**CAP -6412**

**Project -1**

**Removing the reflections from sunglass and original shade reconstruction**

Shreyas. Somashekar, 3671876, shreyas0906@knights.ucf.edu

**Problem Statement:** This project aims to remove reflection from sunglasses in an image and try to reconstruct the original shade of the sunglass.

**Procedure:**

**1.** The images for the project were downloaded from google by searching for the keyword “ sunglasses”, “sunglasses on the beach”, “selfies with sunglasses”. From all this 3100 images were downloaded and called as “positive images”. Similarly 1800 negative images were downloaded which did not contain an object of detection, in our case, sunglasses. To train the haarcascade classifier, we need to have a positive images to negative images in the ratio 2.1

**2.** The downloaded images were cleaned for our requirement, meaning, the images were converted to grayscale. The positive images were downsized to 50X50 and the negative images were downsized to 100X100.

**3.** The downsized images are put together into “pos” and “neg” folders which contain the respective images. A background text file is created which contains all the file names of the negative images as “bg.txt”. Similarly, for the positive images a “info.dat” file is created which contains the file name along with its contents. That is, “1 0 0 50 50” .Which means that the positive image contains 1 object of interest which has to be detected and the dimension of the image is 50X50.

**4.** With all these information we begin to train the haarcascade classifer. We make two directories info and data which contains the information about the positive images and the data directory stores the .xml file after the program finishes to run. To classify the image the following commands are used:

1.$opencv\_createsamples -img filename.ext -bg bg.txt -info info/info.lst -pngoutput info -maxxangle -0.5 -maxyangle 0.5 -maxzangle 0.5 -num 1800

This command means that with a positive image “filename.ext” and background information bg.txt of the negative images, create samples with the information present in info.lst. The created samples are of .png output and are stored in info directory. The positive image is the rotated about x, y, and z axis about -0.5 0.5 -0.5 degrees and superimposed to create samples. 1800 defines the number of negative images to be used.

2. $opencv\_createsamples -info info/info.lst -num 1800 -w 100 -h 80 -vec postives.vec

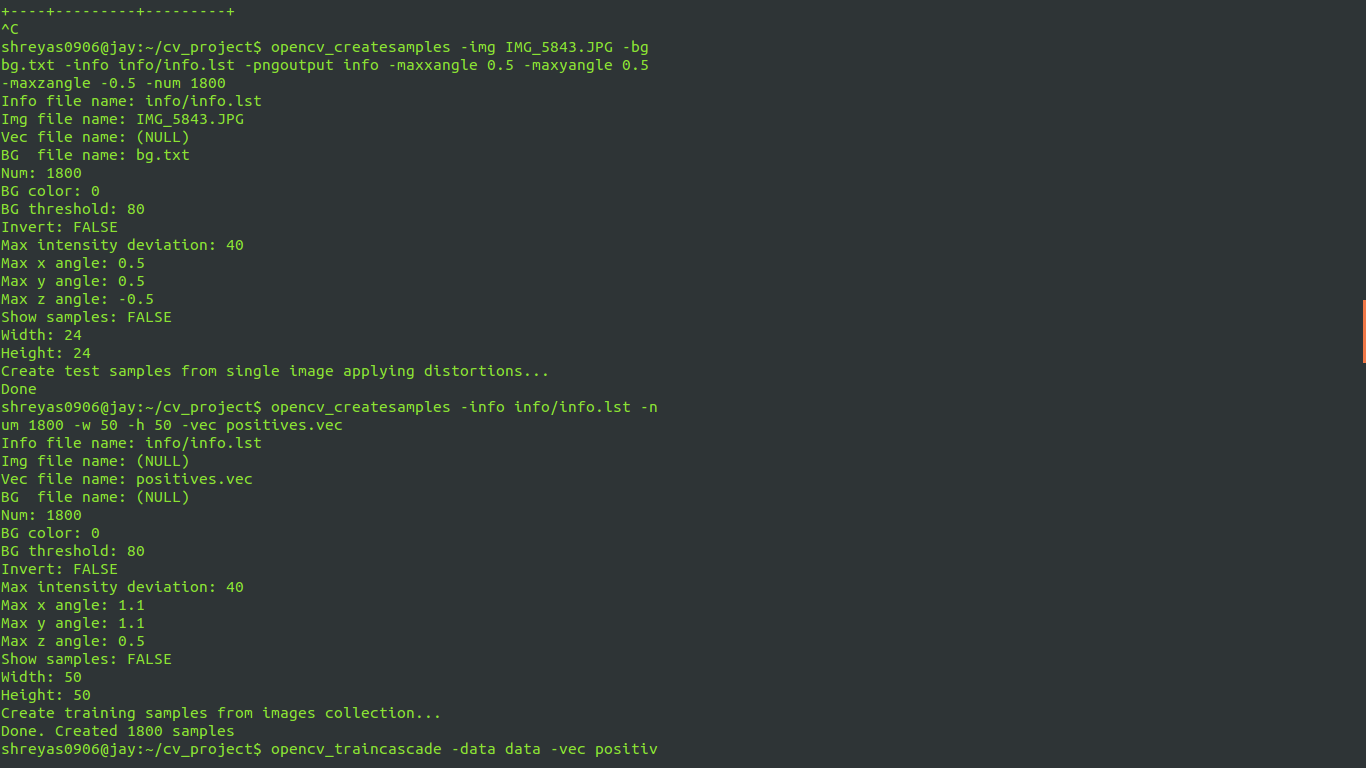
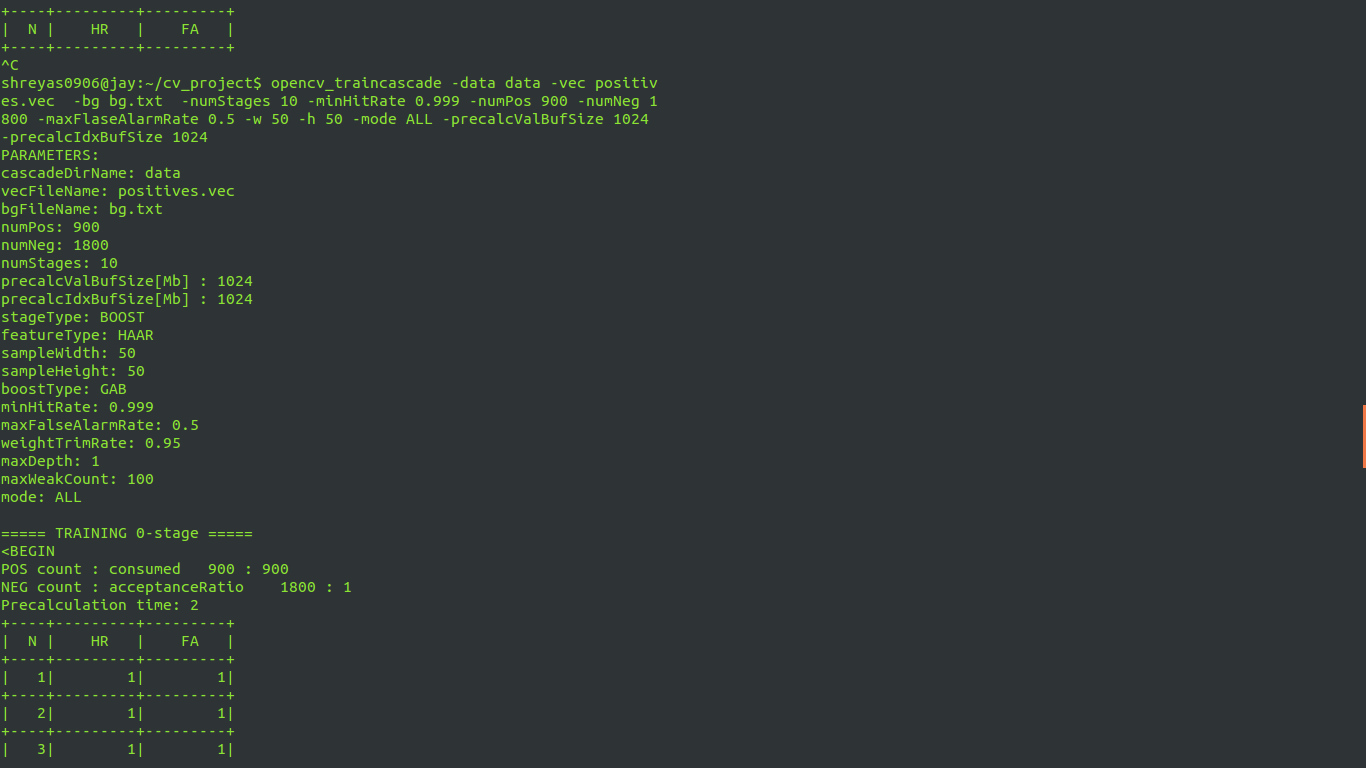
This command generates the vector file for the positive image. The haar cascade window being defined by width of 100 pixels and height of 80 pixels.

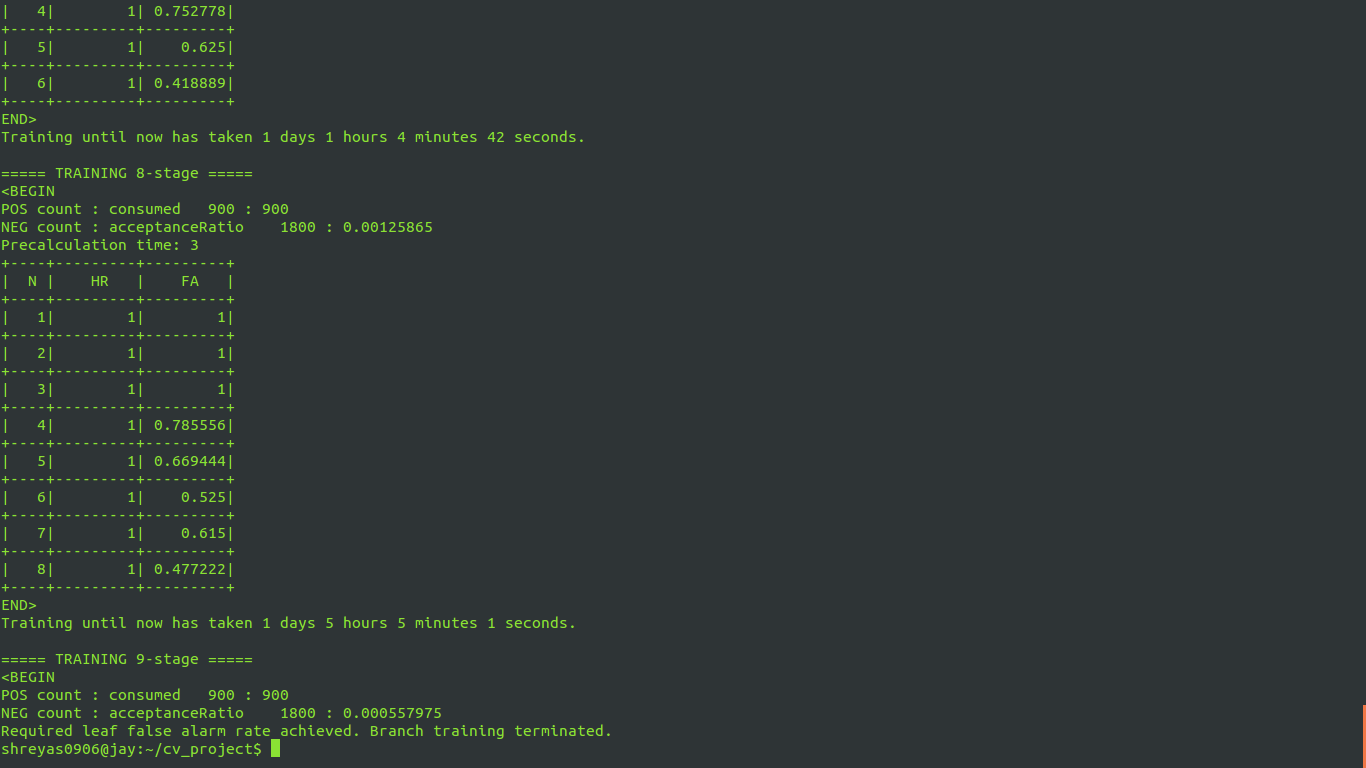
3.$opencv\_traincascade -data data -vec positves.vec -bg bg.txt -numStages 10 -minHitRate 0.999 -numPos 900 -numNeg 1800 -maxFalseAlarmRate 0.5 -w 100 -h 80 -mode ALL -precalcValBufSize 1024 -precalcIdxBufSize 1024.

This command trains the haarcascade with a window witdth of 100 and height of 80.

The Following images will illustrate the commands when run on a server.

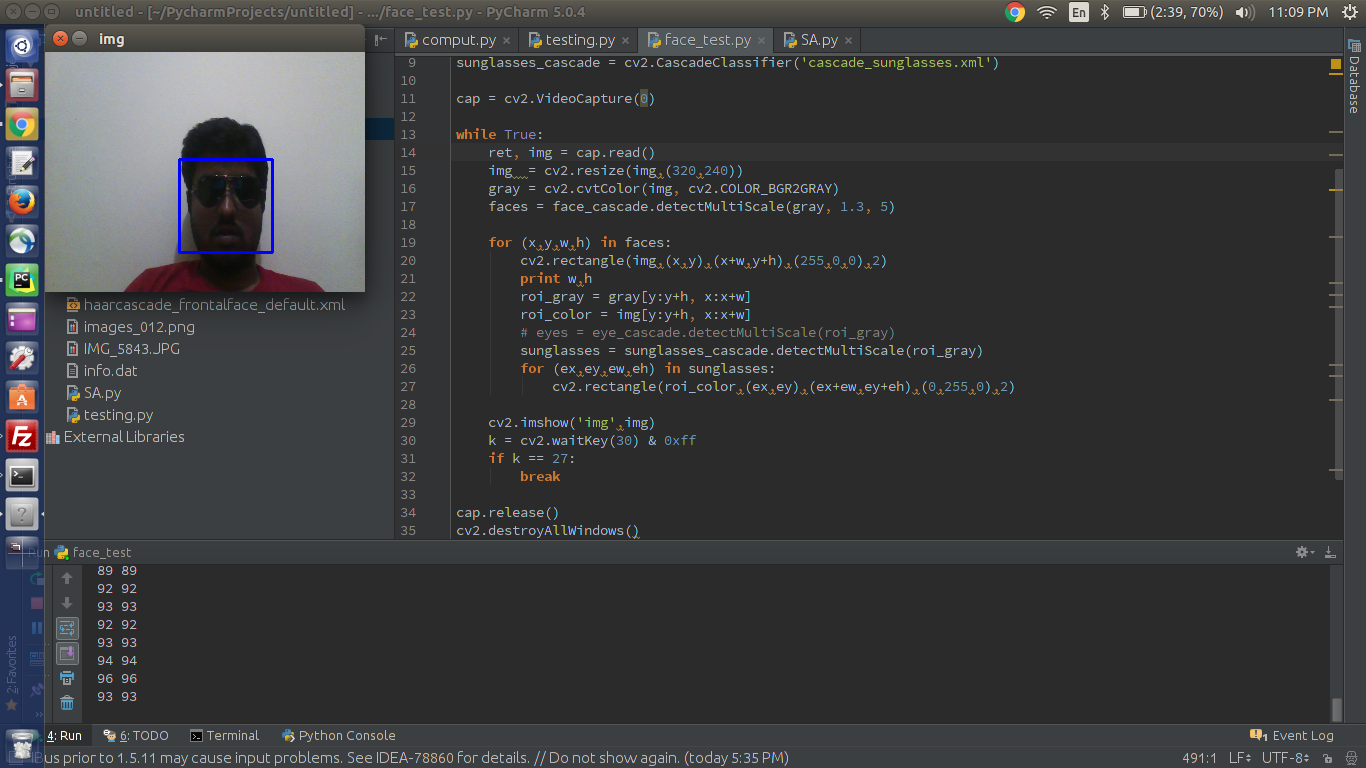
Fig1.

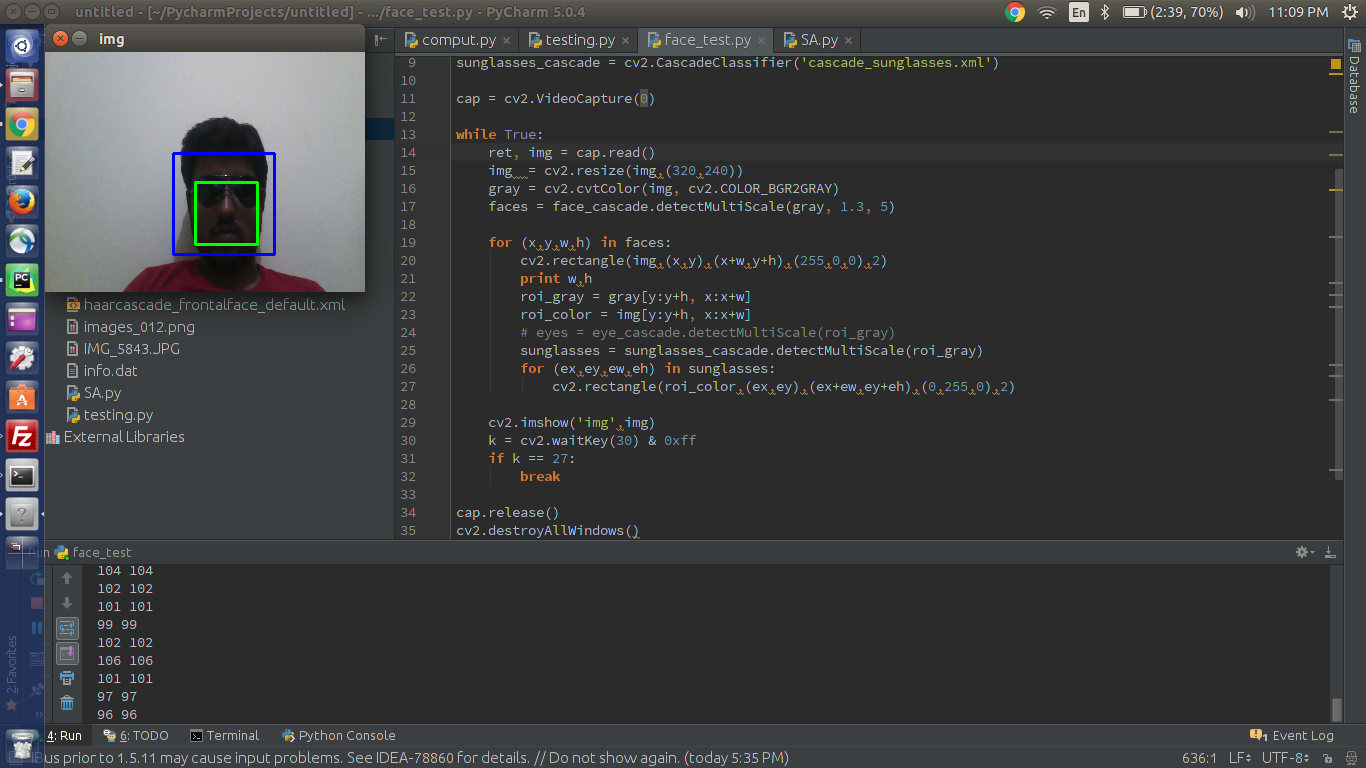
Fig2

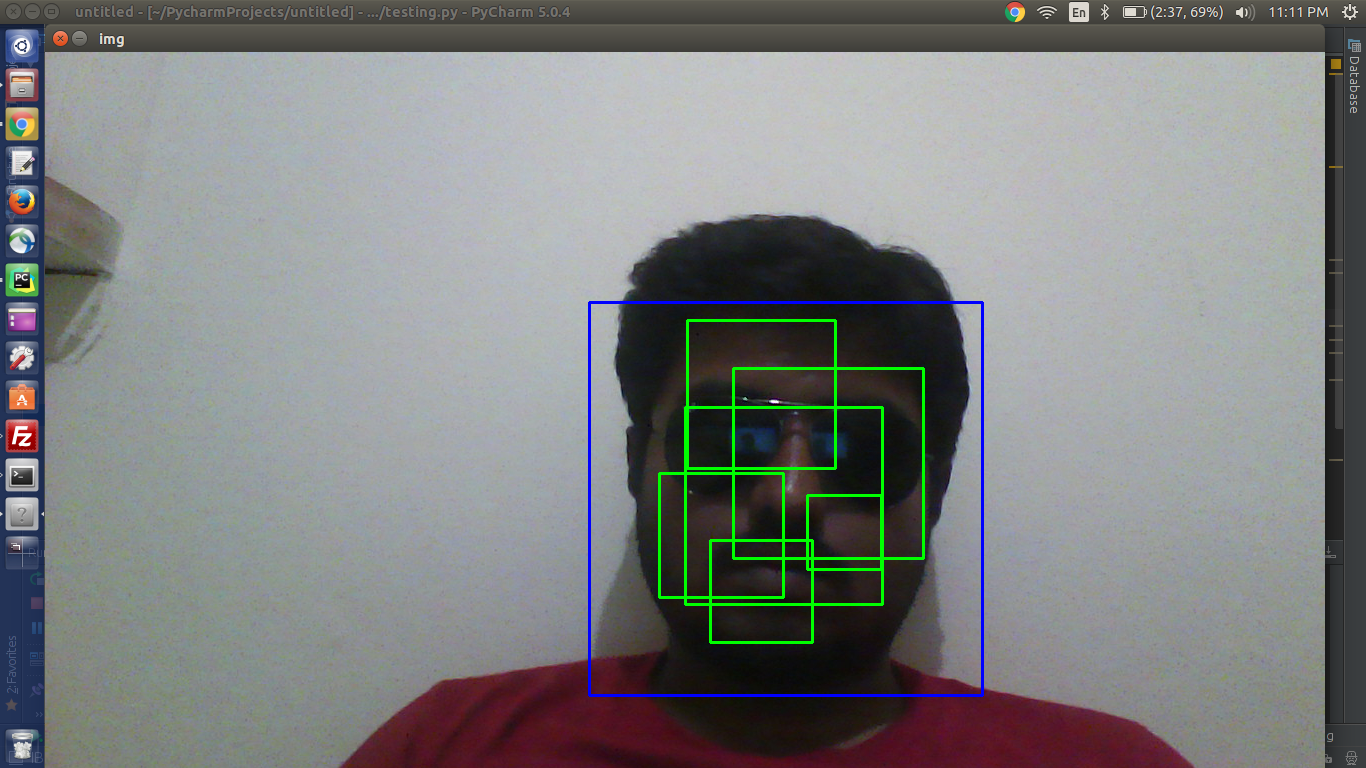
**fig(3)**

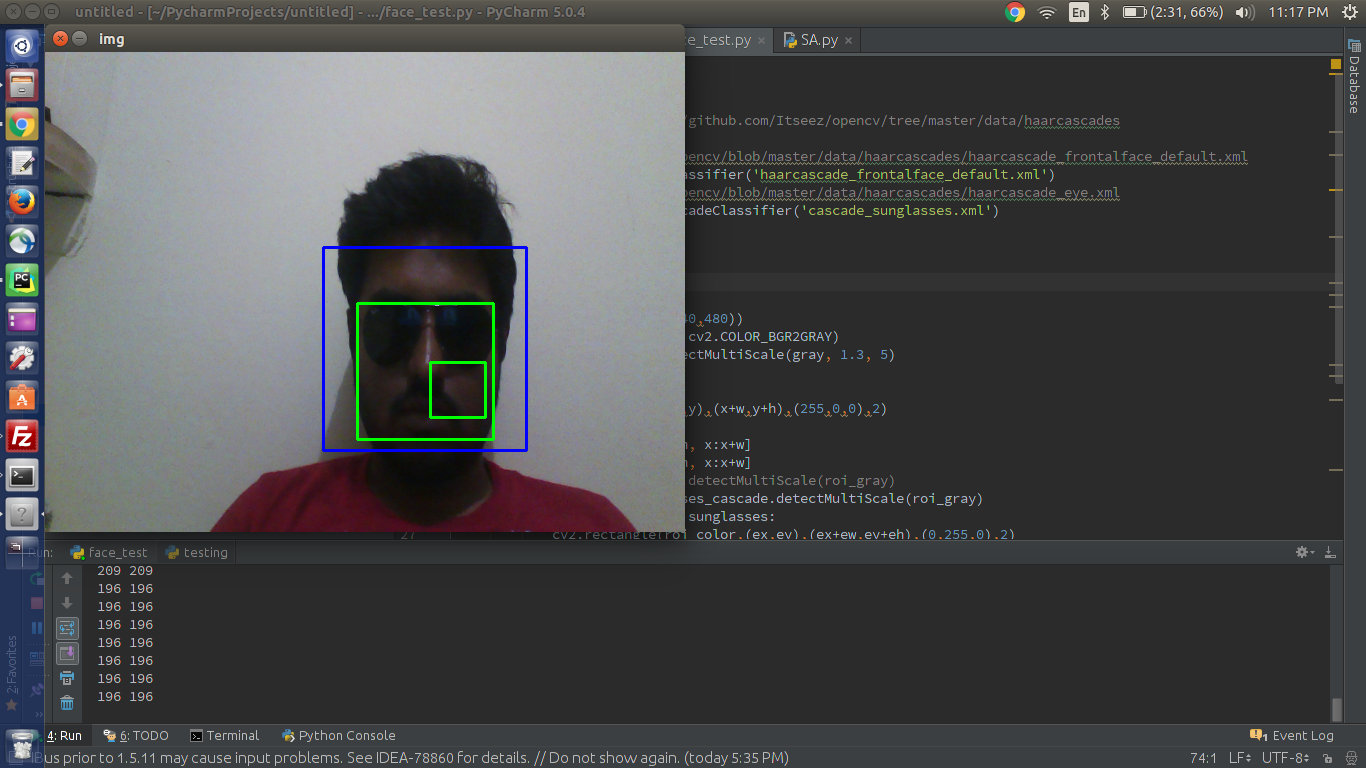
After running the training for hours, we get the cascade.xml file in the data directory. It will also contain all the .xml files for the stages.

Once we obtain the cascade.xml file we try to look for the sunglasses in the image. The following results were obtained and will be discussed later.

**Fig(4): basic face detection **

**fig(5): Sunglasses detected **

**fig(6): sunglasses detected and other facial features**

**Fig(7): sunglasses detected**

From the above figures we can observe that the window of the detected sunglasses needs more refinement and meets the project milestones. The intention behind choosing haarcascades over the methos discussed earlier are that, haar cascades are more acurate and can beused in realtime as well. Though the trainig took a lot of time, the results obtained are good.

Further, the things to be achieved:

varying the window size and making the window size dynamic.

Once the window is being detected, cropping the image out of it.

Applying machine learning techniques to identify and removing reflection

Reconstructing the image.

Run the face\_test.py python file with python2.7 and the cascade\_sunglasses in the same directory.