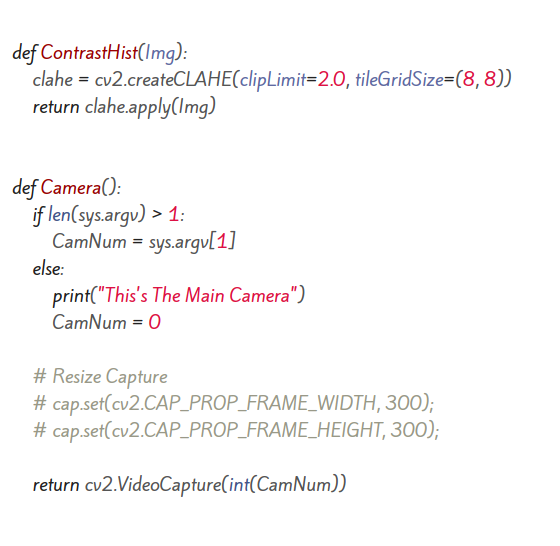


Start Shell script file: combine all python files to run the full project

Command python detect.py 1  
detect.py is our project and the second parameter is the number of camera



Detect Class

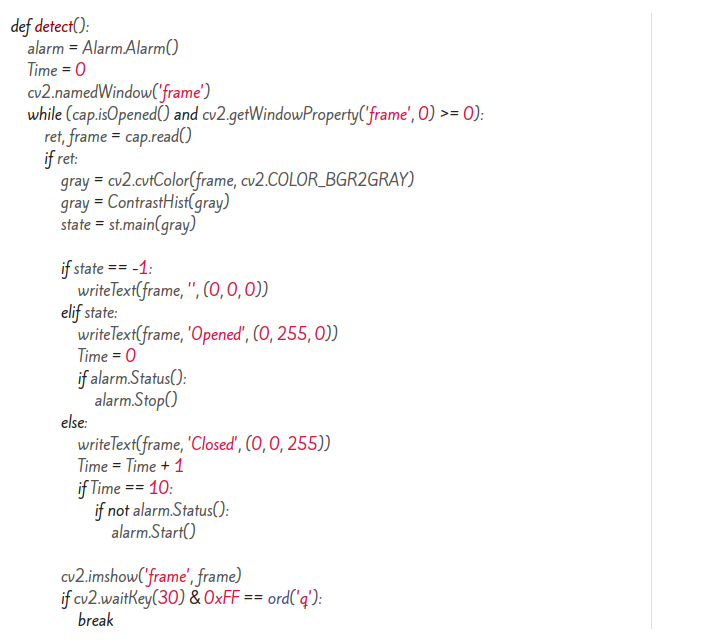
ContrastHist function: takes an image from user then Apply contrast limited adaptive histogram equalization

Camera function : open camera based on user input from the start shell script if their are multiple cameras then the user have the choice to change between cameras if their is no input program will automatically select the main camera if exist

Then return video capture which contain the driver face video



writeText function: which take user image , text which will written on image and color of text



Detect function: contain the main function for detect.py file

Abstract:

- take object of Alarm class to alarm drowsy driver.

- time variable to walk driver after period of time.

- name the window which we will use

- loop depend on capture is opened and the window not closed

- read the current frame of video

- if their is a frame:

\* convert the frame to gray color which will save processing

\* apply contrast limited adaptive histogram equalization

\* call the state class to check if the driver is drowsy

+ state = -1, then their is no person

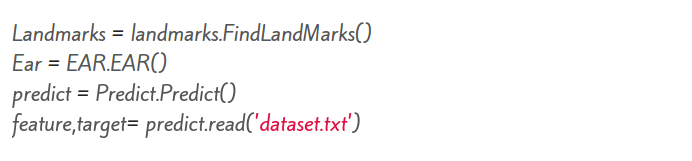
+ state = 1 , then their is an opening eye person in this case the program will set the timer to 0 and write text “opened” and close the alarm if it opened

+ state = 0 , then theirs is a closed eye person in this case the program will write a text “closed” and start the timer if it reaches 10 which equal about 3 seconds then call alarm

\* preview the current image

\* if the user press q key the application will close

- destroy all opened windows if the user closed the application to free cpu and memory



Status Class

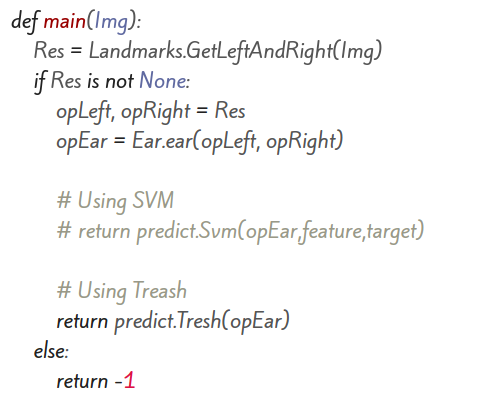
Call classes which we will use to detect if the driver is drowsy

- Landmarks : get facial landmark of driver face

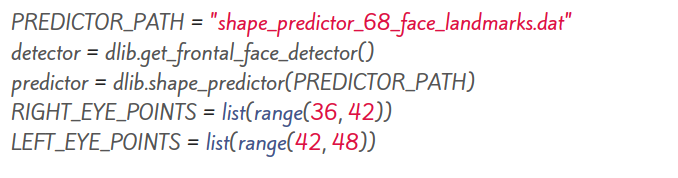
- Ear : compute the ear equation to left and right eyes

- predict : predict if the driver is drowsy

Call the dataset which we manually train on driver face ‘this section only used if we call predict using support vector machine’



Main function : this w=function f=will get the result of facial landmarks then pass it to Eye aspect ratio equation then take the result and passes to predict class to return the current status



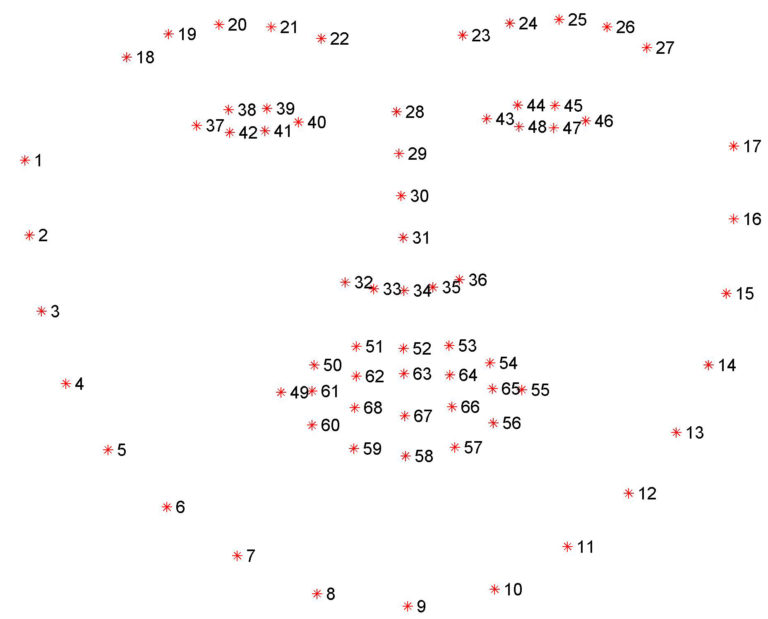
Landmarks Class

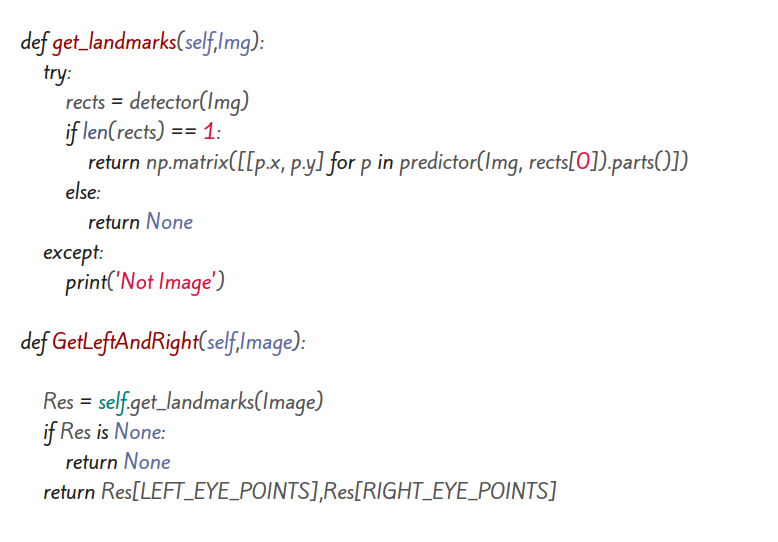
PREDICTOR\_PATH: contains the path of the face landmarks dataset which dlib need to predict the facial landmarks

detector: is a macro contain dlib get frontal face detector method

predictor: is a macro contain dlib predictor method

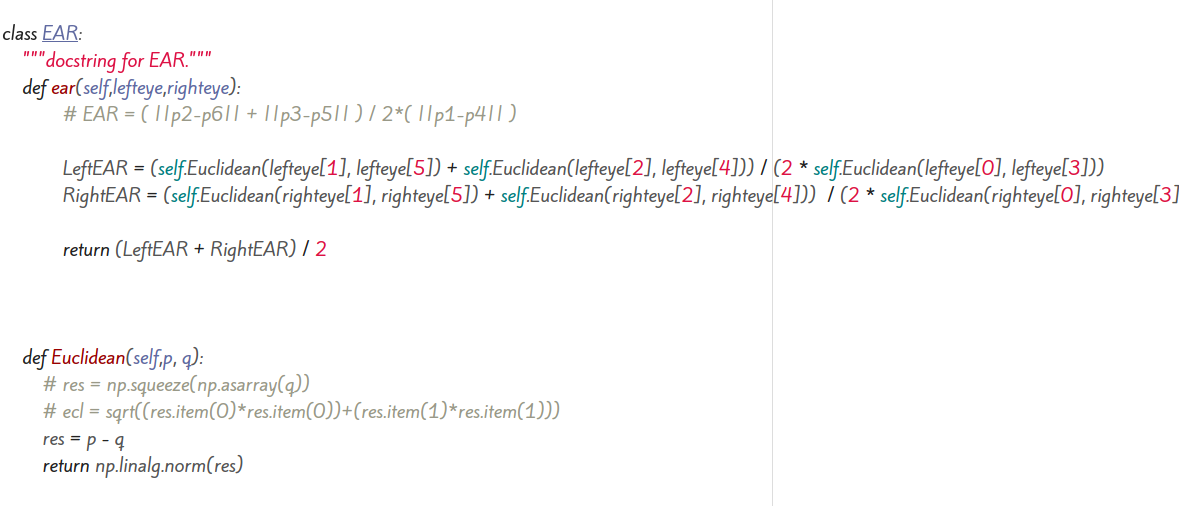
RIGHT and LEFT\_EYE\_POINTS get list of eye pixel location 36 to 42 is the right eye and 42 to 48 is the left eye





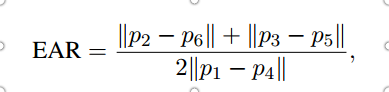
get\_landmarks method: tries to get the rects ‘faces’ which the the dlib detector returns take parameter image of the driver then if the rects numbers is 1 ‘one face’ returns a matrix of all landmarks pixels

GetLeftAndRight method: returns the location of left and right eyes from the landmarks matrix

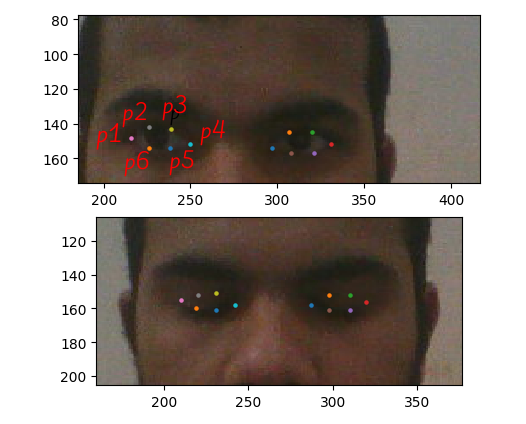


Ear Class

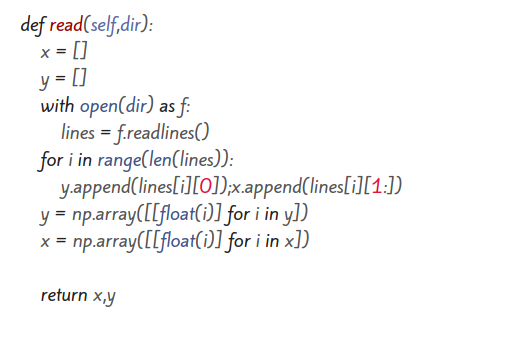
ear method: takes two parameters left and right eye list of pixels then applies the equation :



In which P represent the landmark

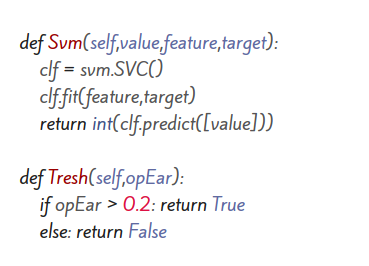


Euclidean method: returns the euclidean distance of the landmark pixel



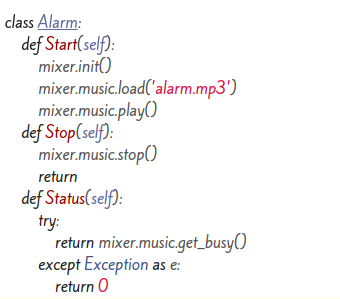
Predict Class

read method: take the directory of dataset trained file and returns a two numpy arrays of the dataset file, first parameter is status second one is ear value



Svm method: applies support vector machine to dataset

Tresh method: applies treasholding which the paper recommended

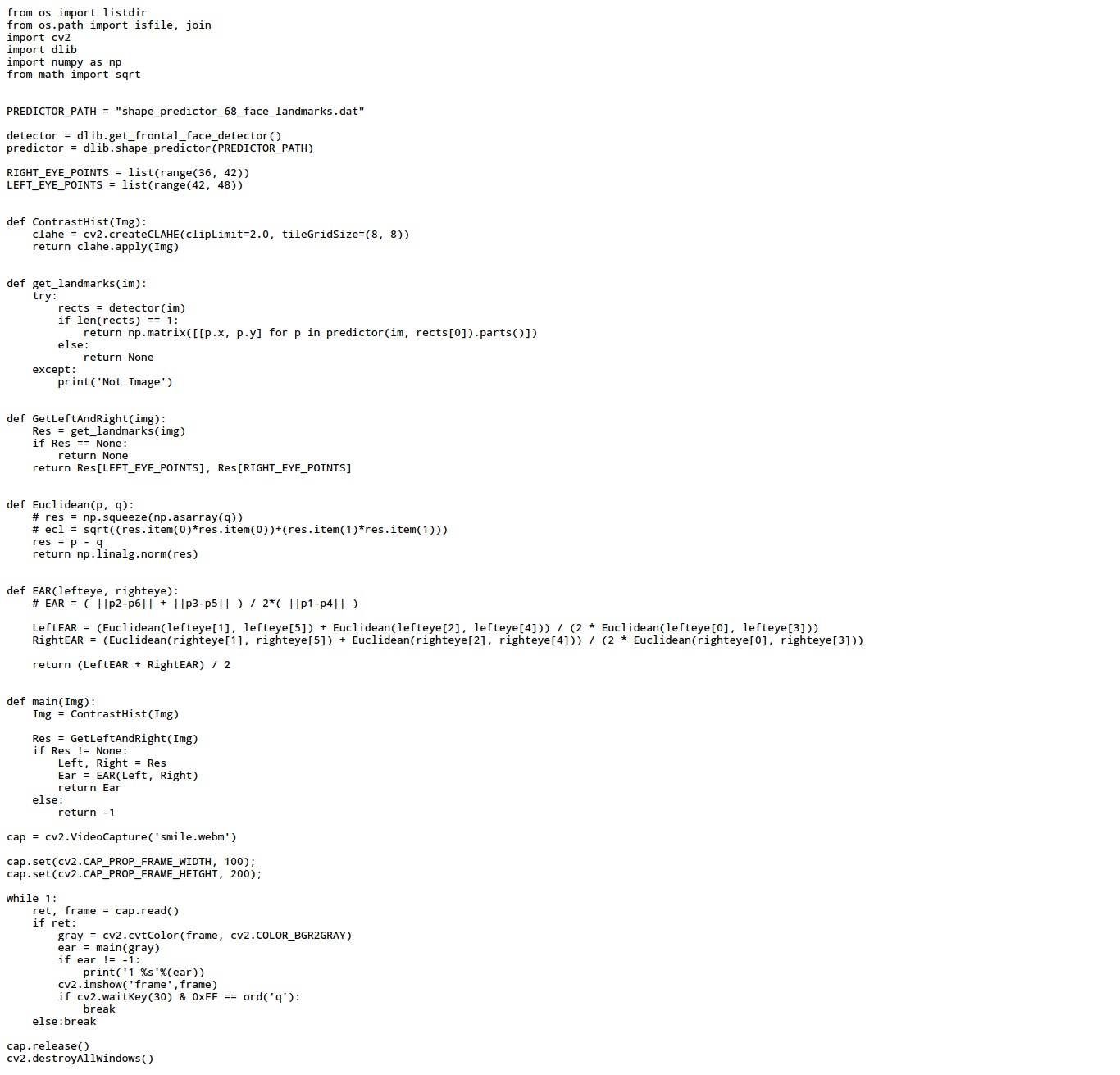


Alarm Class

Start method: starts the alarm

Stop method: close the alarm

Status method: return current alarm status ‘opened/closed’



Training Class

Almost the same as full application but instead of camera to input of video capture we pass the video that we will train on and instead of show video and play alarm we print the result of ear plus the video label ‘0/1/-1’

On Linux i take this output using terminal on exiting data or text file for example:

python training.py >> openeye.txt