**FACE RECOGNITION**

**By PRAKHAR SINGH PARMAR**

School of Computer Science,KIIT

1905261@kiit.ac.in

Supervisor mail id

***Abstract-*** This is a Face Recognition project; so as the project name clearily indicates, that with the use of this python program we can detect facial objects in the video frame captured via webcam of the device. With OpenCV v2 library and Cascade which is a Object Detection algorithm utilizing Machine Learning. We can make a program that detects specified objects in a provided frame.

***Index Terms-*** face detection, OpenCV, haar-cascade, frame

INTRODUCTION

This project is on Face Detection, i.e., it can detect facial structures by identifying various features of a face and combining all that data to accurately detect faces in the frame. This technology has widespread use scenarios in various tasks ranging from very simple to much complex implementation.

* From a very simple scenario face detection can be used in cctv cameras so that in case of any mishap anywhere, people responsible can be easily identified and confronted.
* In this Covid times, this technology can scan if people are wearing masks in public places and fine those who fail to do so.
* Advance face recognition can scan faces in 3D and this data can be used as method of biometric password just like fingerprint.
* In robots and self-operating machines, this can would make them immensely natural and fluent with interacting with humans and their environment.

The machine learning algorithm used in this project is called **haarcascade\_frontalface\_default** which contains heaps of facial structure data and uses it to identify facial features in each single frame. The program uses webcam to capture live video and the program computes through each frame to detect facial structure in real time and upon detection of face, a blue square casing surrounds it.

BASIC CONCEPTS AND TECHNOLOGY USED

The project utilizes a very resourceful library Open CV version2. This library is mainly aimed to provide real-time computer vision. The library has features like GPU acceleration which used GPU along with CPU in order to facilitate processing-intensive tasks such as deep learning, analytics, and engineering applications.

For this face detection, a haarcascade module is used which is provides framework for object detection algorithm. Inside haarcascade, there is a xml file names **haarcascade-frontalface-default.xml** which contains tons of facial data which we can adopt to provide learning data for our own algorithm.

**cv2.VideoCapture(0)** is a object of opencv library used to access webcam and capture data which is further processed inside the programs.

Each frame is capture and converted to grayscale image which is used by **face\_cascade.detectMultiscale()** to find facial features.

**cv2.rectangle()** is used to make a rectangular boundary on the face.

**Cv2.waitkey()** for quitting the program on the press of escape key.

IMPLEMENTATION AND CONCLUSION

The programs works pretty well and the algorithm recognizes the face accurately and is able to keep face in detection even with quick movements of the person. It is a very small and simple code and it also does not consume lots of power so can be implemented at large scale.

It even bypasses hurdles like low camera quality, and can even capture face in very low light situation. It can also squeeze out some amount facial data even if any obstacle is in the way of covering face. Hence it can immensely improve our security standards.

REFERENCES

1. docs.opencv.org
2. [www.tutorialspoint.com](http://www.tutorialspoint.com)
3. [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
4. [www.edureka.com](http://www.edureka.com)
5. [www.pythonprogramming.com](http://www.pythonprogramming.com)
6. <https://youtu.be/-ZrDjwXZGxI>
7. <https://youtu.be/7IFhsbfby9s>