Driver Drowziness Detection System

CSE541: Computer Vision

Rajan Nandha AU2040101

Kenil Shah AU2040111

Yesha Dhivar AU2040215

***Abstract*—Driver fatigue is a major cause of vehicle accidents globally, hence, there is a need to detect drowsiness to save lives and property. This project aims to develop a non-intrusive drowsiness detection system that continuously captures images and measures the state of the driver's eyes. This model is based on the InceptionV3 architecture of the Convolutional Neural Network (CNN). There is an alert system as well which alerts the drowsy driver. The accuracy of the InceptionV3 is found to be 90%.**

***Keywords*— Drowsiness, status of eye, non-intrusive, image capture, Haar-cascade, InceptionV3**

I. Introduction

A camera is located near the driver's seat which will monitor the real-time images of the driver. The dataset is taken from 'MRL Eye Dataset by *Fusek, R.*. Here, we used Convolutional Neural Network (CNN) architecture InceptionV3 for classification of eye status. Further, we developed a real-time drowsiness detection module which will alert the driver if the driver is found to be drowsy or sleepy.

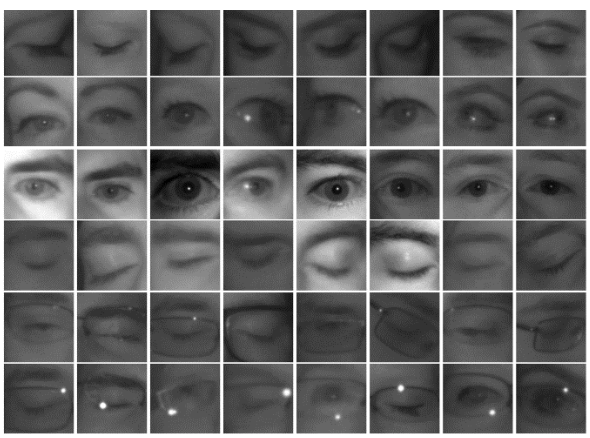

Figure 1. Sample of Dataset

II. Literature Survey

The paper titled 'A Comparative Study on Machine Learning Techniques for Intrusion Detection in Wireless Sensor Networks' presents a comparative analysis of different machine learning techniques for intrusion detection in wireless sensor networks. The paper proposes a drowsiness detection system that uses Electroencephalogram (EEG) signals and machine learning algorithms like Discrete Wavelet Transform (DWT), Support Vector Machine (SVM) and K-Nearest Neighbour (KNN).

A paper titled 'Driver Drowsiness Detection' presents a comparative study of various techniques for driver drowsiness detection and alert systems. One method uses an arithmetic-based approach to detect fatigue by tracking eye movements through a camera. Another system uses the shape predictor algorithm to identify facial features associated with drowsiness. The systems use computer vision and machine learning algorithms such as Viola Jones, AdaBoost, and CAMSHIFT. The proposed systems are non-intrusive, effective, and can be improved by adding various types of sensors. The output of the system is in the form of an alarm or buzzer that alerts the driver when drowsiness is detected. The paper also groups drowsiness detection techniques into driver-based and vehicle-based and surveys various methods in each category.

III. Implementation

A. *Dataset*

We have obtained our dataset from 'MRL Eye Dataset' which contains close and open eye images. This dataset consists of 84898 unique images. It has images of different gender, of people with spectacles, with varied lighting conditions, with varied size of reflections. The dataset contains images captured using three different sensors; Intel RealSense RS 300 sensor, IDS Imaging sensor, and Aptina sensor.

B. *Problem Statement*

Our objective is to build a model which detects the drowsiness of the driver by observing the status of the eye and alert the driver about it.

C. *Pre-processing of Data*

We have done random rotation, shearing, zooming, horizontal shifting, vertical shifting of around 20% to introduce additional variation to the training data. This operations helps in improving the ability of the model to generalize to new and unseen data. We have rescaled the pixel values of images to normalise them to range [0,1]. We have reduced the image size to 80x80 pixels. The images are kept in RGB (Red, Green, Blue) channel only.

D. *InceptionV3 Model*

1. InceptionV3 is a pre-trained model on the ImageNet dataset, which contains millions of images. This means that you can use the pre-trained weights of the model and fine-tune them on our dataset. This also

makes the model robust to variations in lighting, colour and other factors affecting the image quality.

2. It uses a combination of convolutional layers with different kernel sizes and pooling layers to extract features from images at multiple scales. This allows the model to better capture complex patterns and relationships between different parts of the image, which can be particularly useful when detecting subtle differences between open and closed eyes.

3. It can handle input images of varying sizes. This makes it easier to use the model with images that have different resolutions, as our dataset has images of three different resolution.

4. Additionally, it also solves the vanishing gradient problem occurring while backpropagation through the network by using the *'ReLU activation'* function.

E. Tuning of Parameters and Hyperparameters

1. We have kept the number of epochs to be 5. This was so because when the number of epochs was kept 10, the early stopping criteria was met, which caused the training process to stop before completing all 10 epochs. Along with this, we also observed that the validation loss has stopped improving significantly after the 2nd epoch.

2. The batch size was kept 8 to have a faster convergence and avoid overfitting of data.

F. Real-time Module

We have developed a real-time module which would process the real-time images of driver to detect the state of the driver. Here, the frame processing rate would depend on the camera used. Before the state of the eye is predicted, the frame images undergo preprocessing steps like rescaling the pixel values and resizing. It would then observe the state of the eye from the video frames and predict the state using the trained model.

We have kept a confidence threshold for open and close eye, which are 0.9 and 0.3 respectively. A score would be maintained which indicates the number of frames in which the driver is found with close eyes. If this score becomes greater than 15, then an alarm sound plays to alert the driver. This score would increase when a close eye is encountered and would decrease when an open eye is encountered.

IV. Results

Figure 2 shows the change in loss after every epoch for training data and validation data. The orange line depicts the validation loss change and blue line depicts the training loss change. One can see that the validation loss remains consistent after the 3rd epoch. Similarly, the training loss does not change significantly after 2nd epoch and it completely stops improving after the 4th epoch. Due to this reason, the graph has data only till the 4th epoch. The final loss obtained is around 12.72%, 23.57% and 30.77% for training, validation and test data, respectively.

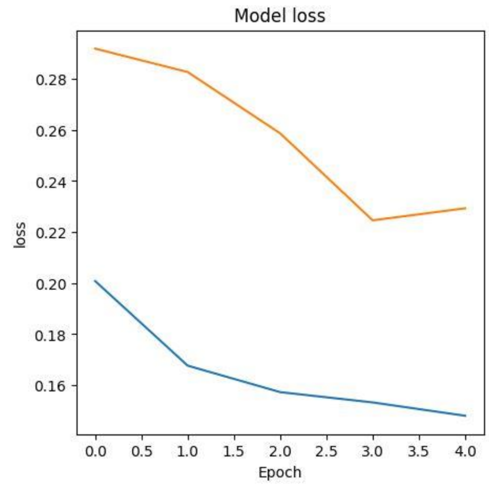

Figure 2. Loss v/s Epoch Graph

Figure 3 shows the change in accuracy after every epoch for training data and validation data. Similar to the Figure 2, the orange line depicts the validation accuracy change and blue line depicts the training accuracy change. One can see that the training accuracy nearly attains the saturation level at the 3rd epoch. However, the validation accuracy seems to improve over the course of 4 epochs, as the loss as well as accuracy stops changing after the 4th epoch. The final accuracy obtained is around 95.38%, 90.33% and 87.54% for training, validation and test data, respectively.

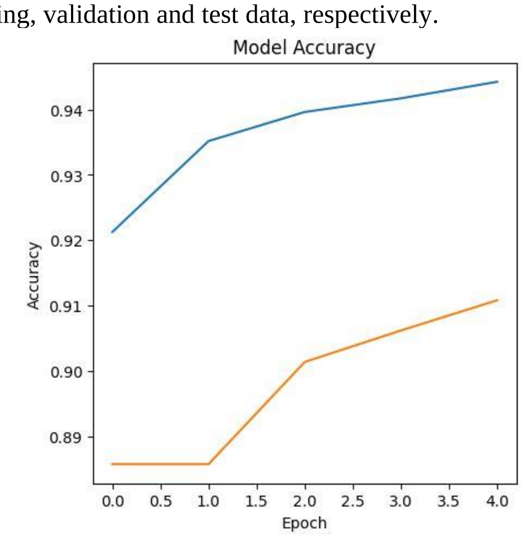

Figure 3. Accuracy v/s Epoch Graph

As shown in Figure 4, we can see that the state of eye along with the score is displayed at the bottom of the frame. This score and state of the eye is updated every time the frame changes. Here, the model can process the frames without significantly affecting the frame rate of the camera. There will be some loss of frame, however it will not significantly affect the result of the prediction.

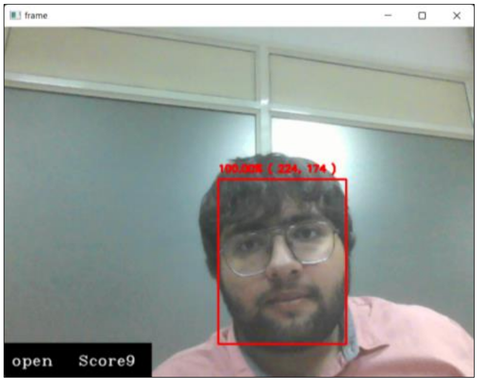

Figure 4. Implementation of Real-time Module (Open Eye)

V. CONCLUSION

We have trained our model on the InceptionV3 Model through transfer learning, to predict the status of the eye. This trained model, with an accuracy of 90% and loss of 23%, is then further used to predict the status of the eye for the real-time module. This real-time module will also have an alert system which would become active when the driver is found to be sleepy. There are various parameters specified in the real-time module which would decide whether the driver is sleepy or not.

VI. REFERENCES

[1] Fusek, R. (2018). *MRL Eye Dataset*. Retrieved from MRL: http://mrl.cs.vsb.cz/eyedataset

[2] Jain, M., Bhagerathi, B., & Sowmyarani, C. N. (2021). *Real-Time Driver Drowsiness Detection using Computer Vision.* Blue Eyes Intelligence Engineering and Sciences Publication (BEIES).

[3] Navya Kiran, V. B., Raksha, R., Rahman, A., Varsha, K. N., & Dr. Nagamani, N. P. (2020). *Driver Drowsiness Detection.* International Journal of Engineering Research and Technology (IJERT).