DRIVER DROWSINESS DETECTION SYSTEM

Department of Computer Science and Engineering

Under the supervision of :
Dr. Rajib Ghosh
Assistant Professor
NIT PATNA

Vineet Raj (2106019)
Aditya Kumar (2106011)
Rupesh (2106005)

Outline

Introduction

Problem Statement

Objective

Proposed Methodology

Dataset

Result

References

Introduction

Driver drowsiness refers to a state of decreased alertness and impaired performance caused by physical and mental fatigue. Factors such as lack of sleep, long driving hours, night driving, or monotonous road conditions can lead to drowsiness, reducing a driver’s ability to respond quickly to traffic events.

In today's fast-paced world, road safety has become a critical concern. One of the leading causes of road accidents is driver drowsiness. Fatigue-related crashes can be just as dangerous as those caused by drunk driving. Therefore, it is essential to understand the importance of detecting driver drowsiness in real-time, which can significantly reduce accidents, save lives, and enhance road safety.

Problem Statement

Accidents caused due to drowsy driving are a major problem in the World.

The National Highway Traffic Safety Administration estimates that drowsy driving was responsible for 72,000 crashes, 44,000 injuries, and 800deaths in 2013[1].

Drowsiness detection technologies have attempted to prevent such incidents by predicting if a driver is falling asleep based on various inputs.

Objective

This project outlines the design and development of a system that focuses on driver’s drowsiness detection and prediction.

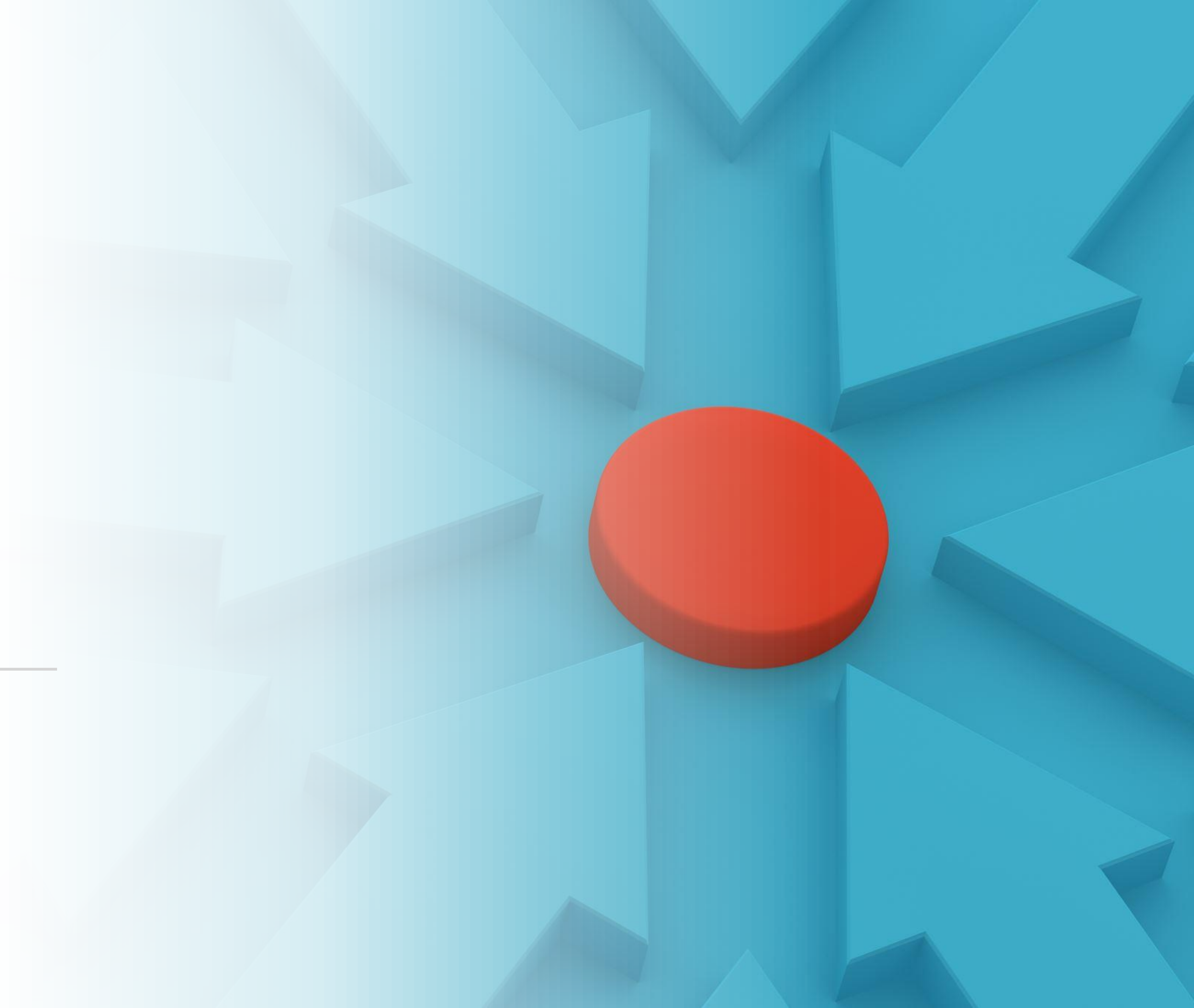
Monitoring the driver behaviour by observing the manoeuvre stability and performance.

Validate and measure the progress by using Specific algorithm.

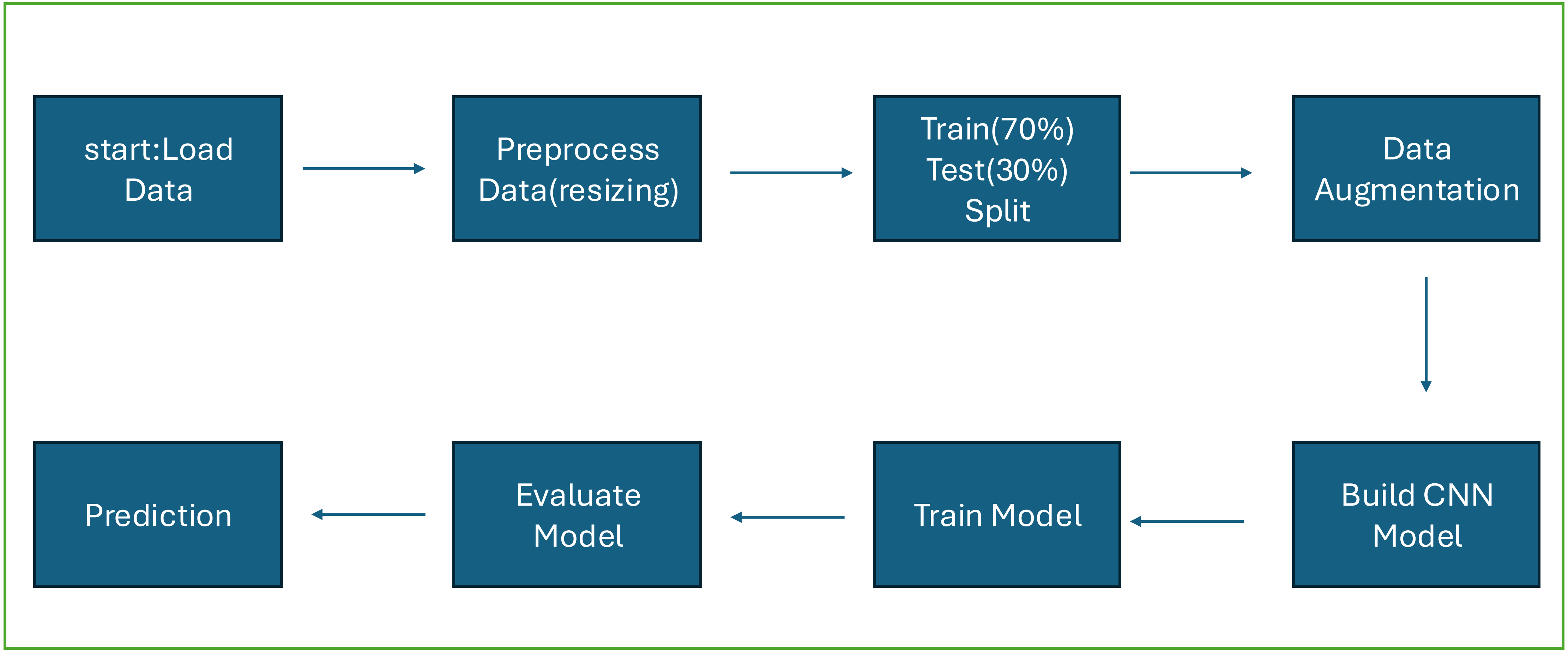
Warning the drivers if the behaviour beyond the thresholds.

Here we will employ machine learning methods to classify the data of actual human behaviour during drowsiness.

Proposed Methodology



Flow Diagram



Method

The proposed model includes the following key components and steps:

Data Collection and Preprocessing

Load images from specified directories (archive/train). Detect faces in the images using Haar Cascade Classifier. Resize the detected faces to a uniform size (145x145 pixels).

Data Augmentation

Apply transformations such as zoom, horizontal flip, and rotation to enhance the dataset and prevent overfitting.

Model Building

Construct a CNN with multiple convolutional and pooling layers to extract features from the input images.

Method(contd)

Model Training

Train the model using the augmented data, utilizing categorical cross-entropy as the loss function and Adam optimizer.

Model Evaluation

Validate the model on a separate test dataset and visualize training/validation accuracy and loss.

Prediction

Implement a prediction function to classify input images into categories: yawn, no_yawn, Closed, and Open.Use the trained model to predict the state of the driver based on input images.

Dataset

- Closed Files : 726
- Open eyes : 726
- Yawn : 726
- No Yawn : 726

Result

```
Epoch 1/50
43/43 [==============================] - 422s 10s/step - loss: 1.1787 - accuracy: 0.4595 - val_loss: 0.8622 - val_accuracy: 0.6747
Epoch 2/50
43/43 [==============================] - 412s 10s/step - loss: 0.6066 - accuracy: 0.7506 - val_loss: 0.6524 - val_accuracy: 0.7180
Epoch 3/50
43/43 [==============================] - 408s 9s/step - loss: 0.4214 - accuracy: 0.8411 - val_loss: 0.3382 - val_accuracy: 0.8668
Epoch 4/50
43/43 [==============================] - 334s 8s/step - loss: 0.3222 - accuracy: 0.8656 - val_loss: 0.4703 - val_accuracy: 0.8270
Epoch 5/50
43/43 [==============================] - 257s 6s/step - loss: 0.3199 - accuracy: 0.8708 - val_loss: 0.2397 - val_accuracy: 0.9048
Epoch 6/50
43/43 [==============================] - 247s 6s/step - loss: 0.2934 - accuracy: 0.8775 - val_loss: 0.2595 - val_accuracy: 0.8858
Epoch 7/50
43/43 [==============================] - 6473s 154s/step - loss: 0.2589 - accuracy: 0.9035 - val_loss: 0.2098 - val_accuracy: 0.9048
Epoch 8/50
43/43 [==============================] - 207s 5s/step - loss: 0.2530 - accuracy: 0.8990 - val_loss: 0.2265 - val_accuracy: 0.8979
Epoch 9/50
43/43 [==============================] - 388s 9s/step - loss: 0.2421 - accuracy: 0.9072 - val_loss: 0.2006 - val_accuracy: 0.9048
Epoch 10/50
43/43 [==============================] - 395s 9s/step - loss: 0.2165 - accuracy: 0.9079 - val_loss: 0.1884 - val_accuracy: 0.9170
Epoch 11/50
43/43 [==============================] - 392s 9s/step - loss: 0.1997 - accuracy: 0.9161 - val_loss: 0.1857 - val_accuracy: 0.9187
Epoch 12/50
43/43 [==============================] - 387s 9s/step - loss: 0.1970 - accuracy: 0.9109 - val_loss: 0.1609 - val_accuracy: 0.9239
Epoch 13/50
...
```

Fig 1

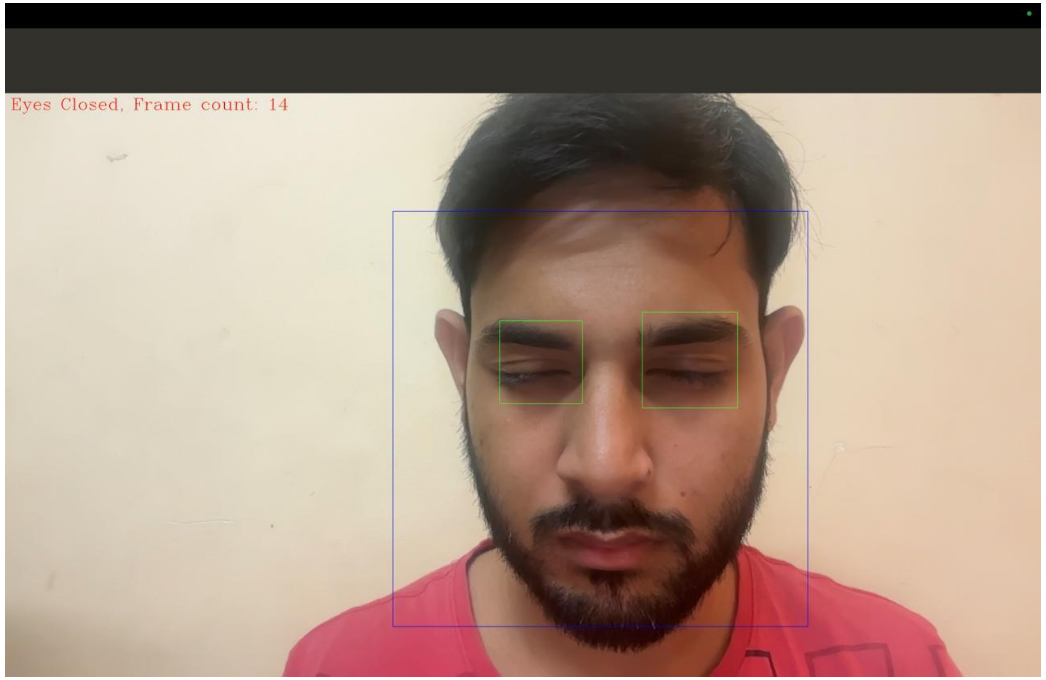
Result(contd)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 143, 143, 256)     7168

 max_pooling2d (MaxPooling2D) (None, 71, 71, 256)      0

 conv2d_1 (Conv2D)           (None, 69, 69, 128)       295040

 max_pooling2d_1 (MaxPooling2 (None, 34, 34, 128)      0

 conv2d_2 (Conv2D)           (None, 32, 32, 64)        73792

 max_pooling2d_2 (MaxPooling2 (None, 16, 16, 64)       0

 conv2d_3 (Conv2D)           (None, 14, 14, 32)        18464

 max_pooling2d_3 (MaxPooling2 (None, 7, 7, 32)         0

 flatten (Flatten)           (None, 1568)              0

 dropout (Dropout)           (None, 1568)              0

 dense (Dense)               (None, 64)                100416
...
Total params: 495,140
Trainable params: 495,140
Non-trainable params: 0
```

Fig 2

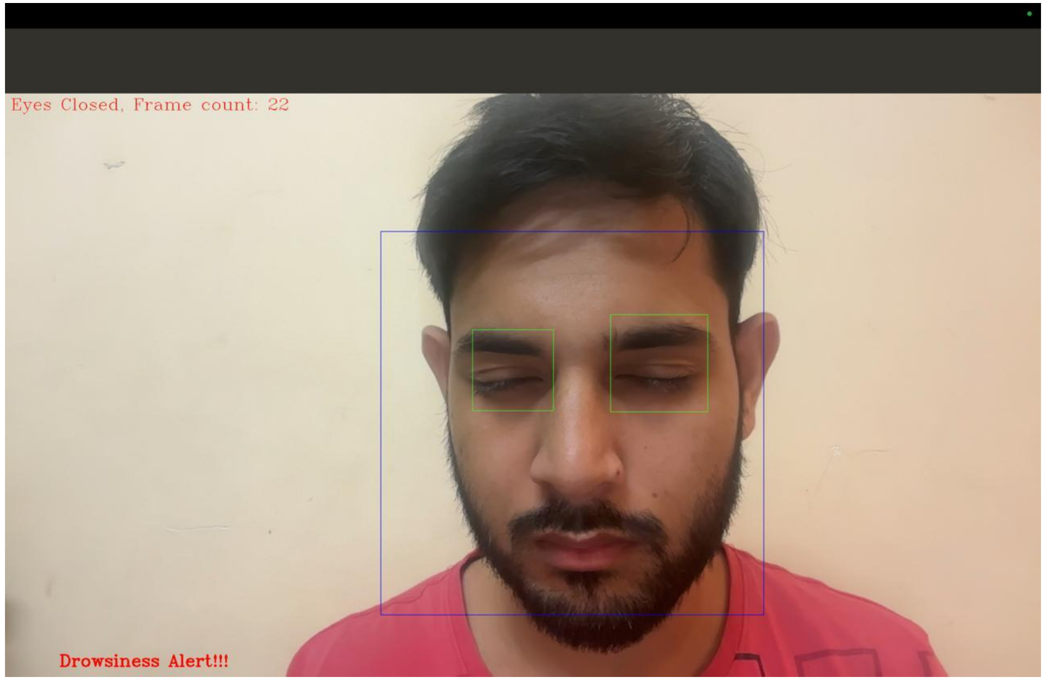
Result(contd)

Number of Frames : 20
Time taken : 5 sec
Frame Rate : 20/5 = 4fps



Alarm not beep

Fig 3



Alarm beep

Fig 4

Result(contd)

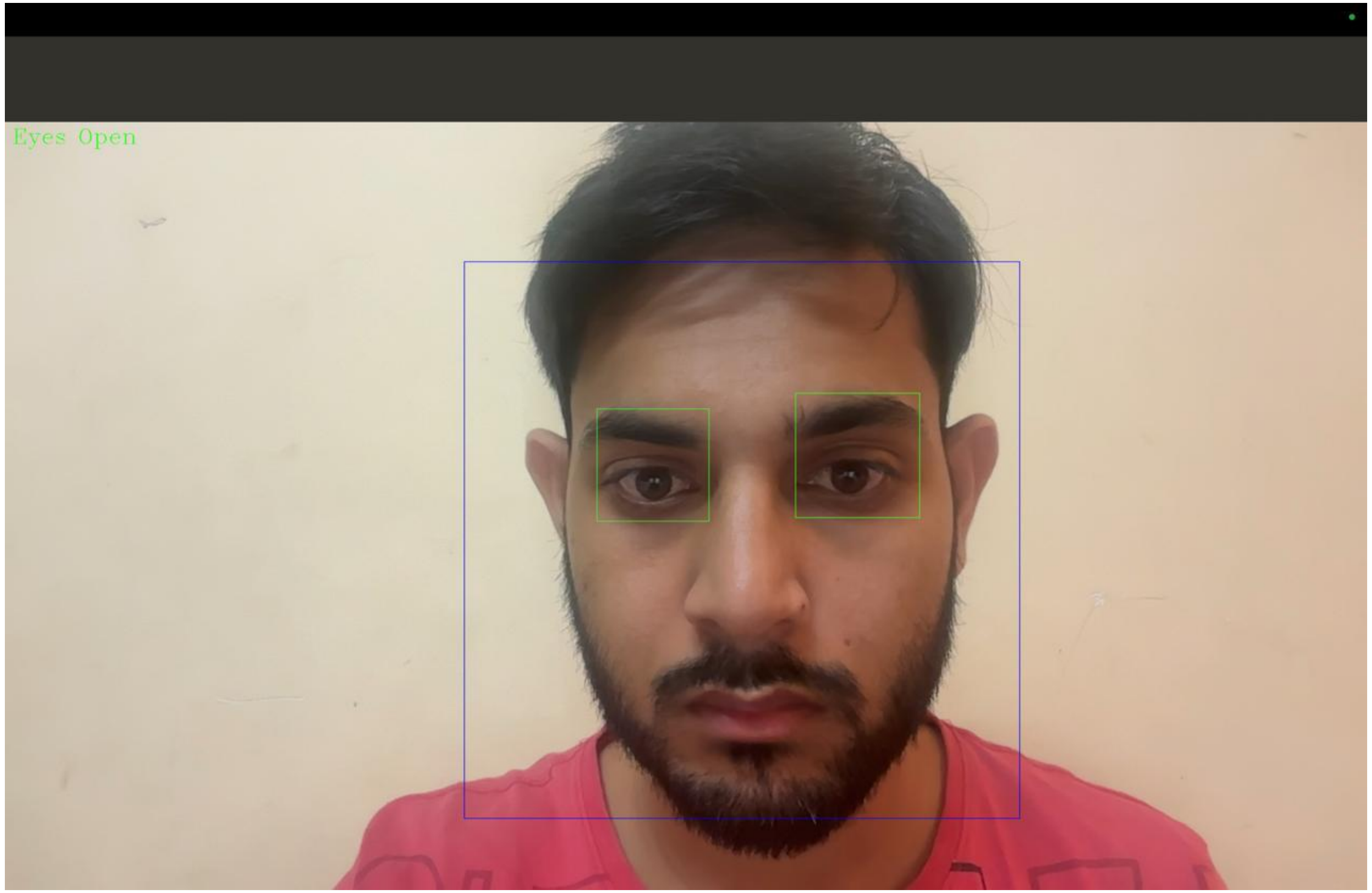


Fig 5

Future Work

1. **Real-time Implementation:** Enable seamless operation with live video feeds from dashboard cameras.

2. **Hardware Integration:** Link the system to vehicles for automatic safety measures like slowing down or emergency alerts.

3. **Multimodal Analysis:** Enhance accuracy by integrating visual data with heart rate and steering behavior sensors.

4. **Mobile Application Development:** Create a smartphone or wearable app for real-time drowsiness alerts.

5. **Driver Behavior Prediction:** Analyze driving patterns to predict and preempt drowsiness.

Conclusion

Developed a system to detect driver drowsiness using machine learning.

Effectively predicts fatigue through behavioral analysis.

Aims to reduce road accidents and enhance safety.

Supports safer driving through timely intervention.

Aims to make a cost-effective product for all type of vehicles

References

Driver drowsiness detection with eyelid related parameters by SupportVector Machine

Hu Shuyan, Zheng GangtieSchool of Astronautics, Beijing University of Aeronautics and Astronautics, 37 Xueyuan Road, Haidian District, Beijing 100083, China