Analysis of V-D based features for distracted driver monitoring system using deep learning.

Dr.Tejaswi Potluri
Assistant Professor,
VNR VJIET,
Hyderabad,
INDIA
Email:
tejaswi_p@vnrvjiet.in

Sarika Nyaramneni
Assistant Professor,
VNR VJIET
Hyderabad,
INDIA
Email:
sarika_n@vnrvjiet.in

Jahnavi Somavarapu,
Assistant Professor,
VNR VJIET,
Hyderabad,
INDIA

Dr.M.Ravikanth
Assistant Professor,
VNR VJIET,
Hyderabad,
INDIA

ABSTRACT

According to the survey, 'Distracted Driving in India: a study mobile phone usage, pattern & behavior' by SaveLIFE Foundation, details that, in the last decade alone, India lost 1.3 million people to road cashes and many are injured. India has the highest number of road crash fatalities, while it is just 1% of the world's vehicles, India accounts for over 10% of global road crash fatalities. The most frightening thing is that this is only one of the many ways a driver is being distracted like drowsiness, procrastination, eating, drinking, talking to the passengers, etc. Advanced safety technology called a Driver Distraction Detection System (DDDS) is used to spot distracted drivers and warn them. This system tracks the driver's actions and surroundings using a variety of sensors, including cameras and accelerometers. To detect whether the driver is distracted, such as by texting or using a mobile phone, eating or drinking, or nodding off, the data is then analyzed by a sophisticated algorithm. The DDDS prompts the driver when they need to focus again, which can greatly lower the likelihood of accidents brought on by distracted driving. Most driver monitoring systems already embedded in vehicles to detect distraction use vehicle-based features such as ADAS(advanced driver assistance systems) .There are other DMS models based on driver monitoring camera. Owing to the cost of ADAS systems, they are available only in premium cars. Our proposed solution is not a substitute for a commercial ADAS system but a step towards low-cost driver safety options. We are not only considering vehicle-based features but also monitoring the driver's behavior using Convolutional Neural Network (CNN) model. Because of its modularity, the system should be easy to fit in any car at an affordable price

Keywords—Driver Distraction Detection System, ADAS, Convolutional Neural Network.

I. INTRODUCTION

Driving is a complex task that requires a high level of attention and focus. However, many drivers are easily distracted by various factors, such as their mobile phones, music, food, or fatigue, leading to a higher risk of accidents on the roads. To address this issue, Driver Distraction Detection System (DDDS) technology has been developed to monitor drivers' behavior and alert them when they become distracted.

One approach to implementing DDDS technology is through the use of both driver monitoring and lane departure detection models. Driver monitoring involves the use of sensors, such as cameras and infrared sensors, to track the driver's face and eyes for signs of fatigue, distraction, or impairment. Lane departure detection, on the other hand, uses cameras or other sensors to detect when a vehicle crosses a lane marker or drifts out of its lane without the driver indicating their intention to do so.

By combining these two models, a DDDS can provide a more comprehensive and accurate assessment of the driver's attention level and overall safety on the road. If the driver's behavior or actions indicate a high level of distraction or drowsiness, the DDDS can provide an immediate alert to the driver to help them refocus their attention and prevent an accident.

The project aims to analyze the effectiveness of vehicle and driver-related (V-D) features in detecting distracted drivers using deep learning. The V-D features will be extracted from a driver's visual appearance and driving behavior. These features will then be used as inputs to a deep learning model to classify the driver's state as focused or distracted.

The objective is to develop a robust and accurate distracted driver monitoring system that can help reduce the risk of road accidents caused by driver distraction. The use of both driver monitoring and lane departure detection models in a DDDS can significantly improve road safety and reduce the risk of accidents caused by distracted or fatigued driving. This technology is an important step forward in the development of intelligent transportation systems that prioritize safety and efficiency on our roads.

II. LITERATURE SURVEY

Vehicle and lane-based distraction detection systems have become increasingly important in recent years due to the growing number of accidents caused by distracted driving. Numerous studies have been conducted to develop effective methods for detecting distractions in vehicles and on the road. In this literature review, we will discuss some of the recent research in this field.

One of the most commonly used methods for detecting distraction in vehicles is analyzing driving behavior. Several studies have shown that changes in driving behavior, such as lane deviation and sudden braking, can indicate distracted driving. For instance, in a study conducted by Lee and colleagues (2019), it was found that distracted driving could be detected by analyzing the vehicle's acceleration and deceleration patterns, steering wheel movements, and lane deviation.

Another method used for distraction detection is the use of physiological sensors. These sensors can detect changes in the driver's physiological state, such as heart rate and skin conductance, which can be indicators of distraction. For instance, a study by Cunha and colleagues (2018) demonstrated that physiological sensors, such as electroencephalography (EEG) and electrocardiography (ECG), could accurately detect driver distraction.

In recent years, the use of machine learning algorithms has become increasingly popular for detecting distractions in vehicles and on the road. These algorithms can analyze large amounts of data and identify patterns that are indicative of distraction. For instance, a study by Yan and colleagues (2020) used machine learning algorithms to analyze eye-tracking data to detect distracted driving.

Lane-based distraction detection systems have also been developed to detect distractions on the road. These systems use cameras and sensors to monitor the driver's lane position and detect lane deviation. For instance, a study by Zhang and colleagues (2020) developed a lane-based distraction detection system that could accurately detect distraction by monitoring the driver's lane position.

“A Fuzzy-Logic Approach to Dynamic Bayesian Severity Level Classification of Driver Distraction Using Image Recognition -2020 [1]” : In a fuzzy logic-based system for driver distraction detection, the inputs are the various sensor data collected from the driver and the vehicle, such as eye gaze, head position, vehicle speed, etc.. The output of the system is a degree of distraction represented by a fuzzy membership value ranging from 0 to 1, with 1 being fully distracted and 0 being fully focused.

“ Detecting Human Driver Inattentive and Aggressive Driving Behavior Using Deep Learning: Recent Advances, Requirements and Open Challenges-2020 [2]” : This paper helped us know the methods of categorizing the distraction detection techniques using CNN ,DBN and RNN algorithms .

“ Selection of Measurement Method for Detection of Driver Visual Cognitive Distraction : A Review [3]” :eye gaze tracking or head position tracking . vehicle speed or lane departure .External or internal distractions.

“ An Efficient Deep Learning Framework for Distracted Driver Detection [4] ”: The proposed model contains two steps. we detect the objects involved in these distracting activities and the ROI of the body parts from the dataset’s images. Pre-Processing of Dataset .We use preprocessed and annotated data to train the Efficient Set model.

“ A Lane Tracking Method Based on Progressive Probabilistic Hough Transform [5] ”:pre-processing, adaptive ROI setting and lane markings detection and tracking using PPHT. Input image to the AROI area PPHT is used to extract the straightest lines in the AROI. Classical K-means clustering algorithm to fix the road lane.two lines (right and left side) only from the lines provided by PPHT with specific angles and lengths.

In [8] application's functionality is compromised when eyes are too close or too far from the camera. However, the application's design and performance align ideally when the eyes are around 50 cm away. This application aligns with the established theory that individuals experiencing drowsiness tend to exhibit an average eye closure time exceeding 400ms, while the eye closure time for alert individuals is typically below this threshold.

Authors of [9] has effectively tackled the issue of driver distraction, covering both visual and cognitive aspects. Through the utilization of real-world driving data, we executed a comprehensive assessment, leveraging external observers to individually evaluate the cognitive and visual distractions experienced by drivers. These metrics of distraction underwent rigorous validation, showcasing strong agreement between different evaluators across both dimensions of distraction. With the evaluated distraction levels serving as dependent variables, they proceeded to develop and train regularized regression models. These models were designed to accurately forecast the perceived degree of distraction faced by drivers. This approach enabled them to better understand and anticipate the impact of various factors on drivers' levels of distraction, ultimately contributing to enhanced safety measures.

The study [10] hinges on a driver's attention-monitoring system, which operates through computer vision. This system diligently observes drivers for signs of frequently shifting gaze, nodding of the head, and instances of eye closure. [11] introduced a robust architecture designed to detect and mitigate driver distractions, encompassing diverse computer vision models and an NLP model. The computer vision models are deployed for tasks such as detecting the driver's head pose and classifying driver actions. The NLP model encompasses trigger word activation, speech recognition, text classification, and entity recognition.

[12] conducted an examination of classification models using feature extraction through convolutional neural networks (CNNs). The SqueezeNet CNN architecture was employed and trained using the transfer learning technique, with image features extracted prior to the classification layer. The classification process involved using the acquired features as input for machine learning algorithms such as k-nearest neighbor (k-NN), support vector machine (SVM), and random forest (RF). [13][15] aimed to create a real-time distracted driver monitoring system that operates across various operating systems and employs deep learning techniques. A convolutional neural network (CNN) is utilized for tasks such as detection, feature extraction, image classification, and generating alerts. The training process of the system involves utilizing both publicly accessible and privately collected data. [14] presents the introduction of a comprehensive deep unsupervised multi-modal fusion network, referred to as UMMFN. The architecture comprises three key components: multi-modal representation learning, multi-scale feature fusion, and unsupervised driver distraction detection. Initially, three embedding subnetworks are constructed to extract low-dimensional features from various sensors. To effectively capture temporal dependencies for each modality and spatio-temporal dependencies across different modalities, we propose a multi-scale feature fusion approach. Lastly, an unsupervised driver distraction detection task is performed using a ConvLSTM Encoder-Decoder model. The UMMFN, which encompasses these three modules, is trained in an end-to-end manner. Authors of[16] conducted an examination to identify the most effective machine learning (ML) methods for detecting different driving distractions, considering the utilization of various sensors and data-capture techniques, specifically focusing on physiological and video camera-based sensors. The statistical analysis revealed that the choice of the most informative feature or modality for detecting driver distraction depends on the specific type of distraction. Overall, video-based modalities proved to be the most informative, with classical ML classifiers achieving high performance using one of these modalities. Conversely, deep learning (DL) classifiers require a greater number of modalities, either all or a pre-selected subset, for constructing effective classifiers.

Overall the literature review helps in classifying the distractions as follows. Visual distractions require you to look away from the road, even if only briefly. As an illustration, consider looking to your phone, GPS, or dashboard controls. Visual distractions pose a special risk to drivers because they may cause them to miss crucial traffic signals or other potential hazards. Manual distractions include removing your hands from the wheel, even for a little period of time. Examples include reaching inside the vehicle, adjusting the radio, or eating while operating a motor vehicle. Manual distractions can be risky because they make it harder for drivers to respond quickly to unforeseen circumstances. Your mind is taken away from the task of driving when you are distracted cognitively. One can daydream, talk on the phone, or even get into a furious argument with another passenger.

III. SYSTEM ARCHITECTURE

The architecture for a Driver Distraction Detection System (DDDS) that uses both Driver Monitoring System(DMS) and Lane Departure Warning System (LDWS) can be divided into three main components:

1. **Data Acquisition:** This component is responsible for collecting data from the DMS and LDWS systems. The DMS system captures data from the driver, such as eye gaze, facial expressions, head position, and vital signs. The LDWS system captures data about the vehicle's position on the road, such as lane markings and speed.

2. **Data Processing:** This component is responsible for processing the data collected by the DMS and LDWS systems. The data is analyzed to detect signs of distraction or drowsiness in the driver, and to detect when the vehicle is drifting out of its lane. The processing component uses machine learning algorithms to analyze the data and make real-time decisions about whether to issue an alert to the driver.

3. **Alert Generation:** This component is responsible for generating alerts to the driver when signs of distraction or drowsiness are detected, or when the vehicle is drifting out of its lane. The alert can be in the form of a visual or audible warning, such as a flashing light or a beeping sound, or it can be haptic feedback, such as a vibration in the steering wheel. The alert is designed to bring the driver's attention back to the road and prevent an accident.

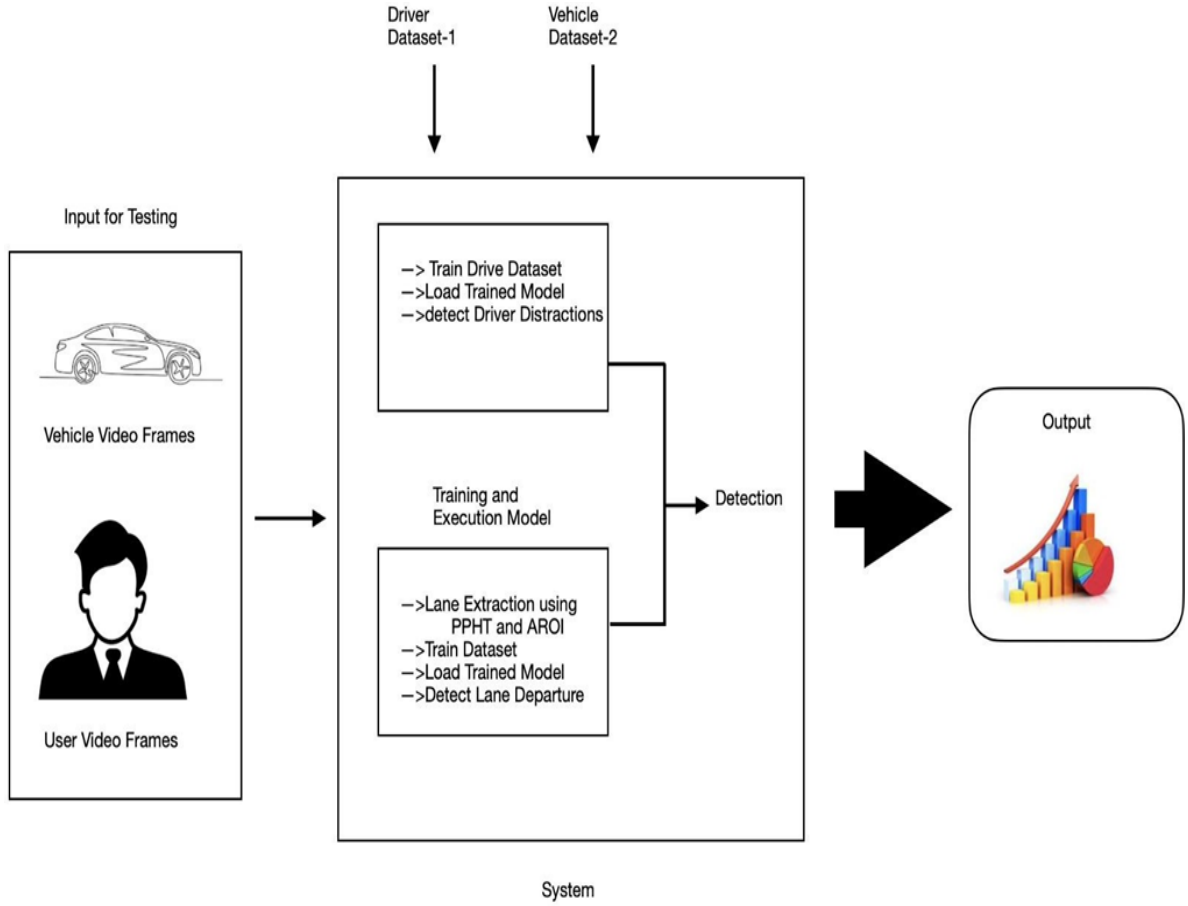

Figure 1: System Architecture of the Methodology.

Overall, the DDDS architecture is designed to integrate the data from the DMS and LDWS systems and use it to make real-time decisions about whether to issue an alert to the driver. The system is designed to be reliable, efficient, and easy to use, and to minimize the risk of accidents caused by driver distraction or lane departure. The architecture can be customized to suit the needs of different vehicles and driving conditions, and can be extended to include additional safety features, such as collision avoidance or adaptive cruise controls.

IV. METHODOLOGY

Authors understood that the existing systems are derived either by monitoring the facial, behavioral patterns of the driver or the vehicular behavior which wouldn't be completely useful leaving few areas of concern.

For example, in [6] Developing and implementing computer vision technology for driver drowsiness detection can be technically challenging, as it requires sophisticated algorithms and processing power to accurately analyze the driver's facial expressions and body posture. The use of cameras in vehicles raises privacy concerns, as the technology can potentially record and store images of the driver and passengers. The technology may generate false alarms if it fails to accurately detect the driver's level of drowsiness. This can be a distraction for the driver and reduce their confidence in the system.

The accuracy of computer vision-based drowsiness [7] detection can be impacted by lighting conditions, such as bright sunlight or darkness, which can make it difficult for the cameras to accurately capture the driver's facial features.

In [1] The technology may not be 100% accurate, leading to false alerts or missed warnings. The development and implementation of an adaptive forward collision warning system can be expensive. The use of cameras and sensors to monitor driver behaviour may raise privacy concerns for some individuals. Some drivers may resist the use of the technology, viewing it as intrusive or an infringement on their personal freedom. The system may require specialized knowledge to install, operate, and maintain, leading to difficulties for some users. There may be legal and ethical considerations.

In [2] A multi-model system can be more complex and difficult to develop and maintain compared to a single-model approach. Running multiple models can require more computational resources and time compared to a single-model approach. The output of a multi-model system can be difficult to interpret and understand, making it challenging to diagnose problems and improve the system.

In [5] Training and deploying deep learning models can be computationally expensive and require high-performance computing resources. Deep learning models require large amounts of annotated data to train, and collecting and annotating such data can be time-consuming and expensive. Deep learning models can be difficult to interpret and understand, making it challenging to diagnose problems and improve the model. Deep learning models can be vulnerable to adversarial attacks, where small, imperceptible changes to the input data can cause the model to make incorrect predictions.

These problems are the main reasons we wanted to use two different models and integrate them together with the help of a function which would help to predict distractions accurately. Authors developed two models, one with the help of driver monitoring and the other focusing on lane departure detection. Authors used two different sequential models for both the distracted driver detection using driver monitoring and lane departure detection. These models require images as input to be tested (frames).

Authors used predefined datasets from Kaggle for training both the models. We need to give two images to the model to check if the driver is distracted, one for each model. We check if the driver is distracted through driver monitoring, if the result is that he is distracted then we display a message on that kind of distraction. If the model predicts safe driving that's when the second model comes into play, and it tests the images and displays if the driver is distracted or not.

A. Data Sets:

Authors identified two datasets, one for each model. For the driver monitoring based model the dataset used is taken from kaggle-"https://www.kaggle.com/competitions/state-farm-distracted-driver-detection/data" and the dataset used for the lane departure detection model is taken from Kaggle as well-"https://www.kaggle.com/datasets/kpgeek/kitti-roadlane-detection-dataset-224-x-224".

B. Cleaning:

Data cleaning is the process of preparing data for analysis by weeding out information that is irrelevant or incorrect. This is generally data that can have a negative impact on the model or algorithm it is fed into by reinforcing a wrong notion. Data cleaning not only refers to removing chunks of unnecessary data, but it's also often associated with fixing incorrect information within the train-validation-test dataset and reducing duplicates.

Here are some key takeaways on the best practices you can employ for data cleaning:

1. Identify and drop duplicates and redundant data.
2. Detect and remove inconsistencies in data by validating known factors.
3. Maintain strict data quality measures while importing new data.
4. Fix typos and fill in missing regions with efficient and accurate algorithms.

C. Models:

A machine learning model is defined as a mathematical representation of the output of the training process. Machine learning is the study of different algorithms that can improve automatically through experience & old data and build the model. A machine learning model is like computer software designed to recognize patterns or behaviors based on previous experience or data. The learning algorithm discovers patterns within the training data, and it outputs an ML model which captures these patterns and makes predictions on new data.

D. Training and Testing:

Machine Learning is one of the booming technologies across the world that enables computers/machines to turn a huge amount of data into predictions. However, these predictions highly depend on the quality of the data, and if we are not using the right data for our model, then it will not generate the expected result. In machine learning projects, we generally divide the original dataset into training data and test data. We train our model over a subset of the original dataset, i.e., the training dataset, and then evaluate whether it can generalize well to the new or unseen dataset or test set. Therefore, training and test datasets are the two key concepts of machine learning, where the training dataset is used to fit the model, and the test dataset is used to evaluate the model.

V. FUNCTIONALITIES

A. **Importing Data**

- Import CSV file.

- Import Zip file.
- Files containing data sent by the driver.

B. **Preprocessing Data**

- Removing inconsistencies in data like null values and outliers.
- Dealing with categorical data.
- Scaling images.
- Blurring images for privacy issues.

C. **Choosing Model**

- Test each model and choose best model according to accuracies.

D. **Detect Fraud Case**

- Data which is out of pattern should be considered as the fraud, return 0 if it is legitimate case ,1 if it is fraud case.

VI. RESULTS AND DISCUSSION

A. **Dataset**

Dataset is used to train models. The model learns from this data set and gives the outputs based on the learning during the testing. We have two datasets 1) Distracted driver dataset 2) Lane dataset.

Convert validation and train data from the paths into several batches. Authors added several layers to the sequential model with the help of the add() method. The first layer is a flatten layer, which is used to reshape the input data into a one-dimensional array, while specifying the input shape to be (256,256,3). This is commonly used to transform the 2D images to 1D vectors. The second layer is a Dense layer with 512 neurons, using the rectified linear unit (ReLU) activation function. This layer is intended to learn complex feature representations of the flattened input. The third layer is another dense layer with 256 units and ReLU activation. This layer is also used to learn additional features and increase the capacity of the model to represent complex patterns in the data. The fourth layer is a dense layer with 128 units and ReLU activation. This layer further reduces the dimensionality of the learned features while still preserving important information. The final layer is a dense layer with 10 units and softmax activation. This layer is used for multi-class classification, where the model predicts the probability distribution over 10 classes(c0-c9). The softmax function ensures that the predicted probabilities sum to one, making it a suitable output layer for classification tasks. Train the distraction detection model using driver monitoring as shown in the figure 2.

```
[ ] history_dense = model_dense.fit(dense_train ,epochs=10,validation_data=dense_val)

Epoch 1/10
141/141 [==============================] - 280s 2s/step - loss: 6.5522 - accuracy: 0.3370 - val_loss: 1.0234 - val_accuracy: 0.6922
Epoch 2/10
141/141 [==============================] - 273s 2s/step - loss: 0.7999 - accuracy: 0.7569 - val_loss: 0.6095 - val_accuracy: 0.8207
Epoch 3/10
141/141 [==============================] - 283s 2s/step - loss: 0.4848 - accuracy: 0.8710 - val_loss: 0.3895 - val_accuracy: 0.8993
Epoch 4/10
141/141 [==============================] - 278s 2s/step - loss: 0.2640 - accuracy: 0.9457 - val_loss: 0.2098 - val_accuracy: 0.9646
Epoch 5/10
141/141 [==============================] - 287s 2s/step - loss: 0.1925 - accuracy: 0.9632 - val_loss: 0.2733 - val_accuracy: 0.9269
Epoch 6/10
141/141 [==============================] - 281s 2s/step - loss: 0.1527 - accuracy: 0.9669 - val_loss: 0.1755 - val_accuracy: 0.9619
Epoch 7/10
141/141 [==============================] - 287s 2s/step - loss: 0.0967 - accuracy: 0.9827 - val_loss: 0.1747 - val_accuracy: 0.9552
Epoch 8/10
141/141 [==============================] - 277s 2s/step - loss: 0.0780 - accuracy: 0.9861 - val_loss: 0.1217 - val_accuracy: 0.9706
Epoch 9/10
141/141 [==============================] - 285s 2s/step - loss: 0.0645 - accuracy: 0.9891 - val_loss: 0.1306 - val_accuracy: 0.9630
Epoch 10/10
141/141 [==============================] - 279s 2s/step - loss: 0.0667 - accuracy: 0.9865 - val_loss: 0.0764 - val_accuracy: 0.9829
```

Figure 2: Training Model

Test the model with the help of the test folder as shown in the figure 3.

```
[ ] test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)

    test = test_datagen.flow_from_directory('/content',
                                             classes=['test'],
                                             target_size=(256, 256),
                                             batch_size = 20,
                                             class_mode = None,
                                             shuffle = False)

    Found 79726 images belonging to 1 classes.
```

Figure 3: Create batches from the test dataset.

Python function load_dataset takes two arguments: train_path and test_path, which are the paths to the directories containing the training and testing images respectively. It loads the image files and their corresponding labels from the directories and returns four lists: train_images which contains the file paths to the training images, train_labels which contains the corresponding labels for the training images, test_images which contains the file paths to the testing images, and test_labels which contains the corresponding labels for the testing images.

The function first initializes empty lists for the training and testing images and labels. It then iterates through the subdirectories in train_path and test_path and assigns labels to each subdirectory based on the starting characters of their names. If the subdirectory starts with 'um', the label is 0; if it starts with 'umm', the label is 1; and for any other subdirectory, the label is 2. Next, for each image file in each subdirectory, it checks if the file has a '.png' extension, and if so, it adds the file path to the corresponding list of image paths (train_images or test_images) and the corresponding label to the list of labels (train_labels or test_labels). The graph in figure 4 represents the relationship between accuracy and epoch number. The accuracy increases with the increase in epoch number and the error would be very less.

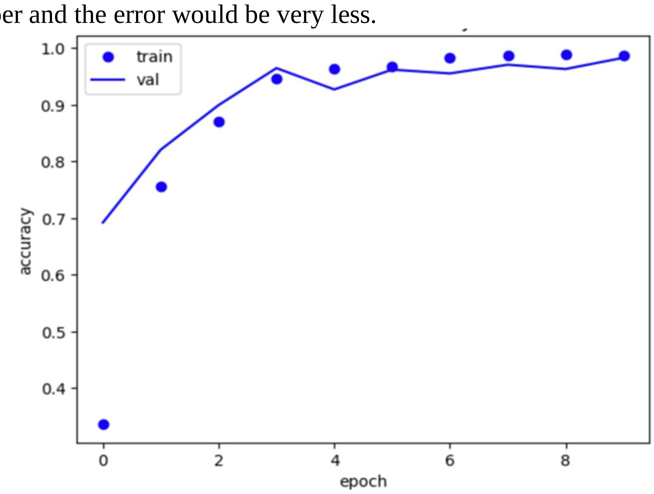

Figure 4: Accuracy versus Epochs.

The graph in figure 5 represents the relationship between loss and epoch numbers which depict the amount of loss during a particular epoch. As the number of epochs increase the loss gets reduced and eventually be negligible.

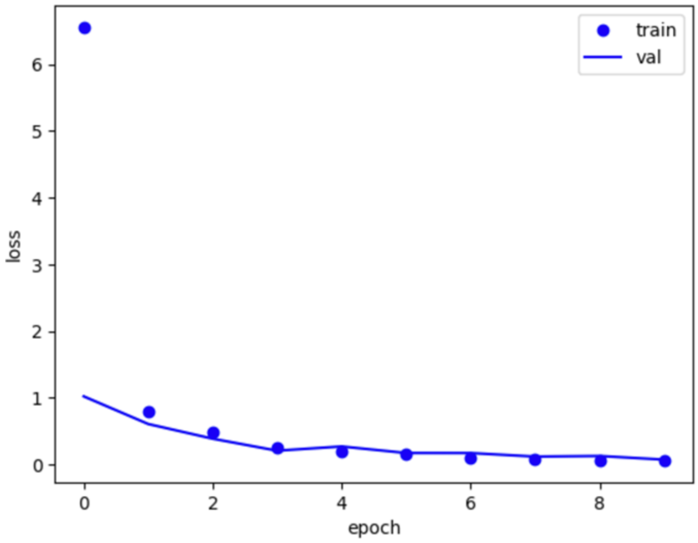

Figure 5: Loss vs Epoch.

The figure 6 helps us depict the way to test two images and predict if the driver is distracted at that instance of time . It is done by passing the images to the res() method which is the combination of the two models , Driver monitoring and lane departure monitoring thus predicting the distraction faced by the driver.

```
[ ] img1="/content/test/img_100300.jpg"
    img2="/content/data_road_224/testing/image_2/um_000008.png"
    res(img1,img2)

    Distraction detected : ! Talking to passenger

[ ] img1="/content/test/img_10.jpg"
    img2="/content/data_road_224/testing/image_2/um_000008.png"
    res(img1,img2)

    Distraction detected : ! operating the radio

[ ] img1="/content/test/img_100008.jpg"
    img2="/content/data_road_224/testing/image_2/um_000008.png"
    res(img1,img2)

    Safe Driving
```

Figure 6: Results.

Figure 7 shows the accuracy of the model. Thus, the two models, driver monitoring and lane departure detection can be merged with a function called res and distraction can be predicted.

```
[ ]  acc1=history_dense.history['accuracy'][-1]
     vacc1=history_dense.history['val_accuracy'][-1]
     loss1=history_dense.history['loss'][-2]
     vloss1=history_dense.history['val_loss'][-2]
```

```
[ ]  print("Accuracy : ",acc1)
     print("validation accuracy : ",vacc1)
     print("Loss : ",loss1)
     print("Validation Loss : ",vloss1)
```

```
     Accuracy :  0.9864503145217896
     validation accuracy :  0.9828507900238037
     Loss :  0.06449799984693527
     Validation Loss :  0.1306404322385788
```

Figure 7: Accuracy and Loss.

VI. CONCLUSION AND FUTURE WORK

A driver distract detection system is an important technology that can help improve road safety by detecting and alerting drivers when they become distracted while driving. The system uses two cameras to detect the distractions faced by the drivers and alert them so that it wouldn't lead to accidents. It is done with the help of monitoring both the driver and the road. The driver distraction detection model helps in identifying the inner disturbances or distractions a driver face and alerts the driver whereas the lane departure detection system helps in alerting the driver if we deviate the lane more than a certain threshold, thus helping to alert the driver which helps them to avoid accidents to a great margin.

By collecting data on driver behavior and distraction, these systems can provide valuable insights into the causes and effects of distraction and help to inform the development of new and improved methods for detecting and mitigating it. In some cases, these systems may be used to enforce laws and regulations that prohibit certain types of distraction, such as using a phone while driving. We can improve efficiency with the help of combining several models using ensemble learning. We can improve the privacy and security measures as we are directly using images to detect if they are distracted. We look forward to integration of driver distraction detection model with other safety systems, such as collision warning to provide a comprehensive safety package that can help prevent accidents and reduce the severity of crashes.

REFERENCES

[1] Adebamigbe Fasanmade; Ying He; Ali H. AlBayatti; Jarrad Neil Morden "A Fuzzy-Logic Approach to Dynamic Bayesian Severity Level Classification of Driver Distraction Using Image Recognition"

[2] Monagi H. Alkinani; Wazir Zada Khan; Quratulain Arshad "Detecting Human Driver Inattentive and Aggressive Driving Behaviour Using Deep Learning: Recent Advances, Requirements and Open Challenges"

[3] Norhasliza M. Yusoff; Rana Fayyaz Ahmad; Christophe Guillet; Aamir Saeed Malik; Naufal M. Saad "Selection of Measurement Method for Detection of Driver Visual Cognitive Distraction: A Review"

[4] Faiqa Sajid; Abdul Rehman Javed; Asma Basharat; Natalia Kryvinska. "An Efficient Deep Learning Framework for Distracted Driver Detection"

[5] Mehrez Marzougui; Areej Alasiry; Yassin Kortli; Jamel Baili."A Lane Tracking Method Based on Progressive Probabilistic Hough Transform"

[6] Mahek Jain, Bhavya Bhagerathi, Sowmyarani C N. "Real-Time Driver Drowsiness Detection using Computer Vision"

[7] Prof. Swati Gade1, Kshitija Kamble2, Aishwarya Sheth3, Sakshi Patil4, Siddhi Potdar "Driver Drowsiness Detection Using Machine Learning"

[8] Widodo Budiharto, Welly Dwi Putra, Yudha Pratama Putra "Design and Analysis of Fast Driver's Fatigue Estimation and Drowsiness Detection System using Android"

[9] Li, Nanxiang & Busso, Carlos. (2015). Predicting Perceived Visual and Cognitive Distractions of Drivers With Multimodal Features. Intelligent Transportation Systems, IEEE Transactions on. 16. 51-65. 10.1109/TITS.2014.2324414.

[10] L. Zhiqiang, W. Peng and Z. Jingjing, "The Study of Driver Distraction Characteristic Detection Technology," 2010 International Forum on Information Technology and Applications, Kunming, China, 2010, pp. 188-191, doi: 10.1109/IFITA.2010.296.

[11] A. AbouOuf, I. Sobh, M. Nasser, O. Alsaqa, O. Elezaby and J. F. W. Zaki, "Multimodel System for Driver Distraction Detection

and Elimination," in *IEEE Access*, vol. 10, pp. 72458-72469, 2022, doi: 10.1109/ACCESS.2022.3188715.

[12] Aldoori, Shafeeq & Taspinar, Yavuz & Koklu, Murat. (2021). Distracted Driving Detection with Machine Learning Methods by CNN Based Feature Extraction. International Journal of Applied Mathematics Electronics and Computers. 9. 116-121. 10.18100/ijamec.1035749.

[13] Hanafi, Mohamad & Nasir, Mohammad & Wani, Sharyar & Abdulghafor, Rawad & Gulzar, Yonis & Hamid, Yasir. (2021). A Real Time Deep Learning Based Driver Monitoring System. International Journal on Perceptive and Cognitive Computing. 7. 79.

[14] Y. Zhang, Y. Chen, and C. Gao, ‘‘Deep unsupervised multi-modal fusion network for detecting driver distraction,’’ Neurocomputing, vol. 421,pp. 26–38, Jan. 2021.

[15] Mcdonald, Anthony & Ferris, Thomas & Wiener, Tyler. (2019). Classification of Driver Distraction: A Comprehensive Analysis of Feature Generation, Machine Learning, and Input Measures. Human Factors The Journal of the Human Factors and Ergonomics Society. 10.1177/0018720819856454.

[16] M. Gjoreski, M. Ž. Gams, M. Luštrek, P. Genc, J. -U. Garbas and T. Hassan, "Machine Learning and End-to-End Deep Learning for Monitoring Driver Distractions From Physiological and Visual Signals," in *IEEE Access*, vol. 8, pp. 70590-70603, 2020, doi: 10.1109/ACCESS.2020.2986810.