

# Neural Network Based Detection of Driver's Drowsiness

Tanvi Naxane

Information Technology

International Institute of Information Technology  
tanvinaxane@gmail.com

Sayli Shrungare

Information Technology

International Institute of Information Technology  
saylishrungare107@gmail.com

Shraddha Bhandarkar

Information Technology

International Institute of Information Technology  
shraddhabhandarkar@gmail.com

Shivani Rajhance

Information Technology

International Institute of Information Technology  
shivaniirajhance12@gmail.com

Mrs. Pranali Deshmukh

Assistant Professor

International Institute of Information Technology  
pranalid@isquareit.edu.in

Tushar Kute

Data Scientist

Mitu Skillogies

tushar@tusharkute.com

**Abstract**—The primary purpose of this paper was to propose a way to alert sleepy drivers in the act of driving. Most of the traditional methods to detect drowsiness are based on behavioural aspects while some are intrusive and may distract drivers, while some require expensive sensors/hardware. Therefore, in this paper, driver's drowsiness detection system is developed and implemented to aid drowsy drivers from falling asleep and to prevent accidents. The system takes images from the device as input. Using these image templates, the trained model starts execution and predicts/classifies whether the face of the person in the image is drowsy or alert. The proposed model is able to achieve accuracy of 99.93% using CNN on trained image dataset.

**Index Terms**—CNN, Alarm, VGG16, Drowsiness Detection.

## I. INTRODUCTION

As we know, road transport is one of the biggest means of transportation in any country. Information available through statistical data, a total of 151,113 people were killed in 480,652 road accidents across India in 2019, an average of 414 accidents per day or 17 road mishaps in an hour. In a year, 40% of highway road accidents occur due to drivers dozing off which leads to a mishap. A driver who falls asleep at the wheel loses control of the vehicle, an action which often results in a crash with either another vehicle or stationary objects. To resolve this issue, we need to understand the 4 root cause of this pressing issue.

- **Fatigue:** One of the reasons of drivers falling asleep can be fatigue. This might happen when the driver has been on the wheel for too long or if he has monotonous heavy work before the ride.
- **Alcohol Intoxication:** Even small amount of alcohol consumption is enough to make the driver drowsy and cause distraction in driving.
- **Monotonous Road:** Another common reason for drivers to feel drowsy or fall asleep is when the roads are without any features or turnings/curves and has the same view which can give the drivers a calming effect and would eventually make them drowsy.
- **Inopportune Time:** Riding during inopportune times, such as during midnight can stimulate sleepiness in drivers.

- **After Food Consumption:** Riding after ingesting a heavy supper or a meal can make the driver fall asleep anytime.
- **Taking Medication:** Many drugs have a reducing reaction, like calming the nervous system, relieving pain, leading to drowsiness.

In order to monitor the state of drowsiness of the driver, following measures have been used widely:-

1. **Vehicle-based Measures** — A number of metrics, including deviations from lane position, movement of the steering wheel, pressure on the acceleration pedal, etc. are constantly monitored and any change or deflection is found in these measures that crosses a specified threshold indicates a significantly increased probability that the driver is drowsy.
2. **Behavioral Measures** — The behaviour of the driver, including yawning, eye closure, eye blinking, head position, etc. is monitored through the camera and the driver is alerted if any of these drowsiness symptoms are detected.
3. **Physiological Measures** — The correlation between physiological signals (Electrocardiogram (ECG), Electromyogram (EMG), Electrooculogram (EoG) and Electroencephalogram (EEG)) and driver drowsiness has been studied by many researchers over the years and have found that these measures are extremely dependable and accurate in detecting driver's drowsiness.

In this work, we developed a system based on the Behavioral Measures considering a drowsy person displays a number of characteristic facial movements and expressions, including rapid and constant blinking, nodding or swinging their head and frequent yawning. Developing a system based on these characteristics provides an efficient method and cooperative insights to see or detect whether a person is drowsy or alert.

We propose to develop a Deep Learning Model which will process the stored image templates/data to predict/classify whether the face in the image is drowsy or alert. Processing various acquired images as input, the trained model will provide a prediction using CNN(Convolutional Neural Network). We have

developed, implemented and compared the performance of various Classification Models such as VGG16 and ResNet50 for our classification problem. We have extracted and cleaned the image dataset from Kaggle, which comprised of images of human faces with several conditions, such as: eyes closed, eyes open, some with spectacles, some with hair coming in front of their faces etc. The image dataset is available on the official website of Kaggle.

## II. LITERATURE SURVEY

Our prime focus for literature survey was literature that addresses the problem statement and to research about developments on the same. Hence, our primary emphasis was based on three measures to detect drowsiness - Physiological Measures, Behavioral Measures and Vehicle-based Measures. In this paper, we have proposed a system that uses Behavioral Measures as means to detect drowsiness, to train the system/model and to finally provide optimized results. It provided accurate and satisfactory results as it was more accentuated towards the characteristics of a drowsy person. Focusing on features such as: blinking eyes too often, keeping eyes closed for a longer period of time, yawning, swaying the head more than usual etc. helped in providing better predictions/results.

[1] The first paper consisted of a survey conducted on state of drowsiness technique. It has comparative study of all three measures. These methods have been studied in detail and the advantages and disadvantages of each have been discussed. To develop an efficient drowsiness detection system, the strengths of the various measures should be combined into a hybrid system which is combination of two or three measures. However, it has not been implemented in real time so we didn't consider working on a hybrid model. The following table provides us with an essential information regarding the three measures. Most experiments using behavioral measures are conducted

TABLE I MEASURES TO DETECT

DROWSINESS			
Measures	Parameters	Advantages	Disadvantages
Vehicle Based	Deviation from the lane position loss of control over the steering wheel movements	Non-intrusive	Unreliable
Behavioral	Yawning, eye closure, eye blink, head pose	Non-intrusive, ease of use	Lightning condition background
Physiological	Statistical & energy features derived from ECG EoG EEG	Reliable	Intrusive

in a simulated environment and the results indicate that it is a reliable method to detect drowsiness. So far various methods have been implemented on the works of drowsiness detection using behavioral measures. Out of which CNN guarantees highest accuracy which is close to 100. Hence, we decided to consider this paper as our base paper.

[2] Franklin Silva and Eddie Galarza proposed a drowsiness detection system based on driver's face image behaviour using a system of Human Computer Interaction implemented on a Smartphone. If it detects that the person is drowsy, then it will alarm the driver and notify the same on the smartphone. The

system used PERCLOS algorithm for eye detection and gave an accuracy of 98.7% for blink rate. The objective of this work was to implement a surveillance system that alarms the driver. We decided to use this aspect in our proposed system to alert the driver.

[3] This paper attempted to address the issue by creating an experiment in order to calculate the level of drowsiness. A requirement for this paper was the utilisation of a Raspberry Pi Camera and Raspberry Pi 3 module, which were able to calculate the level of drowsiness in drivers. The frequency of how often head tilting and blinking of the eyes was captured was used to determine whether or not a driver felt drowsy. With an evaluation on ten volunteers, the accuracy of face and eye detection was calculated up to 99.59 percent. However, it uses Haar-Cascade classifier which is not efficient for huge datasets.

[4] Maneesha V Ramesh, Aswathy K. Nair and Abhishek Kunnath proposed a real time automated multiplexed sensor system which aims in developing an intelligent wireless sensor network to monitor and detect driver drowsiness in real-time. It consists of multiple non-obstructive sensors which continuously monitor the driver's physiological parameters and disseminate the first level alarm to the driver and the passengers. The second level alarm will be disseminated, along with the vehicle identification number and the real-time location coordinates of the driver to the nearby police station or the rescue teams using the available wireless ad-hoc network if the driver's state does not change even after the first level alarm. On the contrary, since our primary aim was to work on behavioral measures, we considered it best not to go with this method, as it is intrusive.

[5] In this paper, proposed by Challa Yashwanth and Jyoti Singh Kirar, Artificial Intelligence-based advanced algorithms were used to detect driver fatigue and the rate at which the driver is drowsy. It uses eye and mouth vertical distances, eye closure, yawning. Although the proposed classifiers are good enough to give reasonable results, still there is a lot of latitude for improvement in their performance. A more robust drowsiness detection classifier can be still be improved by researching on other datasets.

## III. NEURAL NETWORK ALGORITHMS

Deep Learning is a subset of Machine Learning and it primarily involves processing of raw input data through numerous layers of non-linear transformations. Deep Learning focuses on simulating operations and algorithms similar to the human brain. The main reason behind calling it "Deep Learning" is that it involves heavy and large layered network of ANN(Artificial Neural Network) layers. Deep Learning provides automatic/implicit feature extraction and feature selection. It is instinctively capable of deriving or obtaining meaningful information/features from the acquired input data which is further employed for learning, understanding, generalizing and predicting output class from the provided input data. The prolonged task of data pre-processing and feature engineering is cut down since deep learning models implement feature engineering internally which saves a lot of time, energy and work

of the Data Scientists and Programmers. Moreover, a Deep Learning model works best with huge unstructured data and delivers high performance when it is given as input to the model.

Neural networks are a collection of algorithms, modeled loosely to identify relationship among the data via processing which operates in a similar way like the human brain and that are designed to find knowledge/insights in patterns. They interpret sensory data through a sort of machine perception, labeling or clustering raw input. The patterns that are recognizable to Neural Networks are numerical and vectors, in which all real-world data like images, sound, text or statistics must be translated into meaningful data.

Neural Networks are extremely convenient for clustering information and classifying it into various target classes. One can think of them as classification and clustering layer, present above the data you are storing, managing and processing. Neural Networks are found to be useful in grouping data as per the similarities among acquired inputs which they then classify when they have a labelled dataset to train the model on. Additionally, Neural Networks are capable of extracting features from images/series of images/videos which are later fed to the algorithms for clustering and classification. Hence, Deep Neural Networks can be visualized as components of Machine Learning applications which involves Classification, Regression, Reinforcement Learning etc.

As we know, a neural network is made up of several interconnected computational units. Within the neural network, a basic computational unit performing specific computation or processing is called a Neuron. Neuron is mainly a node through which data computation is carried out. They receive input signals from any one the layers and it sends out the output signal to the subsequent layers.

Learning Process of Neural Networks - Learning Process of a Neural Network is divided into two phases - Forward Propagation and Back Propagation. Forward Propagation involves propagation or transferring of information from the input layer to the output layer. This whole one cycle of passing data from input layer to output layer is called Forward Propagation. Whereas, the propagation of information from output layer to the input layer is called Back Propagation. The output from the output layer is computed further to calculate the possible loss. After we quantity or obtain loss/cost/error, our next step is to use an Optimizer. The obtained loss is analyzed and a suitable Optimizer is used to optimize the values of weights (w) and biases (b). Optimizers perform 'Back Propagation' where the model learns. This whole process is repeated until the model delivers best optimized results.

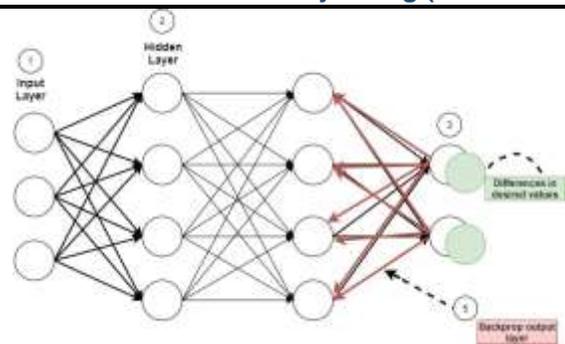


Fig. 1. Learning Process in Neural Network

There are mainly 3 types of layers for Neural Network based Deep Learning models - Input Layer, Hidden Layer and Output layer where the Input Layer is provided with raw input data or external data and the output from the Input Layer is fed to the Hidden Layer for further processing and finally the processed data is then given to the Output Layer which gives us the final output/prediction. Neural Networks consist many hidden layers and they can get vast in structure whereas Shallow Networks tend to have less number of layers which makes them less complex as they require more up-front knowledge of optimal features/characteristics. Deep Learning algorithms depend on employing optimal model selection and result optimization through enough model training. Deep Learning models are considered reliable to solve problems even when the prior knowledge of features is missing or when the labelled data is unavailable or not required. In addition, Neural Networks also supports techniques from Signal Processing which includes non-linear transformations.

There are several types of neural networks, some of the most commonly used are - Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Artificial Neural Networks (ANN). For our system it was very essential to understand all the above mentioned neural networks and get a better understanding of various neural network models.

TABLE II  
COMPARATIVE STUDY OF NEURAL NETWORKS

Parameters	ANN	CNN	RN
Types of Data	Tabular Data, Text data	Image Data	Sequence Data
Parameter Sharing	No	Yes	Yes
Fixed Length Input	Yes	Yes	No
Recurrent Connections	No	No	Yes
Vanishing and Exploding Gradient	Yes	Yes	Yes
Spatial Relationship	No	Yes	No
Performance	Least powerful	Most Powerful	Lower than CNN, Higher than ANN
Application	Facial Recognition, Computer Vision	Facial recognition, text digitization and Natural	Text to Speech conversions

		language processing	
Advantages	Fault tolerance, ability to work with incomplete knowledge	High accuracy in image recognition problems, weight sharing	remembers each and every information, Time series prediction
Disadvantages	Hardware dependency, unexplained behavior of network	Large training data needed	Gradient vanishing, exploding gradient

Since we are using images as input, we fixed CNN for our system implementation. Convolutional Neural Network helped us achieve high accuracy for our proposed model.

A. CNN

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which takes images as input, assigns importance (learnable weights and biases) to various aspects/objects in the image and are able to differentiate/classify data in various categories/classes. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms, while in primitive methods filters are hand-engineered with enough training, ConvNets have the ability to learn these filters/characteristics.

A Convolutional Neural Network typically consists of Convolutional Layers which undergo recurrent downsampling by applying pooling layers. Pooling Layers are responsible for extracting dominant features from obtained input data and transmitting its output to the subsequent layers like normalization layers and fully connected layers.

CNN is majorly used for processing image data and for image classification. With the help of CNN, you can easily build a classification model. An image is nothing but a matrix with several pixel values that represent various features like brightness, shape, size, colour etc. A pixel is a 1-bit number that represents either foreground or background of the image. Every image is a combination of numerous features like size, colour, brightness, shape etc. and our main objective is to segregate or separate out the distinct features/properties from the image to better understand the characteristics of the image. To do this, we need filters. Filters are matrices/arrays with different pixel values used to detect patterns/edges in the image. A filter represents one particular property or a feature which is mapped to the original image to extract dominant features that helps to better generalize the image into the target output class. Filter size should be in the form of N x N, where 'N' is a positive integer representing the size of the matrix filter. The process of employing a Filter/Kernel of the size 2 x 2 or 3 x 3 to the entire input image to map/distinguish/separate out dominant or distinct features from the original image which provides us with meaningful information is called Convolution. Hence, it is named as "Convolutional Neural Network".

- Convolutional Layer - Convolutional Layers are important building blocks of Convolutional Neural Networks. Convolutional Layer implements Convolutional Operation

that undergoes data filtration. Kernels/Filters are applied on the actual image to extract meaningful patterns or features which are later passed to the subsequent layers.

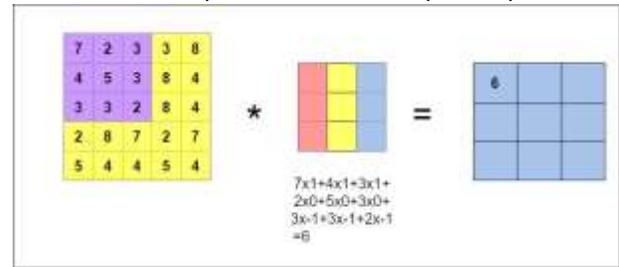


Fig. 2. Convolution Operation

- Pooling Layer - Convolutional Neural Networks include Pooling Layers along with the convolutional layers. Pooling layers are mainly responsible to reduce the dimension i.e the number of parameters and to extract only the dominant features which helps the model to better read, understand and interpret the input image. Pooling layer is applied to the convolutional layer to extract only those features that will contribute more to the model. Pooling layers decreases the overall size, number of parameters and computation time of the image. Pooling layers commonly consists filter sizes of 2 x 2 or 3 x 3. There are 3 types of pooling Minimum Pooling, Maximum Pooling and Average Pooling. Maximum Pooling is found to be widely used for classification tasks.

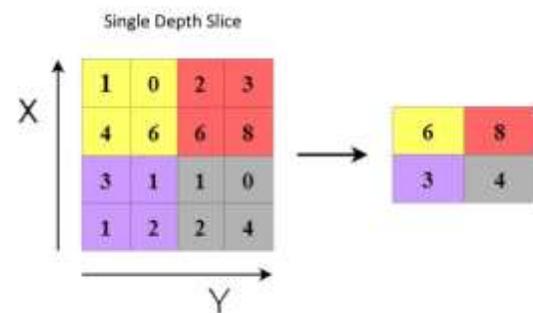


Fig. 3. Max-Pooling Operation

- Fully Connected Layers- Fully Connected Layer, as the name indicates, is the layer where every neuron in one layer is connected to every neuron of another layer. Fully connected layers are used to further generalize the information obtained from the previous layers. The output of convolutional layer or a pooling layer is provided to the fully connected layer. This information in fully connected layer is in the form of a matrix with important pixel values. This matrix is subsequently transformed into a flattened array of matrix values which is then given to the output layer to predict/display the final output.

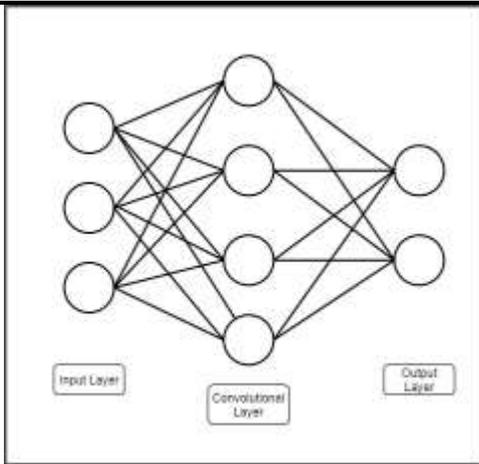


Fig. 4. Fully Connected Layer

- Weights and Biases - Weights and Biases are also referred as 'w' and 'b' and are commonly known as the learnable parameters of any machine learning model. Weight is an integer value which is initially assigned to the neuron which tells us how important the neuron is and they it also represents the strength of the connection between neurons in the network. A Bias is primarily known as Training Error and is a constant unique number associated with the neuron. This bias is added to the weighted sum of inputs and sent forward for further processing. The weights and biases are continuously adjusted and altered to optimize results in every epoch and to fully train the network/model.

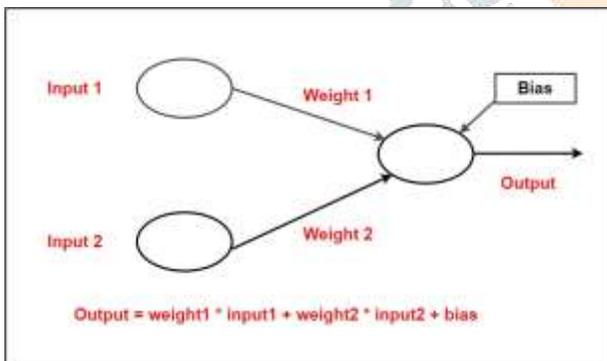


Fig. 5. Weights and Biases in CNN

- Activation Layer - An activation function is encountered in hidden layers when we pass data to the neural network. Activation functions are predominantly used for normalizing the data. They are basically used to achieve Normalization and Non-linearity from input data. Activation Functions are capable of providing an output with certain range or boundaries. They are used at the time of back propagation to regularize and scale the input data. They are responsible for deciding whether a neuron will be fired or not. There are several activation functions available like Sigmoid, Tanh, ReLU, Leaky ReLU, ELU etc. For our proposed system with two target classes (Drowsy/Tired and Alert), Sigmoid Activation Function is used as it delivers best results for binary class classification tasks.

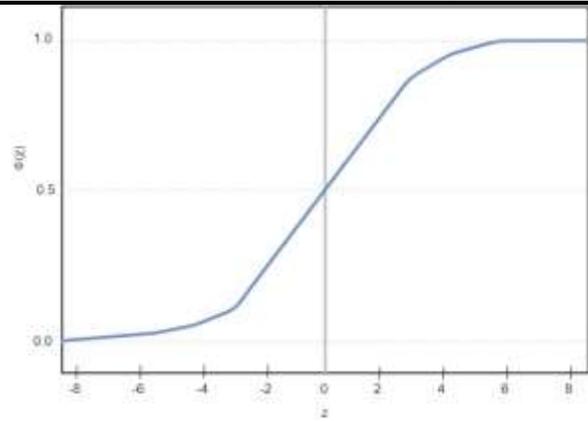


Fig. 6. Sigmoid Activation Function in CNN

There are several models to implement CNN, out of which ResNet and VGG16 are most commonly used for classification. However, due to ResNet being extremely powerful and having very complex and deep architecture we settled on working with VGG16 for image data categorization/classification.

VGG-16 is a Convolutional Neural Network Model comprising of total 16 layers, out of which 13 layers are convolutional layers and 3 layers are fully connected layers. The model surpasses AlexNet by replacing large kernel sized filters with multiple 3x3 sized kernel-sized filters one after the other. VGG16 model loads a set of weights pre-trained on ImageNet dataset having over 20,000 categories. ImageNet consists of variable-resolution images. On that account, the images are down-sampled to a fixed resolution of 224x224. Given a rectangular input image, the images are re-scaled and cropped to a fixed size of 224x224 from the resulting image. The default input size for VGG16 model is 224 x 224 pixels with 3 channels for RGB image. It has convolution layers of 3x3 filter with a stride 1 and max-pool layer of 2x2 filter of stride 2.

#### IV. METHODOLOGY

VGG16 as Transfer Learning – Due to longer training time durations, a shortcut to this is to use/re-use an existing model/pre-trained model and model weights that were developed for standard use. VGG16 happens to be a pre-trained model with 16 layers out of which 13 are convolution layers and 3 are fully connected layers. These pre-trained models can be combined with new model or used directly for carrying out various classification tasks. Transfer Learning is a process of re-using the pre-trained model on a new task/problem statement. It is a machine learning technique where a developed model is used to perform new tasks/solve new problems. This marks the flexibility of transfer learning which allows data scientists/programmers to use the pre-trained models directly for feature extraction, image processing and finally for classification.



Fig. 7. VGG16 Architecture

An image of fixed size 224x224x3 Fig 7, is provided as an input to the first convolution layer. The first two layers have 64 channels. The input image is transferred through the series of convolution layers and filters of fixed size 3x3. Further, MaxPooling is carried out with the help of max-pooling layers with 2x2 pixel window size and stride 2. Three fully connected layers are applied after the convolution layers. The final layer is the Output Layer to predict the final output class/category. For drowsiness detection, Class 1(Drowsy) and Class 2(Not Drowsy/Alert) were provided for prediction from the image dataset which later, made use of Sigmoid Function for predicting the final output class. Obtained output is assessed and evaluated i.e. it is compared with expected output. Then the loss function is used to quantify the deviation from the expected output and information is sent from output layer to input layer to employ back propagation in the neural network. This process will be repeated until the loss/cost is minimized and expected output is obtained.

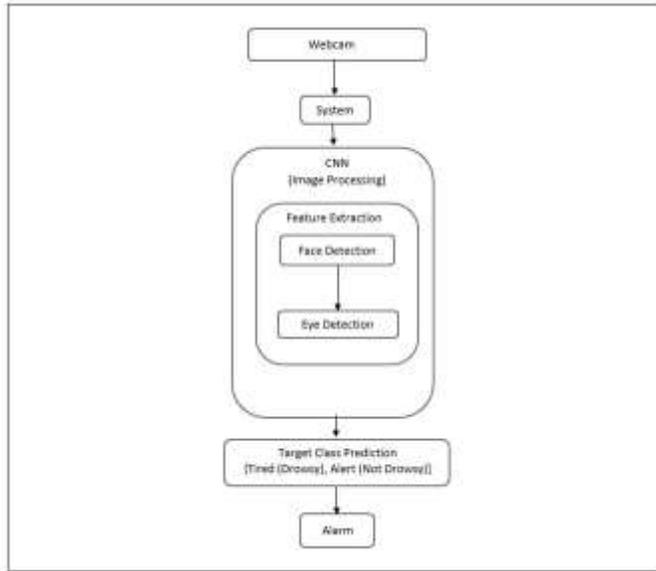


Fig. 8. System Architecture

V. RESULTS

The results for several conditions were calculated and tested in order to achieve high accuracy. These conditions vary from person to person, with person under observation having their eyes closed or open. The person could be someone who wears spectacles due to which it might cause reflection of light and result in a glared image, making it difficult for feature extraction and feature selection. So, in order to avoid these hindrances, the proposed model was tested against the above mentioned constraints as well to achieve satisfactory results. Various conditions with their corresponding results have been depicted in the figures below:-

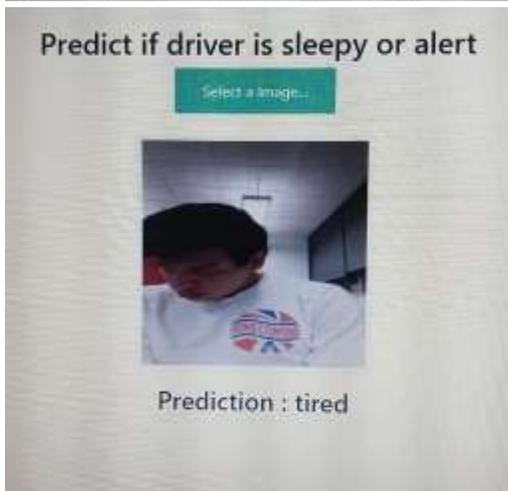


Fig. 9. Results obtained from training image dataset using VGG16 Model

Fig. 12. Person with spectacles - Alert State



Fig. 13. Person with spectacles - Drowsy State



The results have been tabulated in the table below:-

TABLE III  
RESULT

Measure	VGG16	ResNet50
Accuracy	0.9993	0.9795
Loss	0.0127	0.5168
valaccuracy	0.9906	0.4178
valloss	0.0366	0.03759

### VI. FUTURE SCOPE

Our model was able to successfully incur satisfactory results. However, the main challenge faced was training of images in dim/low light as the model finds it difficult to process images in extremely low light which results in poor feature extraction and feature selection and therefore, wrong/inaccurate predictions. Another challenge our model faced was live capturing, processing and training of videos and splitting the captured/acquired video into a series of picture frames for further processing, feature extraction and finally classification. We have included this constraint in our future scope to increase the reliability of our model.



Fig. 10. Alert State • This system can be further utilized and employed to alert/notify the driver during the night, which happens to be the prime time for the drivers to feel sleepy or fall asleep. A night vision camera can be employed for this purpose to deliver accurate results.



• Furthermore, a video dataset or live videos could be used as input to our model which would further divide the acquired video into a stream of images for further CNN processing and classification into any one of the categories or defined classes, making our model more reliable and dependable. Video Dataset can be created

Fig. 11. Drowsy State by the programmer or the team members and can be later used for training purpose to provide better optimized results.

## VII. CONCLUSION

In this paper, we have proposed a Driver's Drowsiness Detection System which takes images as input to our model, processes the images, performs feature extraction and feature selection and finally categorizes/classifies the image into any one of the categories i.e. Drowsy or Alert. Several classification algorithms have been developed and benchmarked. Out of those, VGG16 Model achieves an accuracy of 99.96%.

## ACKNOWLEDGMENT

We would like to extend our sincere gratitude towards our Project Guide, Prof. Pranali Deshmukh and our Project Sponsor, Mr. Tushar Kute for guiding and mentoring us throughout the course of project development and implementation, providing us with the necessary resources, knowledge and constant moral support.

## REFERENCES

- [1] D. Qian et al., "Drowsiness Detection by Bayesian-Copula Discriminant Classifier Based on EEG Signals During Daytime Short Nap," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 4, pp. 743–754, 2017.
- [2] "European Road Safety Observatory (2006) Fatigue," 2006.
- [3] "Driving while Drowsy – What's the Harm? Watch Out for Signs of Sleepiness," *Drowsy Driv. Prev. Resour. Cent.*, 2007.
- [4] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Detecting driver drowsiness based on sensors: A review," *Sensors (Switzerland)*, vol. 12, no. 12, pp. 16937–16953, 2012.
- [5] R. N. Khushaba, S. Kodagoda, S. Lal, and G. Dissanayake, "Driver drowsiness classification using fuzzy wavelet-packet-based feature extraction algorithm," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 1, pp. 121–131, Jan. 2011.
- [6] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Drowsiness detection during different times of day using multiple features," *Australas. Phys. Eng. Sci. Med.*, vol. 36, no. 2, pp. 243–250, Jun. 2013.
- [7] Y. S. Kim, H. J. Baek, J. S. Kim, H. B. Lee, J. M. Choi, and K. S. Park, "Helmet-based physiological signal monitoring system," *Eur. J. Appl. Physiol.*, vol. 105, no. 3, pp. 365–372, 2009.
- [8] A. Lal, S. K. L. and Craig, "A critical review of the psychophysiology of driver fatigue. *Biological Psychology*," vol. 55, no. 2001, pp. 173–194, 2001.