



# FACE & EYES DETECTION WITH OPEN-CV & FLASK

Author : Aniket Tanaji Bhosale

## Abstract

Computer vision is used to detect objects, faces and features of faces. This project focusses on face and eye detection with the help of OpenCV and Haar Cascade algorithms on a basic flask based web application.

aniketbhosale2808@gmail.com  
[Email address]

# 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:**

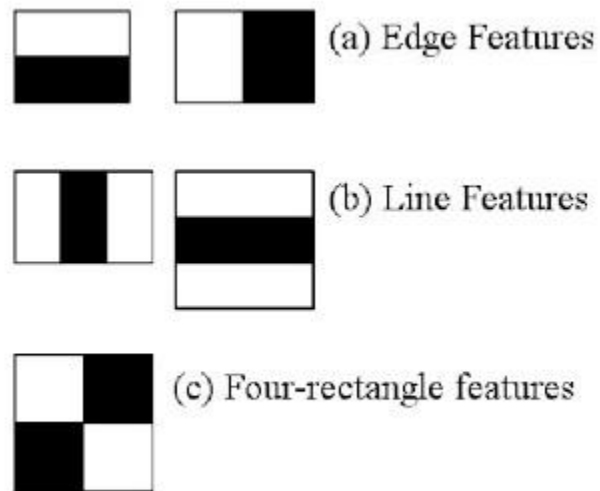
This project aims at detecting faces and eyes in live stream with Haar Cascade Algorithm on a flask application.

## **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.

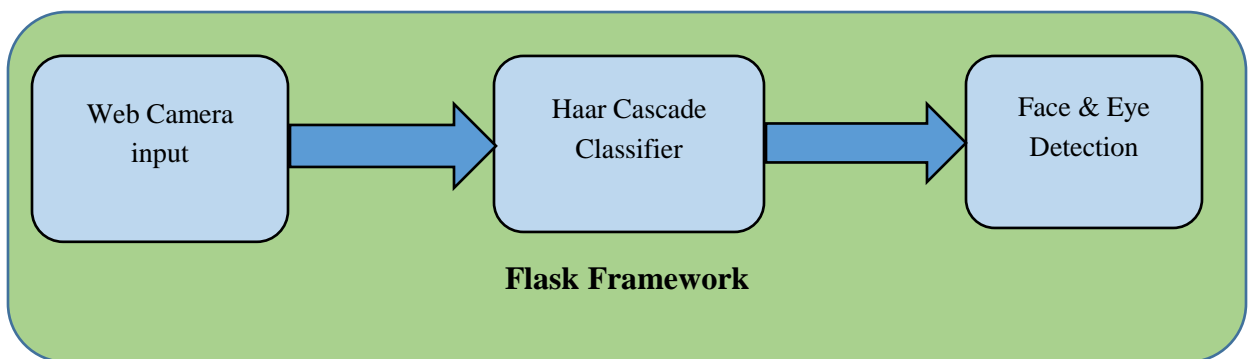
## Face Detection with Haar Cascade:

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

## Flow diagram of the project:



## Requirements:

Programming Language: Python

Packages: Flask, OpenCV, venv.

Code Editor: VSCode

OS: Windows/ Ubuntu.

Version management tool: git

## How to Run the Project:

This project is a computer vision application built on a light weight python web framework flask.

Follow the following steps to execute this project:

1. Open the directory where your source code resides.
2. Open command prompt and activate the virtual environment for the project if it already exists. If it doesn't exist, create a virtual environment with help of venv package and install all required for the project by using requirements.txt and pip.
3. Execute command `python -m flask run` and it will render the application on localhost on default port for flask application i.e. port number 5000.
4. Open your favorite web browser and visit an uniform resource locator, [localhost:5000](http://localhost:5000) or [127.0.0.1:5000](http://127.0.0.1:5000)
5. It will load the default or home page for the web application, which will need access to your machine's web cam if it exists in existing hardware configuration of your machine.
6. The home page will render the view that your web cam can see. If it contain any face or eye which can be detected by Haar Cascade Algorithms, it will detect them and draw squares around those shapes in real time.
7. You can exit the project from the command prompt by pressing *Control + C*.

## About source code:

The source code of this project is made publically available on github.

URL of the repository: [https://www.github.com/aniketbhosale2808/fbs\\_mini](https://www.github.com/aniketbhosale2808/fbs_mini)

All the details about the project can be available on [localhost:5000/about](http://localhost:5000/about)