MIT Academy of Engineering

School of Computer Engineering And Technology

Presentation for Deep Learning Project

Real-Time Driver Drowsiness Detection System Using Deep Learning

By

Seat No.	Name	Roll No.
B224129	Ritesh Kulkarni	260
B224135	Rushikesh Jadhav	270
B224139	Uddhav Patil	276
B224142	Shruti Dhumne	267
B224151	Aditya Karpe	283

Guide

Mrs. P. V. Ugale

Problem Statement

Real Time Driver Drowsiness Detection System .

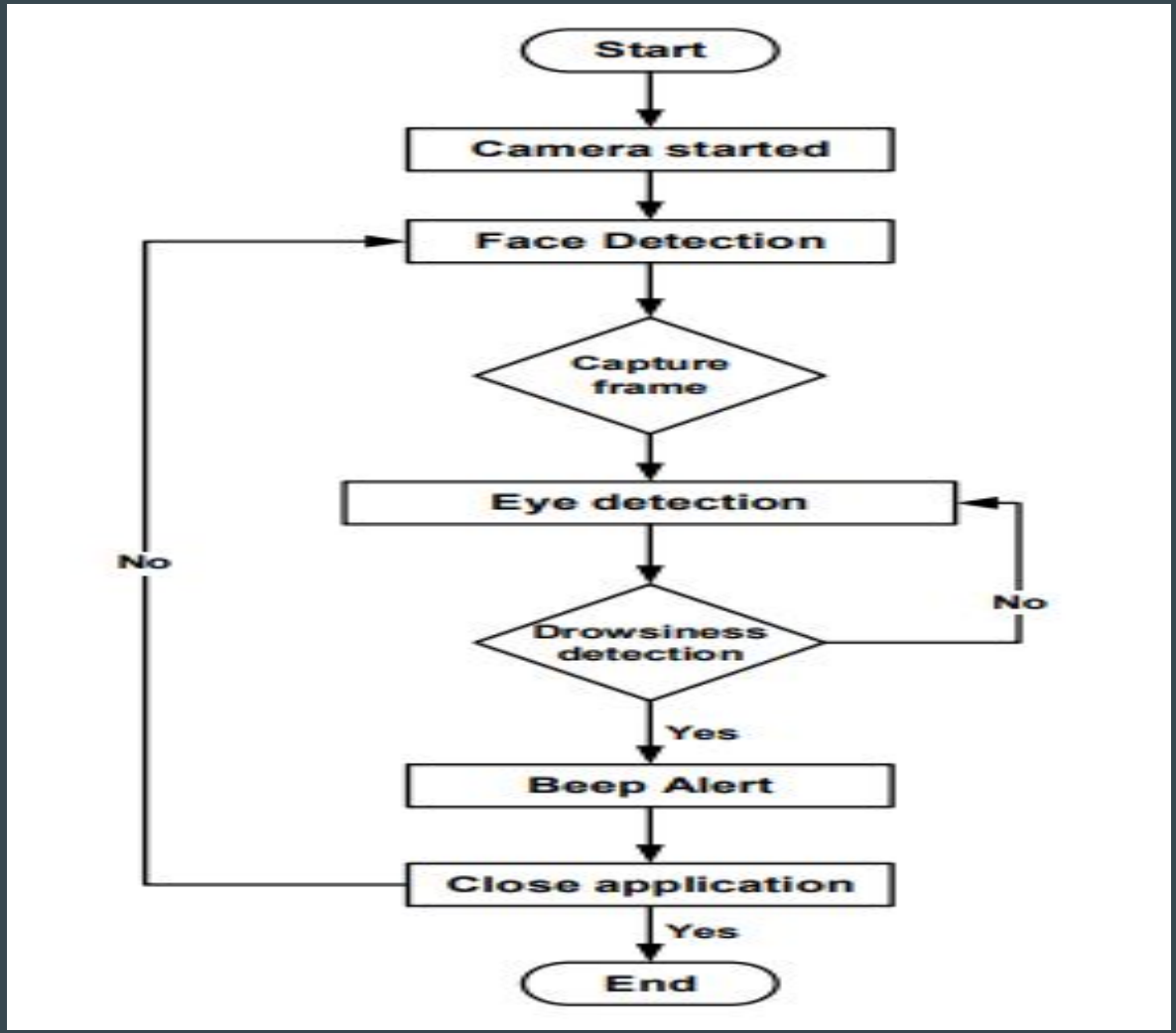
Motivation

- A countless number of people drive on the highway day and night. Taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep.
- Due to which it becomes very dangerous to drive when feeling sleepy.The majority of accidents happen due to the drowsiness of the driver.
- Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving.
- The objective of this intermediate Python project is to build a drowsiness detection system that will detect that a person's eyes are closed for a few seconds. This system will alert the driver when drowsiness is detected.

Abstract

The major aim of this project is to develop a drowsiness detection system by monitoring the eyes. It is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident. In such a case when drowsiness is detected, a warning signal is issued to alert the driver. This detection system provides a noncontact technique for judging different levels of driver alertness and facilitates early detection of a decline in alertness during driving. In such a case when fatigue is detected, a warning signal is issued to alert the driver. The system also can have additional feature of slowing down the vehicle if driver fails to respond to the alarm and ultimately stops the vehicle.

Real-Time Driver Drowsiness Detection System Work Flow

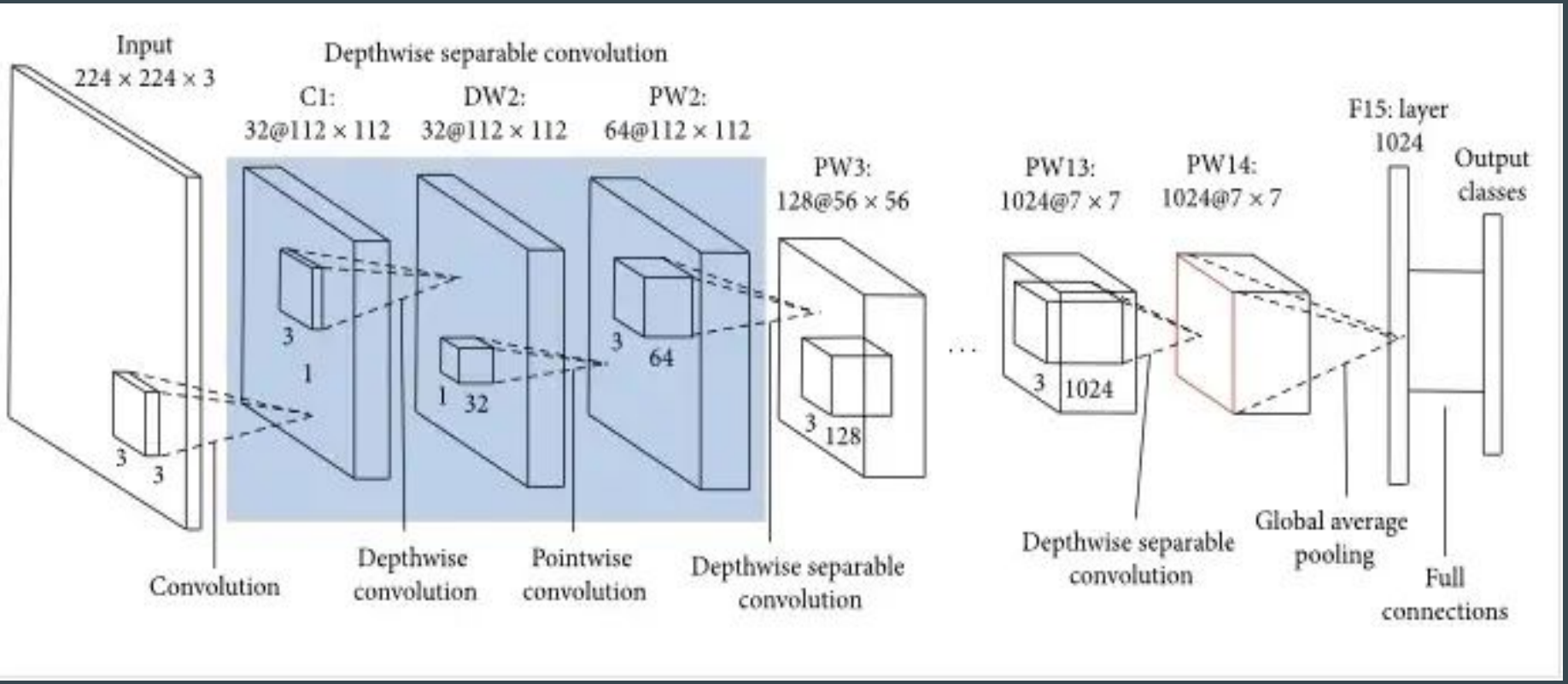


Methodology

MobileNet

1. MobileNets are built on depth wise separable convolution layers.Each depth wise separable convolution layer consists of a depthwise convolution and a pointwise convolution.
2. Counting depthwise and pointwise convolutions as separate layers, a MobileNet has 28 layers.A standard MobileNet has 4.2 million parameters which can be further reduced by tuning the width multiplier hyperparameter appropriately.
3. The size of the input image is **224 × 224 × 3**.

Architecture of MobileNet



Methodology

Haar Cascade

1. Haar cascade classifier is an open cv algorithm. It makes classification between images with an object (i.e face) and images without an object (i.e with non-faces).
2. Initially, several hundreds of images with face and several hundreds of images with non-faces have been given to this classifier.
3. This classifier was then trained by applying machine learning methods like the neural networks to recognize human faces. It then extracted Haar Features from those images and stored them in an xml file.

Haar Cascade

1. Basically for face detection, the classifier looks for the most relevant features on the face such as eyes, nose, lips, forehead, eyebrows because we know that although people have different looks, these features are in the similar positions on the face.

2. Haar features are white and black pixels on the face. Also, we know that the greyscale image of a face does not have completely white and black pixels but here we are considering an ideal case where white pixels are lighter pixels and black pixels are darker pixels.

Methodology

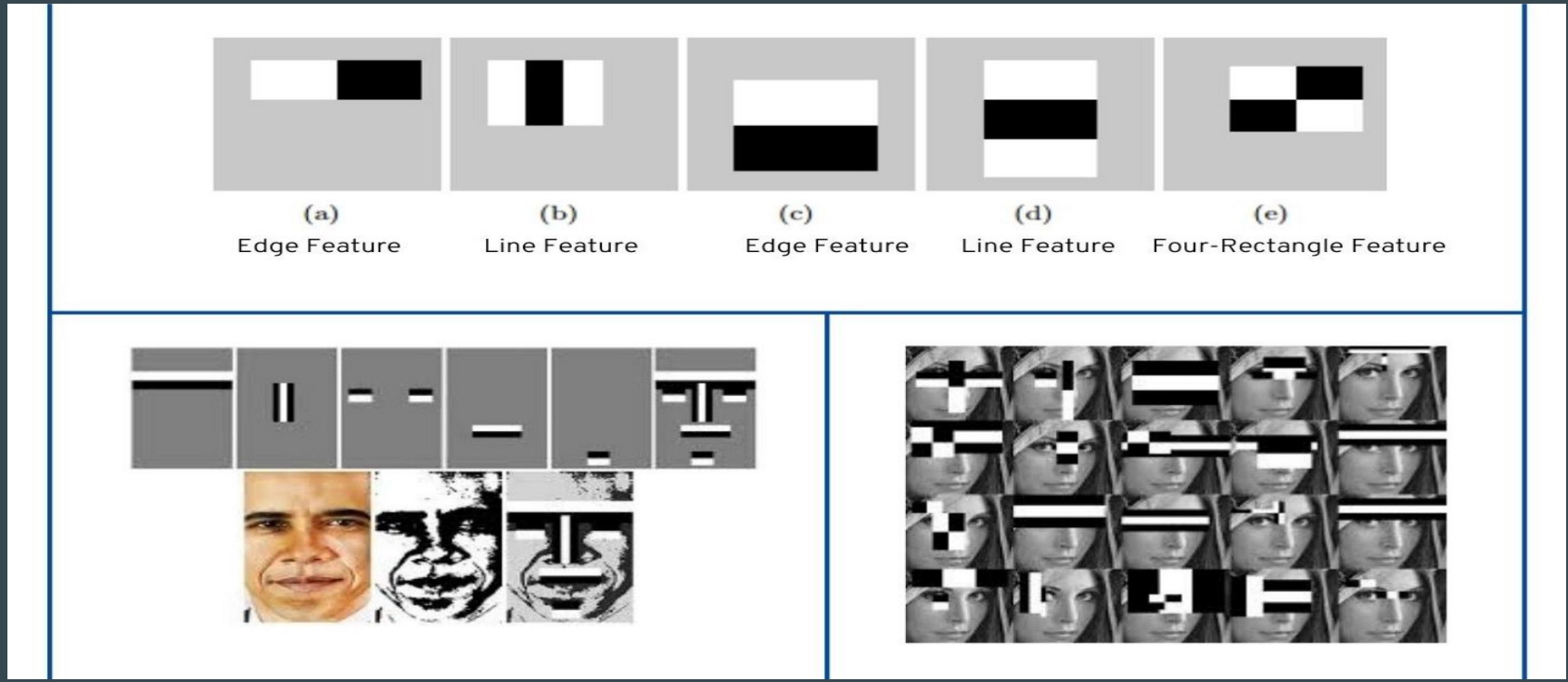


Fig:-Haar Features

Experimental Results

The experimental results showed that Mobilenet achieved around 87% accuracy which is high as compare to other CNN Architectures(VGG16(49%),ResNet(49%)) . This project uses video stream to detect driver-drowsiness, and this system has three part: human face tracking, feature exaction, and evaluation.

Experimental Results

Competitive Study of MobileNet,ResNet,Vgg

Training & Saving The Models

In [25]:
```
mobile_net_model.compile(loss = "binary_crossentropy", optimizer = "adam", metrics = ["accuracy"])
mobile_net_model.fit(X,Y, epochs = 1, validation_split = 0.2)
mobile_net_model.save('mobile_net_model.h5')## training
```

```
50/50 [==============================] - 452s 9s/step - loss: 1.8051 - accuracy: 0.8756 - val_loss: 8.4734 - val_accuracy: 0.44
84
```

In [46]:
```
resnet_model.compile(loss = "binary_crossentropy", optimizer = "adam", metrics = ["accuracy"])
resnet_model.fit(X,Y, epochs = 1, validation_split = 0.2)
resnet_model.save('resnet_model.h5') ## training
```

```
50/50 [==============================] - 1850s 36s/step - loss: 7.7704 - accuracy: 0.4956 - val_loss: 7.4988 - val_accuracy: 0.
5139
```

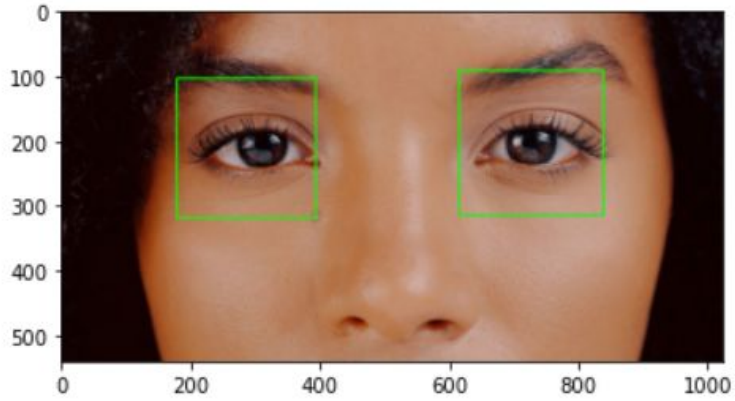
In [47]:
```
vgg_model.compile(loss = "binary_crossentropy", optimizer = "adam", metrics = ["accuracy"])
vgg_model.fit(X,Y, epochs = 1, validation_split = 0.2)
vgg_model.save('vgg_model.h5')## training
```

```
50/50 [==============================] - 1777s 35s/step - loss: 7.7416 - accuracy: 0.4981 - val_loss: 7.4988 - val_accuracy: 0.
5139
```

Resultant Outputs

Positive Prediction Value For Open Eyes

```
In [63]: plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
Out[63]: <matplotlib.image.AxesImage at 0x1f61759c460>
```

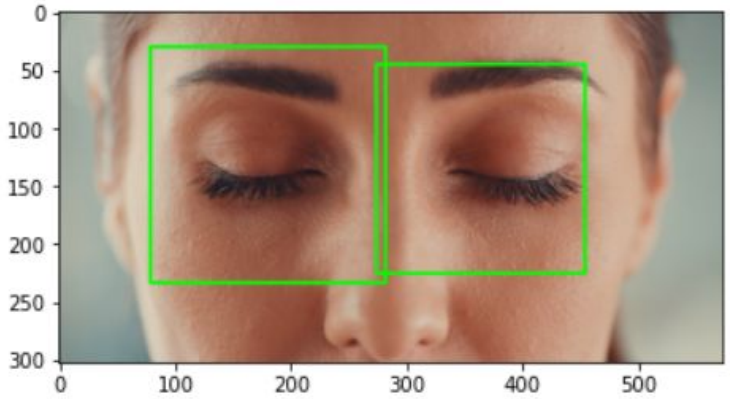



```
In [64]: new_model.predict(final_image)
1/1 [==============================] - 0s 467ms/step
Out[64]: array([[12.225228]], dtype=float32)
```

Resultant Outputs

Negative Prediction Value For Open Eyes

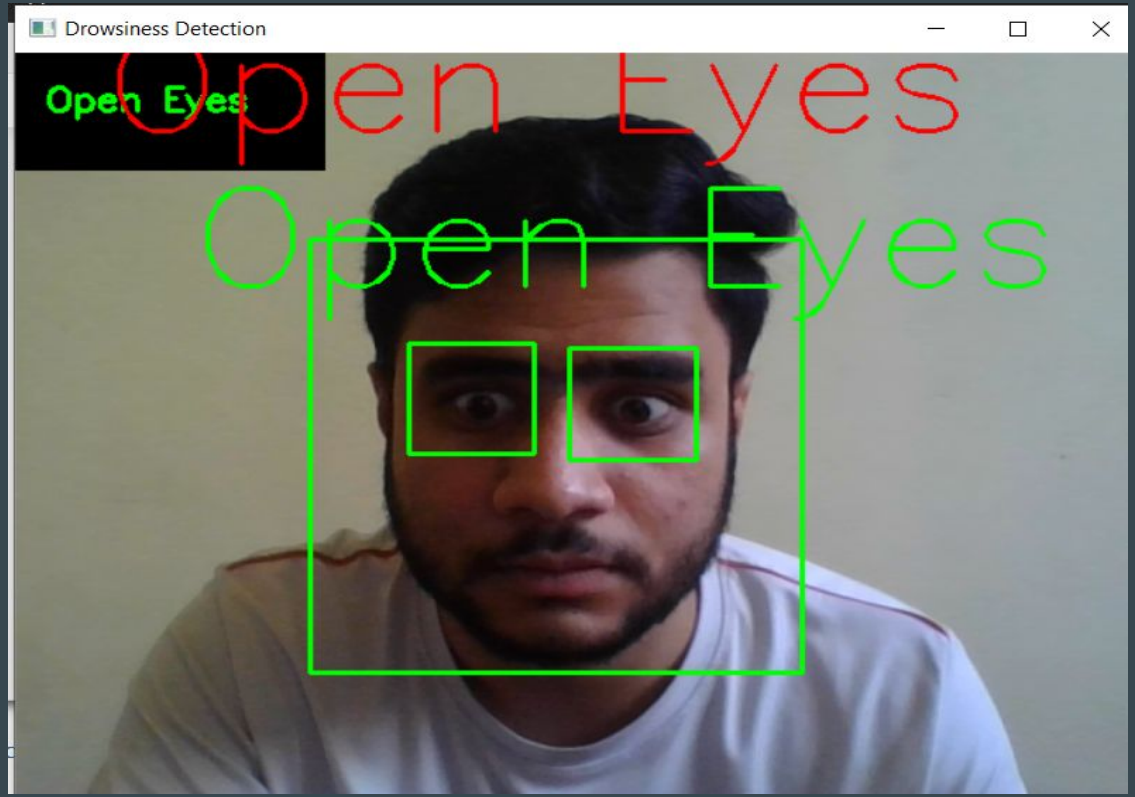
```
In [108]: plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
Out[108]: <matplotlib.image.AxesImage at 0x1f6115c4610>
```




```
In [109]: new_model.predict(final_image)
          1/1 [==============================] - 0s 209ms/step
Out[109]: array([[-11.397864]], dtype=float32)
```

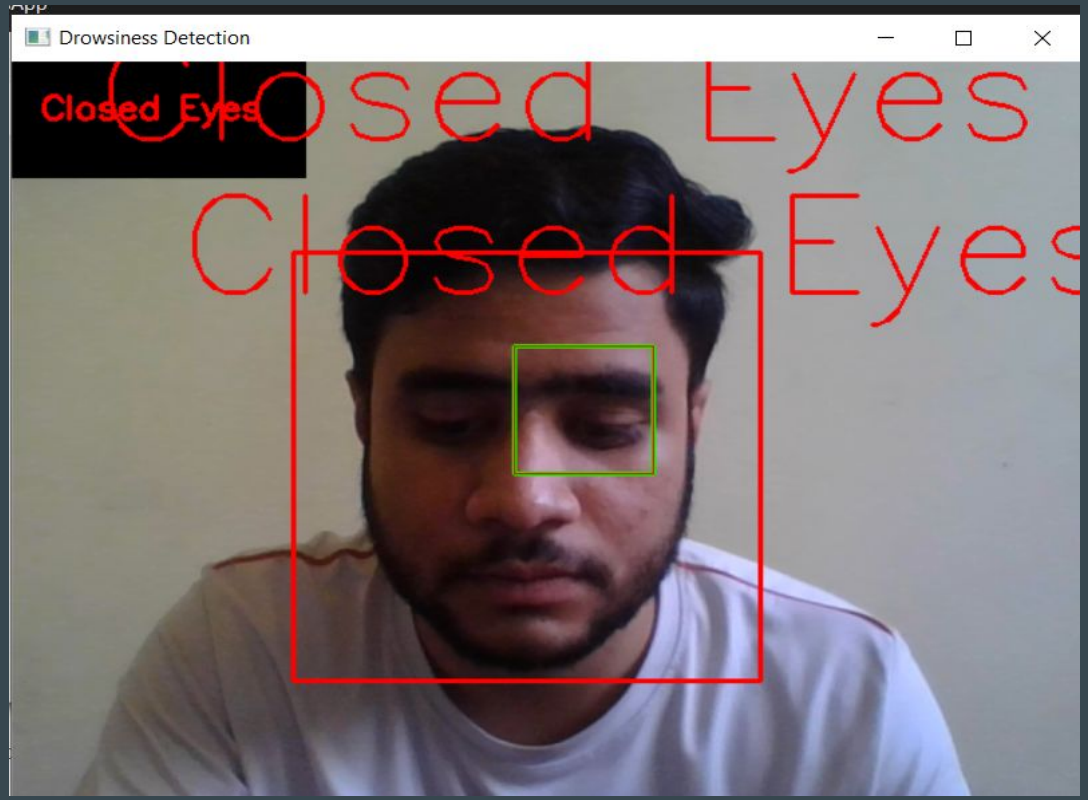
Resultant Outputs

Real Time Drowsiness Detection (Open Eyes)



Resultant Outputs

Real Time Drowsiness Detection (Close Eyes)



Future Scope

1) Capture individual drivers steering activity while drowsy.

2) Conduct additional simulator experiments to validate the algorithm, test additional road conditions, and test a more diversified group of drivers.

3) Test and refine the algorithm based on the road test data, and conduct research on warning systems integrated with the detection system.

Conclusion

1) Image processing achieves highly accurate and reliable detection of drowsiness.
2) Image processing offers a non-invasive approach to detecting drowsiness without the annoyance and interference.
3) A drowsiness detection system developed around the principle of image processing judges the drivers alertness level on the basis of continuous eye closures.With 92% accuracy, it is obvious that there are limitations to the system.

Thank You!!!!