Driver Drowsiness Detection

Kiruthika V[1*], Surya Y[1], S R Shivaritha[1], Taran Sri Arangesh V[1], Senthil Vel K[1]

[1] School of Electronics Engineering, Vellore Institute of Technology, Chennai 600 127, India

Corresponding Author Email: kiruthika.v@vit.ac.in

Received: 02 August 2, 2022

Accepted:

Keywords:
Drowsiness, Transfer learning

ABSTRACT

A survey shows that Sleep-deprived drivers remain responsible for about 40% of road accidents. Many categories such as facial expression, eye movement, and yawning play an important role in analyzing the drowsiness of a driver, along with non-invasive methods that depend upon the skill of the driver. Hence, a machine learning model needs to be developed to understand various parameters and functions in order to alert and detect drowsiness.The objective of the project is to create a drowsiness detection system that can recognize when a person's eyes are closed for a brief period. The project uses a CNN model and transfers learning techniques using renet50 and VGG16 to predict whether a person feels drowsy or not based on whether the eyes are closed or open. When drowsiness is detected, this system will warn the driver. Hence, we propose a system to analyze different architectures to compare and contrast the accuracy and performance of different metrics for the same features. A multilevel classification model using CNN is implemented in this system.

1. INTRODUCTION

The objective of the project is to create a drowsiness detection system that can recognize when the person's eyes are closed for a brief period. The project uses 3 CNN models to predict whether a person feels drowsy or not based on whether their eyes are closed or open. When drowsiness is detected, this system will warn the driver. To develop CNN, Resnet50, and VGG16 models with the acquired data and observe the performance and accuracy of each model. To predict the drowsiness of the test data using the CNN models. Survey shows that Sleep-deprived drivers remain responsible for about 40% of all road accidents. With this project, we can potentially reduce the number of crashes related to drowsy driving. Traffic management can be maintained by reducing accidents. The features used in this project are: CNN (convolutional neural network), VGG16 (Visual Geometry Group), RESNET50 (Residual Neural Network)

2. LITERATURE REVIEW

Zuzana et al. 2020, in the paper titled “Driver Drowsiness Detection Using Convolutional Neural Networks,” use Convolutional Neural Networks and Deep Learning techniques to solve the problem of monitoring, detecting, and modeling critical situations and emotional stress of a smart car driver. VGG Face was chosen since it was trained on human faces, a dataset-like dataset for this project. For these problems, a model VGG Face is a CNN divided into 11 blocks, where 8 blocks are convolutional blocks, where each block contains one or more convolution layers followed by ReLu activation layer and/or max pooling layer. The developed model was trained with a custom dataset divided into batches of 32 during a 50 epoch-long process and with the nadam optimizer and categorical cross-entropy loss function. The VGG face layers were not retrained but froze and a head model was tuned only. The results of the model were 90.77% training data accuracy. The limitations were that when the model was tested on a completely new person, which was not included in the dataset, the neural network struggles to identify drowsiness in a person’s facial expression and suggests a person is aware most of the time, an efficient gesture that is recognized in other people is slight head falling and the whole training process on the hardware took about 15 hours of training.

Viswapriya et al. (2021), in the paper titled, "Machine learning approach for driver-drowsiness detection based on eye state," the authors used deep learning to detect the face and extract the eye region from the face images. Histogram equalization and canny-edge detection algorithms are used in this work. Along with this, an app was made to alert the driver with an alarm sound depending upon the driver's status. The facial landmarks are detected using image processing with canny edge detection. The eye aspect ratio (EAR) helps to classify the images as sleepy or nonsleepy after pre-processing of data. The pre-processing technique is that the image resizes and image enhancement is done with the help of the Histogram Equalization algorithm has been included here. Background noise can also be removed. The system was able to achieve about 90% accuracy. The major drawbacks were when the captured image was blurred due to movement, the pre-processing is not efficient and hence classification was not accurate. This system's lack of a transfer learning approach is the reason behind poor performance. Since image processing is the base of the project, the accuracy of prediction comes with the price of the quality of the image, which is not standard. The proposed system effectively identifies the state of the driver and alerts with an alarm and notification with alarm in the App, when the model predicts a drowsy output state continuously.

B. V. Bharath Chandra et al. 2021, in the paper titled “A Comparative Study of Drowsiness Detection from EEG Signals Using Pretrained CNN Models” uses electroencephalogram signals to detect drowsiness with sufficient reliability. The scalograms which describes the time-frequency characteristics of eye blinking segments were taken. Pretrained Convolutional

Neural Network based architectures viz. ResNet-152, ResNet101, VGG16, VGG19, AlexNet were used to distinguish three states of the driver namely "Sleepy or Drowsy", "Asleep" and "Awake". Of all these pretrained models analysed, AlexNet with Adam optimizer gave highest accuracy of 85.7% while the validation accuracy using AlexNet was 73% and 68% respectively for Adam and SGD optimizers. The drawbacks of this method are that the eye blinking frequency might drastically change for each of the person using it, the accuracy of this model is low in the other pretrained models such as ResNet-152, ResNet 101, VGG16, and VGG19. The features used in this model is very limited and many other features can be added for increase in accuracy.

In the paper by Alexis Arcaya Jordan et al, "Deep Learning for Eye Blink Detection Implemented at the Edge" a convolutional neural network (CNN)-based solution in smart connected glasses to detect eye blinks and use them to estimate the driver's drowsiness level This system uses a Eye blinking detection algorithm. The current threshold-based drowsiness detection algorithm is based on blinking signal time series coming from IR sensors. The contributions of this paper are first, an accuracy improvement for eye blink detection based on IR sensor signals and leveraging CNN. The drowsiness evaluation is performed through three successive phases: 1) eye blink detection, 2) metric computation, and 3) drowsiness index estimation. Data were collected from five subjects while driving to create training and test datasets. This system has recorded an accuracy of 90%. Raw data were collected from IR sensors at a sampling frequency of 100 Hz. Despite the lower inference rate, CNN models provide better accuracy than the threshold-based solution. The CNN model can avoid false positives by discriminating look down events through deactivation around the positive slope of the signal. Limitations of this system is that even though the sensitivity and the overall accuracy are better, it is observed that 1-convolution layer CNN models do not improve the specificity offered by the threshold-based algorithm. Research gap of this project is the memory footprint which needs to be quantized. The lack of data to increase the generalizability of the model. Data augmentation is required. There is a gap to enable eye blink detection based on IR sensor signals on a wearable embedded system. 1. Only yawning is used for classification in this paper. Eye detection and yawning detection can give more accuracy in the results than the method used in the model.

Mahek Jain, Bhavya Bhagerathi, Sowmyarani C N et al. Volume-11 Issue-1 (2021), in the paper titled, "Real-Time Driver Drowsiness Detection using Computer Vision" used Open CV for detection and classification of Drowsiness detection. They used a dataset of 300 indoor and as many outdoor images. The dataset covers a large variety of identities, face size, lighting conditions, pose, etc. For the purpose of yawn detection, a YAWN value is calculated using the distance between the lower lip and the upper lip, and the distance will be compared against a threshold value. EAR (Eye Aspect Ratio) computes the ratio of distances between the horizontal and vertical eye landmarks which is required for detection of drowsiness. The face is localized in the image using facial landmark detection. Then, shape prediction methods are used to detect important features on the face. Face detection is done by OpenCV built in HAAR cascades, which are pre-trained. Five test cases were conducted while doing this project for drowsiness and yawn detection of the driver. According to the methodology of the system, when the eyes are closed for more than the set threshold number of frames or when the driver is yawning, then the driver is feeling tired. The research has concluded the significance of Haars Cascades for driver drowsiness detection with a classification accuracy of 85%. Accuracy drops down a little when an obstacle is present (e.g., Hat). Ambient lighting conditions are very essential for getting proper results. In case, the user's eye closure and yawn occur simultaneously, a voice alert is raised but the system behaves in an erroneous and unsynchronized fashion. The limitation of the paper is that it is observed that the system's accuracy decreases in bad lighting conditions. Furthermore, the research has concluded the necessity of a technique which focused on the use of outer factors for measuring fatigue and drowsiness with precision.

S. Jansi Rani1, Anand Rajasekharan, Chandrasekhar A R et al. Volume 7 Issue 1 (2020), in the paper titled, "Drowsiness Detection System using Machine Learning" used machine learning techniques for detection and classification of Drowsiness of the driver. HOG- Linear SVM system aims at successfully identifying drivers who are in a state of drowsiness and immediately wake them up by sounding an alarm. This system requires a camera that is to be used to obtain a live running video stream of the driver behind the wheel. Histogram of Oriented Gradients (HOG) feature descriptor which uses the distribution of directions of gradients as features and tends to be more accurate and faster in operation than other algorithms. By covering the detection window with a dense (in fact, overlapping) grid of HOG descriptors and making use of the combined feature vector in a standard Linear SVM based window classifier gives us our human detection chain. The next step involved after successful face detection is recognizing the facial landmarks and extraction of the desired facial landmarks. A pre trained dataset (iBUG 300-w) is used for eye detection. To detect if the driver 's eye is closed or not, and to also successfully differentiate between standard eye blinks and eyes being closed during a state of drowsiness, a single, scalar quantity called eye aspect ratio (E.A.R) is computed hat reflects whether the eye is closed or not. The E.A.R value remains constant when the eye of the driver is open, but it starts to reduce to a value close to zero when the eye starts to close. The average duration of a person 's eye blink is 100-400 milliseconds, hence if the driver is in a state of drowsiness, their eye closure time is beyond this interval. In this system, the threshold is set 5 seconds, and if this is crossed the alarm is sounded and an alert regarding this will pop. The system has a face and eye blinking detection formula based upon the Histogram of Oriented Gradients (HOG) image descriptor and a Linear Support Vector Machine (SVM) facial detector. These are precise enough to reliably estimate the positive photos of the face and level of eye openness. Limitations of this project is that HOG with SVM gives the required results but HOG with other classifications give better results and accuracy. Thus, the eye aspect ratio (E.A.R) is calculated for every frame and as soon as the E.A.R value comes close to zero for more than the threshold time the alert is sounded along with the corresponding message.

3. EXISTING SYSTEM

There are different approaches to identify driver drowsiness. HM-LSTM Network, SVM, behavioral, DCCNN, Logistic regression, ECG signal etc. The HM-LSTM network is a temporal model to detect drowsiness. This uses the blink sequence and frequency of blinks to detect drowsiness. The major drawback of such specific systems is that these parameters vary from individual to individual nevertheless many external factors such as intensity of light, dust and foreign particles that change the blink velocity, frequency and rates. Such systems cannot be generalized.

Behavioral measures are non-invasive and more practical than physiological measures. In most models the facial feature: Eye closure analysis Eye blink rate, Yawning analyses, Facial expression analysis is performed. These features might seem to result in high performing models but it is not generalized.

Logistic regression method based on the machine learning approach implements a unique algorithm for each person to analyze their ECG. Since a driver does not always wear the same clothing material, the impedance matching between the body sensors and the clothing varies. This variation in material of clothes leads to more noise and a lower ECG amplitude.

The research gap found common in most of these systems is the lack of proper dataset to bridge the gap between different features used in different architectures. Standardized dataset to detect and extract reliable constraints. A model to understand the data for a wider range of human race.

4. PROPOSED SYSTEM

In the proposed system for drowsiness detection, the aim was to create a model that incorporates transfer learning techniques. A comparison study was conducted between 3 architectures namely resnet50, vgg16 and CNN. In this project the features used to detect drowsiness are yawning and the eye state. Both the features are monitored and checked before determining the state of the driver.

CNNs are used for image classification and recognition because of its high accuracy. In the CNN model we created a multilevel classification model to identify the drowsiness with the help of yawning and eye state. The layers in the CNN are created sequentially. The dimensions are rescaled in order to prevent overfitting of data which often results in false positives indicating a bad model. There are 4 classes to which data can be segmented into.

Auto Tuning is done to automatically Tune Convolutional Neural Networks for Improved Transfer Learning. Different pretrained networks are used to check the quality of the model. ResNet-50 has more parameters, there are 50 layers present in it. The pretrained model can classify the images into the 4 classes. VGG16 is an object detection and classification algorithm which can be used for transfer learning. Softmax is used as the activation function for multi-class classification in VGG16 as it needs to be classified into more than 3 classes.

The major research gaps for existing systems is the lack of a proper dataset that can work for all conditions such as: low intensity of light, unstable unfocused data, different eye size, colours etc. Due to the implementation of transfer learning concepts we were able to compare and contrast the results for different parameters of different architecture.

5. RESULTS AND DISCUSSION

This section shows the results from the proposed system, and the discussions surrounding this.

A. Dataset Description:
For this system, we make use of a 2467 files of training dataset consisting of no_yawn, yawn, open and closed eye 616 each. Similarly, 433 files for testing. But with regards to this project our region of interest (ROI) includes only the eyes and lips region which can be extracted using python slicing techniques.

B. Experimental Results:
This section shows the results of the proposed system using different models:

CNN MODEL:

```
## Lets train our CNN
retVal = MyCnn.fit(training_ds, validation_data= testing_ds, epochs = 10)

Epoch 1/10
20/20 [==============================] - 86s 4s/step - loss: 1.2834 - accuracy: 0.4406 - val_loss: 0.7154 - val_accuracy: 0.762
1
Epoch 2/10
20/20 [==============================] - 86s 4s/step - loss: 0.5552 - accuracy: 0.7365 - val_loss: 0.4746 - val_accuracy: 0.799
1
Epoch 3/10
20/20 [==============================] - 87s 4s/step - loss: 0.3679 - accuracy: 0.8204 - val_loss: 0.3476 - val_accuracy: 0.833
7
Epoch 4/10
20/20 [==============================] - 88s 4s/step - loss: 0.2737 - accuracy: 0.8833 - val_loss: 0.3462 - val_accuracy: 0.836
0
Epoch 5/10
20/20 [==============================] - 87s 4s/step - loss: 0.1960 - accuracy: 0.9193 - val_loss: 0.3487 - val_accuracy: 0.859
1
Epoch 6/10
20/20 [==============================] - 88s 4s/step - loss: 0.1724 - accuracy: 0.9266 - val_loss: 0.2452 - val_accuracy: 0.900
7
Epoch 7/10
20/20 [==============================] - 88s 4s/step - loss: 0.1327 - accuracy: 0.9505 - val_loss: 0.2316 - val_accuracy: 0.921
5
Epoch 8/10
20/20 [==============================] - 88s 4s/step - loss: 0.0747 - accuracy: 0.9720 - val_loss: 0.2365 - val_accuracy: 0.928
4
Epoch 9/10
20/20 [==============================] - 88s 4s/step - loss: 0.0701 - accuracy: 0.9765 - val_loss: 0.1724 - val_accuracy: 0.951
5
Epoch 10/10
20/20 [==============================] - 88s 4s/step - loss: 0.0486 - accuracy: 0.9830 - val_loss: 0.1650 - val_accuracy: 0.958
4
```

Figure 1 Epochs of CNN model

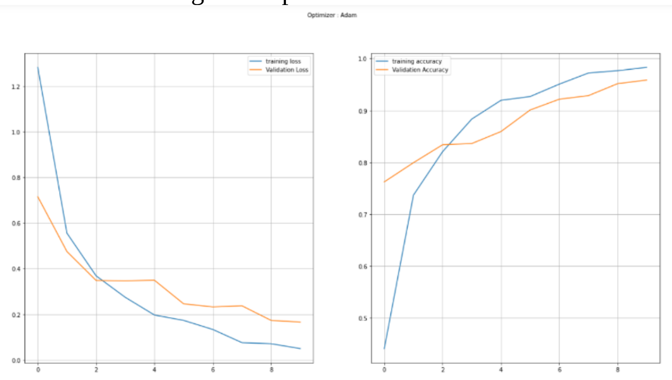

Figure 2 Training and Validation Graph

Using Transfer Learning:
VGG16:

```
result=model.fit(train_batches,
                 validation_data = valid_batches,
                 epochs= NUM_EPOCHS,
                )

Epoch 1/5
20/20 [==============================] - 487s 24s/step - loss: 3.2264 - acc: 0.8050 - val_loss: 0.4708 - val_acc: 0.9446
Epoch 2/5
20/20 [==============================] - 521s 26s/step - loss: 0.3828 - acc: 0.9526 - val_loss: 0.4024 - val_acc: 0.9630
Epoch 3/5
20/20 [==============================] - 533s 27s/step - loss: 0.1391 - acc: 0.9814 - val_loss: 0.3459 - val_acc: 0.9561
Epoch 4/5
20/20 [==============================] - 510s 25s/step - loss: 0.0160 - acc: 0.9951 - val_loss: 0.4189 - val_acc: 0.9630
Epoch 5/5
20/20 [==============================] - 506s 25s/step - loss: 0.0147 - acc: 0.9964 - val_loss: 0.2270 - val_acc: 0.9723
```

Figure 3 Epochs for VGG16

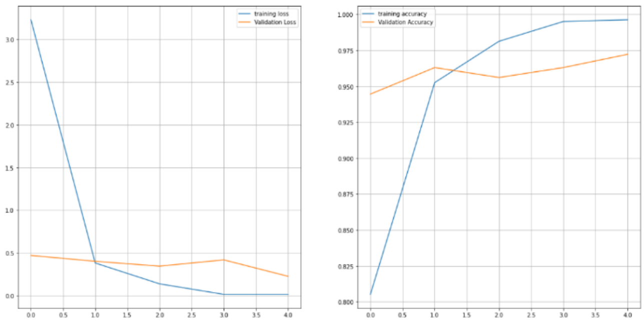

Figure 4 Training and Validation Accuracy

RESNET50:

```
res=model.fit(training_ds, validation_data= testing_ds, epochs = 10)
9792
Epoch 5/10
20/20 [==============================] - 183s 9s/step - loss: 0.0705 - accuracy: 0.9927 - val_loss: 0.5266 - val_accuracy: 0.
9838
Epoch 6/10
20/20 [==============================] - 179s 9s/step - loss: 0.0139 - accuracy: 0.9976 - val_loss: 0.3063 - val_accuracy: 0.
9861
Epoch 7/10
20/20 [==============================] - 176s 9s/step - loss: 0.0015 - accuracy: 0.9992 - val_loss: 0.2210 - val_accuracy: 0.
9815
Epoch 8/10
20/20 [==============================] - 181s 9s/step - loss: 0.0216 - accuracy: 0.9968 - val_loss: 0.2011 - val_accuracy: 0.
9861
Epoch 9/10
20/20 [==============================] - 181s 9s/step - loss: 0.0102 - accuracy: 0.9980 - val_loss: 0.1159 - val_accuracy: 0.
9861
Epoch 10/10
20/20 [==============================] - 188s 9s/step - loss: 0.0780 - accuracy: 0.9907 - val_loss: 0.7477 - val_accuracy: 0.
9654
```

Figure 5 Epochs for Resnet50

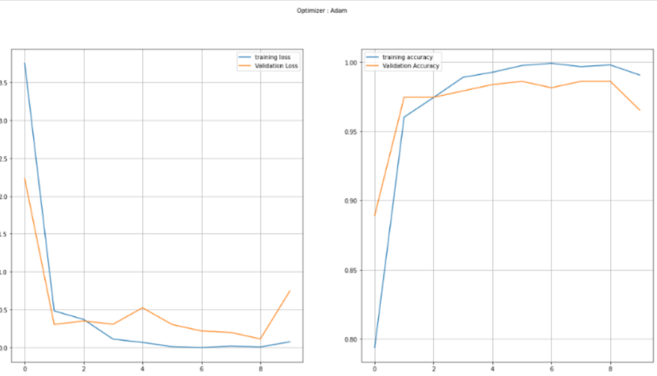

Figure 6 Training and Validation Accuracy

Performance Analysis:
To achieve the expected results, a large number of pictures were used and their accuracy in drowsiness and yawn detection was analyzed.

Model Used	**Training Accuracy**	**Testing Accuracy**
CNN	98.30%	95.82%
VGG16	99.64%	97.23%
Resnet50	98.80	98.61%

CNN was faster than the rest of the models but accuracy was low compared to other models. VGG16 took the longest time to run but had almost same accuracy as Resent50 but Resnet50 has better Training Accuracy and Testing Accuracy compared to VGG16 and CNN even though it was slower than CNN it was faster than VGG16.

System Testing:
We used 40 test images and sent it to our model and then predicted the output. We can see the results that we obtained below.

Using CNN

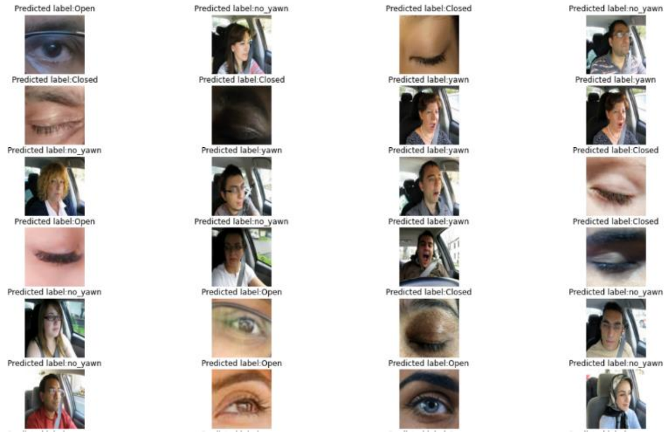

Figure 7 Prediction using CNN

Using VGG16

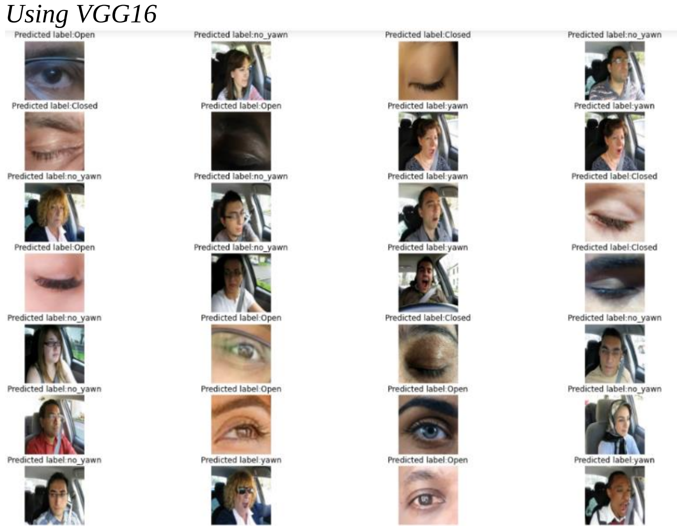

Figure 8 Prediction using VGG16

Using Resnet50

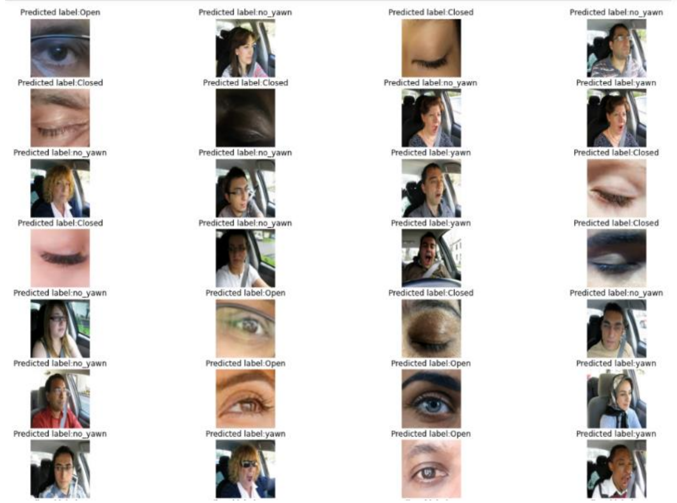

Figure 9 Prediction using Resnet50

Using OpenCv to find the ROI of the eye and then use the image for prediction.

```
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x2ae54876760>
```

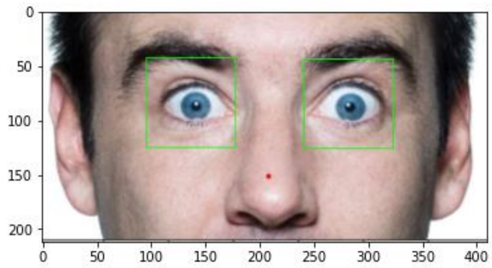

Figure 10 ROI detection using openCV

```
plt.imshow(cv2.cvtColor(eyes_roi, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x2ae548abbb0>
```

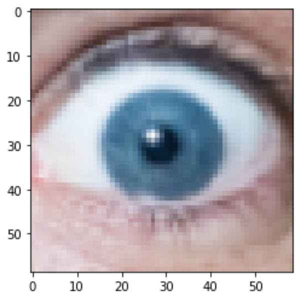

Figure 11 ROI detection using openCV

```
final_img = cv2.resize(eyes_roi, (256,256))
final_img = np.expand_dims(final_img, axis=0)
data = np.array(final_img)
result = model.predict(data)
```

```
for mem in result:
    print(class_names[np.argmax(mem)])
```

```
Open
```

Figure 12 Prediction from the detected ROI

We can see that our model was able to find the ROI of the eye and detect if it was open or closed.

Hence, we can see that in CNN model 2 of the predicted outputs are false, but we were able to change this by using transfer learning. VGG16 was able to increase the accuracy than CNN but still there was one false prediction hence using Resnet50 we were able to change that too. Resnet50 was able to predict almost all the test dataset as positive output. Hence, we can see that Resnet50 worked the best out of all the 3 models in a comparatively less time.

6. CONCLUSION AND FUTURE WORK

A comparison study between 3 models, 1 CNN and 2 transfer learning models (VGG16 and Resnet 50) was analyzed and it was seen that the Resnet50 model was best out of these and gave the best Training Accuracy and Testing Accuracy compared to VGG16 and CNN. Hence the features for eye and yawn were extracted and then trained to create a model using multi class classification, in our case 4 classes were used. The model is capable of detecting drowsiness by monitoring the eyes and mouth. The whole project is designed to decrease the rate of accidents and to contribute to the technology with the goal to prevent fatalities caused due to road accidents. The future work of this paper can be focused on the use of outer factors for measuring fatigue and drowsiness. The outer factors may be weather conditions, state of the vehicle, time of sleeping and mechanical data. Driver drowsiness is among the major threats to road safety, and in the case of commercial motor vehicle operators, the problem is particularly severe. The factors that contribute to this serious safety issue are twenty-four-hour services, unpredictable conditions of the environment, high annual mileage, and an increase of work schedules that are demanding. One important step of preventive measures that are needed to solve this problem is by continuously observing the driver's drowsiness state and giving information about their state to the driver so that they can take necessary action

7. REFERENCES

INTERNATIONAL JOURNALS AND CONFERENCES

1. Z. Képešiová, J. Cigánek and Š. Kozák, "Driver Drowsiness Detection Using Convolutional Neural Networks," 2020 Cybernetics & Informatics (K&I), 2020, pp. 1-6
2. S. E. Viswapriya , Singamsetti Balabalaji , Yedida Sireesha, 2021, A Machine-Learning Approach for “Driver-Drowsiness Detection based on Eye-State”, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 04 (April 2021),
3. B. V. Bharath Chandra, C. Naveen, M. M. Sampath Kumar, M. S. Sai Bhargav, S. S. Poorna and K. Anuraj, "A Comparative Study of Drowsiness Detection From EEG Signals Using Pretrained CNN Models", 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1-3
4. Alexis Arcaya Jordan, Alain Pegatoquet, Andrea Castagnetti, Julien Raybaut, Pierre Le Coz. “Deep Learning for Eye Blink Detection Implemented at the Edge”. IEEE Embedded Systems Letters, Institute of Electrical and Electronics Engineers, 2021, 13 (3), pp.130-133.
5. Mahek Jain, Bhavya Bhagerathi, Sowmyarani C N , “Real-Time Driver Drowsiness Detection using Computer Vision”, Volume-11 Issue-1 (2021)
6. S. Jansi Rani1, Anand Rajasekharan, Chandrasekhar A R,“Drowsiness Detection System using Machine Learning” Volume 7 Issue 1 (2020)