



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

# **Real Time Face Detection Using OpenCV**

**A Project Report**

*Submitted by*

**MD**

**SHAKEEB**

**ASHRAF**

**(1613101393)**

*in partial fulfilment for the award of the  
degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**Under the Supervision of**

**MS. DEEPIKA SHERAWAT**

**Asst.Professor**

**APRIL / MAY- 2020**



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

## **SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

### **BONAFIDE CERTIFICATE**

Certified that this project report is “**Real Time Face Detection Using OpenCv**” the bonafide work of “**MD SHAKEEB ASHRAF(1613101393)**” who carried out the project work under my supervision.

#### **SIGNATURE OF HEAD**

**Dr. MUNISH SHABARWAL,**  
**PhD (Management), PhD (CS)**  
**Professor & Dean,**  
**School of Computing Science & Engineering**

#### **SIGNATURE OF SUPERVISOR**

**Mr. Deepika Sherawat**  
**Asst. Professor**  
**School of Computing Science &**  
**Engineering**

## **ABSTRACT:**

Face detection is one of the fields in the computer vision industry and by considering the past events many applications have already been developed in the field of face detection and still many researches are going on.

The live detection of faces is of use especially nowadays and surveillance cameras can be easily found in the streets and in the shops, factories, hospitals, military, all are based on the live detection of faces. The use of surveillance camera with the live face detection we can expect to see the actual condition of the person and detect any face on the video.

By using live detection of faces we can detect that there is a human presence nearby whenever our camera caught the human so it is very useful for security purposes .This live face detection method can be applied to the surveillance cameras as well as the webcam of your computer, by applying the live face detection on the webcam we can detect who is in front of the computer and this applies to more than one face also. By applying other techniques along with the face detection we can track the person also. As for the implementation of this research is the detection of the faces using JavaScript.

Detection of faces is a field in the image processing field. Face Detection based on the identifying and finding the location of the image in the window or in some frame regardless of the size, position and condition of the image.

There are also some problems occur in the face detection:

1. Occlusion faces
2. Poses.
3. Lighting factor.

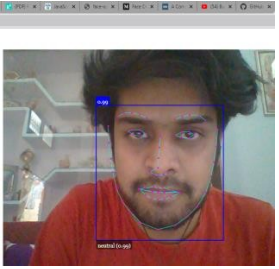
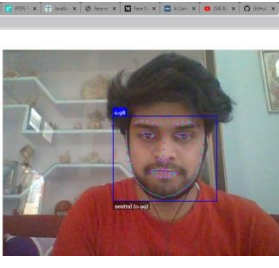
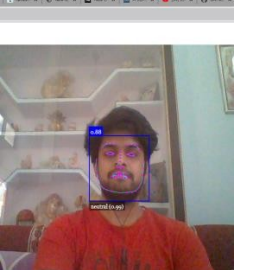
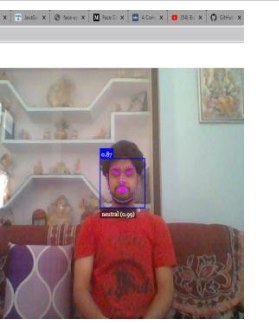
There are various researches in the field of image processing on the face. As there are various researches are already performed on the face detection, various algorithm have already been applied and various results have already been shown and shared with us.

Keywords: - Face Detection; Webcam; Surveillance camera; JavaScript.

<b>Table of Content</b>	
<b>Contents</b>	<b>Page Numbers</b>
<b>LIST OF TABLE</b>	<b>5 to 7</b>
<b>LIST OF FIGURES</b>	<b>7</b>
<b>LIST SYMBOLS, ABBREVIATIONS AND NOMENCLATURE</b>	<b>7</b>
<b>CHAPTER</b>	
INTRODUCTION	8 to 9
LITERATURE REVIEW	9 to 10
MAIN THEME	10 to 15
CONCLUSION AND FUTURE ENHANCEMENTS	15
<b>REFERENCES</b>	<b>16</b>


## LIST OF TABLES:

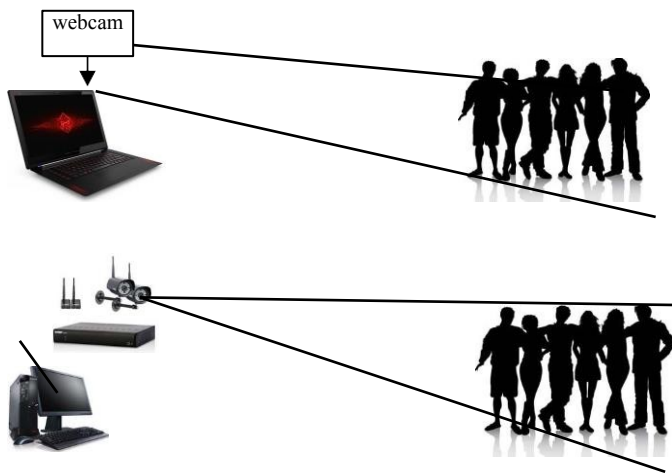
TABLE 1:


Conditions	Expected Results	Observations	Results
<p><b>Condition</b></p> <p>1. Camera at a distance of 50 m</p>	the camera can display objects and detect human face		Accepted
<p><b>Condition</b></p> <p>2. Camera at a distance of 100 m</p>	the camera can display objects and detect human face		Accepted
<p><b>Condition</b></p> <p>3. Camera at a distance of 150 m</p>	the camera can display objects and detect human face		Accepted
<p><b>Condition</b></p> <p>4. Camera at a distance of About 150m</p>	the camera can display objects and detect human face		Accepted

Conditions	Expected Results	Observations	Results
<b>Condition</b> 1. Normal Lighting conditions.	the camera can display objects and detect human face		Accepted
<b>Condition</b> 2. Dim lighting	the camera can display objects and detect human face		Accepted
<b>Condition</b> 3. Dark Lighting	the camera can display objects and detect human face		Accepted

TABLE 3:

Conditions	Expected Results	Observations	Actual Results
1. When we put an object over the face.	The camera can detect the human face.		In this pic it does not detect the face But in other case, it detects the face.



Conditions	Expected Results	Observations	Actual Results
2. In dark, multiple faces should be detected.	The camera can detect all the faces.		Failed, it did not detect multiple faces

**LIST OF FIGURES:**

**List of Symbols, Abbreviations and Nomenclature:**

Name of Abbreviation Used	Full Forms
MTCNN	Multi-task Cascaded Convolutional Neural Network
SSD	Single shot Multibox Detector
VS code	Virtual Studio Code
DCT	Discrete Cosine Transform
CNN	Convolutional Neural Network
RGB	Red Green Blue
API	application program interface





## **INTRODUCTION:**

The industry of processing the image using the computer is growing rapidly, where the image used as a digital technology. Image (Picture) is a combination of points, lines, shapes and colors to create a replica of a physical object .(animal , plants, human , fruits etc).

Detection of faces is a field in the image processing field.

Face Detection based on the identifying and finding the location of the image in the window or in some frame regardless of the size, position and condition of the image.

There are also some problem occur in the face detection such as occlusion faces, poses and lighting factor.

There are various researches in the field of image processing on the face. As there are various researches are already performed on the face detection, various algorithm have already been applied and various results have already been shown and shared with us .

As there are several ways that are proposed for the method of face detection which are assumed to be a great milestone in this field still many times it has been observed that when we use these approaches for same purpose on a fixed data it is of not much use because no method was able to give the good performance on the overall basis when evaluated.

In this research we are applying the live face detection, such that it can detect the faces based on the information we are providing it in the real time system, due to this property it can be used in surveillance cameras since the surveillance cameras use live face detection technique and my objective is also to apply this technique to the webcam of the computer. Apart from all the techniques that have been used previously, we are using JavaScript.

There are various libraries available also in JavaScript that are very useful in the process of image processing and can be used as a very good resource for the face detection (for eg. Pico.js and tracking.js etc) these are the API implemented in the browser. The algorithms are Authentication in such a way that in some cases they can recognize the special cases too. It also gives the facility of facial motion capture digital cameras use such libraries for auto focus.

Surveillance cameras were previously created for the purpose of security to investigate criminal crime, robbery and for the illegal activities but as the technology emerges it can not only monitor the images but it also includes the recording system can be saved in a hard drive.

## **Literature Review:**

### A. Previous Researches:

Face detection is a very famous field in the image processing world as it is very famous, various researches have already been done using various technologies and algorithms in this field. From the Python libraries to the applying machine learning algorithms for the face detection. In this paper I have proposed the face detection using JavaScript. I have used face-api.js that leverages the TensorFlow.js.

**A1.** Divya Meena and Ravi Sharan in 2016 used a face detection method in which they have used the Viola Jones Algorithm and used the result of the algorithm for the face recognition. For the face detection using Viola Jones algorithm it was concluded that in occluded image face cannot be detected and false positive value is also given. If the image is static then only it detects the face.

**A2.** M Hanuma Teja in 2011 proposed Real Time Face detection using Face Template Matching and DCT Energy analysis; he changed the RGB image into the YCbCr color space by applying the transformations and used DCT Energy Analysis. It was concluded that the system was quite able to distinguish between the live face and the photographic imitation. He used 2MP camera and it was stated that better results could be achieved by using multiple cameras.

**A3.** Wang Liting , DING Xiaoqing, FANG Chi in 2009, Proposed Face Live Detection Method based on Physiological Motion Analysis. They used Eye Contour Extraction algorithm to extract the data of the eye and then detect the face so the conclusion was, the algorithm was able to achieve real time performance for facial video sequences.

**A4.** Faizan Ahmad, Aaima Najam and Zeeshan Ahmad have proposed Image based Face Detection and Recognition. They have used Adaboost classifier with Haar and Local Binary Pattern features whereas Support Vector Machine classifier is

used with Histogram of Oriented Gradients for face detection. They have used Eigenfaces for Face Recognition . The conclusion was Haar Like Features reported relatively well but it has much false detection than LBP which could be considered as the future work for surveillance.

**A5.** Tao Ni , Shaoyuan Zou , Zhifei Kong , Lingtao Huang , Hongyan Zhang , Le Sun in 2019 have proposed Research on Living Face Detection Based on Dual Camera Contour Matching , they have used thermal-imaging camera which uses multi-line convex hull detection algorithms to speed up the detection.

The contour matching algorithm is also used to calculate the distance between two contours to determine whether it is a living face, Compared with previous traditional template matching , this method does not require a priori template training and user corporation.

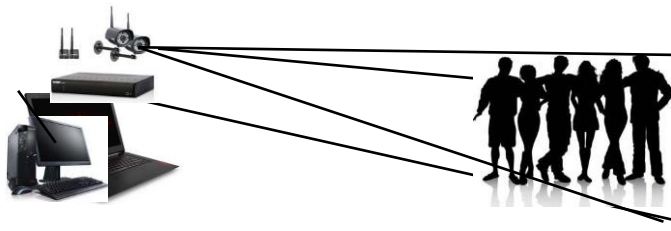
## **MAIN THEME:**

### **Theoretical Review:**

**1. Image:** The image is a representation, likeness and imitation of an object .It is combination of points, lines, shapes and color to create replica of physical object. Digital Image is as a function of two dimensional light intensity  $f(x, y)$  where  $x$  and  $y$  represents the spatial coordinates and the value of  $f$  at a point  $(x, y)$  is proportional to the brightness of image at that point.

**2. Convolutional Neural network:** ConvNet and CNN is a deep learning algorithm which can take an input image , assign importance to various objects in the image and be able to differentiate one from the other . The preprocessing required in a ConvNet is much lower as compared to other classifications algorithms. While in primitive methods filter are hand –engineered, with enough training, ConvNets have the ability to learn these filter.

**3.Face-api.js:** It implements convolutional neural network to solve for face detection and recognition of faces and face landmarks . The face-api.js leverages TensorFlow.js and is optimized for the desktop and mobile web.



## **Face Detection:**

### **JavaScript Face Detection with face-api.js:**

Face-api.js implements convolutional neural network to solve for face detection and recognition of faces and face landmarks. The face-api.js leverages TensorFlow and is optimized for the desktop and mobile web.

Recent releases to face-api.js have mentioned consistency with the latest versions of TensorFlow.js.

In the below figure the webcam was used.

As we can use this live face detection technique to the webcam as shown in the figure above , it can also be used in the surveillance camera's . Figure below shows the diagram for live face detection for the surveillance camera for the security purposes.

## **Implementation:**

I have used VS Code for the implementation of the code and a live plugin. After running the program output window first want us the access of using the webcam for the detection and after providing the access it detects the face.

I have created two files and implemented the face-api.js file:

## **1. Index.html (File):**

In the html file I have included the **face-api.min.js** and also the **script.js** so that I can use their functions and functionalities.

## **2. Script.js:**

This file contains all the functions that detects the faces in the window. It has event listener for the video and several functions have also applied to draw the dimensions around the faces detected, such as drawDetection, drawFaceLandmarks and drawFaceExpressions. With the help of all these functions we can detect the faces.

## **3. Face-api.min.js:**

It is a JavaScript API that implements CNN for the face detection. face-api.js implements multiple face detectors for different user cases.

The most accurate face detector is SSD (Single shot Multibox Detector) which is basically a CNN based MobileNet V1.

It also implements a Tiny Face detector basically a tinier version of Tiny Yolo V2. The last one is MTCNN (Multi-task Cascaded Convolutional Neural Network. The Network returns the bounding boxes of each faces, with their corresponding scores, e.g. the probability of each bounding boxes , as it might be that an image that does not contain any face at all. It should also be noted that it works for one person also in the conditions such as in dim light and also in normal day light.

## **4. Live Server Plugin:**

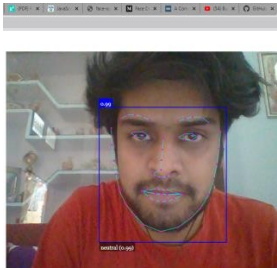
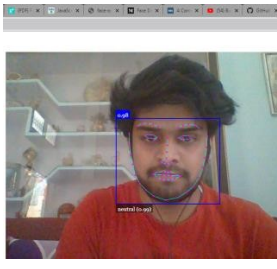
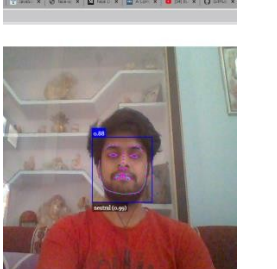
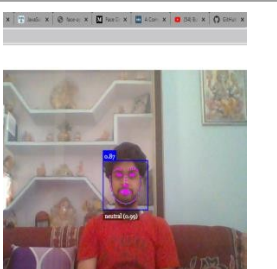
In order to get the live changing results. A VS code live server plugin is also used to run the program. It depicts all the live changes that is happening in the code and changes the output in the real time according to the code has changed.

After the face detection if we also want to implement the face recognition , we can do that because this face-api.js returns 68 point face landmarks ,after receiving the data from the face-api.js we can also implement the face recognition but in this paper we have done only the detection part but if we talk about its use in the further processing such as in face recognition , it can be proved useful as well.

## Results and Conditions.

Tables for different conditions:

1. Table 1 represents the camera is placed at different distances from the face.

Conditions	Expected Results	Observations	Results
<p><b>Condition</b></p> <p>1. Camera at a distance of 50 m</p>	the camera can display objects and detect human face		Accepted
<p><b>Condition</b></p> <p>2. Camera at a distance of 100 m</p>	the camera can display objects and detect human face		Accepted
<p><b>Condition</b></p> <p>3. Camera at a distance of 150 m</p>	the camera can display objects and detect human face		Accepted
<p><b>Condition</b></p> <p>4. Camera at a distance of About 150m</p>	the camera can display objects and detect human face		Accepted

As shown in the above table I have placed the camera at different distances and we got very good results , all the conditions have been accepted.

Table 2:  
It shows the different lighting conditions.

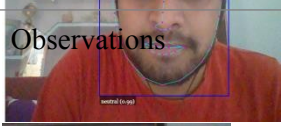


Conditions	Expected Results	Observations	Results
<b>Condition</b> 1. Normal Lighting conditions.	the camera can display objects and detect human face		Accepted
<b>Condition</b> 2. Dim lighting	the camera can display objects and detect human face		Accepted
<b>Condition</b> 3. Dark Lighting	the camera can display objects and detect human face		Accepted

Table 3:



Conditions	Expected Results	Observations	Actual Results
1. When we put an object over the face.	The camera can detect the human face.		In this pic it does not detect the face. But in other cases, it detects the face.



Table 4:

Conditions	Expected Results	Observations	Actual Results
2. In dark, multiple faces should be detected.	The camera can detect all the faces.		Failed, it did not detect multiple faces

## **Conclusion and Future Enhancements:**

The result of this research shows that use of face-api.js in JavaScript can be used for the live detection of faces. Based on the results it is obtained that this method can only detect the faces within 300 m approx and I have also obtained some false results when it comes to dark lighting conditions because in dark lighting it is unable to detect multiple faces sometimes, so it is unreliable for dark lighting conditions and lastly when we put an object over our face it does not detect the face.

So, I hope for the future researches in this area would probably generate some good results and be able to detect multiple faces when it comes to dark lighting conditions, be able to detect the faces even greater distances than 300 m and able to detect the face even when it covered.



## **REFERENCES:**

1. Divya Meena and Ravi Sharan (2016), An Approach to face detection and Recognition, IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE – 2016)
2. M Hanuma Teja, Real-Time Live Face Detection using face template matching and DCT Energy Analysis (2011 – International Conference of Soft Computing and Pattern Recognition {SoCPaR})
3. Wang Liting, DING Xiaoqing, Fang Chi (2009 ) Face Live Detection Method Based on Physiological Motion Analysis.
4. Tao Ni, Shaoyuan Zou, Zhifei Kong, Lingtao Huang, Hongyan Zhang, Le Sun, Research on Living Face Detection Based on Dual Camera Contour Matching, (2019), National Natural Science Foundation (NNSF)
5. Faizan Ahmad, Aaima Najam and Zeeshan Ahmad, Image-based Face Detection and Recognition, Department of Computer Science, COMSATS Institute of Information Technology.
6. Dhanar Intan Surya Saputra, Kamal Miftahul Amin, Face Detection and Tracking using Live Video Acquisition in Camera Closed Circuit Television and Webcam, (2016) 1<sup>st</sup> international Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE).
7. Ali Akbar Singh, Zulfiqar Ali Zaidi, Bhawani Shankar Chowdhry, Jawaid Daudpoto, (2016), Real Time face detection using raspberry pi and MATLAB, IEEE 10<sup>th</sup> conference on Application of Information and Communication Technologies (AICT).
8. Arundhati Das, Mameeta Pukhrambam, Ashim Saha , Real Time robust Face detection and tracking using extended Haar Features and improved boosting algorithm. (2015 – International Conference of Green Computing and Internet of Things) (ICGCIoT).