Face & Eyes Detection with

Open-cv & flask

**Introduction:**

**Object detection is one of the most promising applications of computer vision, a field of Artificial Intelligence (AI) which lets computers to derive meaning data patterns from the digital images and other visual inputs like webcam and CCTVs. These data patterns can be used to take actions or recommendations based on that information.**

OpenCV (Open Source Computer Vision Library) is an open source computer vision & machine learning software library. OpenCV is an Apache2 licensed product which lets its users to utilize and modify the code. OpenCV is built to provide common infrastructure for Computer Vision & to accelerate the use of machine perception in the commercial product.

OpenCV has more than 2500 optimized algorithms which can be used to recognize the human faces, identify objects, classify human actions, track moving objects, extract 3D models of objects etc.

OpenCV has a community of more than 47K active users and more than 18M downloads. It has C++, Python, Java and MATLAB interfaces & supports Windows, Linux and Android & Mac OS. OpenCV mostly deals with the real-time computer vision applications and whenever MMX & SSE instructions are available, OpenCV takes advantage of it.

**Introduction to Mini Project:**

This project aims at detecting faces and eyes in live stream with Haar Cascade Algorithm on a flask application.

**About Haar Cascade:**

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paula Viola & Michael Jones in their paper, ‘Rapid Object Detection using a Boosted Cascade of Simple Features’ in 2001.It is a machine learning based approach where a cascade function is trained from a lot of positive & negative images. It is also used to detect objects in other images.

**Face Detection with Haar Cascade:**

The algorithm needs lots of positive images which contain faces and negative images which do not contain faces to train the classifier. Then we need to extract values from it.



**OpenCV implements AdaBoost and it can be considered as one of the most efficient applications of AdaBoost, hidden under Haar Cascade classifier.**

**Flow diagram of the project:**

**Flask Framework**

Haar Cascade Classifier

Web Camera input

Face & Eye Detection

**Requirements:**

Programming Language: Python

Packages: Flask, OpenCV, venv. OS: Windows/ Ubuntu.

Code Editor: VSCode Version management tool: git