means you can implement using idea of inheritance and polymorphism.

OpenCV

OpenCV is a synonym of Open Computer Vision Library, which has at least 500 algorithms, documentation and sample code for real time computer vision. OpenCV is originally developed by Intel and launched in 1999.

3.2 HARDWARE REQUIREMENTS

Hardware	**:**	Pentium Based System with a minimum of P4
RAM	**:**	1GB(minimum)

Pentium 4

Pentium 4 is the intel processor that was released in the November 2000.The P4 processor has a viable clock speed that now exceeds 2GHz – as compared to the 1GHz of the Pentium3.

RAM

Random Access Memory is a form of computer storage that stores data and machine code currently being used. A RAM device allows data item to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory

4. PROJECT REQUIREMENTS

4.1 Functional Requirements

Sequence of Operations

1. Image is imported by providing the location of the image.
2. Then, the picture is transformed from RGB to grayscale because it is easy to detect faces in grayscale.
3. Image Segmentation, which is used for detecting of multiple objects in a single image so that the classifier can easily detect the objects and faces in the picture.
4. The next step is to use haar-like features algorithm which is proposed by viola-jones for face detection.
5. This algorithm used for finding the location of human faces in frame or image.
6. The next step is to give the coordinates which makes a rectangle box in the image/frame to show the location of face.
7. After this, it can make a rectangle box in the area of interest where it detects the face.

4.2 Non Functional Requirements

Performance Requirements

The performance is independent on the Video Input. Performance depends solely on the accuracy of the classifier which does the prediction.

5. DESIGN

5.1 High Level Requirements

Webcam will be opened from that a face will be detected. If there is a face it displays “Faces found ” otherwise “No faces found”.

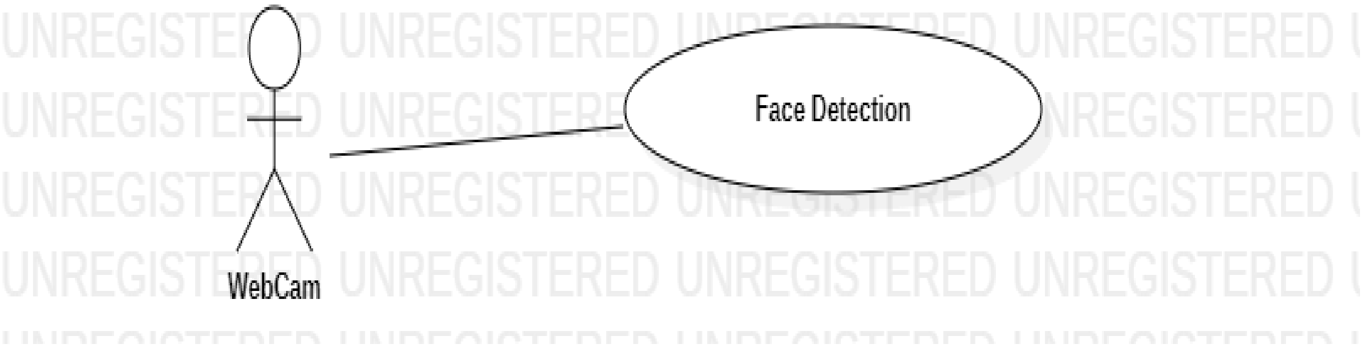


Figure 5.1: Usecase Diagram

5.2 Low Level Requirements

Webcam will be opened from that a face will be detected. When it detects the face, If there is a face it gives the no. of faces otherwise 0 faces. After this, it will display whether the live video has faces or not. Then after detecting the faces, the name which was given as input, is the image name which will be stored in the same location.

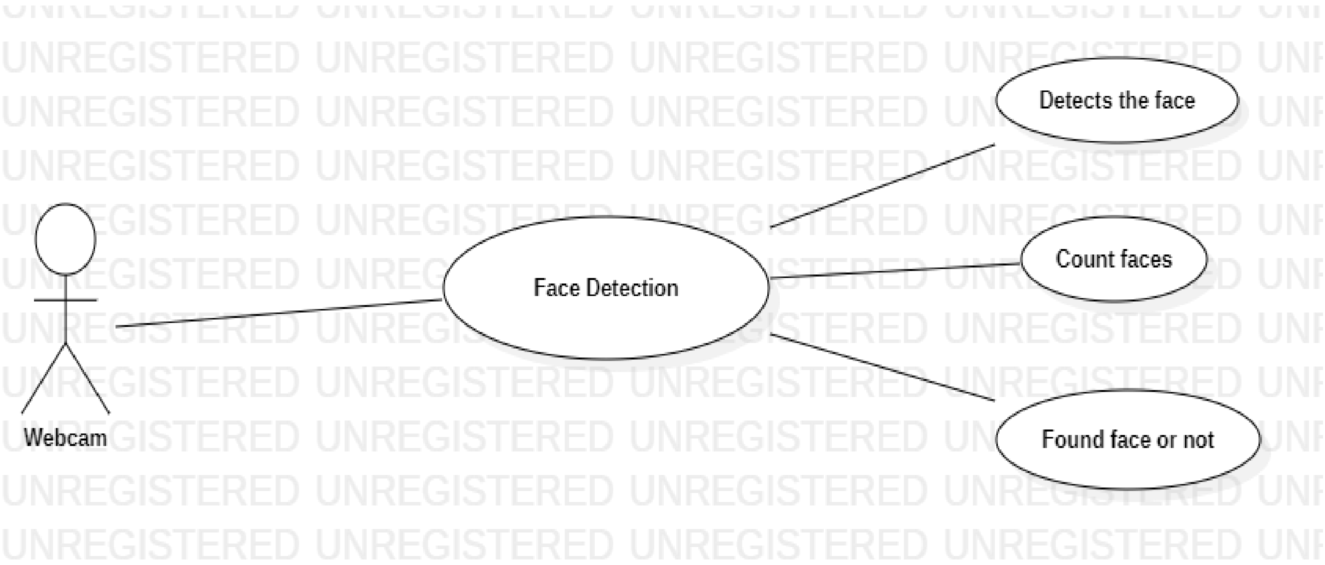


Figure 5.2: Usecase Diagram

6. IMPLEMENTATION

6.1 CODE SNIPPET

```
import cv2
classify = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
name = input()
photo= cv2.VideoCapture(0)
i=0
while(photo.isOpened()):
    ret, image = photo.read()
    #Convert a Image to GrayScale to detect faces easily in a grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    face = classify. detectMultiScale (gray,scaleFactor=1.1,minNeighbors=5,minSize=(30, 30))
    for (a, b, c, d) in face:
        cv2.rectangle(image, (a, b), (a+c, b+d), (0,255,0), 2)
    if ret == True:
        cv2.imshow("'Face Detection using Webcam'  - 'Press e to capture the image' ",image)
    for i in range(1):
        cv2.imwrite(name + '.jpg',image)
    if cv2.waitKey(20) & 0xff==ord('e'):
        break
if(len(face) >= 1):
    print("Face Count = {0}".format(len(face)))
    print("Faces Found")
else:
    print("No face found")
photo.release()
cv2.destroyAllWindows()
```

7. TESTING

High Level Test

Test Case	**Description**	**Output(Passed/Not Passed**
H_01	Face Detection	Passed

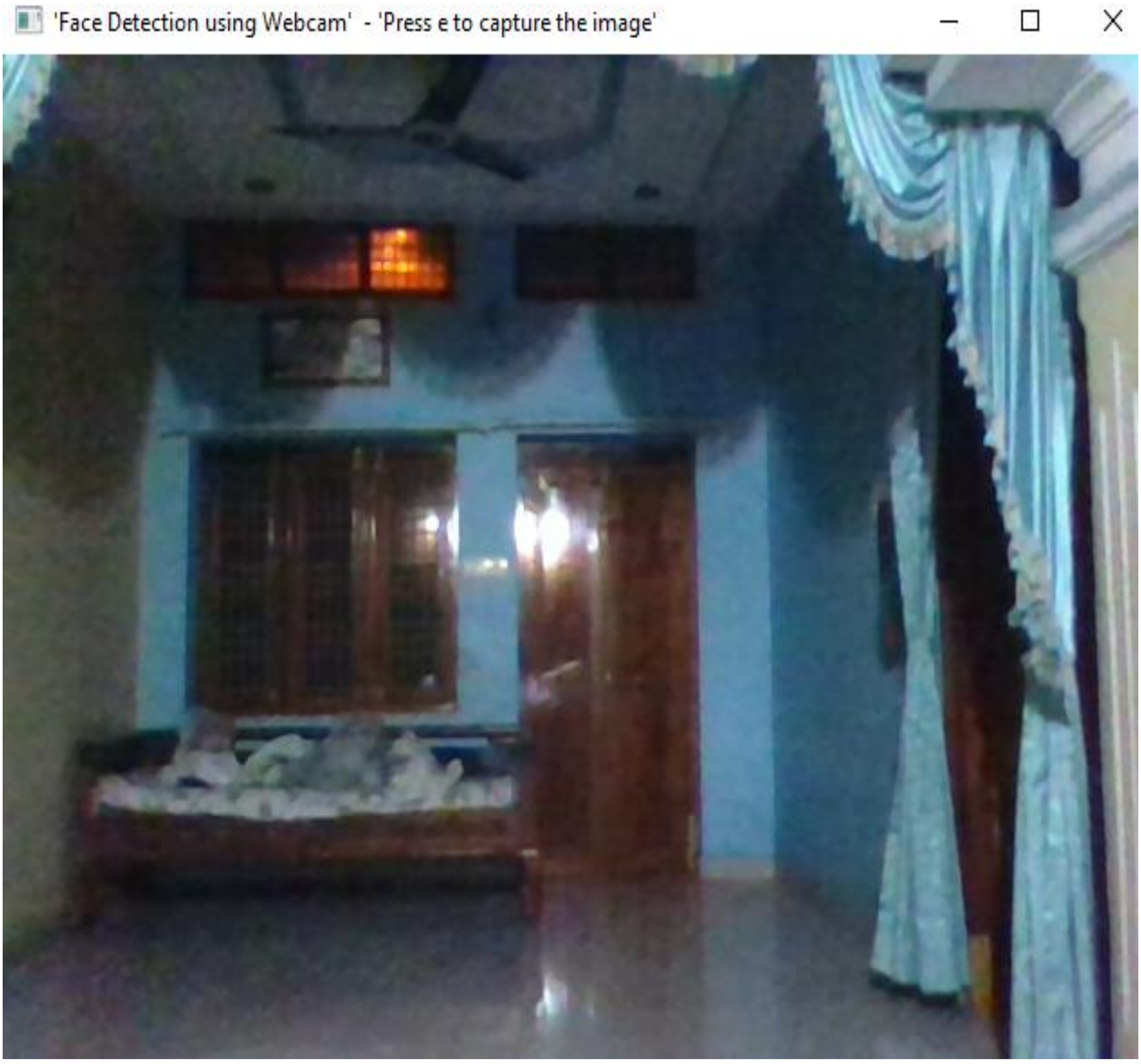
Low Level Test

Test Case	**Description**	**Output(Passed/Not Passed**
L_01	HaarCascade Classifier	Passed
L_02	Webcam	Passed
L_03	Image Name	Passed

Note: These test cases are done manually by running the code.

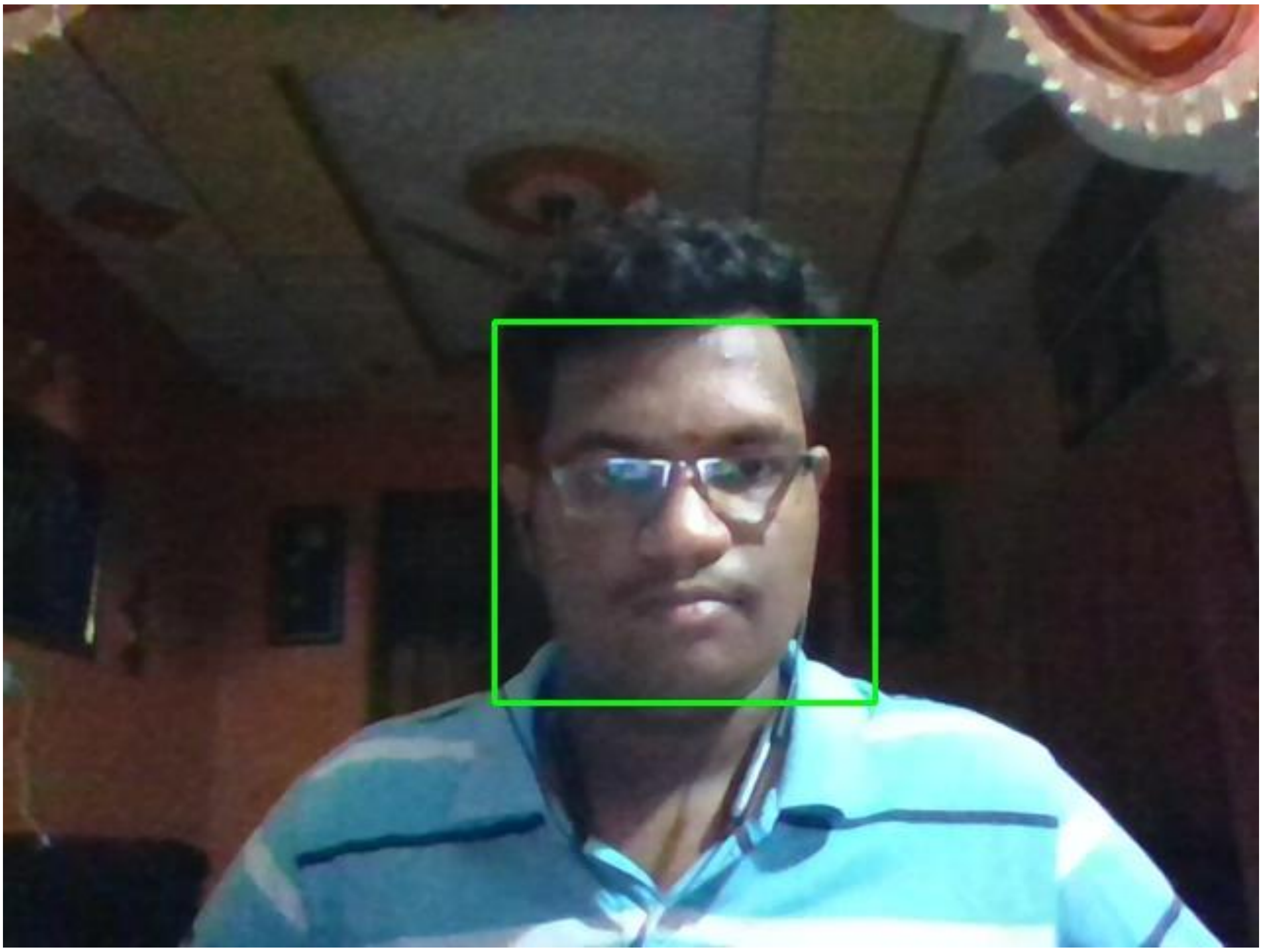
8. OUTPUT SCREENS

Webcam Window:

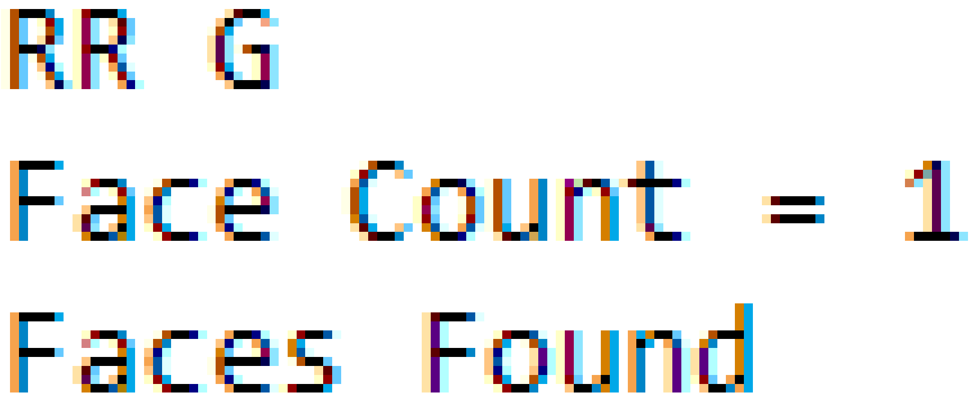


'Face Detection using Webcam' - 'Press e to capture the image'

Face Detection:

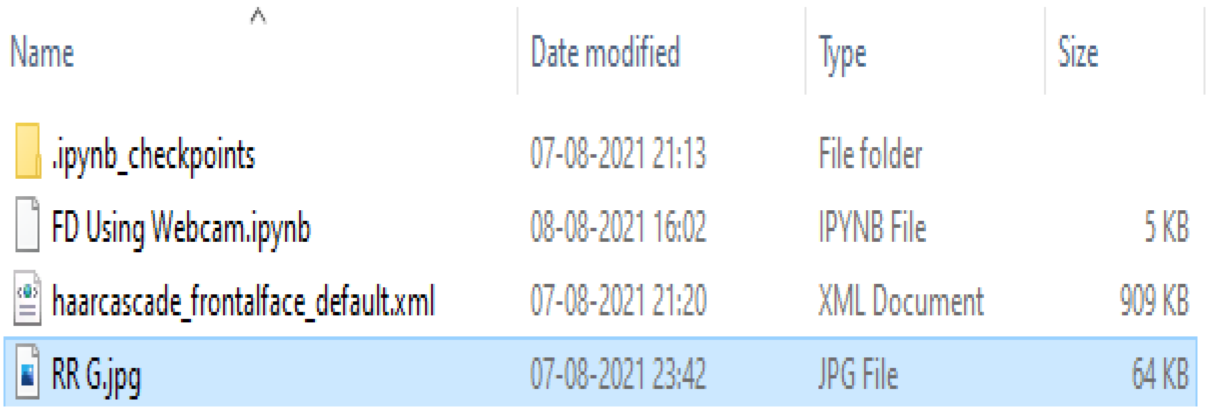


Count & Face found or Not:



```
RR G
Face Count = 1
Faces Found
```

Image Name:



Name	Date modified	Type	Size
.ipynb_checkpoints	07-08-2021 21:13	File folder	
FD Using Webcam.ipynb	08-08-2021 16:02	IPYNB File	5 KB
haarcascade_frontalface_default.xml	07-08-2021 21:20	XML Document	909 KB
RR G.jpg	07-08-2021 23:42	JPG File	64 KB

9. CONCLUSION

Face detection is an important step in face recognition, which is one of the more representative and classic application in computer vision. The aim is to develop and propose a system to detect human faces in digital images effectively, no matters what person’s ethnic, pose. Input images may be varied with face size, complex of background and illumination condition.

The haar-like algorithm is also used for feature selection or feature extraction for an object in an image, with the help of edge detection, line detection, centre detection for detecting eyes, nose, mouth, etc. in the picture. It is used to select the essential features in an image and extract these features for face detection. The next step is to give the coordinates of x, y, w, h which makes a rectangle box in the picture to show the location of the face or we can say that to show the region of interest in the image.

The implemented face detection (with an eye detection system) could be used for simple surveillance applications such as ATM user security, Public places or at any other important places, while the implemented manual faces detection.

10. BIBLIOGRAPHY

1. Face Detection For Beginners - https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9
2. http://www.peter-lo.com/Teaching/U08096/200912-B1