A Deep Learning Approach To Detect Driver Drowsiness

Madhav Tibrewal
4th year, B.Tech Computer Science & Engineering student
SRM Institute of Science & Technology, Kattankulathur
Chennai, India-603203

Aayush Srivastava
4th year, B.Tech Computer Science & Engineering student
SRM Institute of Science & Technology, Kattankulathur
Chennai, India-603203

Dr. R. Kayalvizhi
Assistant Professor, Dept. of Computer Science & Engineering
SRM Institute of Science & Technology, Kattankulathur
Chennai, India-603203

Abstract—**In the current times, we can see drastic changes in how humans manage their time. The natural sleep cycle of human beings is therefore been disturbed. Due to a lack of sleep and irregular sleep cycles, humans tend to feel drowsy at any time of the day. With these poor work-life timings, people can find it difficult to carry out the activities like driving which requires a healthy and properly functioning state of mind and body. Drowsiness is one of the major causes of road accidents in today's time. According to the Central Road Research Institute (CRRI), tired drivers who drowse off while driving are responsible for about 40% of road accidents. Several misfortunes can be avoided if the driver is alerted in time. This paper explains the working on making a complete drowsiness detection system which works by analyzing driver's state of eyes to further deduce the drowsiness state of the driver and alert the driver before any serious threat to road safety.**

Keywords—Drowsiness Detection, Fatigue Detection, Classification, Driver Monitoring System, Road Safety, Perclos

I. INTRODUCTION

One of the main reasons for untimely deaths today is road accidents. Most of the time the drivers would lose their alertness and meet with unfortunate accidents. This loss of the state of alertness is due to fatigue and drowsiness of the driver. This situation becomes very dangerous when the driver is alone. The ultimate reason for the loss of the state of alertness is accidental micro-sleeps (i.e. temporary lapse of consciousness which occurs when a driver is drowsy and is fatigued). Drowsiness or fatigue is one of the main reasons of low road safety and some severe injuries, economy loss, and even deaths. Collectively, these situations increase the risk of road accidents. Using computer for automatic fatigue detection, several misfortunes can be avoided. The drowsiness detection systems continuously analyze the driver's condition and warns before any unfortunate situation arises.

Due to the accidents being caused due to the fatigue state of the drivers, several methods have been developed to detect the driver's drowsiness state and warn accordingly. Each method has its advantages as well as disadvantages. There have been some great works in this field but we can have some space for future improvements.

This paper's aim is to identify driver drowsiness while addressing the issue of late warning due to analysis in discrete time periods by existing solutions. This paper shows the design and implementation of a Deep Learning based model for eye state (open/close) classification, and the integration of the above ideas into a driver drowsiness detection system.

II. STATE OF THE ART (LITERATURE SURVEY)

Dasgupta et al. improve the probability of predicting drowsiness state correctly by composing multiple separate drowsiness tests together [1]. Three separate stages of sleep verification are performed. These stages are based on visual, sound and touch inputs. A linear SVM is used in the first stage for visual classification of eyes as open or close.

The composition method actually helps and this model achieves quite a good accuracy of 93.33%. But this comes at the cost of the time taken to finish the tests. This paper estimate that at the worst case it will take 20 seconds from onset of sleep for the application to finish all its tests and sound the alarm. It also requires manual interaction from the driver in the final stage of verification which could be distracting during the non-drowsy state. 92.86% accuracy is still possible by having only the first two stages of verification.

A good chunk of 10 seconds can be shaved off by only keeping the first two tests with only 1% hit to accuracy. Additional few seconds can be saved by some smart engineering at the first stage. Waiting for many seconds for the model to convince itself that the driver is indeed asleep can be dangerous and potentially life-threatening when it could have alerted the driver earlier. Ultimately the same task can be achieved within seconds of the onset of sleep as can also be corroborated from other works.

Another important consideration is support for low-performance devices. Jabbar et al. focus on making a minimal network structure for practical use in low-performance devices [2]. Instead of using some state-of-the-art neural network techniques like CNN, it uses a Multilayer Perceptron Classifier. This paper first used Dlib to recognize facial landmarks for input to the model. This paper use the National Tsing Hua University (NTHU) Driver Drowsiness Detection Dataset. A total of 22 subjects of various ethnicities are used. The subjects are divided into two portions. 18 are used in training whereas 4 are used for evaluation. A model based on facial landmarks is thus trained.

The accuracy achieved by this model is 81%. But the main achievement is the resulting model size which is just 100 KB, allowing for use in older devices. Newer dataset and frameworks can allow us to train more accurate and performant classifier networks. Still, some improvement can be made using more sophisticated networks like CNN- As this paper is two years old, some lighter-weight frameworks have been developed since which can allow us to consider better algorithms.

You et al. have quite a fresh take. Majority of projects on the subject focus on making a one-size-fits-all solution. Which has an advantage of its own that it is pretty much plugged and play for the end-user. This paper, however, takes a different approach of tailoring some key portions to end-users' [3]. The tradeoff is that the algorithm has to perform offline learning on the driver before it can be used online. This results in visible improvement like more accuracy compared to other papers on the subject.

Dlib's DCCNN is used for face detection. Dlib's "68 point facial landmarks" is used for calculating the eye aspect ratio. In the end, a linear SVM is trained on the particular driver's eye aspect ratios. The SVM trained for the user's eye aspect ratio on open and closed eye examples. This is necessary because the eye aspect ratio, in particular, is a feature that is highly dependent on individuals. A generalized model for this problem would actually have lesser accuracy. So training on a person's features who is going to use the product is a good idea. However, It is believed the training of sorts required here needs assistance and is difficult to perform alone by the driver himself.

Although the performance is pretty decent and the accuracy is quite good at 94%, the need to train personalized SVM for every driver before use could be a deal-breaker. Other solutions are easier to get started with because they can be used without requiring training from the end-user.

Deployment on heavy vehicles also presents some unique sets of challenges. Mandal et al. aim to build a framework which can utilize existing "dome cameras" on buses which are used to record driver's action for security purposes [5]. This paper proposes to utilise the chain of detectors which narrow the search area till eyes are in focus. This paper usez the method called Percentage of Eyelid Closure (PERCLOS) to estimate and classify fatigue level.

This paper specifically studies detection in special cases where the camera is placed obliquely (top right or top left) to the driver's seat in a heavy-weight vehicle like bus or truck. Because of the placement of the camera on heavy vehicles, this paper has to tackle background changes so that the vision-based eye detection is more accurate. This paper uses a pedestrian image database for head-shoulder detection instead of a simple face detector which has a higher diversity of backgrounds to have a more accurate way to detect the head-shoulder region of the driver in varied conditions and on obtaining a rough area, apply the usual face detection techniques to proceed. Hence using this approach the algorithm is robust to varied backgrounds resulting from placing the camera at a distance.

The algorithm is robust to varied backgrounds resulting from placement of the camera at a distance. The good thing is that existing ``dome camera" in commercial vehicles can be utilized thereby not needing any physical modification on vehicles as they are currently running. Whereas this also limits our ability to use this in other situations where the camera is not placed obliquely. This loss of generalization is the price to be paid for the benefit of not needing any hardware changes, which is frankly not too bad.

Deng et al. use the angle of the nearer eye and mouth to determine drowsiness. This paper also use a pre-processing step to automatically identify low lighting conditions and enhance the image if so [4].

Deng et al. also do their processing of video data on a cloud server. Cloud processing for such a latency-sensitive task will pose more difficult problems. High network latency can severely affect the video frames that can be processed per second and could also introduce synchronization problems. Also, the solution is bound to fail in real-world usage where driving in remote places with poor or no internet connectivity could mean the app doesn't work at all in the best case or the driver has a false sense of security in the worst case.

Lets have a look at some vehicular feature-based models now. Interestingly in addition to academic studies [7], [8], there are products from well-known car makers [12].

Li et al. use data from sensors attached to the vehicle to detect the drowsiness in driver [7]. This paper attempts to make a vehicle-based system using a steering wheel angle (SWA) for drowsiness detection. During experiments, data is collected from steering wheel angles and a video of the driver's face is also recorded. After recording the SWA signals, it calculates the approximate entropy (ApEn) features to create the dataset for training a model. The recorded video is then used to label the segments of data by timestamp as "sleepy" or "awake". Some advantages of this approach are that no camera is required to keep focusing the driver's face and it can easily be incorporated into cars without changing the current looks of the car.

This model has achieved a good accuracy of 78.01% overall. This type of system is pretty efficient but it ignores the importance of human facial expression even though the same is used to actually label their data through the study of driver's face by medical experts.

Li et al. use data from sensors attached to the vehicle to detect the drowsiness in driver [8]. The main method used here is steering wheel angles (SWA) and yaw angle (YA) information to detect a driver's drowsiness state. Their approach is to analyse the features gathered from SWA and YA under different fatigue conditions, and then calculate the approximate entropy (ApEn) features of a small sliding window on time series data. The algorithm calculates the ApEn features of fixed window SWA and YA time series by using the nonlinear feature theory of dynamic time series. The paper uses the fatigue features as input to a (2-6-6-3) multi-level backpropagation Neural Networks Classifier to predict the driver's drowsiness state. Approximately a 15-h long experiment on a real road has been conducted, and segmented and labelled the data received with three levels, namely awake, drowsy and very drowsy, with the help of experts.

To determine the fatigue level of the data, this paper uses facial video expert evaluation method which is a standard and most applicable method up to now. This method takes a group of well-trained experts to score the fatigue status of drivers based on the features like facial expressions, head positions and some other important features. This method is pretty robust because of its evaluation criteria, which has acquired good and consistent results when examined by experts and some other works as well, which are facial-based instead of SWA-based. This method of fatigue detection at different levels: awake, drowsy and very drowsy, has achieved an accuracy of 88.02% during 15-h long training on a working road. It has low false alarm rates as well for drowsy (7.50%) and very drowsy (7.91%). This accuracy can be improved considering more features like the vehicle's lateral position.

Volvo uses a system called "Driver Alert Control" in their cars [12]. This system uses a camera to detects the highways side markings and compares the section of roads with the driver's steering wheel movements. If the vehicle does not follow the road evenly then the car alerts the driver.

The system is not intended to determine driver fatigue though. Volvo have themselves stated that they do not take driver fatigue directly into consideration and instead try to indirectly infer it from steering wheel movements. It is stated that in some cases driving ability is not affected despite driver fatigue. So there may be false negatives, i.e. no alarm is issued to the driver, and the driver himself is responsible to ensure that they are properly rested.

Bartolacci et al. study the more physiological aspects such as sleep quality, effects of drowsiness in normal working, tiredness coping mechanisms, sleepiness at different times [6]. This paper compares sleeping habits and coping mechanisms between younger and older people. They can show that age is the only factor which can be used to accurately determine cognitive driving-related abilities. The conclusion can be made that young people would benefit from a companion product that can detect and alert them of drowsiness symptoms while driving. The lifestyle of subjects should also be taken into account. It would be better to conduct the tests while enforcing a strict and uniform sleep-wake schedule for the subjects during the testing period.

Gwak et al. investigate the feasibility of classification of the alert states of drivers, mainly the slightly drowsy state, using a hybrid approach which is a combination of vehicle-based, behavioural, and physiological signals to implement a drowsiness detection system [9]. Firstly, this paper measures the drowsiness level, driving performance (i.e. vehicular signals), physiological signals (i.e. EEG and ECG signals), and behavioural features of a driver by using a driver monitoring system, driving simulator, and physiological measurement system. Next, this uses machine learning (ML) algorithms to detect the driver’s alert and drowsy states, and constructs a dataset from the extracted features over a period of 10 s. It has used ML algorithms like Decision Tree (DT), the majority voting classifier (MVC), and random forest (RF). Finally, this paper used the ensemble algorithms for classification.

This paper analyses vehicular parameters like longitudinal acceleration, vehicle velocity, lateral position, SWA, the standard deviation of lateral position (SDLP), time headway (THW), and time to lane crossing (TLC). SWA is used as steering smoothness evaluation index. SDLP is used as steering control evaluation index. THW is the difference between the time taken by the preceding vehicle and the test vehicle to reach the same point on the road. TLC is amount of time required to reach the edge of the lane, which has an assumption that the velocity and steering angle of the vehicle are constant at a certain point while driving. Behavioural features like eye blink using an eye mark camera, percentage closure of eyes (PERCLOS), seat’s pressure distribution using a pressure sensor. Physiological features like EEG signal using EEG cap and ECG signal using ECG body device were used in this paper to analyze the nervous system activities. The raw signals were filtered and transformed for investigation.

This paper results has reached 82.4% accuracy using hybrid methods to alert in slightly drowsy states, and 95.4% accuracy to alert in moderately drowsy states. Using random forest algorithm it achieved 78.7% accuracy to alert slightly drowsy states when exculding physiological indicators and obtained 89.8% accuracy to alert the moderately drowsy states. This paper represents that hybrid sensing methods using no-contact sensors are quite feasible for implementing driver drowsiness detection systems.

Guede-Fernandez et al. propose a drowsiness detection method based on respiratory signal changes [11]. It has used an inductive plethysmography belt to obtain respiratory signals and has real-time signal processing to classify the driver as awake or drowsy. The algorithm proposed in this paper analyses the respiratory rate variability (RRV) in order to predict the driver’s resistance against falling asleep. The algorithm is designed to detect early fatigue symptoms and warn the driver accordingly.

This paper have sampled data from 20 adult volunteers, aged between 20 and 60 years. This paper has also taken the weights of the subjects into account and sampled the data likewise. Also checked for any medical issues associated with the subjects like if the subjects have sleep disorders, consumed alcohol in the past 6 hrs. It uses a driving simulator to carry out the experiment as a real road experiment could be dangerous since the subjects were in a sleepy state. It uses three Respiratory inductive plethysmography (RIP) band sensors placed different positions namely thoracic, diaphragm and abdominal to ensure best possible signals. The experiment was carried out in two conditions: subjects are not slept for past 24-h and subjects with at least 6-h of sleep last night. Trained experts were observing the subject’s activity to classify them as drowsy or awake. At the same time, the experts also classified the breathing process as good or bad from the respiratory signals each minute.

The proposed algorithm uses Thoracic Effort Derived Drowsiness index (TEDD) for drowsiness detection. This algorithm is capable to analyse diaphragmatic and abdominal effort signals also, and only thoracic effort signals. The algorithm mainly analyses the variability in the respiratory rate along the time and also the presence of important artefacts in the respiratory signal. After generating the dataset, a quality signal classification algorithm is used which classifies the signal as good and bad. To improve the accuracy of the system the algorithm is optimized by tuning certain parameters like WLD and ThTedd. The algorithm overall achieved a specificity of 96.6%, a sensitivity of 90.3% and Cohen’s Kappa agreement score of 0.75.

Warwick et al. has used wireless devices which can be wore on different body parts to detect drowsiness[10]. This paper attempts to design a drowsiness detection system by using a biosensor called BioHarness 3 manufactured by Zephyr Technology, which can be worn on the body itself to measure driver’s physiological signals. The system is designed in two phases: In the first phase, the driver’s physiological data is collected using the biosensor and then it is analyzed to find the most important parameters from it. In the second phase, an algorithm to detect drowsiness is designed and a mobile app is made to warn the driver using and alarm. The biosensor can ne wore to the chest by wearing a shirt, using a holder or a strap. The sensor transmits the data to the smartphone and the algorithm analyses the data and predicts the driver as drowsy or normal.

Finally, the detection system is tested extensively by metrics like a false positive and false negative on different groups of people with different backgrounds, ages, sec, etc. In the future phases, this paper plans to carry out more real environment experiments to collect more data for analysis and carry out better prediction by designing a better and more robust algorithm.

III. Proposed Work

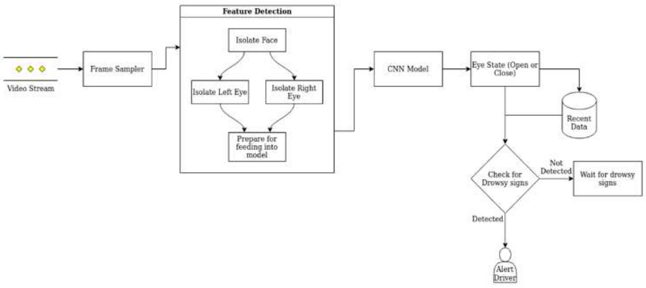


Figure 1. System Workflow

A. Data Acquisition

This paper is based on the MRL Eye Dataset [13]. This is a large scale dataset of 84k human eye images captured in various driving conditions using three different sensors. *Figure 2* shows different sensors used to capture images.

B. Data Pre-Processing

- Firstly, segregating the open and close images into folder

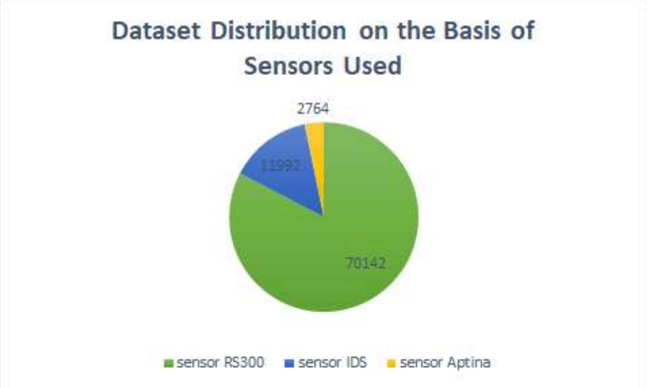


Figure 2. Dataset distribution based on sensors used

  - Eye/
    - Open/
    - Close/
- Next, importing images and resized them to 32*32 pixels

C. Model Designing and Training

Firstly, designing of a CNN model with four convolution layers, three pooling and two dense layers is done and then training the above model with different configurations to achieve best results. Several efforts have been made to make our model as small as possible. More on this in Implementation.

D. Drowsiness Detection

Firstly, the video stream is inputted from a camera using OpenCV library. On every sampled video frame, facial landmark detection using dlib's api is being performed. Next, isolating the left and right eye from the isolated face and using it to feed the trained model.

For each eye image in frame, the predicted eye state in a queue is stoed, and then used to analyze using Percentage of Eyelid Closed (PERCLOS) value, and then predict if the driver is drowsy or not.

IV. Implementation

The dataset consists of 84,898 eye images of 37 subjects (33 men and 4 women). Images were taken in different lightning and reflective conditions. *Figure 3* shows some of the conditions in which images were taken.

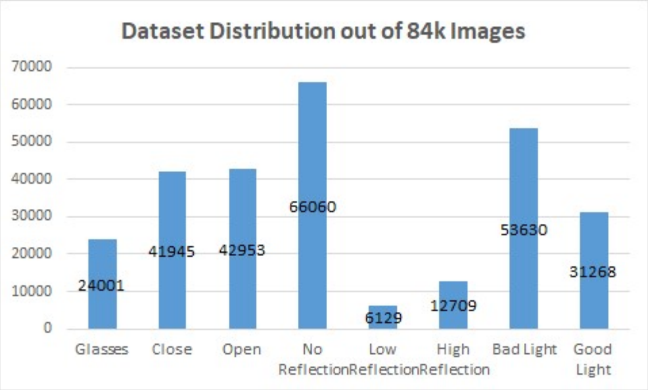


Figure 3. Dataset distribution based on environment conditions

After segregating the images into open and close folder structure, the images were imported in different random test and validation splits but 70% and 30% split worked best. Since the infrared images were of different dimensions, the images are resigned to 32*32 pixels.

B. Model Designing and Training

At first, it started off by using six convolution layers with filters numbers as 32, 32, 64, 64, 128, 128 of size 3*3, four max pooling layers of size 2*2, and two dense layers of sizes 256 and 128 respectively. This architecture was very complex in terms of images provided and hence it was underfitting. So some regularization techniques like L1-L2 regularizations, dropouts and batch normalization were used. Although these techniques improved the model but it was still not significant enough and our model was very large. Then, less complex model were obtained by dropping off some convolution layers and after trying several architectures and training the model on those the best working architecture for the model is selected, which is shown in *Figure 4*.

It consists of four convolution layers with filter numbers as 10, 20, 50 and 2 respectively of sizes 3*3 for first three and 1*1 for last, three max pooling of size 2*2 and two dense layers of size 4 and 1 respectively.

This new model is 22 times smaller in size (only 191kB) and it has only 11,293 total trainable parameters when compared to the older models with six convolution layers which had 566,817 total trainable parameters.

The model could not pass the 47%-50% accuracy mark initially, which was due to our model keep overshooting (i.e. not reaching global minima) due to higher learning rate. Reducing the learning rate to 0.001 after trying out 0.01 and

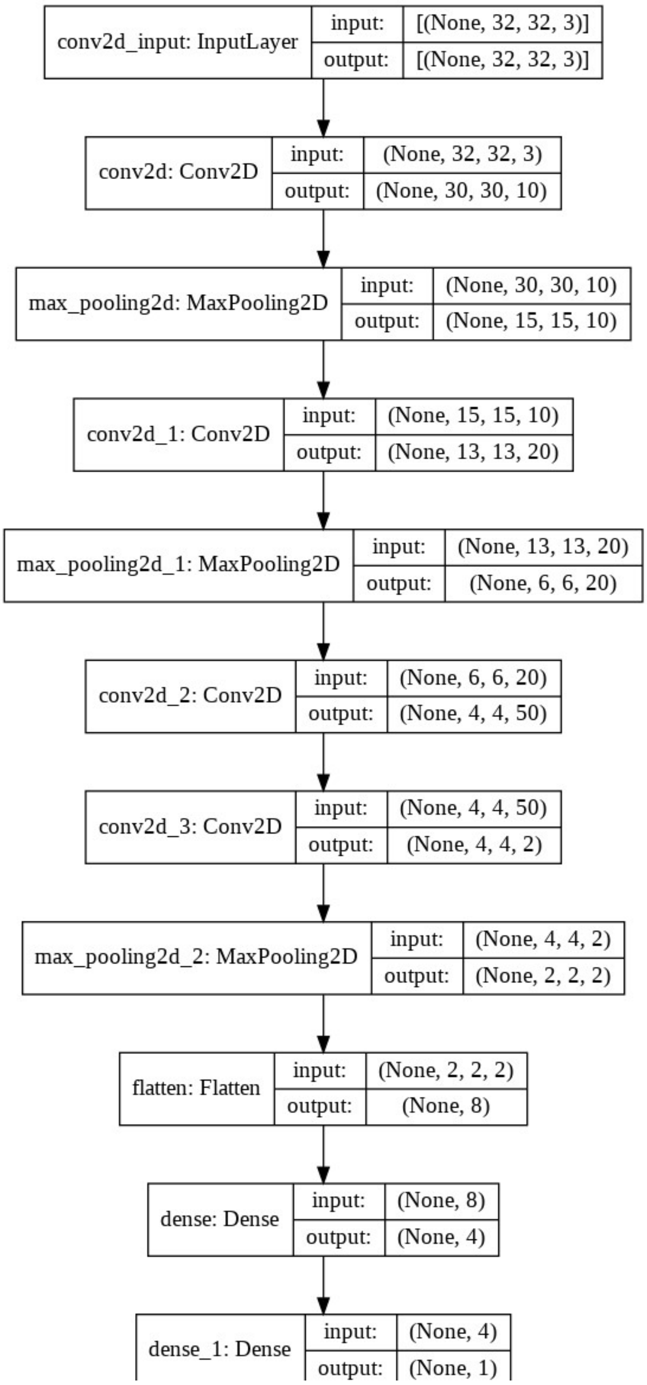


Figure 4. *Final model's architecture*

0.001 helped to reach an accuracy of 95%. Adam is used as the optimizer and mean squared logarithmic error as the loss function.

Even after getting decent results our models validation loss and accuracy was fluctuating a lot, which was due to high batch size. On trying out different batch sizes of 25, 20, 10, and 5, it is found out that validation accuracy and loss fluctuations is decreasing with smaller batch sizes. Hence, batch size of 5 was selected finally.

Epochs higher than 10 started to introduce fluctuations again. So epoch is selected as 10. *Figure 5* shows model's accuracy and loss graph.

We also compared our model with the onnx model we found on OpenVino website under public trained models section [14] and found out our worked significantly better although the onnx one had same accuracy as ours. We suppose it can be the result of higher noise in the onnx model due to several factor like batch size, learning rate etc. Images predicted as Open and Close is shown in *Figure 6*.

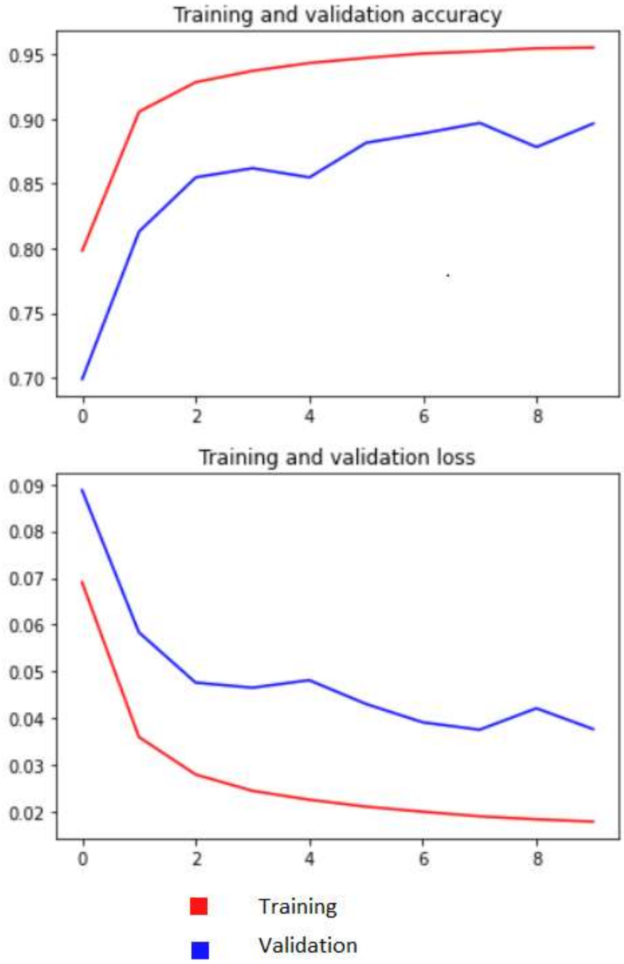
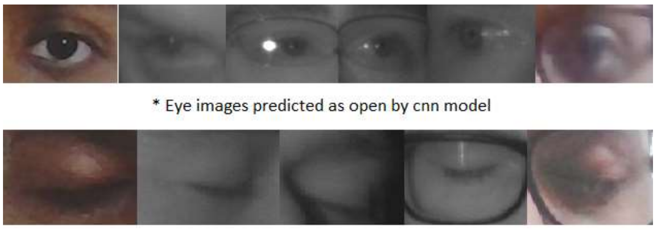


Figure 5. *Training and Validation accuracy and loss*


* Eye images predicted as open by cnn model

* Eye images predicted as close by cnn model

Figure 6. *Open and Close eye state prediction*

C. Drowsiness Detection

This approach uses OpenCV library to input video stream of size 640*360 pixels from a camera. It is then sampling each frame into greyscale from the live camera feed and then feeding it to the feature extractor which is using dlib's api to detect face and predict facial landmarks in each frame.

After getting all the facial landmarks from the feature extractor it is feeding it to the eye extractor which is cropping out a rectangular eye section from the live input frame. Further, it is feeding the extracted eye image to the eye classifier after pre-processing it to the required dimensions (i.e. 32*32*3).

The algorithm was performing analysis on a single threaded python program initially, which resulted in jitter and frame losses during analysis. During single thread running, our code was taking 200ms-220ms approximately to analyse one frame. Hence, multi-threading which improved the model's performance to 170ms-195ms to analyse one frame. This low increase in performance was due to python's GIL (Global Interpreter lock) which synchronizes the execution of thread so that only one thread can execute at a time, and since the tasks performed on each thread were computationally heavy multi-threading did not give any significant benefit over single-threading. To further increase our model performance, multi-processing is introduced to utilize multiple cores and run time consuming stages parallelly by pipelining the tasks to save time. This approach achieved a huge performance boost of almost 100% with each frame now being analysed in 55ms-95ms. The pipeline structure is shown in *Figure 7*.

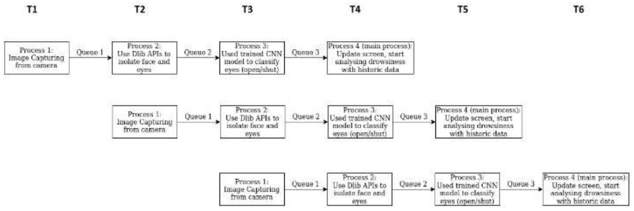

T1
T2
T3
T4
T5
T6
Process 1: Image Capturing from camera
Queue 1
Process 2: Use Dlib APIs to isolate face and eyes
Queue 2
Process 3: Used trained CNN model to classify eyes (open/shut)
Queue 3
Process 4 (main process): Update screen, start analysing drowsiness with historic data
Process 1: Image Capturing from camera
Queue 1
Process 2: Use Dlib APIs to isolate face and eyes
Queue 2
Process 3: Used trained CNN model to classify eyes (open/shut)
Queue 3
Process 4 (main process): Update screen, start analysing drowsiness with historic data
Process 1: Image Capturing from camera
Queue 1
Process 2: Use Dlib APIs to isolate face and eyes
Queue 2
Process 3: Used trained CNN model to classify eyes (open/shut)
Queue 3
Process 4 (main process): Update screen, start analysing drowsiness with historic data

***Figure** 7. Systems pipeline structure*

Finally, it is storing every eye state classification into a queue to process and analyse the queue further to calculate PERCLOS and predict the driver as drowsy if the PERCLOS value crosses Threshold Value (TV). Instead of using a discrete window frame approach for PERCLOS calculation as in [1], this approach is using a sliding window approach to lose minimum information, calculating PERCLOS value every second for the 2 second window frame and then predicting the driver as drowsy if the PERCLOS value crosses the Threshold Value (TV) which is set at 25%. The Framework of the entire system is shown in *Figure 10*.Working models output is shown in *Figure 8* and *Figure 9*.

$$P = \frac{E_c}{E_c+E_o} * 100\% \quad (1)$$

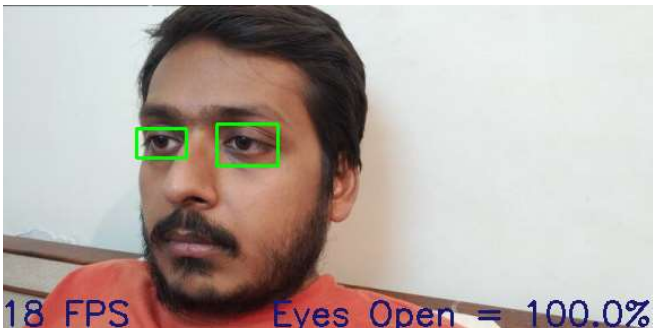


***Figure 8.** Normal state prediction by the system*

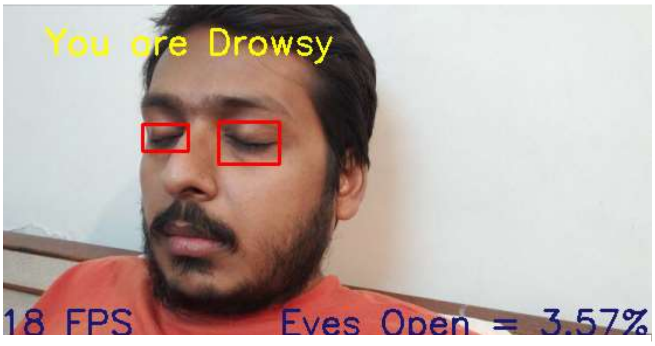


***Figure 9.** Drowsy state prediction by the system*

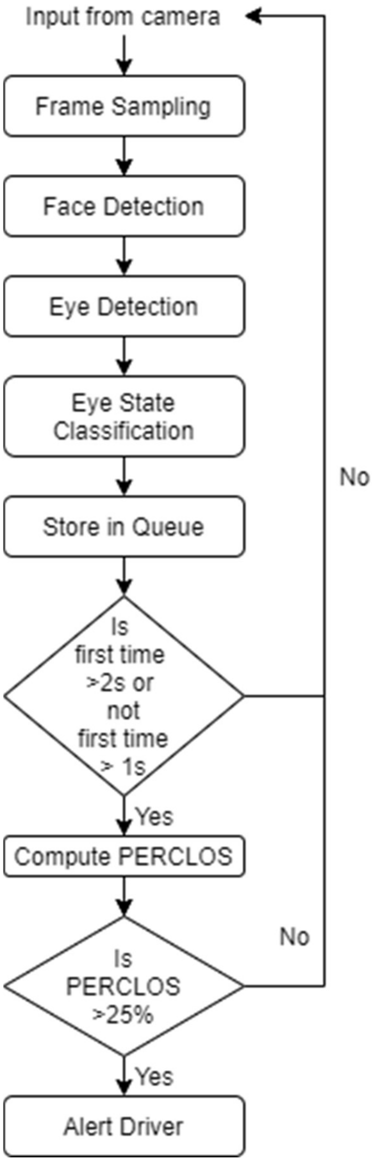


***Figure 10.** Entire system's framework*

V. Results Discussion

Our Classification model has achieved 95% training accuracy and 90% validation accuracy. The final model is quite small in size (191kB) and ideal for use in low powered devices. Our processing pipeline can process 15-18 frames per second compared to running on single-processor which could only process 5 frames per second. Finally, our drowsy detection system can successfully predict 94% of the times if the driver was drowsy or not when it is tested on different users which can be seen in *Table 1*.

Conditions	No of Test Cases	No of Correct Predictions	Success Percentage
No glasses	25	25	100%
Glasses	25	22	88%

Table 1. Final system's accuracy

$$Mean\ Accuracy = \frac{100 + 88}{2} = 94\%$$

VI. Conclusion

In this paper, we have taken a look at existing solutions on drowsiness detection from varied angles. Many successful high accuracy models have been made but there is still room for improvement. This paper shows a solution which analyzes driver's eyes for detection of drowsiness by extracting eyes from each frame using dlib's api and then feeding it to the eye classification model to predict eye state as open or close. Finally, storing the predicted values in a queue and analyzing it using a sliding window approach to calculate PERCLOS value and predict if the driver is drowsy or not when PERCLOS crosses TV. There is still room for improvement since the eye classification model has problems with classifying eye images with glasses that has high light reflection. Further improvements can be made in the classification model by filtering and creating a better dataset to train it and detect reflections on eyeglasses better to predict state of eyes more accurately in noisy conditions. Infrared cameras can also be used to detect eyes better in dark conditions.

References

[1] A. Dasgupta, D. Rahman, and A. Routray, "A smartphone-based drowsiness detection and warning system for automotive drivers," IEEE Trans. Intell. Transp. Syst., vol. 20, pp. 4045-4054, Nov. 2019, doi:10.1109/TITS.2018.2879609.

[2] Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., and Jiang, S., "Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques," in Proc. Comp. Sci. (Elsevier), E. Shakshuki and A. Yasar, 2018, vol. 130, pp. 400-407. doi:10.1016/j.procs.2018.04.060.

[3] You, F., Li, X., Gong, Y., Wang, H., and Li, H. (2019). "A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration." IEEE Access, 7, pp. 179396 - 179408, Dec. 2019, doi:10.1109/access.2019.2958667.

[4] W. Deng and R. Wu, "Real-Time Driver-Drowsiness Detection System Using Facial Features,", IEEE Access, 7, pp. 118727 - 118738, Aug. 2019, 10.1109/ACCESS.2019.2936663.

[5] Mandal, B., Li, L., Wang, G. S., and Lin, J., "Towards Detection of Bus Driver Fatigue Based on Robust Visual Analysis of Eye State," IEEE Trans. Intell. Transp. Syst., 18(3), 545 - 557, Mar. 2017, doi:10.1109/tits.2016.2582900.

[6] Bartolacci, C., Scarpelli, S., DAtri, A., Gorgoni, M., Annarumma, L., Cloos, C., Giannini, A. M., and De Gennaro, L., "The Influence of Sleep Quality, Vigilance, and Sleepiness on Driving-Related Cognitive Abilities: A Comparison between Young and Older Adults," Brain Sci. (MDPI Open Access), 10(6), 327, Jun. 2020, doi:10.3390/brainsci10060327.

[7] Li, Z., Li, S., Li, R., Cheng, B., and Shi, J., "Online Detection of Driver Fatigue Using Steering Wheel Angles for Real Driving Conditions," Sensors (MDPI Open Access), 17(3), 495, Mar. 2017, doi:10.3390/s17030495.

[8] Li, Z., Chen, L., Peng, J., and Wu, Y., "Automatic Detection of Driver Fatigue Using Driving Operation Information for Transportation Safety," Sensors (MDPI Open Access), 17(6), 1212, May. 2017, doi:10.3390/s17061212.

[9] Gwak, J., Hirao, A., and Shino, M., "An Investigation of Early Detection of Driver Drowsiness Using Ensemble Machine Learning Based on Hybrid Sensing," Applied Sciences (MDPI Open Access), 10(8), 2890, doi:10.3390/app10082890.

[10] Warwick, B., Symons, N., Chen, X., and Xiong, K., "Detecting Driver Drowsiness Using Wireless Wearables," 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems. Oct. 2015, doi:10.1109/mass.2015.22.

[11] Guede-Fernandez, F., Fernandez-Chimeno, M., Ramos-Castro, J., and Garcia-Gonzalez, M. A., "Driver Drowsiness Detection Based on Respiratory Signal Analysis." IEEE Access, 7, 81826 - 81838, Jun. 2019, doi:10.1109/access.2019.2924481.

[12] Driver Alert Control (DAC), Volvo 2018 V40 Manual (Driver Support), July 2018. [Online]. Available: https://www.volvocars.com/en-th/support/manuals/v40/2018/driver-support/driver-alert-system/driver-alert-control-dac (accessed Aug. 29, 2020).

[13] MRL Eye Dataset, January 2018. [Online]. Available: http://mrl.cs.vsb.cz/eyedataset (accessed Jan. 20, 2021).

[14] OpenVino Public Pretrained models, April 2020. [Online]. Available: https://docs.openvinotoolkit.org/latest/omz_models_model_open_closed_eye_0001.html (accessed Mar. 2, 2021).