PROJECT REPORT

CIS – 721

FALL 2016

BY

VIVEK KATTA

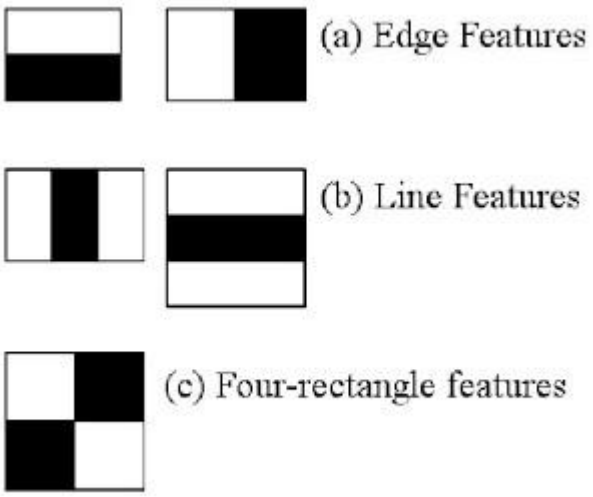
CONTENTS:

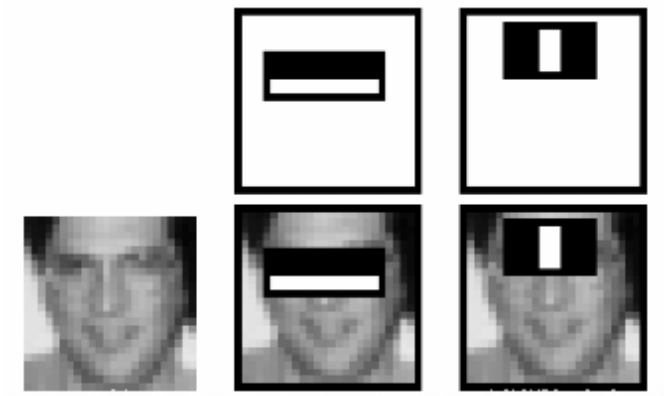
Introduction

Drunk driving is one of the major causes of accidents in US. In 2014, alone 9,967 people were killed in alcohol-impaired driving crashes, accounting for nearly one-third (31%) of all traffic-related deaths in the United States. There are many organizations that are working to make the people aware of this situation and educate them in this issue. The government is also working very hard by implementing laws very strictly. So, to add to this effort I created a device that can detect if the driver is impaired not only if they are drunk but also for any other reason. We may have a chance to decrease these fatalities. Using eye detection techniques, we can determine the state of the driver. By finding rate of blinking by the driver we can calculate to see if he/she is fit to drive. We can use this in all kinds of motor vehicles.

Implementation

Object Detection using Haar feature-based cascade classifiers is a very effective object detection method. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. For this, haar features shown in below image are used.





The first feature selected focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. By using many such features we can detect any property of a person like smile, eyes, face etc.,

For this project I have developed a program to identify the eyes and face of a person. We can detect if the person has opened the eyes or closed them also. So, for every certain amount of time we keep checking if the person has blinked more than he usually does. If they do, then we give out an alert.

For the purpose of demo, I have decided to check the blinking for every 20 seconds that is approximately 100 iterations. And if the person (in this case myself) blinks more than 20 times I have raised an alert. The output of the program if eyes are open is a message sating the eyes are open and the coordinates of the eyes. If eyes are closed we give out a message saying eyes are closed.

Code

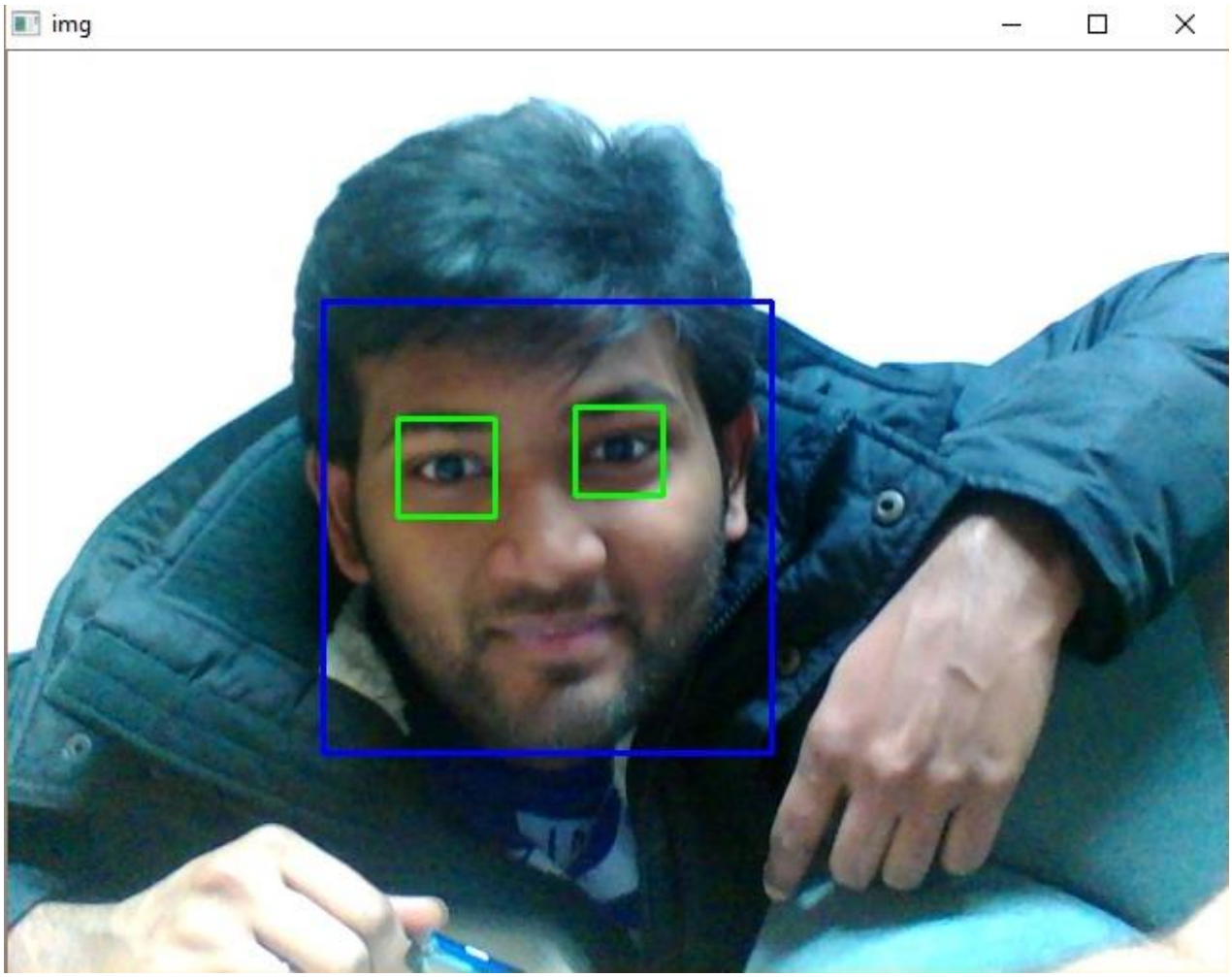
```
import numpy as np
import cv2
from __builtin__ import bool
from ctypes.wintypes import BOOLEAN


# multiple cascades: https://github.com/Itseez/opencv/tree/master/data/haarcascades

#https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml
class EyeBlinkDetector:
    def blink(self):

        face_cascade = cv2.CascadeClassifier('classifiers/haarcascade_frontalface_default2.xml')
        #https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_eye.xml
        eye_cascade = cv2.CascadeClassifier('classifiers/haarcascade_eye2.xml')

        cap = cv2.VideoCapture(0)
        i=0
        cl=[]
        while 1:
            ret, img = cap.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, 1.3,5)
            i=i+1
            if i%100==0:
                print len(cl)
                if len(cl)>10:
                    print "ALERTALERTALERTALERTALERTALERTALERTALERT",i
                    break
                cl=[]
```

```
            '''
            eyes = eye_cascade.detectMultiScale(gray,1.3,5)
            print eyes
            blink=False
            if (len(eyes)==0):
                blink = True
                print "blinked"

            else:
                print "open bro"
            for (ex,ey,ew,eh) in eyes:
                cv2.rectangle(img,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
            cv2.imshow('img',img)
            k = cv2.waitKey(30) & 0xff
            if k == 27:
                break


        '''
```

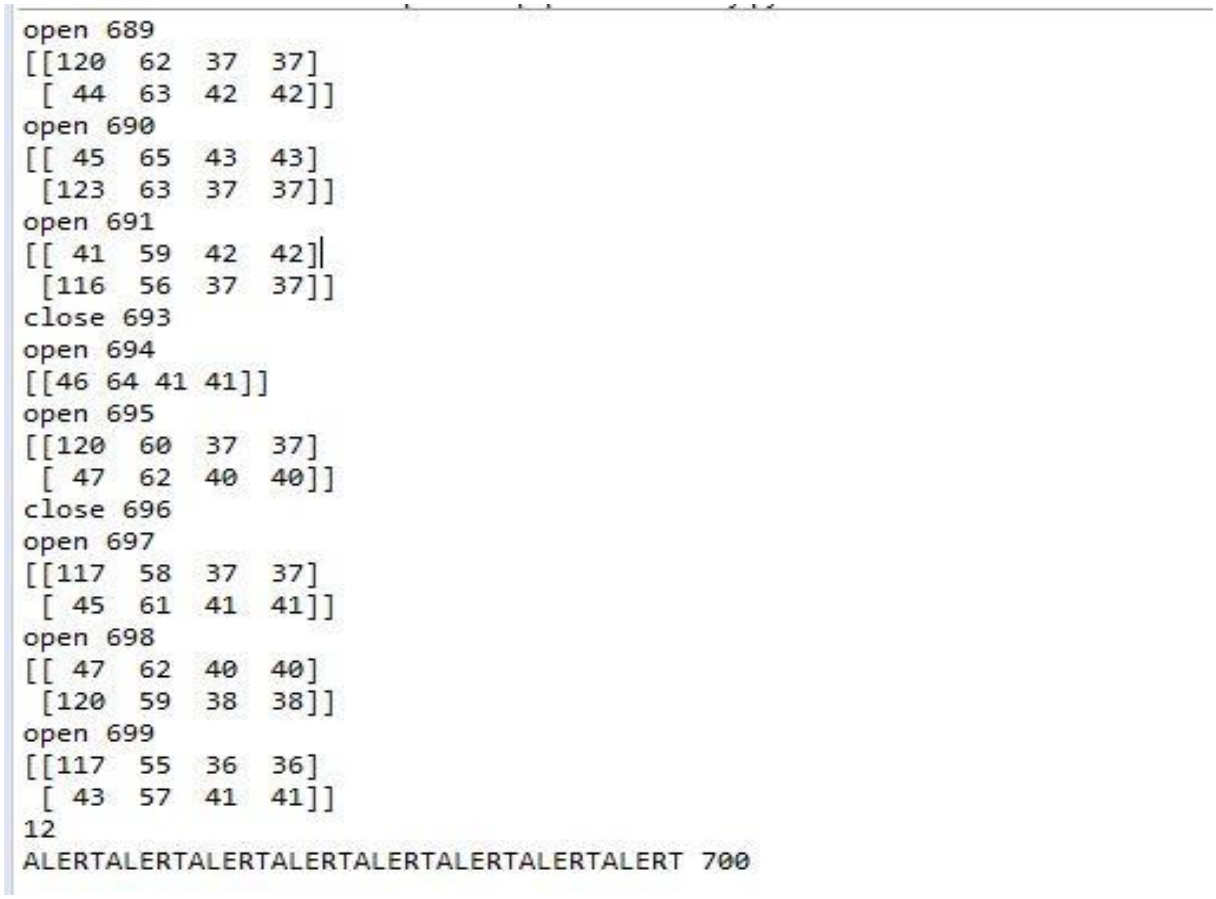
```
            for (x,y,w,h) in faces:
                cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
                roi_gray = gray[y:y+h, x:x+w]
                roi_color = img[y:y+h, x:x+w]

                eyes = eye_cascade.detectMultiScale(roi_gray)
                blink=False
                if (len(eyes)==0):
                    blink = True
                    print "close",i
                    cl.append("1")
                else:
                    print "open",i
                    print eyes
                for (ex,ey,ew,eh) in eyes:
                    cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

            cv2.imshow('img',img)
            k = cv2.waitKey(30) & 0xff
            if k == 27:
                break

        cap.release()
        cv2.destroyAllWindows()

cd=EyeBlinkDetector()
cd.blink()
```

By importing cv2 we can use the inbuilt packages for detecting eyes and face.

Demo

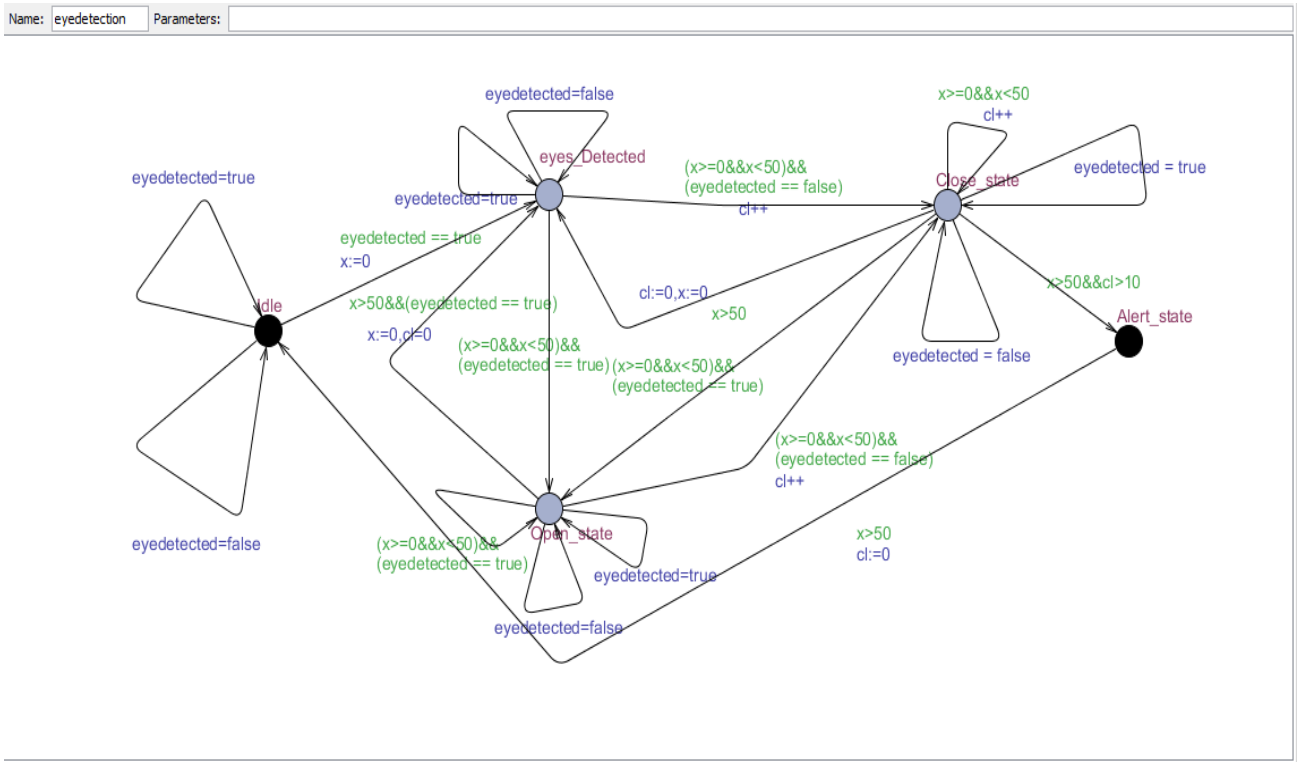


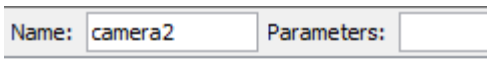
As shown in the image we can identify eyes of a person along with his face.

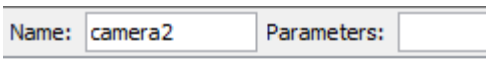


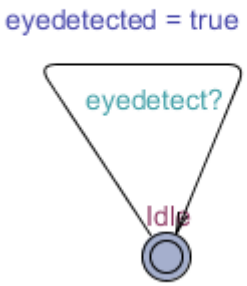
As can be observed from the above image if the eyes are opened we get a message "open" and the coordinates. If the eyes are closed, we can see the message "close" and the number of iteration. When the count of blinks reach 20 we can see the alert message.

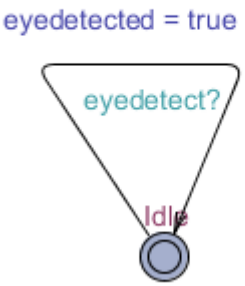
UPPAAL MODEL



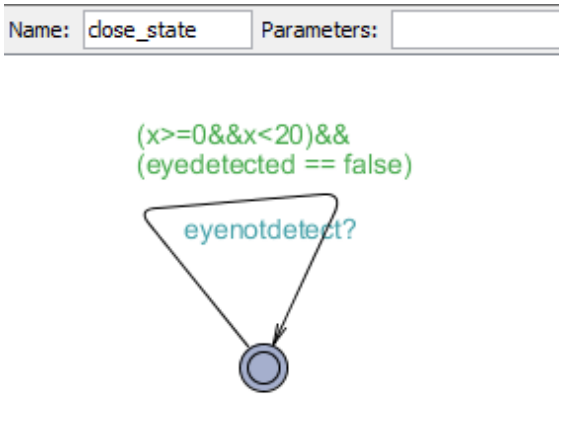


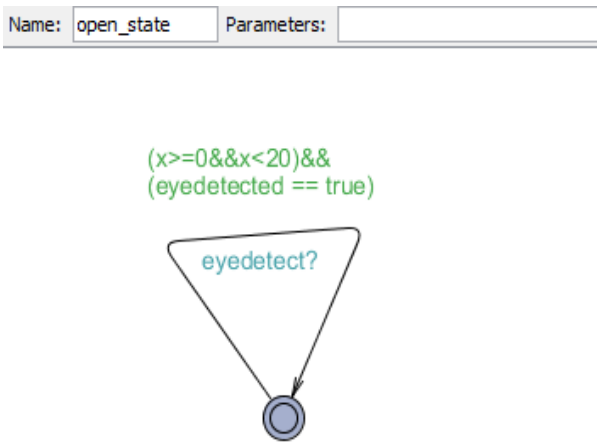




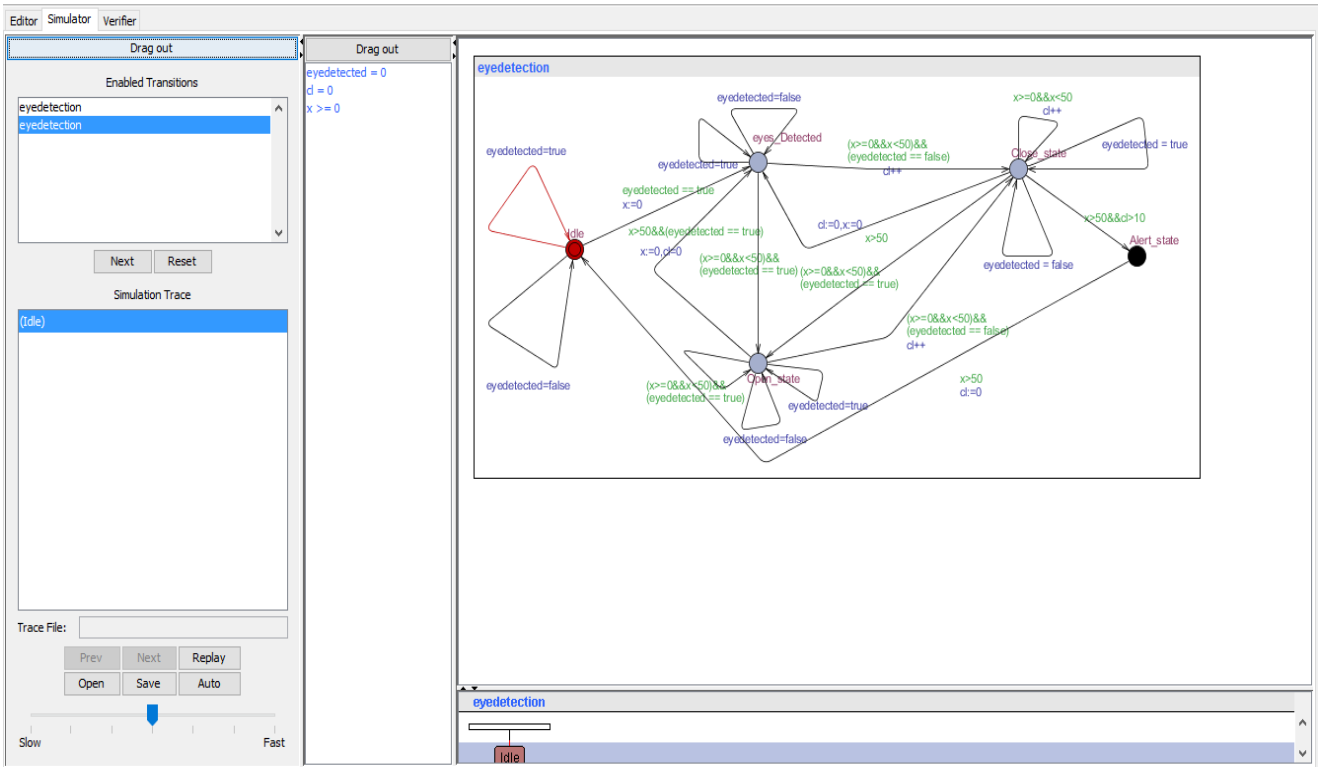


The above model is a depiction of the camera states. First one is if the eyes are detected other one is if they are not detected.

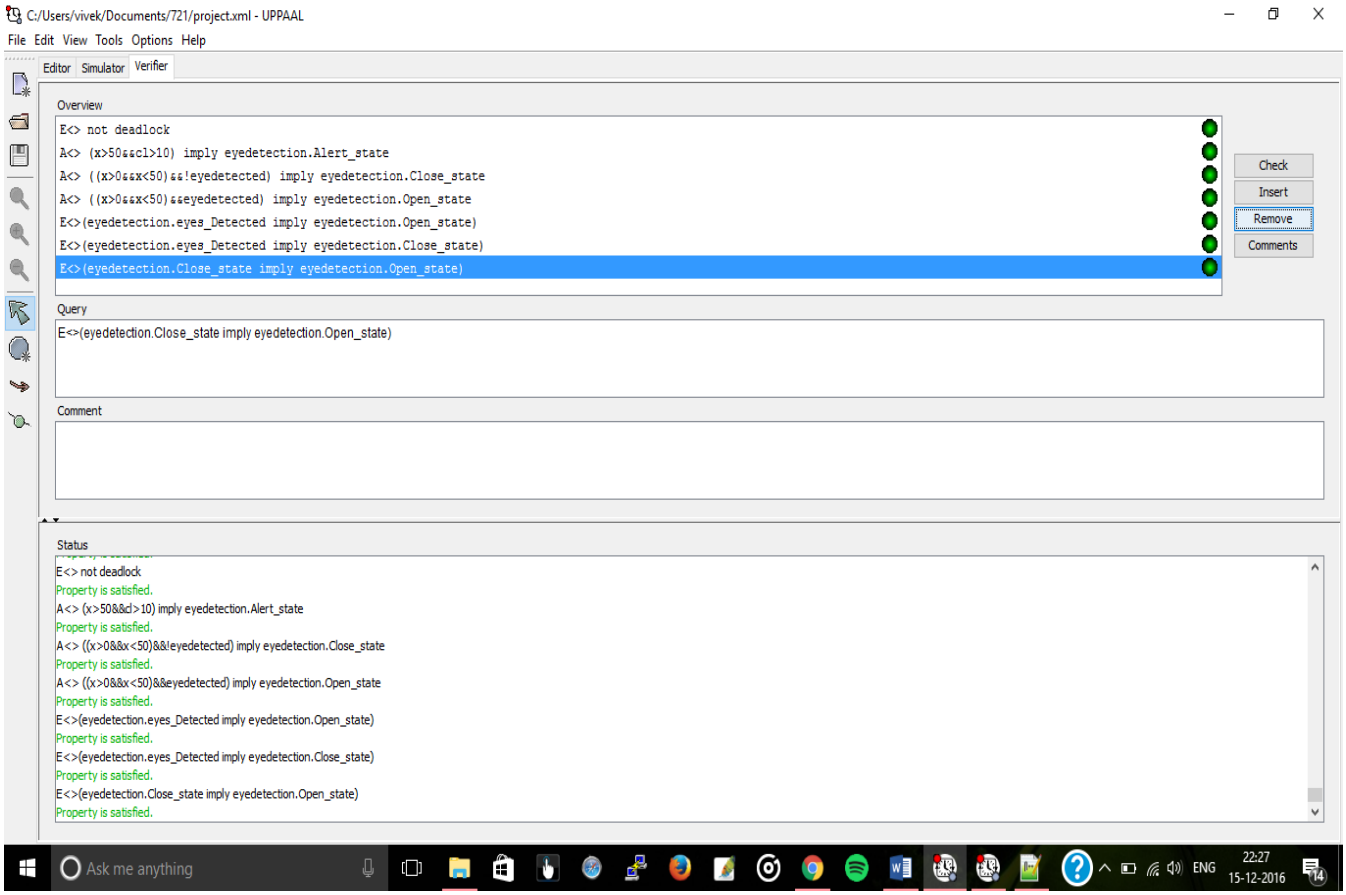




These two states are for the open and close states.



This is the simulator without any errors.



Some of the queries that I checked are as shown above. I have checked for deadlocks and the state reachability.

Conclusion

As stated in this project we can successfully detect if a person is blinking their eyes and calculate the rate at which they are blinking. In the future we can work on the hardware to make it compact so that it can fit on any helmet for motor cycles or bikes.

References

http://www.cdc.gov/motorvehiclesafety/impaired_driving/impaired-drv_factsheet.html

"Rapid Object Detection using a Boosted Cascade of Simple Features"

by Paul Viola and Michael Jones.

http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html