INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

www.ijraset.com

Call: ⓒ08813907089 | E-mail ID: ijraset@gmail.com

Drowsy Driver Detection using Deep Learning

Priyanshu Singh[1], Vishal Kumar Singh[2], Vishal Verma[3], Sakshi Malhotra(Guide)[4]
Department of Information Technology, Galgotias college of engineering and technology, Greater Noida

Abstract: The face is an essential body feature that reveals a lot of information. When a driver is fatigued, their facial expressions change from what they would be in a normal state, such as their eyes blinking more frequently than usual. In this study, we used CNN algorithm to recognize signs of driver drowsiness such as blinking and length of eye closure. CNN algorithm uses video pictures. Due to the drawbacks of existing techniques, we propose a unique face-tracking algorithm to enhance tracking accuracy. Based on human key characteristics, we applied a unique detection algorithm i.e. CNN. Then, using these parts of the face, the drivers' drowsiness is evaluated by watching their eye movements. We have calculated the system's accuracy, which is around 97%.
Keywords: CNN, Facial Key Regions, Feature Extraction, Face Tracking, Machine Learning.

I. INTRODUCTION

Driver drowsiness is one of the main factors that lead to fatalities in traffic accidents. Driver tiredness and drowsiness are common after lengthy periods of continuous driving because drivers readily become exhausted. According to studies, driver fatigue is a major factor in accidents. The statistics for accidents brought on by driver sleepiness vary among nations. The key problem is creating technologies for recognizing driver drowsiness to reduce accidents. In India, there were 4,552 documented incidents in 2016 that claimed thousands of lives as a result of sleepy drivers, according to a report from the "Ministry of Road Transport & Highways" (Road incidents in India 2016). For instance, many vehicles, such as loaded trucks, are driven primarily at night. The drivers of such automobiles who operate them for such a lengthy period continuously grows the risk of these kinds of events. In order to decrease the incidence of these mishaps and accidents, research into detecting driver drowsiness is still ongoing. Physiological, vehicle, and behavioral approaches are frequently used to detect sleepy drivers. The level of exhaustion is determined by physiological techniques such as electrocardiograms, pulse rate and heartbeats. The use of the steering wheel movement, acceleration and accelerator pattern, are examples of vehicle-based approaches. Eye blinking, eye closure and other behaviors are examples of behavioral approaches. To address this global issue, a solution that consecutively captures driver's pictures, in real time and sends them to a model, and detects drowsiness using EAR (Eye Aspect Ratio) and ECR (Eye Closure Ratio) has been presented and put into practice. The system's computed value alerts the driver to stop and relax for a while..

II. BACKGROUND DETAILS

An easy way to comply with IJRASET paper formatting requirements is to use this document as a template and simply type your text into it.

A. CNN Algorithm

CNN (Convolutional neural network) is a kind of artificial neural networks that are frequently used in tasks involving natural language processing, image and video identification, and other types of machine learning.
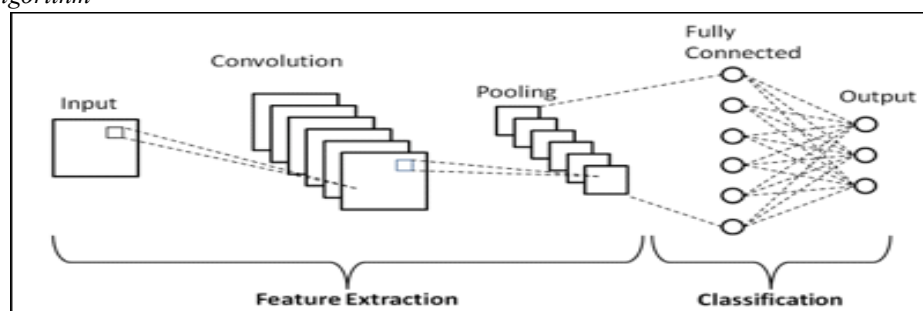
1) Layers Of CNN Algorithm

Figure 1: Layers of CNN algorithm.

The CNN algorithm consists of several layers of interconnected nodes that perform various operations on the input data, such as convolution, pooling, and activation. Here is a brief overview of these layers:

a) *Input Layer:* This layer takes the input data, which in the case of image recognition is a matrix of pixel values.

b) *Convolutional Layer:* This layer applies a set of filters (also known as kernels) to the input data. Each filter slides over the input matrix, performing a dot product between its values and the corresponding values of the input matrix. The result is a set of feature maps that highlight different aspects of the input data.

c) *Activation Layer:* This layer applies a non-linear activation function (such as ReLU) to the output of the convolutional layer. This helps to introduce non-linearity into the model and improve its ability to learn complex patterns.

d) *Pooling Layer:* This layer reduces the spatial dimensions of the feature maps by taking the maximum or average value within a sliding window. This helps to make the model more robust to variations in the input data and reduces the number of parameters that need to be learned.

e) *Fully Connected Layer:* This layer takes the flattened output of the previous layers and applies a set of weights and biases to produce the final output. This layer is similar to the one in a regular neural network and is used to make the final prediction.

2) Benefits of CNN Algorithm

Some of the benefits of using CNNs are:

a) *Efficient Feature Extraction:* CNNs use a series of convolutional and pooling layers to extract features from the input images. This helps in reducing the dimensions of the input data and capturing important patterns and details that are relevant to the task at hand.

b) *Ability to Handle large Datasets:* CNNs are capable of handling large amounts of data, making them suitable for complex image and video processing tasks. They can also learn and adapt to new data over time, making them ideal for applications that require continuous learning.

c) *High Accuracy:* CNNs have been shown to achieve high levels of accuracy in a wide range of image and video processing tasks, including image classification, object detection, and segmentation.

d) *Robustness to Variations:* CNNs are robust to variations in input data such as changes in lighting, orientation, and scale. This is because the convolutional layers are able to capture the essential features of an image regardless of its variations.

e) *Transfer Learning:* CNNs trained on large datasets can be used as a starting point for training on new datasets. This is known as transfer learning and can significantly reduce the amount of time and resources required for training a new CNN model from scratch.

B. Datasets

In machine learning (ML), datasets are a critical component for training and evaluating models.

Here are some common types of datasets used in ML:

1) *Training Dataset:* This is the dataset that is used to train a machine learning model. It consists of a set of input data and corresponding output labels, and the model learns to predict the output labels based on the input data.

2) *Validation Dataset:* This dataset is used to tune the hyperparameters of a machine learning model and to prevent overfitting. It is used during the training process to evaluate the model's performance on data that it has not seen before.

3) *Test Dataset:* This dataset is used to evaluate the performance of a machine learning model after it has been trained. It is typically a separate dataset from the training and validation datasets and contains data that the model has not seen before.

4) *Public Datasets:* These are datasets that are available to the public for research purposes. They can be used for benchmarking and evaluating machine learning models.

Here we have used three categories of dataset for training and validation of images

- Cropped images
- F-opened images
- F-closed images

III. PROPOSED WORK

A driver drowsiness system is designed to detect and alert drivers when they are becoming drowsy or distracted while driving. One way to detect drowsiness is to use eye-tracking technology. By tracking a driver's eye movements and identifying signs of drowsiness such as heavy eyelids, the system can alert the driver to take a break or pull over. Another approach is to use machine learning algorithms that analyze a driver's behavior and driving patterns to determine if they are becoming drowsy. The system can then provide an alert or take action to prevent an accident. Facial recognition technology can be used to detect signs of drowsiness or distraction, such as closing or opening eyelids.

A. Implementation

Implementing a driver drowsiness system involves several steps, including importing modules, setting parameters, importing datasets from directories, datasets splitting, data augmentation, defining and designing models, training models, plotting graphs, saving and testing models.

Detailed explanation of the above steps is discussed below:

1) *Importing Modules:* In machine learning (ML), modules are used to extend the functionality of the programming language being used. These modules contain pre-written code that can be imported into your ML project, which can save time and effort in writing the code from scratch. Here, we have imported various modules like face_recognition, tensorflow, pandas, numpy, cv2 and os.

2) *Setting Parameters:* In machine learning (ML), setting parameters is a critical step in building and training a model. Parameters are values that are set before training begins and can greatly affect the performance of the model.
   Here we have set the following parameters:
   batch_size_val = 80
   epochs_val = 100
   noOfClasses=2
   imageDimesions = (80, 80,3)
   testRatio = 0.2
   validationRatio = 0.2

3) *Importing Datasets:* In machine learning (ML), importing datasets is a crucial step in building and training models. Datasets contain the input and output data used to train and evaluate ML models.
   Here we have imported three categories of image dataset
   - Cropped images
   - F-opened images
   - F-closed images

4) *Dataset Splitting:* Dataset splitting is an essential step in machine learning (ML) that involves dividing a dataset into two or more subsets. The purpose of data splitting is to create a training dataset that is used to build a machine learning model, and a separate testing dataset that is used to evaluate the performance of the model.

5) *Data Augmentation:* Data augmentation is a technique used in machine learning to increase the size of a training dataset by generating new data from existing data. The goal of data augmentation is to create a more diverse and representative dataset that can improve the performance of machine learning models. Here we have used Image Augmentation this include flipping, rotating, scaling, cropping, adding noise, changing brightness and contrast, and color shifting for an image.

6) *Defining and Designing Models:* In machine learning (ML), defining and designing models involves specifying the structure and behavior of the model that will be used to learn patterns from the data. The model architecture determines how the input data is transformed into the output data, and how the model will be trained to optimize its performance. Here we have used CNN (Convolutional Neural Network) model that is commonly used for image classification tasks. It consists of convolutional layers that learn local patterns in the input images, and pooling layers that down sample the output of the convolutional layers.

7) *Training Models:* Training models in machine learning involves using algorithms to learn patterns from the data and make predictions or classifications.

8) *Plotting Graphs:* Plotting graphs is an important part of machine learning because it allows us to visualize the data and the performance of the models.
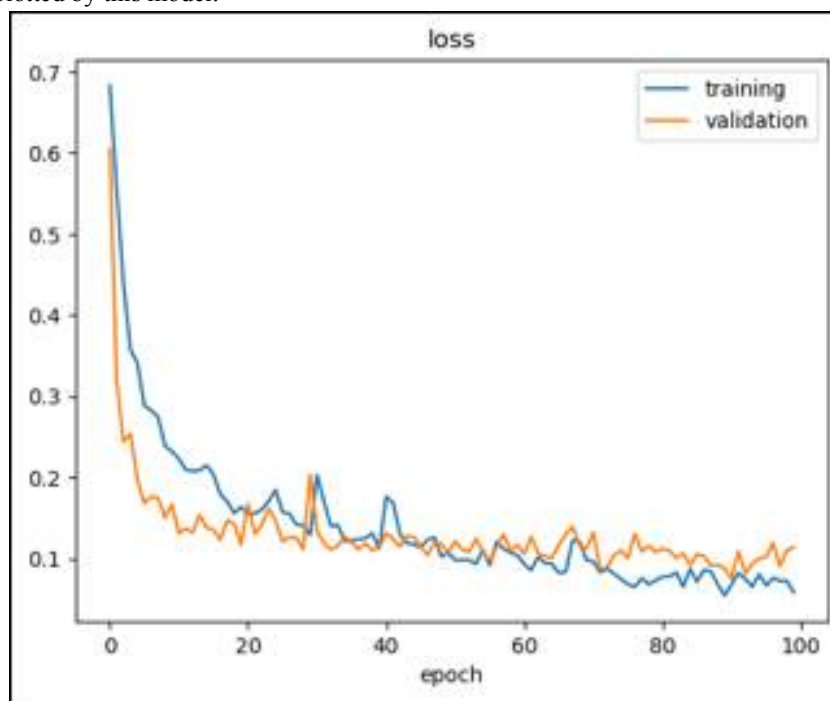
The following graphs are plotted by this model:

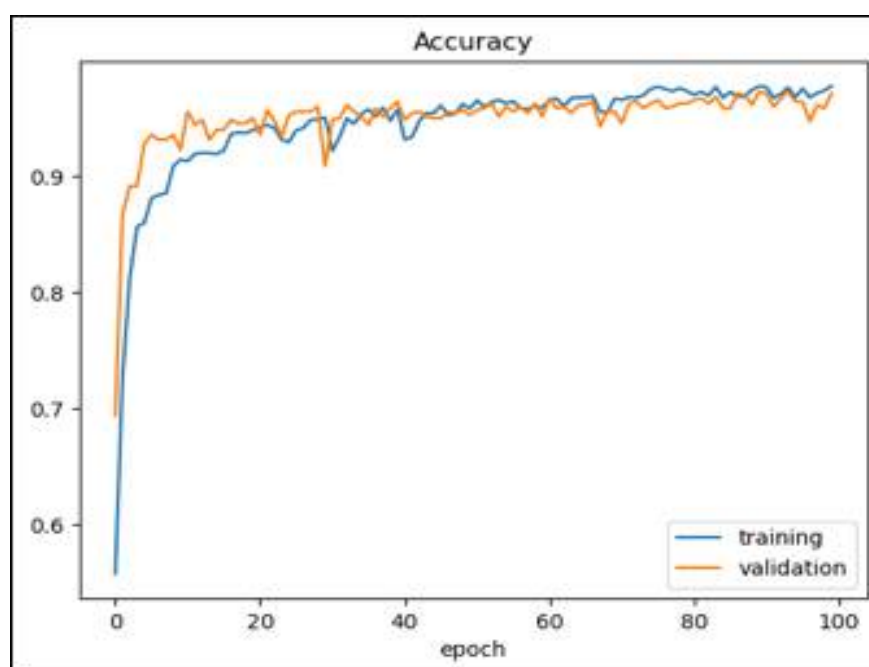Figure 2: Loss graph of Training and Validation

Figure 3: Accuracy graph of Training and Validation

9) *Saving and Testing Models:* Saving and testing models is an important part of machine learning because it allows us to reuse the trained models and evaluate their performance on new data.

B. Formulas Used

Formulas for EAR (Eye Aspect Ratio):

- Establish the max x and y coordinates of the eye

  x_max = max([coordinate[0] for coordinate in eye])

  x_min = min([coordinate[0] for coordinate in eye])

  y_max = max([coordinate[1] for coordinate in eye])

  y_min = min([coordinate[1] for coordinate in eye])

- Establish the range of x and y coordinates

  x_range = x_max - x_min

  y_range = y_max - y_min

- Calculate the coordinates of a square

  if x_range > y_range:

     right = round(.5*x_range) + x_max

     left = x_min - round(.5*x_range)

     bottom = round(((right-left) - y_range))/2 + y_max

     top = y_min - round(((right-left) - y_range))/2

  else:

     bottom = round(.5*y_range) + y_max

     top = y_min - round(.5*y_range)

     right = round(((bottom-top) - x_range))/2 + x_max

     left = x_min - round(((bottom-top) - x_range))/2

IV.SYSTEM WORKFLOW
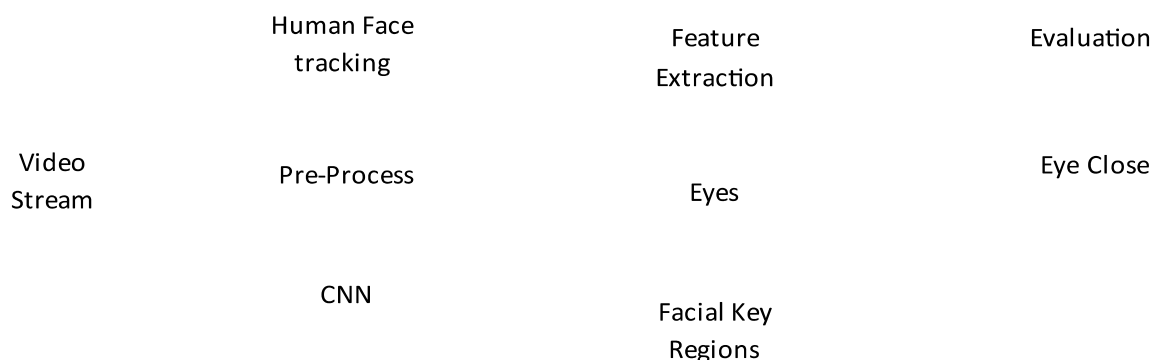
Figure 4: System workflow

The workflow of driver drowsiness detection systems is designed to help and ensure the safety of both the driver and other road users by alerting drivers when they are at risk of falling asleep at the wheel.

The system workflow for driver drowsiness detection is briefly described below:
1) The system captures input from webcam to determine the driver's state. This includes the real time video streaming of the driver's face.
2) During the detection process, the quality of images is affected and features of the human face become unclear if the illumination intensity within the cab is changed during driving
3) We use the histogram equalization (HE) algorithm to improve the brightness of the image frame.
4) Then, the model based on CNN algorithm analyzes the video and detects the driver's degree of drowsiness. In this stage, three main parts are analyzed: the driver's face tracking, facial key-region recognition (eye opening and closing), and driver's fatigue state.
5) Then, the model estimates the driver's state when the states of the eye's changes.
6) Finally, the model produces the result and a warning tone is generated if the driver is observed to be drowsy.

V. RESULTS

This section presents the performance evaluation of the proposed approach by performing an empirical analysis of obtained results. First, the system collects the real-time data of the drivers depicted by Figures 5(a), 5(b) and 6(a), 6(b) and 7. It then determines drowsiness of the drivers based on the EAR values that are computed based on the images captured of the user and its response from the model.
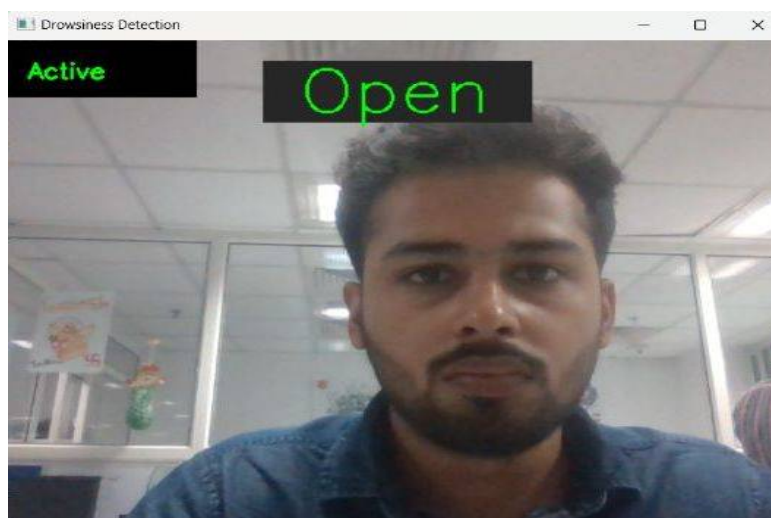

Figure 5(a): Result when eyes are open (without spectacles)

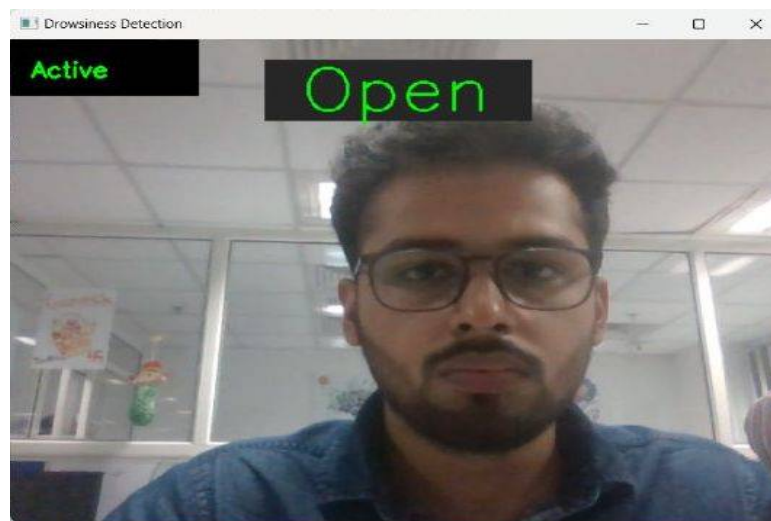Figure 5(b): Result when eyes are closed (without spectacles)

Figure 6(a): Result when eyes are open (with spectacles)

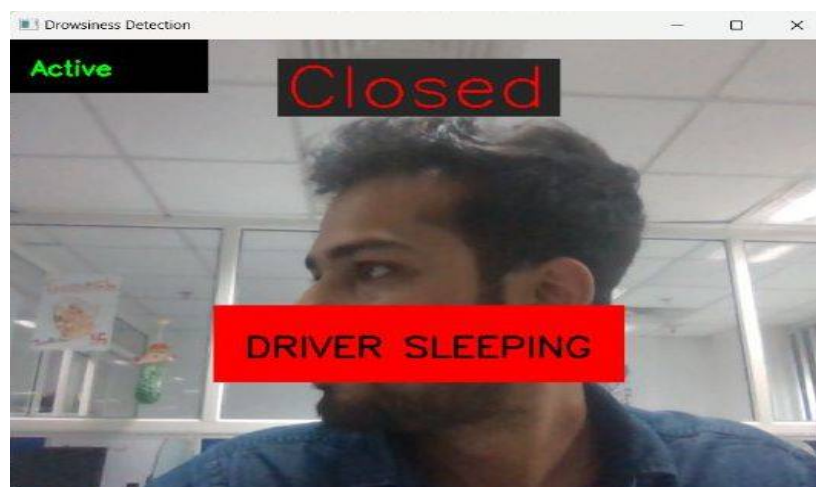Figure 6(b): Result when eyes are closed (with spectacles)

Figure 7: Result when Head tilt

VI. CONCLUSIONS

In this work, a real time system that monitors and detects the loss of attention of drivers of vehicles is proposed. The face of the driver has been detected by capturing facial landmarks and warning is given to the driver to avoid real time crashes. Non-intrusive methods have been preferred over intrusive methods to prevent the driver from being distracted due to the sensors attached on his body. The proposed approach uses Eye Aspect Ratio and Eye Closure Ratio with adaptive thresholding to detect driver's drowsiness in real-time. This is useful in situations when the drivers are used to strenuous workload and drive continuously for long distances. The proposed system works with the collected data sets under different conditions. The facial landmarks captured by the system are stored and machine learning algorithms have been employed for classification. The system gives best case accuracy of 97.8% for random forest classifier.

VII. ACKNOWLEDGMENT

REFERENCES

[1]   Elena Magan, M. Paz Sesmero, Juan Manuel Alonso-Weber and Araceli Sanchis (2022, January). Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images.

[2]   Md. Tanvir Ahammed Dipu, Syeda Sumbul Hossain, Yeasir Arafat, Fatama Binta Rafiq(2021, July). Real-time Driver Drowsiness Detection using Deep Learning.

[3]   Jabbar, R., Shinoy, M., Kharbeche, M., Al-Khalifa, K., Krichen, M., & Barkaoui, K. (2020, February). Driver drowsiness detection model using convolutional neural networks techniques for android application. In 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT) IEEE.

[4]   Reddy, B., Kim, Y. H., Yun, S., Seo, C., & Jang, J. (2017). Real-time driver drowsiness detection for embedded system using model compression of deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops

[5]   S. Park, F. Pan, S. Kang and C.D. Yoo, "Driver drowsiness detection system based on feature representation learning using various deep networks", The ACCV Workshop on Driver Drowsiness Detection from Video 2016, 2016.

[6]   Y. Xie, K. Chen, and Y. L. Murphey, "Real-time and Robust Driver Yawning Detection with Deep Neural Networks," Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018, pp. 532–538, 2019.

[7]   S. Mehta, S. Dadhich, S. Gumber, and A. Jadhav Bhatt, "Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio," SSRN Electronic Journal, pp. 1333–1339, 2019.

[8]   C. S. Wei, Y. T. Wang, C. T. Lin, and T. P. Jung, "Toward Drowsiness Detection Using Non-hair-Bearing EEG-Based Brain-Computer Interfaces," IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2018.

[9]   N. C. for Statistics and Analysis, "Crash Stats: Drowsy Driving 2015," October 2017. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812446

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)