

# HAAR CASCADE DOCUMENTATION

In this project, I applied face detection and eye detection to some photos I took using OpenCV with Python. OpenCV is an open source software library that allows developers to access routines in API (Application Programming Interface) used for computer vision applications. The version I used was developed for Python called OpenCV-Python.

## CONCEPT

Before we continue, we must differentiate between face recognition and face detection. They are not the same, but one depends on the other. In this case face recognition needs face detection for making an identification to “recognize” a face.

### What is “Face Detection”?

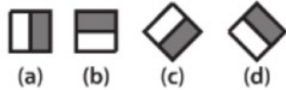
Face detection is a type of application classified under “computer vision” technology. It is the process in which algorithms are developed and trained to properly locate faces or objects (in object detection, a related system), in images. These can be in real time from a video camera or from photographs. An example where this technology is used are in airport security systems. In order to recognize a face, the camera software must first detect it and identify the features before making an identification.

Face detection uses classifiers, which are algorithms that detects what is either a face(1) or not a face(0) in an image. Classifiers have been trained to detect faces using thousands to millions of images in order to get more accuracy. OpenCV uses two types of classifiers, LBP (Local Binary Pattern) and Haar Cascades. I will be using the latter classifier.

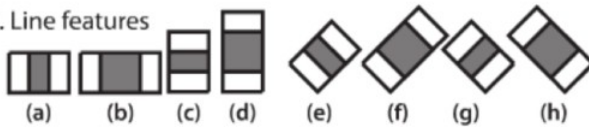
A Haar Cascade is based on “Haar Wavelets” which Wikipedia defines as: A sequence of rescaled “square-shaped” functions which together form a wavelet family or basis.

It is based on the Haar Wavelet technique to analyze pixels in the image into squares by function. This uses machine learning techniques to get a high degree of accuracy from what is called “training data”. This uses “integral image” concepts to compute the “features” detected. Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers.

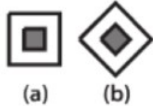
### 1. Edge features



### 2. Line features



### 3. Center-surround features



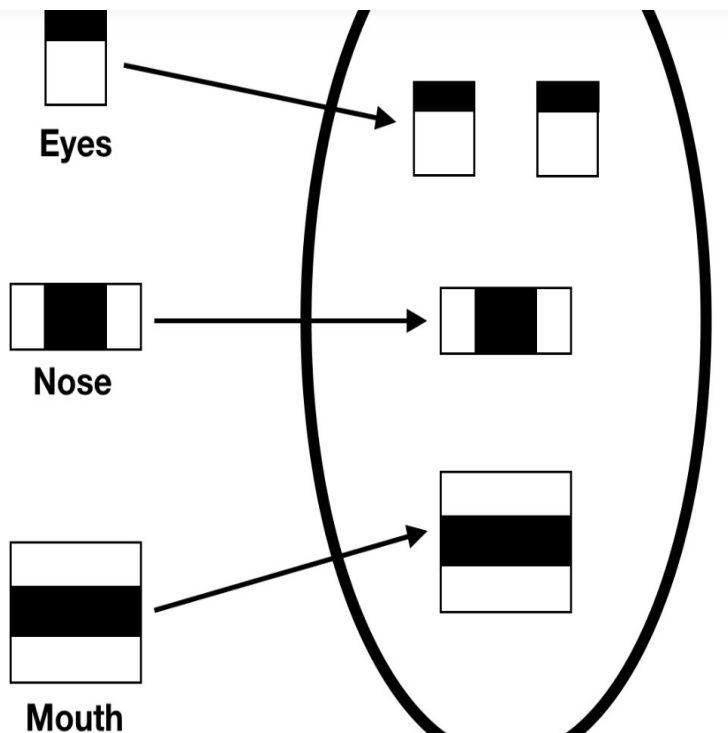
**Face Detection** determines the locations and sizes of human faces in arbitrary (digital) images.

In **Face Recognition**, the use of Face Detection comes first to determine and isolate a face before it can be recognized.

Haar Cascades use machine learning techniques in which a function is trained from a lot of positive and negative images. This process in the algorithm is feature extraction.

*Haar Cascade Eye detection* follows the same, conceptually.

Here below is a theoretical face model.



The training data used in this project are XML files

## BASIC REQUIREMENTS

make sure you have PIP (Python Package Index) installed with Python and run the following command

**pip install opencv-python**

**XML Files for haar cascade (frontal face and eyes)**

## A FEW TERMS FOR BETTER UNDERSTANDING

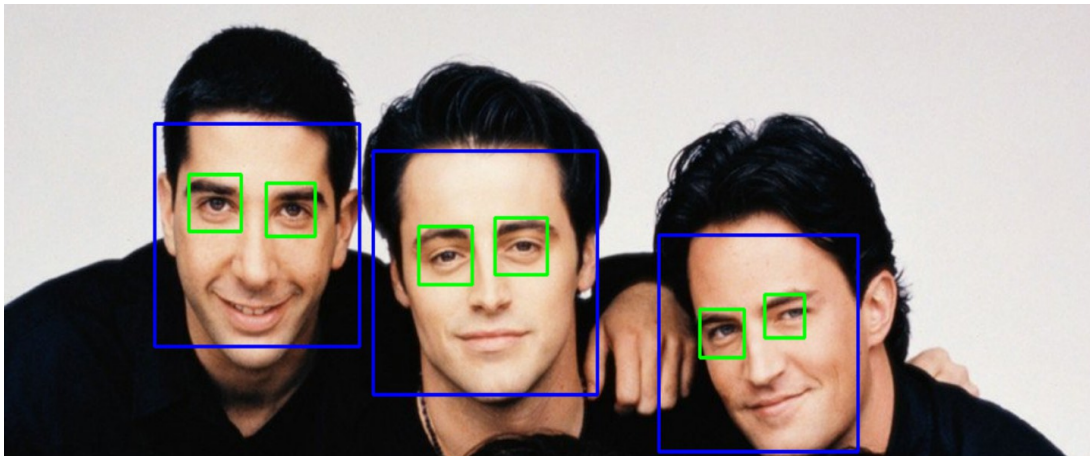
We are going to use the detectMultiscale module from OpenCV. What this does is create a rectangle with coordinates (x,y,w,h) around the face detected in the image. This contains code parameters that are the most important to consider.

**scaleFactor:** The value indicates how much the image size is reduced at each image scale. A lower value uses a smaller step for downscaling. This allows the algorithm to detect the face. It has a value of x.y, where x and y are arbitrary values you can set.

**minNeighbors:** This parameter specifies how many “neighbors” each candidate rectangle should have. A higher value results in less detections but it detects higher quality in an image. You can use a value of X that specifies a finite number.

**minSize:** The minimum object size. By default it is (30,30). The smaller the face in the image, it is best to adjust the minSize value lower.

## RESULT



## REFERENCES

<https://www.superdatascience.com/blogs/opencv-face-detection>

<https://becominghuman.ai/face-detection-using-opencv-with-haar-cascade-classifiers-941dbb25177>