2023 4th IEEE Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 6-8, 2023

Enhanced Driver's Drowsiness Detection System using CNN model

Mandapati Ankitha
Department of CSE
VR Siddhartha Engineering College
Vijayawada, India
ankithamadapati2003@gmail.com

Sandeep Vemuri
Assistant Professor
Department of CSE
VR Siddhartha Engineering College
Vijayawada, India
sandeep.vemuri@gmail.com

Suraneni Lowkya Gayathri
Department of CSE
VR Siddhartha Engineering College
Vijayawada, India
lowkyagayathri@gmail.com

Jarabala Sindhu Bhargavi
Department of CSE
VR Siddhartha Engineering College
Vijayawada, India
jarabalasindhubhargavi@gmail.com

2023 4th IEEE Global Conference for Advancement in Technology (GCAT) | 979-8-3503-0525-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/GCAT59970.2023.10353308

***Abstract*--- Many accidents are frequently caused because of driver's drowsiness. While driving at night, people may fall asleep unknowingly due to fatigue, making it possible for the driver to lose control of the vehicle at times. The system then uses machine learning methods and methodologies to identify the indicators of the driver's tiredness. Even while some older algorithms and techniques can detect a person's drowsiness, our system uses Deep Learning algorithms and techniques to do so with more accuracy and across a wider range of environmental situations. There are many benefits by using Convolutional Neural Network (CNN) for sleepiness detection, including real-time processing, tolerance to changes in lighting, and the capacity to capture intricate spatial characteristics. When compared to earlier image processing methods, utilizing CNN as a detection algorithm for processing the video is the greatest approach in achieving reliable results. In this model, Haar cascade classifier is used for facial features extraction, Shi-Tomasi corner detection and Lucas-Kanade optical flow algorithm are used to detect the head movements of the driver. The model gave good results with 97% percent accuracy.**

Keywords— Deep Learning, CNN (Convolutional Neural Network), Machine Learning, Haar Cascade, Shi-Tomasi corner detection, Lucas-Kanade optical flow.

I. INTRODUCTION

Driving while fatigued is extremely risky and can result in accidents. According to a report from the Ministry of Road Transport and Highways Transport Research Wing [10], road accidents resulted in the deaths of 1,53,972 people in 2021. Driver sleepiness causes numerous accidents every year. Drowsiness detection for drivers is therefore necessary in order to help avoid these collisions. A drowsiness detection system can warn a driver to stop or take a break before they get too groggy to drive safely by monitoring their level of awareness. Overall, a driver's fatigue detection system can help to reduce the risk of accidents, save lives, and increase everyone's level of road safety.

According to the literature studied, drowsiness can be detected in many ways and it is mainly grouped into three categories: physiological based, vehicle based, and by using image processing. The physiological methods include Heart Rate monitoring, brain activity, pulse rate, and other physiological methods. Electroencephalogram (EEG) method uses brain activity through electrical signals, and Electroocoulogram (EOG) is used to record movements of the eye that measures corneo-retinal standing potential. Electromyography (EMG) is used to record the activities of muscles through connected electrodes. These methods require lots of equipment and are very uncomfortable to wear while driving. And second way is based on the driving pattern of a driver there is a difference between a normal driver driving pattern and a drowsy driver driving pattern, based on the sudden changes in the driving like acceleration, brakes are used to detect driver's drowsiness. The third way of using image-processing is efficient way and cost-effective. Different algorithms are used in image processing and most commonly Convolutional Neural Network (CNN) is used because of its working nature that is connected using many layers and nodes. An image or video of the driver's face and eyes is captured by a camera as input data for a conventional CNN-based driver drowsiness detection system. In this paper, we used CNN as our model. The information is then enhanced by preprocessing to reduce noise and highlight crucial elements like the driver's eye movements, facial landmarks, and other gestures. The CNN model, which consists of numerous layers of interconnected nodes that can learn and extract features from the input data given, is then fed the processed data. Once trained, the CNN model may be used to assess fresh video frames or photos in real time and forecast the level of tiredness of the driver. To prevent accidents, the system can play a warning sound, automatically slow down or stop the vehicle, or even alert the driver if it notices signs of intoxication. Here we proposed a model that is built upon CNN and by adding a head tilt or fall module by using corner detection of images and optical flow between the frames of video sequence in real-time.

II. RELATED WORK

In this section, we briefly discussed the overview of the previously employed approaches and their advantages.

Aishwarya Biju et al. [1] developed a method using two-stage CNN which can classify if the eyes are open or closed. The YOLOv3 real-time object detection approach, which uses a Darknet general network with 53 levels, is used to identify the face region. The output is downsized to 299X299 to fit into the Inception-v3 architecture, which comprises 11 stacked layers and includes generic CNN layers. By calculating the scalar product between the kernels and the original picture areas, the layer calculates the point mappings. In order to speed up computations, CNN uses pooling layers Max or Average) to reduce the size of the point mappings. Before conducting actions on each region,

this subdivides the input image into several regions. a specified upper limit that is applied during Max Pooling.

I. Jahan et al. [2] suggested a model called "4D: A Real-Time Driver Drowsiness Detector Using Deep Learning" as a cutting-edge method for applying deep learning techniques to identify driver drowsiness in real-time. To track and identify sleepiness, the suggested system, called "4D," combines a deep learning architecture with real-time visual processing. The authors use CNNs (convolutional neural networks) to analyze the driver's facial expressions, eye movements, and head attitude as seen on a camera mounted inside the car. This approach used pretrained networks such as VGG16 and VGG19 that reduces to computations.

R. Alharbey et al. [3] implemented a model combining two different techniques the first one using machine learning approach that takes EEG signals as a data input that are transforms into Discrete Wavelet Transform (DWT) to remove noise and PCA is used to extract important features are fed into different machine learning classifiers, KNN, SVM Gaussian NB to detect drowsiness. The second approach using CNN with three different architectures the first model based on CNN using 4 convolutional layers with the filters 16, 32, 64, 128 having input size 224 X 224, second model using ConvLSTM implemented as a 2D recurrent network, and third model using a hybrid model that combine both CNN and ConvLSTM modalities that will take video segments as input. Performance of different models can be observed and improved by comparing the parameters. It builds a robust model by combining different features, EEG signals and facial features.

Zuopeng Zhao et al. [4] suggested Fully automated driver fatigue status detection. The level of driver weariness is assessed via driving pictures. The suggested method extracts the region of interest (ROI) utilising feature points using the multitask cascaded convolutional network (MTCNN) architecture for face detection and feature point localization. Using EM-CNN, ROI images can also be used to identify the states of the lips and eyes. Two tests are used to identify fatigue: the amount of mouth opening (POM) and the rate of eyelid closure over the pupil over time (PERCLOS). The proposed EM-CNN can successfully detect driver drowsiness from real time capturing. The proposed algorithm EM-CNN performed with accuracy 93.623 and 93.643 percent respectively. It out-perfomed other CNN-based architectures.

Magan et al. [5] suggested a model to detect by using deep learning techniques, they employed a recurrent and convolutional neural network, while the second one extracts numerical features from images using deep learning techniques, which are given input to fuzzy-logic system that can reduce false positives. To identify whether or not the driver displays signs of tiredness, two distinct methods are devised, with a focus on minimizing false positives. Different approaches are used in this model.

J. R. Paulo et al. [6] The two methods for tiredness detection utilizing EEG signals in a sustained attention driving task are described in this study, with an emphasis on cross-subject zero calibration and consideration of pre-event time windows. Brain-computer signals frequently use EEG signals. They said that due to extremely low signal-to-noise ratios and cross-subject discrepancies that demand a lot of data, these systems are still challenging to develop. Here, they investigated drowsiness detection based on recurrence plots or angular fields as spatiotemporal visual representations for deep convolutional neural network (CNN) classification to address this study domain. But as we discussed earlier EEG requires lots of equipment to wear as it is not comfortable to the driver. Advancement on domain that explored using CNN.

V. Uma Maheswari et al. [7] proposed a model using different image processing techniques. The integrated method put forward in this study is based on the computation of the newly proposed vector FAR (Facial Aspect Ratio), which is similar to EAR and MAR and the PERCLOS (Eye and Mouth Closure Status). This assists in determining the condition of closed eyes or an opening mouth, such as when yawning, in addition to any frames with hand actions like nodding or covering an open mouth with the hand, which are instinctive human habits used to manage exhaustion. They suggested a new dataset called Eye and Mouth closed Dataset and used the NTHU and YawDD datasets.

In this the author Islam A. Fouad et al. [8] tried to build a robust model by capturing electroencephalography (EEG) signals and applying the signals to different machine learning algorithms. The process includes, the signals fed from the electrodes having twelve subjects are pre-processed to extract features and these are labelled as drowsy or not drowsy by applying machine learning algorithms. And the author has used Naive Bayes, Support Vector Machine (SVM), Random Forest Analysis (RFA), and K-Nearest Neighbor (KNN) that made KNN to get highest accuracy as 100 percent. The system takes less electrodes to note down the signals and proved those are enough to classify if driver is drowsy or not.

III. PROPOSED MODEL

The suggested system approach, project architecture, and dataset used in the project are all described in this section.

A. Architecture:

This section describes about the overall structure of the proposed system. The figure 1 depicts the architecture of the system that will describe about each module. Firstly, the processed dataset is collected and the facial region is detected from collected dataset using Haar Cascade algorithm after pre-processing the extracted features are split into training and testing data and data augmentation is performed which will create more amount of data. Training the CNN model with training data and evaluating model using test data that gives the final results. The model is saved and used in real time by capturing a sequence of video images predicting the label for every frame and detecting is there is any head fall or tilt. If the frames are detected as closed for consecutively 10 frames or head is tilted then alarm is played to alert the driver.

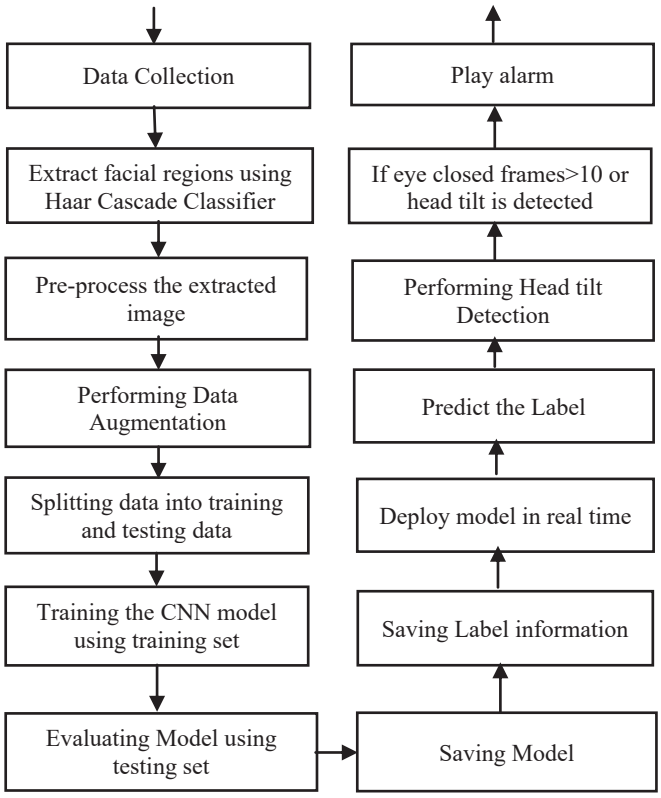

Fig. 1. Proposed system

B. Methodology:

The system methodology describes each individual modules precisely and explains the methods used in them.

1) Face Detection Module:

Since the background and other elements of the image are unnecessary, they have been deleted, and the model will only receive the facial region. We use computer vision technology, which can recognise humans in digital films and photographs to achieve this, Haar Cascade Face detection classifies facial features from the input photographs. And for extracting eye features we use Haar cascade eye classifiers for both left and right eye. Figure 2.a and 2.b depicts the facial regions extracted using Haar face classifier and Figure 2.c and 2.d represents the eye regions extracted using Haar left eye and right eye classifier.

Algorithm-1: Facial Features Extraction using Haar Cascade Classifier:

1. Define functions that will take the path of the input images directory and Haar cascade classifier path.
2. Define the categories such as "yawn" or "no_yawn", "closed" or "open"
3. Iterate over the images in the category directory and store them in an array.
4. Detect the facial region from the images.
5. From each detected face extract the ROI (Region of Interest) using the coordinates.
6. Resize the ROI to the desired size of 145X145.
7. Append the resized face array and its corresponding class index to the list created at start.
8. Return the resized array and labels as output from face detection function.
9. End

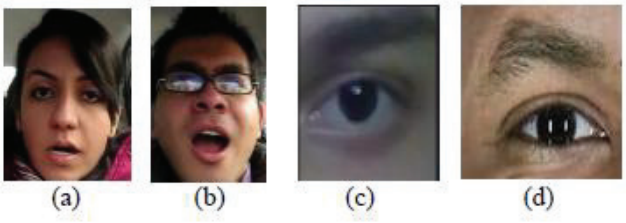

Fig. 2. Extracted Facial Regions and Eye Regions

2) Head tilt detection module:

Since it is difficult to detect eyes if the head falls because of drowsiness we have added a head fall detection. Using the detected face from Haar Classifier by applying optical flow algorithm the head movement is detected between the frames.

Algorithm-2: Head tilt or fall detection:

1. The settings for corner detection using the Shi-Tomasi [11] method are defined, including the maximum number of corners to be detected, the quality level, the bare minimum distance between corners, and the block size.

   $$R=\min(\lambda_1, \lambda_2)$$

   If this value is greater than a threshold value then it is detected as a corner. R is the distance between coordinates λ_1 and λ_2 which are corner points.
2. Parameters for the Lucas-Kanade optical flow algorithm are defined, including the window size, maximum pyramid levels, and termination criteria.
3. Optical flow is calculated between consecutive frames using the Lucas-Kanade method as shown in the following Figure 3. Calculates distance moved in x, and y directions by comparing the face center in current and previous frames.
4. Based on the calculated movement distances, the code determines if a gesture (falling) is detected using a threshold value.
5. Calls the alarm function if the distance is greater than the threshold value.
6. End

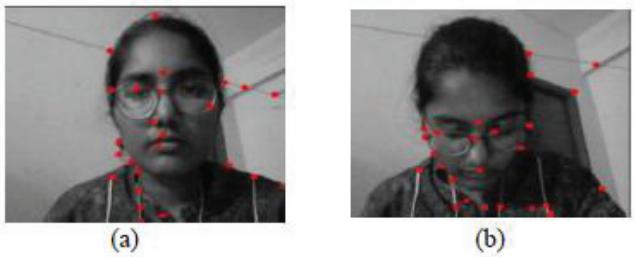

Fig. 3. Corner detection on sample images the red points showing the corners of the image.

3) Data pre-processing module

In order to make the data useful for training and evaluation, it must be cleaned, formatted, and organised. The

training set is subjected to data augmentation methods like rescaling, zooming, horizontal flipping, and rotation.

Algorithm-3: Data Pre-processing:

1. Append the yawning set and extracted eye regions set to create new set.
2. The data is converted into an array and reshaped into 145 pixels.
3. The labels are transformed into a binary representation using LabelBinarizer() so that it can fit into model.
4. The dataset is split into 70% training and 30% testing data.
5. Apply data augmentation on the training and tesing data using keras ImageDataGenerator. It rescales the pixel values from 0 to 1, applies zoom, flips horizontally and rotates upto 30 degrees.
6. Finally, the training dataset is converted into numpy array.
7. End

4) Classification using CNN model

CNN will be trained on the supplied dataset as a classification model. These models can capture complicated relationships between input qualities and drowsiness labels and may learn hierarchical representations from traits or raw data.

Algorithm-4: Classification using CNN model:

1. Define a sequential model using keras.
2. Add a convolutional layers on the input data with 256,128,64,32 filters and relu activation function with kernel size 3X3.
3. And add a max pooling layer following the convolutional layers with pool size 2X2.
4. Add a flatten layer that flattens multi-dimensional to one-dimensional vector.
5. Add a dropout layer that prevents over-fitting.
6. Add a dense layer with relu and softmax as activation functions that finalize the classification.
7. Compile the model categorical entropy as a loss function and the optimizer adam.
8. Train the model using fit that takes the train generator as an input by specifying number of epochs.
9. Save the model.

The Figure 4 represents the summary of the model containing total parameters, number of layers, shape of the output.

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 (MaxPooling2D)	(None,34,34,128)	0
conv2d_2 (Conv2D)	(None,32,32,64)	73792
max_pooling2d_2 (MaxPooling2D)	(None,16,16,64)	0
conv2d_3 (Conv2D)	(None,14,14,32)	18464
max_pooling2d_3 (MaxPooling2D)	(None,7,7,32)	0
flatten (Flatten)	(None,1568)	0
dropout (Dropout)	(None,1568)	0
dense (Dense)	(None,64)	100416
dense_1 (Dense)	(None,4)	260
Total params	-	495,140
Trainable params	-	495,140
Non-Trainable params	-	0

Fig. 4. Model summary

5) Model validation and testing

The model's performance, including its loss, accuracy, precision, and recall, will be verified. While accuracy evaluates the general accuracy of the model's predictions, precision and recall provide information on the model's ability to detect sleeping events accurately and avoid false positives or negatives.

Algorithm-5: Model validation and testing:

1. Take the image frame from captured video.
2. Detect the face and eyes region from the frame.
3. Pre-process the image.
4. Predict the class.
5. If predicted label for the frame is closed for consecutively >10 then play alarm or head tilt occurred play the alarm.
6. Otherwise continue to capture.
7. End

IV. PERFORMANCE MEASURES

After training the model we will get the accuracy how actually its predicted on the test data. Here Figure 3 represents the classification report that says about the parameter’s recall, precision, accuracy, f1-score and support.

	Precision	Recall	F1-Score	Support
Yawn	0.95	0.89	0.92	45
No yawn	0.79	0.98	0.87	45
Closed	0.99	0.94	0.96	216
Open	0.97	0.98	0.98	229
Accuracy	-	-	0.97	535
Macro Avg	0.92	0.95	0.93	535
Weighted Avg	0.96	0.96	0.96	535

Fig. 5. Classification Report

Figure 6 represents the Confusion matrix is a performance measurement tool. It is frequently used in statistics and machine learning to evaluate the precision and efficiency of a classification algorithm. The counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model on a specific dataset. Figure 6 represents Confusion Matrix.

Fig. 6. Confusion Matrix

After training is completed based on the accuracy of the trained model and the validation accuracy that can be taken from model with the epochs values we can drawn a graph as shown in Figure 7. This contains the both training accuracy that is blue line and validation accuracy that is red line on the graph. Figure 7 Represents Accuracy Graph.

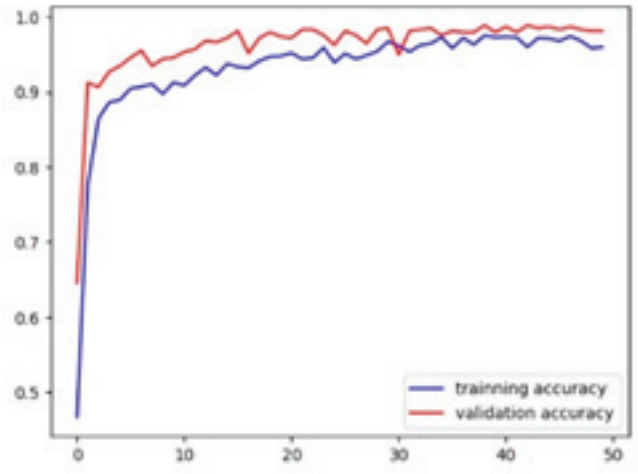

Fig. 7. Accuracy Graph

After training is completed based on the loss of the trained model and the validation loss that can be taken from loss function with the epocs values we can drawn a graph as shown in figure 8. This contains the both training loss that is blue indicate loss red line indicate validation loss. Figure 8 represents loss graph.

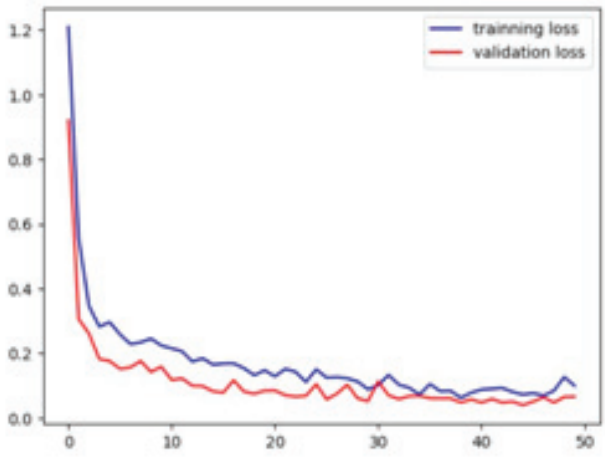

Fig. 8. Validation Loss Graph

V. EXPERIMENTAL RESULTS

The outputs for the model are shown below that are extracted from real time capturing video with the hardware requirements containing 8GB Ram capacity and 4GB SSD capacity with camera attached to it. The alarm sound is played whenever a warning appears on the screen and alert the driver this model can be integrated with any mobile application can be used easily using mobile camera or can be integrated with hardware features internally in the vehicle.

The Figure 9 contains the images of Open eyes taken as input to the model for detecting drowsiness. As the eyes are not closed so it shows the green mark and displayed a label as 'Eyes Open'. And yawning is detected and a warning is displayed.

Fig. 9. Detected Facial and Eye regions

And yawning is detected and a warning is displayed in Figure 10 shown below.

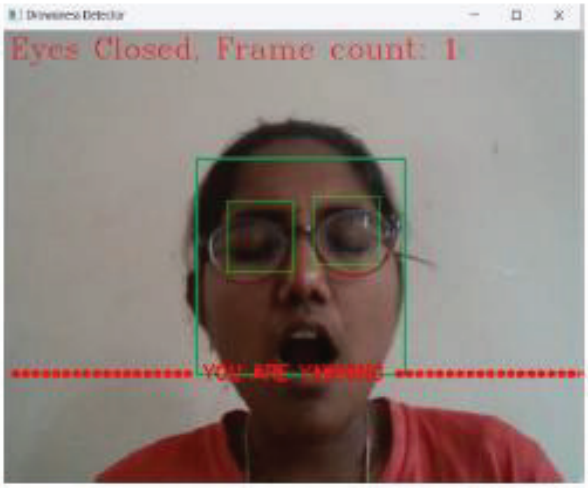

Fig. 10. A warning for detected yawning

Figure 11 contains the sleeping stage with completely closed eyes after a number of frame counts if count exceeds the threshold value an alert message is displayed and plays alarm.

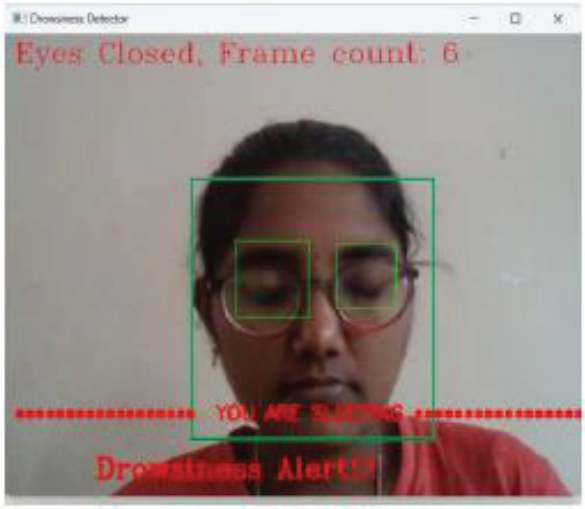

Fig. 11. Drowsiness Alert

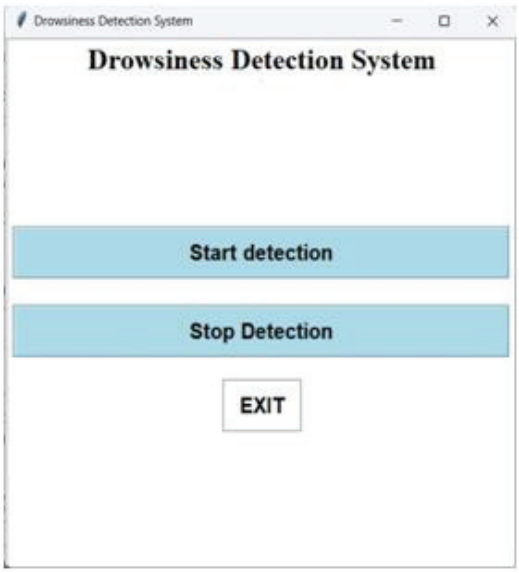

Fig. 12. Graphical User Interface

And Figure 12 represents the Graphical User Interface we have developed using Python Tkinter library that shows a start and stop buttons simply and an exit button to stop the execution of the program.

VI. CONCLUSION AND FUTURE WORK

Using the improved CNN algorithm, this system can identify signs of tiredness in the driver such as opened and closed eyes, mouth characteristics, and head movements. A crucial topic of research in the field of driver safety is the identification of driver fatigue. The deep learning method CNN (Convolutional Neural Network) has demonstrated good results in identifying driver sleepiness. Images and videos of drivers who exhibit various degrees of tiredness are used to train the model. The CNN model learns to discriminate between the two based on the patterns and attributes it extracts from the photos and videos that are classified as being either sleepy or alert. Overall, by lowering the likelihood of accidents brought on by fatigued driving, CNN models for driver drowsiness detection have the potential to dramatically increase driver safety. And in the future, the system can be extended using multiple models and adding hand gesture detection normally people used to stop yawning, objects detection like cool drinks, mobile and some are used to wear spectacles while driving these can be added to make the system reliable.

REFERENCES

[1] A. Biju and A. Edison, "Drowsy Driver Detection Using Two Stage Convolutional Neural Networks," 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Thiruvananthapuram, India, 2020, pp. 7-12, doi: 10.1109/RAICS51191.2020.9332476.

[2] I. Jahan et al, "4D: A Real-Time Driver Drowsiness Detector Using Deep Learning," Electronics, vol. 12, no.1, p. 235, Jan. 2023, doi 10.3390/electronics12010235.

[3] R. Alharbey, M. M. Dessouky, A. Sedik, A. I. Siam and M. A. Elaskily, "Fatigue State Detection for Tired Persons in Presence of Driving Periods," in IEEE Access, vol. 10, pp. 79403-79418, 2022, doi: 10.1109/ACCESS.2022.3185251.

[4] Zuopeng Zhao, Nana Zhou, Lan Zhang, Hualin Yan, Yi Xu, Zhongxin Zhang, "Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN", Computational Intelligence and Neuroscience, vol. 2020,Article ID 7251280, 11 pages, 2020.

[5] Magan, E.; Sesmero, M.P.; Alonso-Weber, J.M.; Sanchis, A. Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images. Appl. Sci. 2022, 12, 1145. https://doi.org/10.3390/app12031145.

[6] R. Paulo, G. Pires and U. J. Nunes, "Cross-Subject Zero Calibration Driver's Drowsiness Detection: Exploring Spatiotemporal Image Encoding of EEG Signals for Convolutional Neural Network Classification," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 29, pp. 905-915, 2021, doi: 10.1109/TNSRE.2021.3079505.

[7] V. Uma Maheswari, Rajanikanth Aluvala, M.V.V Prasad Kantipudi, Krishna Keerthi Chennam, Ketan Kotecha, and Jatinderkumar R. Saini, "Driver Drowsiness Prediction Based on Multiple Aspects Using Image Processing Techniques," IEEE Access, vol. 8, pp. 151819-151830, 2022.

[8] Islam A. Fouad, A robust and efficient EEG-based drowsiness detection system using different machine learning algorithms, Ain Shams Engineering Journal, Volume 14, Issue 3, 2023, 101895, ISSN 2090-4479, https://doi.org/10.1016/j.asej.2022.101895.

[9] Perumandla, D. (2020, September 27). Drowsiness_dataset. Kaggle. Available: https://www.kaggle.

[10] Reasons People fall asleep while driving and what can be done about it. Hindustan Times.(2023,April7). Available: https:// www.hindustantimes.com/lifestyle/health/drowsy-driving-reason s-people-fall-asleep-while-driving-and-what-can-be-done-about it-101.html/.

[11] Shi-Tomasi Corner Detector & Good Features to Track. (April 24,2023). Available: https://opencv24-python tutorials.eadthe docs.io/en/latest/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html.

[12] W. Deng and R. Wu, "Real-Time Driver-Drowsiness Detection System Using Facial Features," in IEEE Access, vol. 7, pp. 118727-118738, 2019, doi: 10.1 109/ACCESS.2019.2936663.

[13] A. Altameem, A. Kumar, R. C. Poonia, S. Kumar and A. K. J. Saudagar, "Early Identification and Detection of Driver Drowsiness by Hybrid Machine Learning," in *IEEE Access*, vol. 9, pp. 162805-162819, 2021.

[14] Ratnesh Kumar Shukla, Arvind Kumar Tiwari, Ashish Kumar Jha, "An Efficient Approach of Face Detection and Prediction of Drowsiness Using SVM", Mathematical Problems in Engineering, vol. 2023, Article ID 2168361, 12 pages, 2023. https://doi.org/10.1155/2023/2168361.

[15] Li, Yongkai, Shuai Zhang, Gancheng Zhu, Zehao Huang, Rong Wang, Xiaoting Duan, and Zhiguo Wang. 2023. "A CNN-Based Wearable System for Driver Drowsiness Detection" Sensors 23, no. 7: 3475. https://doi.org/10.3390/s23073475.