FACE RECOGNITION SYSTEM

Team Members

DESHIREDDY KRISHNAREDDY (20MIS1069)

APPIREDDY GOWTHAM REDDY (20MIS1175)

Face Recognition System

The face recognition is a technique to identify or verify the face from the digital images or video frame. A human can quickly identify the faces without much effort. It is an effortless task for us, but it is a difficult task for a computer. There are various complexities, such as low resolution, occlusion, illumination variations, etc. These factors highly affect the accuracy of the computer to recognize the face more effectively. First, it is necessary to understand the difference between face detection and face recognition.

Face Recognition: The face recognition algorithm is used in finding features that are uniquely described in the image. The facial image is already extracted, cropped, resized, and usually converted in the grayscale.

Algorithm Used

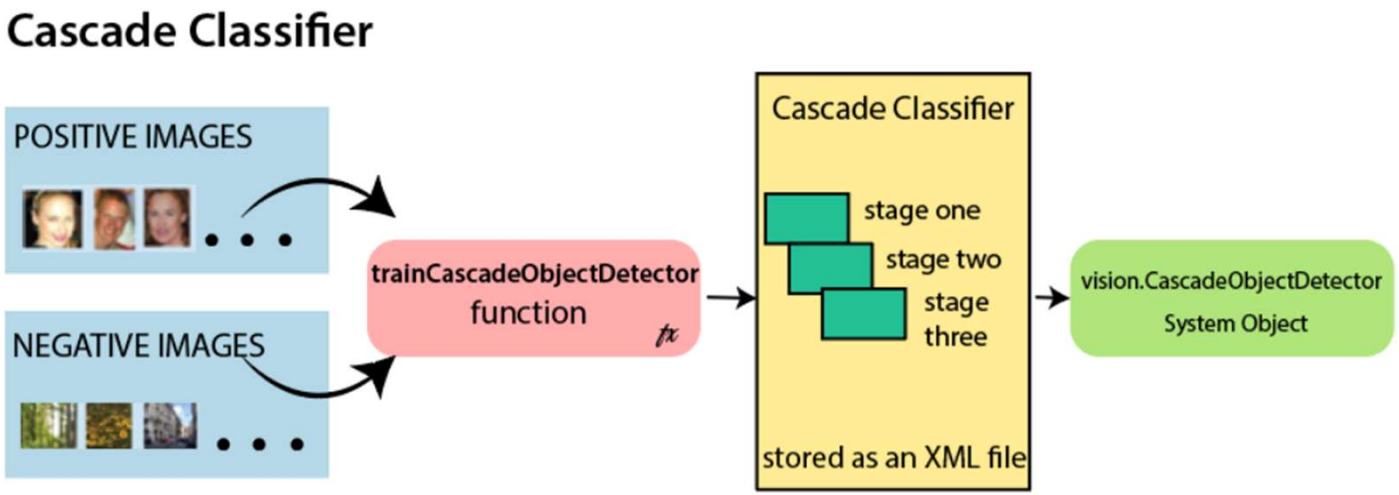
HAAR Cascade Algorithm

- The HAAR cascade is a machine learning approach where a cascade function is trained from a lot of positive and negative images. Positive images are those images that consist of faces, and negative images are without faces. In face detection, image features are treated as numerical information extracted from the pictures that can distinguish one image from another.
- We apply every feature of the algorithm on all the training images. Every image is given equal weight at the starting. It founds the best threshold which will categorize the faces to positive and negative. There may be errors and misclassifications. We select the features with a minimum error rate, which means these are the features that best classifies the face and non-face images.
- All possible sizes and locations of each kernel are used to calculate the plenty of features

HAAR-Cascade Detection in OpenCV

- OpenCV provides two applications to train cascade classifier **opencv_haartraining** and **opencv_traincascade**. These two applications store the classifier in the different file format.
- For training, we need a set of samples. There are two types of samples:
- **Negative sample:** It is related to non-object images.
- **Positive samples:** It is a related image with detect objects.
- A set of negative samples must be prepared manually, whereas the collection of positive samples are created using the **opencv_createsamples** utility.

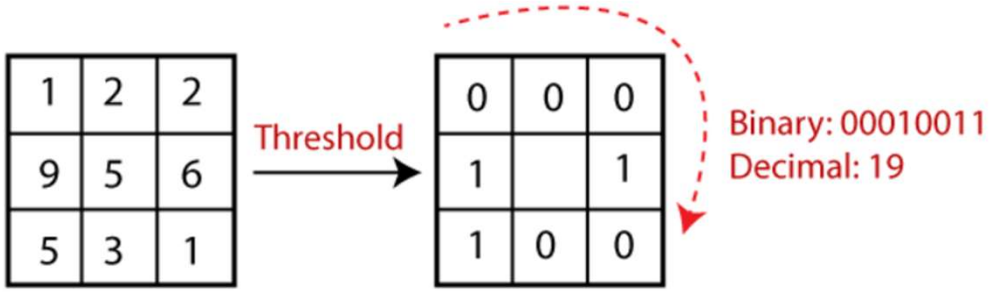
Cascade classifier



LBPH

- Local Binary Pattern Histogram algorithm is a simple approach that labels the pixels of the image thresholding the neighborhood of each pixel. In other words, LBPH summarizes the local structure in an image by comparing each pixel with its neighbors and the result is converted into a binary number. It was first defined in 1994 (LBP) and since that time it has been found to be a powerful algorithm for texture classification.
- This algorithm is generally focused on extracting local features from images. The basic idea is not to look at the whole image as a high-dimension vector; it only focuses on the local features of an object.

LBPH



Extracting the Histograms from the image

- **Extracting the Histograms from the image:** The image is generated in the last step, we can use the **Grid X** and **Grid Y** parameters to divide the image into multiple grids, let's consider the following image:
- There are various approaches to compare the histograms (calculate the distance between two histograms), for example: **Euclidean distance, chi-square, absolute value,** etc. We can use the Euclidean distance based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n} (\text{hist}\,1_i - \text{hist}2_i)^2}$$

- **CascadeClassifier()**: This class is used to load the trained cascaded set of faces which we will be using to detect faces for any input image.
- **Imcodecs.imread()/Imcodecs.imwrite() :** These methods are used to read and write images as Mat objects which are rendered by OpenCV.
- **Imgproc.rectangle() :** Used to generate rectangle box outlining faces detected, it takes four arguments – input_image, top_left_point, bottom_right_point, color_of_border.

Thank You