A Deep Neural Network Approach for The Detection of Driver Distraction: USA

MSc Research Project
Data Analytics

Sai Chethan Singu
Student ID: x18181937

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sai Chethan Singu
Student ID:	x18181937
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	17/12/2020
Project Title:	**A Deep Neural Network Approach for The Detection of Driver Distraction: USA**
Word Count:	9252
Page Count:	25

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th December 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Deep Neural Network Approach for The Detection of Driver Distraction: USA

Sai Chethan Singu
x18181937

Abstract

The increase in the number of accidents has been raising for the last two decades with the increase in the number of vehicles and the people traveling on roads. One of the major reasons for these accidents is due to the distraction of the driver with different activities as part of their driving. That is drivers being involved in different activities has increased the probability of accidents. In order to address the problem, an alert system that could detect this distraction will be useful. The existing works that are involved in this detection are studied and the drawbacks of these techniques are analyzed. With the research gap that is found, a neural network model has been implemented to detect these activities of the driver. The dataset that is considered to complete the empirical analysis of the approach is the State Farm dataset. Several deep learning models were implemented i.e Vanilla CNN, Xception , Mobile Net, ResNet50 and VGG16. Each of these models were implemented and analyzed for its performance in detecting distracted driver activities. The optimized vanilla neural network has outperformed among other models with an accuracy of 97.7%. As part of the work, future directions are also given alongside the review of the findings collected.

Keywords: Distracted Driver Detection, ResNet50, Vanilla CNN, VGG16, Xception.

1 Introduction

The number of accidents that are occurring on roads is increasing for the past two decades. The survey conducted by the World Health Organization states that there has been an increase of 57 percent in the number of accidents that occurred in the past 10 years Baheti et al. (2018). This increase in the number of accidents has raised the number of people getting injured and losing lives due to these accidents. According to the National Traffic monitoring administrator, the reason behind these accidents includes distracted drivers and the loss of attention of the drivers while they are driving the vehicles. From the statistics that they have derived on finding the reason behind the rise in the number of deaths and injuries due to these road accidents is distracted driving. About 63% of the accidents that occur are only due to driver distraction Tran et al. (2018).

1.1 Background and Motivation

The research on driver distraction detection and alerting for the same has been traditionally carried out using computer vision techniques. These involve image processing and video processing, through which the drivers are detected whenever they are involved in different activities while driving. With the increase in the number of accidents and the level of distraction or the frequency of distraction the drivers of the vehicles are having, the research interest towards this problem has increased. This research on driver distraction would add a lot of benefits to the drivers and the people traveling in the vehicles to have a safe journey. That is the stakeholders of this application will be a large group, involving all the users of transportation involving road transport. With the introduction of technological advances like Artificial Intelligence (AI), researchers started, making use of them to solve the problem of driver distraction. The only solution to this problem is to make the driver alert once they start distracting and notify them to concentrate on driving activity. Finally, the technique that is being used to alert the drivers should be feasible both computationally and in terms of accuracy, so that the number of accidents can be reduced saving people from losing lives and getting injured during the accidents.

1.2 Research Question

A computationally efficient and accurate system that alerts drivers when they are distracted from driving activity is essential to reduce the number of accidents. There are existing techniques that have been devised to alert the driver, but they have their own drawbacks in completing the objectives efficiently and effectively. A deep survey of all these existing techniques will be carried out as part of the work and understand the pros and cons of these techniques. Once these are understood, the primary objective of the project is to develop a deep learning-based distracted driver detection that incorporates convolution neural network models.

RQ : *"To what extend can deep learning techniques using dense neural networks (Vanilla Neural network, Xception model, MobileNet, ResNet50, VGG16) enhance detection of distracted drivers?"* The performance of the models will be compared using the performance metrics.
Sub RQ : Can further tuning of the parameters, improve the metrics (Accuracy and Loss)?

1.3 Research objectives and contributions

The research question that is formulated is investigated and objectives through which the research question can be achieved are formulated as depicted in Table 1. The major idea of the project is to be able to detect the distracted driver whenever they perform any other activity other than driving. In short, the problem can be considered as a classification problem in which one of the classes is whether the driver is concentrating on driving activity. And the other classes can be considered as the activities that the drivers are performing apart from driving. So, the objective is making use of the images of the driver that are captured by the camera that fits in the vehicle and make use of them to detect the activity of the driver and classify them according to their features. If the driver is found to be distracted, the driver needs to be alerted. Overall, the application of the method or technique devised in this work is to facilitate the alert system to be computationally efficient along with good accuracy while alerting the driver.

Table 1: Summary of the project objectives and description

ID	Objective	Description	Evaluation
1	Literature Review	Collect and carry out critical evaluation of peer-reviewed papers on detecting the distracted driver.	Critical review and evaluation of existing works.
2	Data collection and data pre-processing	Collecting the suitable benchmark or the dataset that is apt. Pre-process the data for further steps including data cleaning and data visualization.	Exploration of the data collected and preliminary analysis of the data.
3	Construction of neural network by preparing and loading weights	Constructing the convolutional neural network with the preparation of weights for the model to classify and load the weight to the model	
4	Training the model	Prepare training dataset and provide the same to the model for training it.	
5	Implementation of deep learning classification neural networks models		
5.1	Implementation of Vanilla CNN model	Three approaches namely Vanilla CNN, optimized Vanilla CNN and Vanilla CNN with data augmentation.	
5.2	Implementation of Xception model		
5.3	Implementation of MobileNet model		
5.4	Implementation of Resnet50 model		
5.5	Implementation of VGG16 model		
5.6	Enhancing the model based on loss functions	Parameter tuning which involves weight adjusting or the other hyper parameters.	Evaluate the model loss and accuracy after tuning.
5.7	Testing the model	Provide validation and test dataset for testing the model.	
6	Evaluation and Results Analysis	Analyse the results and compare them with the state-of-the-art approaches.	Accuracy and Loss.

The remaining of the report is organized as follows. The following chapter 2 consists of survey and the basic analysis of different works that have been done. Chapter 3 will give subtleties on the technique which incorporates the depiction of approach and methodology. Chapter 4 comprises the implementation of different techniques and is followed by Chapter 5 that presents the evaluation and results of the deep neural network models. Then chapter 6 gives a summary of the work that is done alongside the assessment of methodology and a future direction wherein the approach can be improved.

2 Literature Review

Distracted driver detection is one of the significant reasons behind the accidents and the loss of life and property. This has raised with the increase in the usage of vehicles and the level of transportation requirements by the people. One common solution suggested to prevent this problem of distraction is being attentive and avoiding all kind of secondary activities that are carried out parallelly while driving the vehicles. Several approaches were proposed by researchers to detect these activities and alert the drivers of the vehicle accordingly. The following sections provide details on the existing works along with the achievements and challenges that the approaches are facing.

2.1 Introduction

In this section, all the related works that are carried out towards detecting the distracted driver and alerting them are reviewed and described. Overall, these techniques involve computer vision which include video processing and image processing along with the artificial intelligence approaches involving machine learning and deep learning.

2.1.1 Computer Vision

Greenberg et al. (2003) worked towards the detection of the secondary events that are carried out by drivers while driving the vehicles. Different activities are being performed by adults to teenagers while driving the vehicle. These majorly involve the phone calls and the messages or talking to the co-passengers in the vehicle. All the other activities that are done apart from the major driving activity is considered as the secondary event. The work carried out by the authors has considered the event-driven approach through which these events are detected, and they have developed an alert system that could make use of the events that are detected. Along with distracted driver detection, the authors also considered the lane violation which is also a cause for the accidents that are occurring on the roads Liao et al. (2016). This made them more severe as the distraction caused the volition of the lane and made it even worse.

Liang et al. (2007) focused on detecting the distraction of the driver through the machine learning approaches. The major concentration that they have considered is to find out the exact approach in machine learning that could suit different approaches to detect the driver distractions. As numerous activities can involve the distraction of the driver, they need to be properly detected so that they will be able to identify whether the driver is distracted or not. To ensure the distraction of the driver, they have conducted their empirical analysis in the relative environment and they employed the support vector machine algorithm of machine learning to classify the activity of the driver as the distracting activity or the abnormal activity. The result obtained by the authors as part of this has made it clear that they have found the corner case detection where the head movement of the driver always vary from the actual posture or position. Finding head in different position has made them to extract the features and analysis was done using support vector machine. With this, they were able to achieve satisfactory results.

King et al. (2006) have carried out their research in detecting the distracted driver through a different approach than the existing computer vision or any other artificial intelligence approaches. They have made use of the Electroencephalography signals through which the body movements and the concentration of the driver towards the driving activity are achieved. This made it clear for them to understand that the visual conspiracies are not only involved along with the physical status that is carried out by the drivers is also taken into consideration while detecting their distraction.

2.1.2 Machine Learning

Kutila et al. (2007) have carried out the work in their research towards developing the machine learning approach, this will enable the model to make use of all the feature extraction carried out by the authors in detecting the distracted driver. This paper compares the results that they have obtained with the machine learning model against the approaches that involves the computer vision kind of approaches where the image features are extracted through OpenCV and then they are further used to extract the feature vectors that involved the detection of distracted driver. The machine learning model that the authors have proposed is the support vector machine, in this approach the authors considered the problem to be treated as the classification problem and made use of the classifier formulated using the support vector machine to detect the distracted driver. With this approach, the authors were able to an gain accuracy of 75% whereas the traditional or the state-of-the-art approach was able to pull up 60-68% accuracy in detecting the distracted driver.

Researchers carried out the work using machine learning approaches extensively in detecting the distracted driver apart from the computer vision which is the image processing and video processing approaches. Jo et al. (2011) in this work towards distracted driver detection has made use of machine learning approaches to solve the problem of distracted driver detection. This work focused on solving the problem by treating it as one of the classification problem. In this approach the classification of the driver is considered as the binary where the driver is attentive or the driver is distracted through the features that are extracted were considered and they made use of them to classify the images of the current event or activity performed by the driver into one of these Dong et al. (2010). The techniques used by Jo are component analysis and discriminant analysis to capture the exact features that the machine learning classification model needs to consider to classify the activity of the driver.

The detection of the drowsiness of the driver is considered as the problem Dwivedi et al. (2014). Back then the computer vision is one of the trending approaches to solve the problems that can be using images and videos. One among them is this distracted driver detection. In his work majorly the drowsiness of the driver is considered to be detected and it is one of the major distractions that contribute a large portion to the number of accidents occurring. The earlier approaches were majorly focused on identifying the rate at which the eyes are blinked and how the eyes of the driver wide open and all these are considered, but when these are considered as only features they have certainly made the model to overfit and did not provide the better results. So, in this work, the authors have taken the convolution neural network technique and made use of the facial features from the images and the frames in the video captured to detect the drowsiness of the driver. The results have proven that they got better performance than the sole computer vision, but the model has still room for improvement and enhancement.

Researchers started focusing on the facial features and the head pose or the gaze of the driver along with the focus on their eyes and how they are visually focusing on the driving activity are considered. The work by Vicente et al. (2015) has identified the major features from the face of the person to be extracted and determine the head position and the gaze of the driver while driving the vehicle. The common input for this method is also the images and the videos captured by the camera in the vehicle. The algorithms capture the pose of the driver's head and the expression of the person who is driving are considered as input and based on this model which is machine learning is formulated to detect the attention of the driver Yan et al. (2015). The attention of the driver is calculated based on these features extracted.

2.2 Analysis

Researchers have also made use of the hardware related inputs along with the computer vision-based approaches that were used to detect the distraction of drivers Fernández et al. (2016). These include the visually based sensors and the approaches that could extract and identify the features in making use of them towards the distracted driver detection. They have formulated the low-cost approach which does not need to include any extra processing capabilities beyond the ones that the vehicle already has, these involves an extra sensor and the camera along with a small processing chip that would run the technique that is formulated by taking the input from these sensors and the camera. There were several approaches taken by the researchers using machine and deep learning to detect the action of drivers while they are in driving. These were discussed in the following sections.

2.2.1 Machine Learning Approaches

The approach devised by Liao et al. (2016) has focused on the cognitive distraction where the vehicle being driven and the results that are focused on the driver distraction have impacted the way the vehicle is being driven. This is a completely different approach than the previous works as most of them focused on the visual distraction and the movements of the driver instead of how the vehicle is being moved or driven by the driver due to their distractions. The technique employed is based on the support vector machine through which the features of the driver are extracted and are compared with the regular activities and the features that are formulated during this time to ensure that they have better reach for the detection of driver distraction. Due to the cognitive distraction that the drivers have while driving the vehicles.

The work carried out by Baheti et al. (2018) has considered, convolution neural network as their approach to address distracted driver detection. The approach that they have considered will take up the vision as the input which includes the continuous feed of images which are the frames that are divided from the video that is taken by the cameras in the vehicle. The cameras that are set up will provide the necessary input and the system will take them and feed them to the network that is formulated for detection. The model is based on the modified VGG16 model. This model initially was taken as the pre-trained standard VGG16 model, but it did not perform well which made them move towards enhanced or the modified VGG16 model with batch normalization and regularization as the major changes. With all the changes made they were able to achieve an accuracy of 75%. But still, when the results are considered, there can be a lot of wrong detections which can lead to low performance.

Sanjay and the other researchers in Baheti et al. (2018) work had been considered as their base and the authors Jaco and Andries worked on the challenges that the approach of using a VGG16 and enhanced VGG16 model are facing. They have identified that overfitting of the model, with the data that they have for training the model, is one of the key drawbacks that the approach has. This is addressed by Andries and Jaco using data augmentation through which the data and the class augmentation is carried out from the extracted features has made it easy for the mode to carry out the extraction activity and compare the required features and make use of them in detecting the distracted driver and classify them Cronje and Engelbrecht (2017). This made the network that is being used more generalized and gave a network that is free from overfitting the data and the features extracted by the model.

Koesdwiady et al. (2017) in their work have made use of the deep learning approaches along with the machine learning techniques to detect the driver distraction, they have considered the approach is completely focused on the neural networks but just to make a comparison of the results that they have obtained they also took into consideration the machine learning approaches so that the analysis can be presented on the approach that they have provided and the one that is already existing. The deep learning model that the authors made use of is the VGG19 model where the model is tested in. different techniques that involved the standard VGG19 and the one that is enhanced by the authors to suit the input and other parameters that are required in detecting the distracted driver.

2.2.2 Deep Learning Approaches

The authors have taken into consideration different classes of the drivers and the activities that are carried out by the driver are considered from different classes and these will ensure that whether they are distracted or not based on the activity into which they are classified. Along with this, as they have conducted their empirical analysis through the machine learning approaches, they involve the XGboost model this is one of the classifiers where the decision trees as an ensemble model that is involved in the part of classification. From these techniques, the deep learning approach involving the VGG19 has provided an accuracy of 80% whereas the machine learning approach has reached an accuracy of 75% but the challenge is that they have overfitting in the learning that the models have carried out.

The work carried out by Munif and Bandar towards distracted driver detection is based on the deep learning approach. They have identified the challenges that the existing work has in detecting the distracted driver and based on these they addressed them using artificial intelligence techniques Xing et al. (2017). The input they have considered for the detection is the images that are captured by the camera that is set up in the vehicle Alotaibi and Alotaibi (2019). All the images that are considered are two-dimensional and these images are processed further by the neural network to extract the features and formulate a model to detect the distracted driver. The neural network that they have made use of is based on the inception model which is a pre-trained neural network with a residual block of layers. With the initial stand, they were able to achieve a very slow learning curve and they enhanced it further by making use of the recurrent neural network to improve the performance of the model.

Basubeit and along with two other researchers have focused on developing a network that is based on deep learning to detect the distracted driver. The technique is solely dependent on the convolution neural network. The major contribution from their work is the division of the distraction types. That is all the earlier work was focused on the categorization or classification of the distraction based on the events or the activities that the driver is carrying out which are from talking on the phone to texting and talking to others in the vehicle. But in this work they have classified the distraction based on the action involved by the driver where the activity can be manual, the activity could be visual where the eyes of the person who is driving are involved Baheti et al. (2018). Apart from the manual and the visual, there can also be a cognitive type of distraction where the activity that needs to be performed as part of driving is not done properly. To detect all these types of activities and alert the driver, a visual feed of the driver is taken as input and the technique formulated will consider this as input and based on it to classify the driver activity as one of the above-given types of distractions and if occurred it will alert the driver. The model they have made use of is VGG model through which they were able to achieve 86% accuracy in detecting the distracted driver.

2.3 Research Gap

The research conducted on the existing works that are carried out towards distracted driver detection has majorly focused on three different approaches. These approaches involve computer vision techniques, machine learning approaches, and deep learning approaches. Initially, the computer vision techniques where image processing and video processing are employed towards the detection of different activities that the driver have been performing while driving, they were able to detect the distracted drivers, but the problem with these approaches is they need a lot of processing and computing capabilities and this involved slower detection of the distraction and this is not suitable to a real-world problem. But still, this paves the path to the usage of machine learning approaches where the researchers have made use of the features that are extracted from the images to get the machine learning models trained to detect the distracted driver and classify them. They have focused on the distracted driver problem and solved them by considering it as a classification problem and this made the problem much simpler to have the machine learning models better accuracy. But still, the accuracy that the machine learning model has achieved is in the range of 60-75% which is not really good when a real-world scenario is considered as the problem with the distracted driver being detected as non-distracted would lead to severe loss Liu et al. (2015). When the deep learning models and the neural network in it are approached by the researchers to detect the distracted drivers, they have provided better results than the machine learning.

2.4 Comparision and Conclusion

Table 2: Comparison of Related work

Authors	**Approach**	**performance**	**Description**
Wollmer M. (2011)	Long Short-Term memory recurrent neural network	86.6% accuracy	The approach used is based on the recurrent neural networks which involved the long-short term memory.
Mogelmose et. al., (2012)	Computer vision-based approach	68% accuracy in detecting the distracted driver	The approach provided the survey of the existing techniques and the vision-based approach that employed video processing to detect the distracted driver.
Tango et. al., (2013)	SVM	69%	Made use of machine learning approaches compared with the computer vision techniques.
Das et. al., (2015)	Hand detection along with the machine learning approaches		Provided details on dataset and the details on existing approaches.
Mosa et. al., (2017)	Capacitive Sensors with convolution neural networks	82% accuracy	Hand detection through the capacitive sensors was the approach and the data captured through the sensors was used by the CNN to detect the distracted driver.
Sheng et. al., (2018)	VGG16, AlexNet, ResNet	GoogleNet gave better accuracy of 73%	Made use of three different deep neural networks to detect the distracted driver through the image captured by cameras.
Xing et. al., 2018	ResNet	81% accuracy	Made use of enhanced ResNet with hyper parameter tuning and modularised weights.
Basubeit et. al., 2019	VGG16	86% accuracy	Made use of enhanced VGG16 approach with batch normalization and regularization.

The above Table 2 outlines the comparison of the work previously done by the researchers with the methods incorporated towards the distracted driver detection. Based on the comparison of different methodologies, neural networks outperforms the machine learning and other techniques with further tuning of parameters.

3 Methodology

3.1 Approach

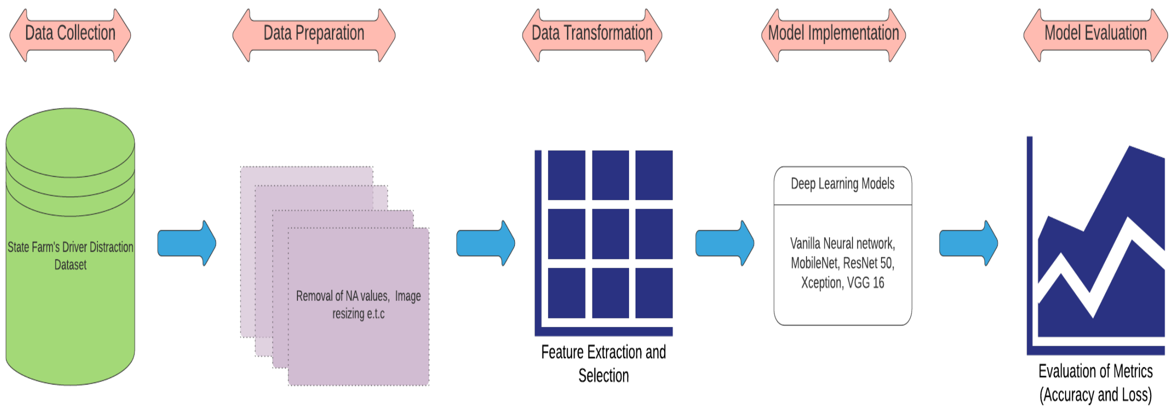

Figure 1: Driver Distraction Methodology Approach

The research question focuses on how a deep neural network can be utilized to detect the distracted drivers. The approach majorly concentrates on the deep neural network to detect the distracted driver. The deep neural network is one of the majorly used neural network techniques which are formulated from bio-inspired networks. They can be understood as one type of Artificial Neural Networks. From the introduction of such networks, they have been performing exclusively on the images and the application areas focusing on the image processing and video processing. The problem of distracted driver detection does belong to the computer vision stream towards the image processing or video processing. This is where the best suit for the application of convolution neural networks for the problem chosen has been concluded.

The neural network that is selected belongs to the vanilla network model, before selecting the model the first step that needs to be considered is to identify the receptive field. That is even before moving to the architecture of the model, it is likely that the receptive field needs to be identified. Once the processing of the model or the network formulated is completed the outcome of the model should match the receptive field that is identified from the dataset available. The major approach that is incorporated while developing the vanilla network is to make use of the convolution and the transition blocks in the network which facilitates the formulation of the receptive field. Figure 1 depicts the major steps that are involved as part of methodology. The methodology that was used is KDD (Knowledge Discovery in Databases) which has multiple stages for retrieving valuable information from massive databases.

The first step is to complete the data collection and required pre-processing on the data that is collected. Once the data is collected the next step that needs to be done is to formulate the network and make sure that the required features are extracted and selection of the same is completed. As the model chosen is vanilla CNN the architecture will be formulated, and the network will be constructed. Once it is constructed, the model will be tuned and the parameters are estimated accordingly. Any further optimization required to improve the accuracy of the classification or remove any overfitting that comes up with the model. Results will be analyzed and as a part of the parameter tuning, it will be further improved.

3.2 Data Collection

The dataset that is selected for the proposed approach is the state farm dataset which does suit the problem that is being dealt with. The dataset is formulated by the CDC motor vehicle society where they have been surveying the reasons for the increase in the number of accidents, how and why they have occurred. With the survey that they have conducted they were able to identify that distracted driver is one of the major reasons that increased the number of accidents. From the statistical analysis that they have conducted, found that in total 425,000 people injured, and 3,000 people killed by distracted driving every year. This is one of the saddest facts that motivated them towards the formulation of the data that they have collected and analyze the data for further facts which can be used to alert the drivers and reduce the number of accidents.

Coming to the details of the dataset, the dataset is downloaded from the Kaggle repository where the dataset is hosted. The data collected contains two different folders one contains the images that are categorized based on the class to which the driver activity belongs to. And the other csv file demonstrates the list of images and the subject to which the image belongs to. The image folder contains a total of 100000 images which are captured from different activities that the driver performs while driving the vehicle. The vehicle considered here is the car and the activities that the driver performs are classified into 10 different activities in total. Out of the ten activities that are formulated, nine of them are the activities of distracted driver and one activity is the driver focusing solely on the driving activity without any distraction.

3.3 Data Preparation

The dataset selected is the state farm dataset and the size of the dataset is about 5 GB. It contains two major data folders. One folder contains the images of the driver belonging to the 10 different classes into which they have been classified based on the actions that are being performed by the driver. Whereas the other folder depicts the list of the images that are available in the dataset and the label that has been assigned to the image. This serves as the ground truth for the models that are formulated as part of the detection of the driver activities.

3.3.1 Pre-Processing

The data collected is in the form of images it does require exploration and pre-processing to be completed before implementing any model on the data. As part of the pre-processing the images collected have their ground truth label associated with the image name and the class to which they belong. This is used to compare the value that is predicted by the model for the class that the image belongs to. This requires the detection of any null values or NA values and they need to be filled in manually or they can either be dropped if they are very less in number. This is required, as for the images, where the ground-truth is not available the block of code that compares the predicted label may not be evaluating the model properly. Coming to the images, resize of the image needs to be completed so that the images and the dimension of it when given as input would suit the model and the architecture that is formulated at its best. Once the images are resized, they are saved further into the data collection folder and the images are divided into training and testing the data that needs to be used further.

3.4 Data Transformation

3.4.1 Feature Extraction

As part of the feature extraction, a sequential convolution neural network model or the other approaches will be incorporated to complete the feature extraction. As in the convolution neural network all the features that the image has, and the model requires will extract through the operations carried out in its layers. The extraction of the feature needs the input to the input layer with the number of rows and columns in the image along with the type of the image that is being given as input. The first layer that is involved in the network will facilitate the extraction of the features and in-depth selection and processing of the features extracted will be carried out by further layers.

3.4.2 Feature Selection

All the features that are extracted from the images through the initial layers of the network are further processed in the deeper layers to find the optimal features that could facilitate better feature vector formulation and completion of the classification task that is required to train or test the images that are given to the model. As there can be numerous features extracted by the model but there will be a set of features that are highly correlated with the output that the model needs to formulate.

3.4.3 Parameter Tuning

Once the features are extracted and selected the next step that needs to be taken is to evaluate the model based on the training and testing accuracy that they can provide. Loss can also be one of the parameters based on which the model can be evaluated. These metrics that are evaluated on the training and testing data that is formulated will help to analyses the performance of the model. Then the model may either face a low accuracy or an extremely high accuracy which can be overfitting the model with all the features and the attribute values that are considered as part of the features.

This needs to be further corrected as the model may not be the best one without this parameter correction. This step deals with the tuning of the parameters and the addition of the extra layers for any further processing that is needed. Apart from these two tunings, one can also change the way the data is being taken for training where the data augmentation can solve the problem of overfitting when the model is facing this issue. This step of the methodology deals with these issues and formulates the enhanced model from the existing base model. Further the enhanced model will also be evaluated with the new parameters added and can complete the procedure further.

4 Model Implementation

This chapter deals with the implementation of the models on the dataset collected, all the steps that are planned as part of the methodology are depicted in this chapter along with the required environment and the setup for the empirical analysis on the model.

4.1 Environment

The environment that is used to implement the distracted driver methodology is python where the libraries in python are made use of utilize the layers in the models and formulate the network from the layers that are required. Coming to the hosting environment, the code is deployed in the google drive to make use of the Google Colaboratory and also used Anaconda Navigator to host in the local environment.

4.2 Data Visualization

As we have the images belonging to different categories the count of images that belong to each class is visualized as depicted in figure 2. There are total of 10 classes ranging from C0 to C9.

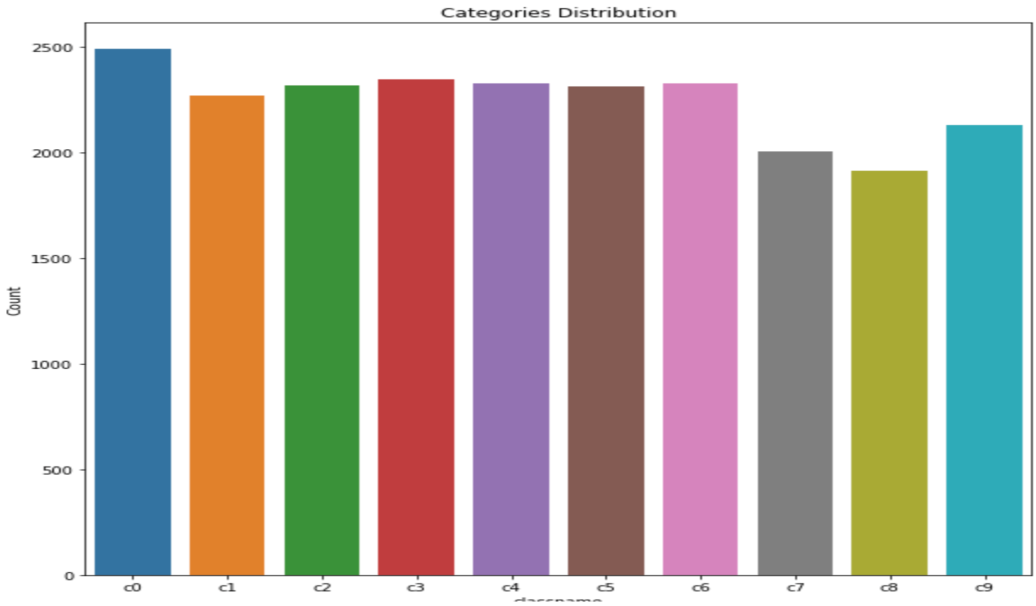

Figure 2: Class Distribution

4.3 Vanilla CNN Model

Vanilla model is the methodology that is proposed novel than the existing approaches is to use the method of vanilla convolution neural network model. This model is a kind of sequential model where the layers that are of two dimension convolution are added into the neural network. Coming to the formulation of the network, the convolution two dimensional layers are added and the max pooling after each of these two-dimensional convolution layers are associated with the max poling layers. A set of four such layers are formulated and the output is flattened through the flatten layers and at the end, dense and dropout layers are also added into the network. Once the model is formulated, the training and testing are carried out on the training and testing data.

4.4 Optimized Vanilla CNN

Once the results are observed for the base vanilla model that is formulated, the optimized model is formulated by modifying the network in the base vanilla CNN in the previous section. To optimize the model batch normalization is involved in the set of convolution neural network layers are incorporate, so that the features and the input that the layers are taking up is normalized before moving to the two-dimensional convolution and the max pooling as in the base model. Once these results are observed another variation of the model is implemented where the model is further improved using a variation in the data that is being used to train the model which is presented as part of the next section.

4.5 Vanilla CNN with Data Augmentation

As the neural network's performance improves when there is enough data to get the model trained, in this perspective, data augmentation would make use of the data that is available to create the artificial or virtual data from the available training images which will be used by the model. Once the data is augmented the data is used by the CNN model that is optimized from the base vanilla model and the results were analyzed and compared.

4.6 Xception Model

Xception model is the next model that is incorporated as part of the implementation, the model is considered from the base Keras models and the extra layers that are required to this base model are formulated and added. The initial base model considered is the Image net as the weights that are required to compute the base model. To the base model that is formulated using these parameters, extra layers with the stochastic gradient are added with the dense layers and the batch normalization followed by the dropout and dense layers which suits the chosen or the available input size and matches the required output to classify the given image into the action that is predicted by the model. To compile the model that is formulated, Adam optimizer is utilized with the categorical loss as the entropy because the problem being considered here is the classification problem. Figure 3 depicts the Xception model that is enhanced for the current problem being researched. Later the NumPy arrays that are formulated for the training and the testing data are considered for the further fit of the model and evaluating with the test data that is available.

4.7 MobileNet Model

MobileNet model is the next model that is incorporated as part of the implementation, the model is considered from the base Keras models and the extra layers that are required to this base model are formulated and added. The initial base model considered is the ImageNet as the weights that are required to compute the base model. For the base model extra layers with the stochastic gradient are added with the dense layers and the batch normalization. Along with these parameters the SoftMax function is used as an

Model: "functional_1"

Layer (type)	Output Shape	Param #
Image_input (InputLayer)	[(None, 224, 224, 3)]	0
xception (Functional)	(None, 7, 7, 2048)	20861480
global_average_pooling2d (Gl	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 1024)	1049600
batch_normalization_4 (Batch	(None, 1024)	4096
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
predictions (Dense)	(None, 10)	5130

Total params: 24,543,282
Trainable params: 24,486,706
Non-trainable params: 56,576

Figure 3: Xception Model with Extra Layers

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, None, None, 3)]	0
conv1_pad (ZeroPadding2D)	(None, None, None, 3)	0
conv1 (Conv2D)	(None, None, None, 32)	864
conv1_bn (BatchNormalization	(None, None, None, 32)	128
conv1_relu (ReLU)	(None, None, None, 32)	0
conv_dw_1 (DepthwiseConv2D)	(None, None, None, 32)	288
conv_dw_1_bn (BatchNormaliza	(None, None, None, 32)	128
conv_dw_1_relu (ReLU)	(None, None, None, 32)	0
conv_pw_1 (Conv2D)	(None, None, None, 64)	2048
conv_pw_1_bn (BatchNormaliza	(None, None, None, 64)	256
conv_pw_1_relu (ReLU)	(None, None, None, 64)	0
conv_pad_2 (ZeroPadding2D)	(None, None, None, 64)	0
conv_dw_2 (DepthwiseConv2D)	(None, None, None, 64)	576
conv_dw_2_bn (BatchNormaliza	(None, None, None, 64)	256
conv_dw_2_relu (ReLU)	(None, None, None, 64)	0
conv_pw_2 (Conv2D)	(None, None, None, 128)	8192
conv_pw_2_bn (BatchNormaliza	(None, None, None, 128)	512
conv_pw_2_relu (ReLU)	(None, None, None, 128)	0
conv_dw_3 (DepthwiseConv2D)	(None, None, None, 128)	1152
conv_dw_3_bn (BatchNormaliza	(None, None, None, 128)	512
conv_dw_3_relu (ReLU)	(None, None, None, 128)	0
conv_pw_3 (Conv2D)	(None, None, None, 128)	16384
conv_pw_3_bn (BatchNormaliza	(None, None, None, 128)	512
conv_pw_3_relu (ReLU)	(None, None, None, 128)	0
conv_pad_4 (ZeroPadding2D)	(None, None, None, 128)	0

Figure 4: MobileNet Model Summary

activation function that needs to be considered a part of the model layers activation. To compile the model, Adam optimizer is used. Figure 4 depicts the MobileNet model that is enhanced for the current problem being researched. For the better fit of the model NumPy arrays were formulated.

4.8 ResNet50 Model

ResNet50 model is the next model that is incorporated as part of the implementation, this model is also considered from the base Keras models. ResNet which is otherwise known as Residual Networks is commonly used for a number of Computer Vision problems. In this implementation along with the other parameters, the SoftMax function is also used as an activation function. To compile the model that is formulated Adam optimizer is utilized with the categorical loss as the entropy because the problem being considered here is the classification problem. Figure 5 depicts the ResNet50 model that is enhanced for the current problem being researched. Later the NumPy arrays that are formulated for the training and the testing data.

4.9 VGG16 Model

VGG16 model is the next model that is incorporated as part of the implementation, the model is considered from the base Keras models and the extra layers that are required to this base model are formulated and added. The initial base model considered the ImageNet as the weights that are required to compute the base model. To the base model that is formulated using these parameters, extra layers with the stochastic gradient are added with the dense layers and the batch normalization followed by the dropout and dense layers which suits the chosen or the available input size and matches the required output to classify the given image into the action that is predicted by the model.

Along with these parameters the SoftMax function is used as an activation function that needs to be considered a part of the model layers activation. To compile the model that is formulated Adam optimizer is utilized with the categorical loss as the entropy because the problem being considered here is the classification problem. Figure 6 depicts the VGG16 model that is enhanced for the current problem being researched. Later the NumPy arrays that are formulated for the training and the testing data are considered for the further fit of the model and evaluating with the test data that is available. Categorical cross-entropy and accuracy are the metrics that are being as part of the model layers to predict and evaluate the model that is formulated.

```
Model: "functional_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
Image_input (InputLayer)     [(None, 224, 224, 3)]     0         
_________________________________________________________________
resnet50 (Functional)        (None, 7, 7, 2048)        23587712  
_________________________________________________________________
flatten (Flatten)            (None, 100352)            0         
_________________________________________________________________
predictions (Dense)          (None, 10)                1003530   
=================================================================
Total params: 24,591,242
Trainable params: 24,538,122
Non-trainable params: 53,120
_________________________________________________________________
```

Figure 5: ResNet50 with Extra Layers Summary

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
Image_input (InputLayer)     [(None, 224, 224, 3)]     0         
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792      
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928     
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0         
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856     
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584    
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0         
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168    
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080    
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080    
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0         
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160   
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808   
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808   
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0         
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808   
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808   
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808   
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0         
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
```

Figure 6: VGG16 Model Summary

5 Model Evaluation and Results

This chapter discusses the results that are obtained from each model that is implemented and analyses the performance of the models.

5.1 Vanilla CNN Model

The results that are obtained based on the Vanilla CNN model are calculated concerning the accuracy and loss metrics. Model training and validation are performed and the accuracy, loss that is obtained as a part of the model are depicted in Figure 7. The accuracy that is obtained with this model is almost 100%. This accuracy results in overfitting and that was further addressed with optimization.

5.2 Optimized Vanilla CNN

The results that are obtained based on the Vanilla CNN model are calculated with respect to accuracy and loss. Model training and validation are performed and the accuracy, loss that is obtained as a part of the model are depicted in Figure 8. The accuracy that is obtained by tuning the parameters is 97.7% which shows that it is better than the base Vanilla CNN model. So further, model is enhanced with data augmentation technique. With this sub research question (Sub RQ) has been answered.

5.3 Vanilla CNN with Data Augmentation

The results that are obtained based on the Vanilla CNN are calculated for the accuracy and loss. Model training and validation are performed and the accuracy, loss that is obtained as a part of the model are depicted in Figure 9. The accuracy that is obtained with this model is 96.51% but there is no overfitting as the data augmentation is employed as a part of the model. When compared to all the three approaches using Vanilla CNN, optimized vanilla neural network gave the highest accuracy of 97.7%.

5.4 Xception Model

As part of the model training and further evaluation, the available data is split up into test and validation data which is further used for the training and the testing purpose respectively. As part of the Xception model, the model that is formulated as part of the implementation is further fit with the data generator formulated with the train data which contains a set of the images and their corresponding labels for further validation and to be used as the ground truth for the model to calculate the loss and the accuracy. The batch size that is considered as part of the model is 64 and the generator that is formulated using these parameters is sued further to evaluate the model with the test or the validation data formulated. The results that are obtained through this model are depicted in Figure 10, which depicts both the accuracy and the loss for training and validation. The accuracy that is obtained with this model is 83%.

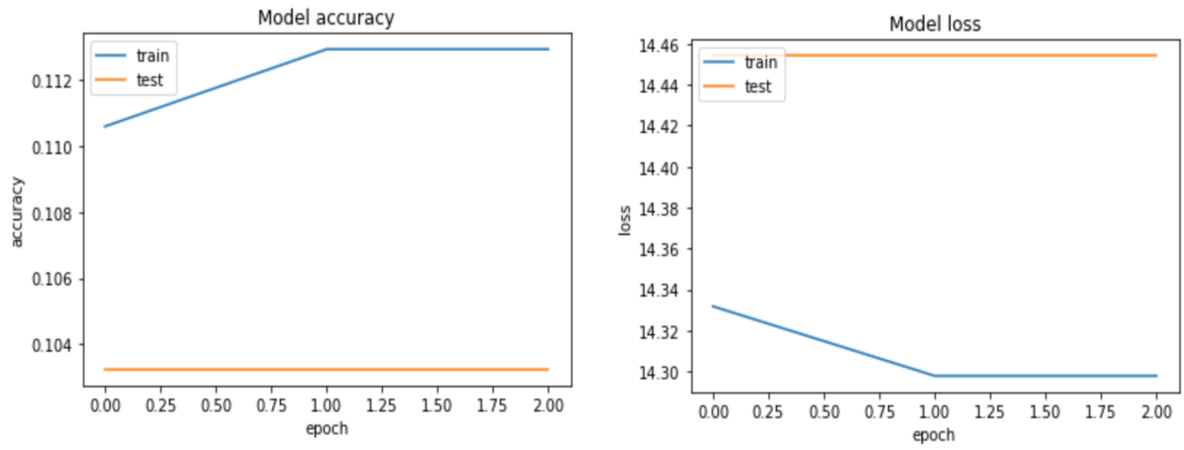

Figure 7: Vanilla CNN Results

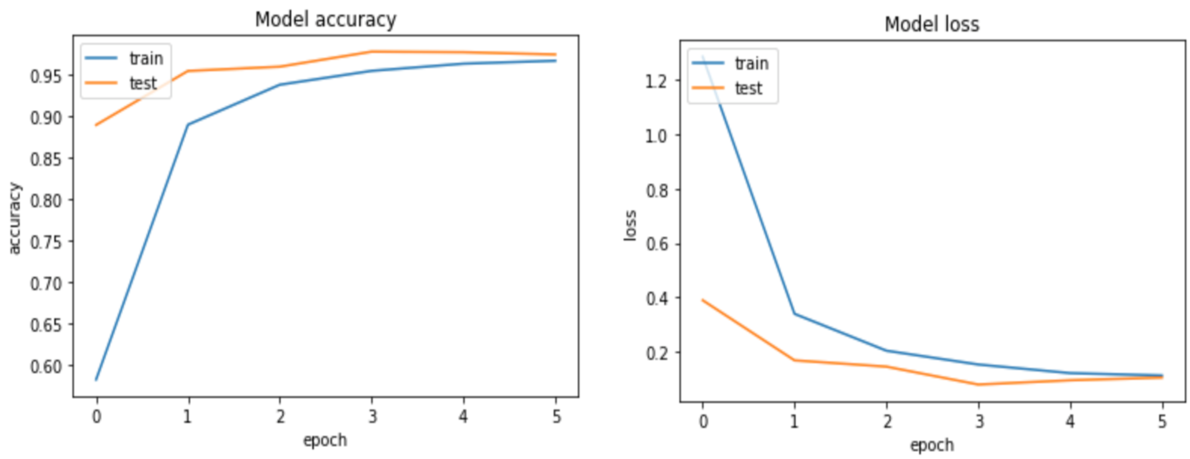

Figure 8: Optimized Vanilla CNN Results

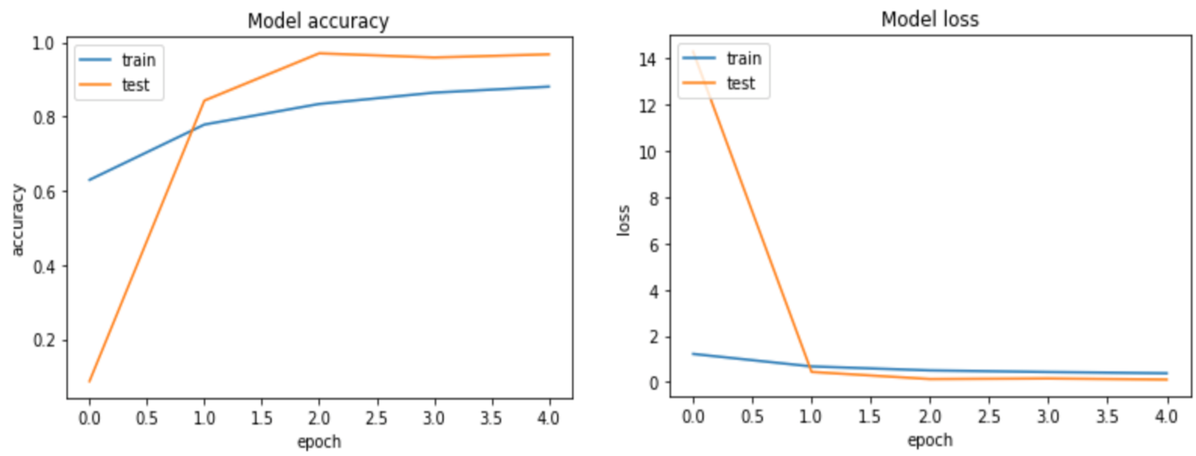

Figure 9: Vanilla CNN with Data Augmentation Results

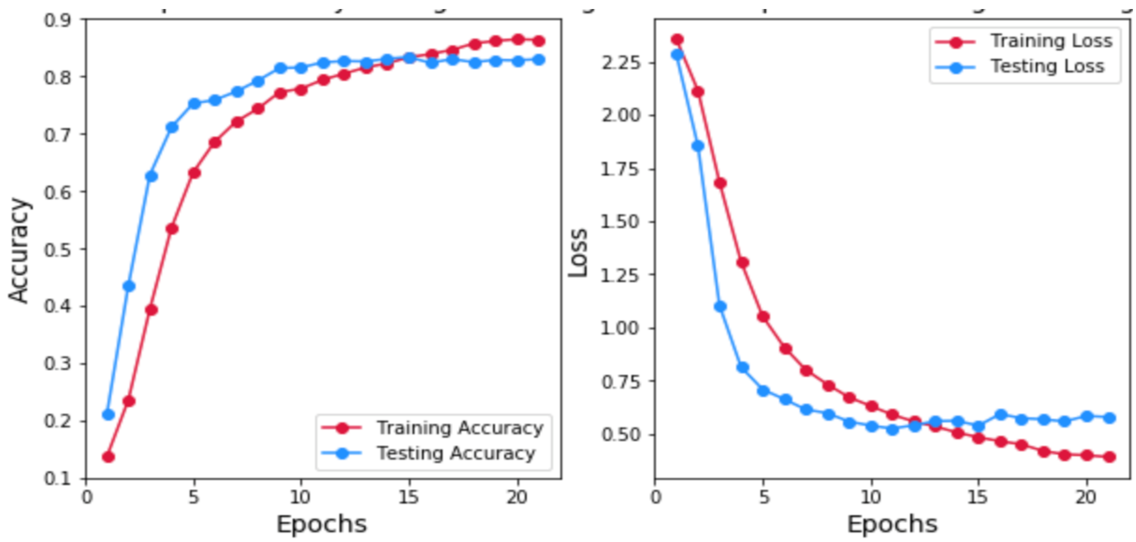

Figure 10: Xception Model Results

5.5 MobileNet Model

Mobile Net model that is formulated as part of the implementation is further fit with the data generator formulated with the train data which contains a set of the images and their corresponding labels for further validation and to be used as the ground truth for the model to calculate the loss and the accuracy. The results that are obtained through this model are depicted in Figure 11, which depicts both the accuracy and the loss for training and validation. The model considered here is the base model and no extra layers are considered as the results show that it has given almost satisfactory results and did not undergo any overfitting or underfitting of the data. The accuracy that is obtained with this model is 84%.

5.6 ResNet50 Model

In ResNet50 model that is formulated as part of the implementation is further fit with the data generator formulated with the train data which contains a set of the images and their corresponding labels for further validation and to be used as the ground truth for the model to calculate the loss and the accuracy. The batch size that is considered as part of the model is 64 and the generator that is formulated using these parameters issued further to evaluate the model with the test or the validation data formulated. The results that are obtained through this model are depicted in Figure 12, which depicts both the accuracy and the loss for training and validation. The accuracy that is obtained with this model is 83.64%.

5.7 VGG16 Model

The model that is formulated as part of the implementation is further fit with the data generator formulated with the train data which contains a set of the images and their corresponding labels for further validation and to be used as the ground truth for the model to calculate the loss and the accuracy. The batch size that is considered as part of the model is 64 and the generator that is formulated using these parameters is sued further to evaluate the model with the test or the validation data formulated. The results

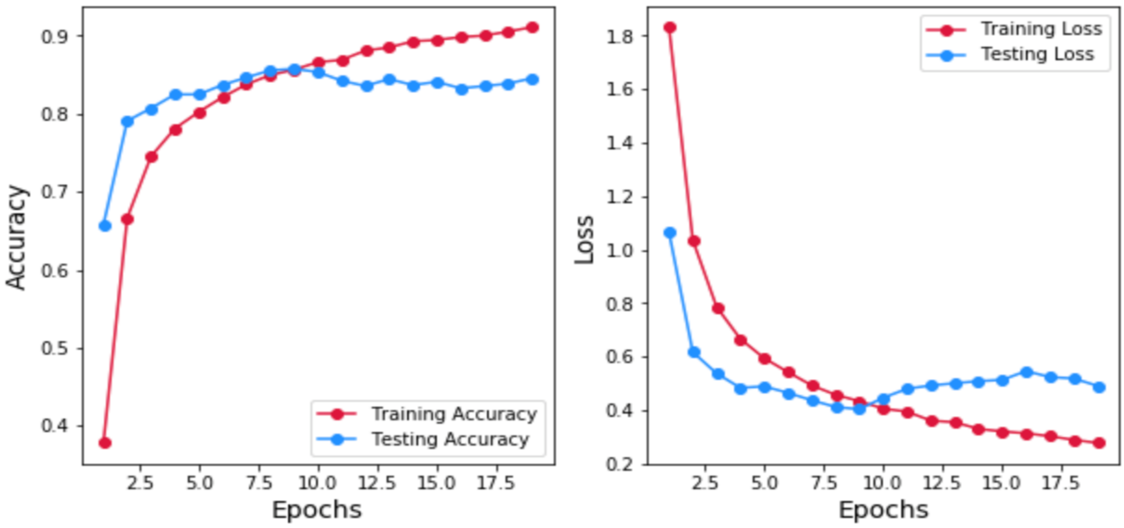

Figure 11: MobileNet Model Results

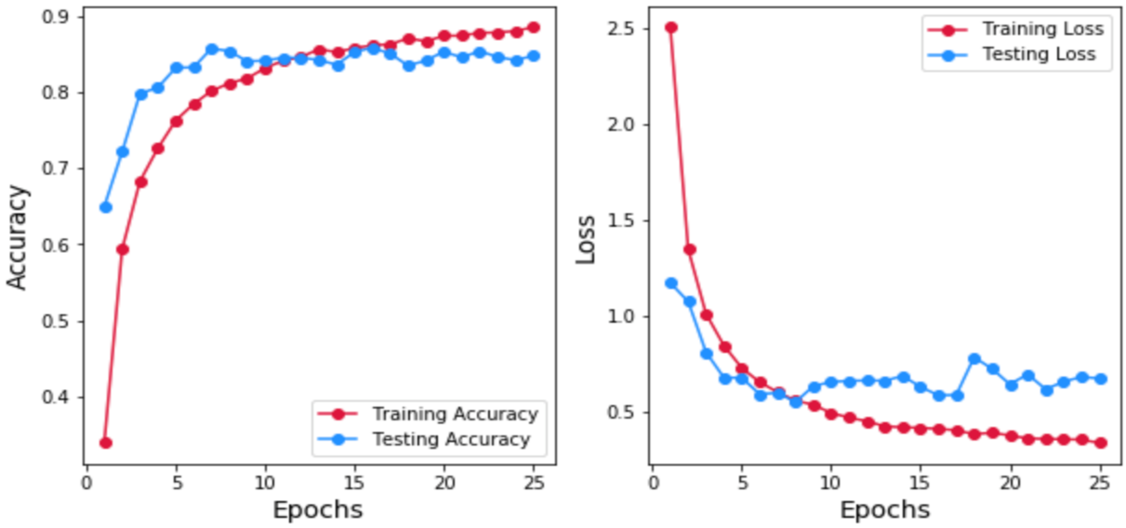

Figure 12: ResNet50 Results

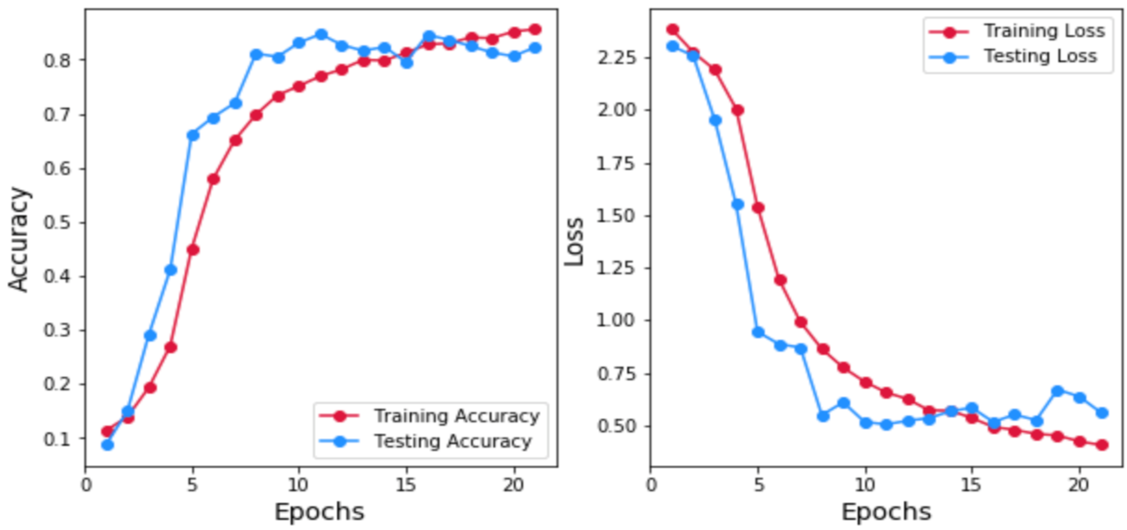

Figure 13: VGG16 Results

Table 3: Comparision of implemented models

Model	**Accuracy**
Vanilla CNN Model	100% but overfitting
Optimized Vanilla CNN	97.7%
Vanilla CNN with Data Augmentation	96.51%
Xception Model	83%
Mobile Net Model	84%
ResNet50 Model	83.64%
VGG16 Model	73.05%

that are obtained through this model are depicted in Figure 13, which depicts both the accuracy and the loss for training and validation. The model considered here is the base model and no extra layers are considered as the results show that it has given almost satisfactory results and did not undergo any overfitting or underfitting of the data. The accuracy that is obtained with this model is 73.05%. With this the research question (RQ) has been answered.

5.8 Discussions

From the empirical analysis conducted a novel model which is based on the sequential neural network architecture is incorporated. The first implementation that is carried out involves the base vanilla convolution neural network model which includes convolution two-dimensional layers along with the batch normalization and dropout layers. The results that are observed depict that it predicts the distracted driver actions at an accuracy of almost 100%. With this as the base model, the further optimization is carried out. This is where the optimized Vanilla CNN model is considered and this has given a performance of 97.7%. To still improve the accuracy of the model without overfitting, data augmentation technique was used. With the data augmentation involved the model has achieved an accuracy of 96.51%. Out of three approaches using Vanilla CNN, optimized Vanilla CNN gave the highest accuracy of 97.7% and with that, it outperforms the other deep learning techniques. Finally, Vanilla CNN gave satisfactory results when compared to the other deep learning architectures. This shows that Vanilla CNN is also suitable for detecting the driver distraction along with the state-of-art techniques. Later different models that have varied neural network architectures are involved, these include the Xception model, MobileNet, ResNet50, and VGG16. The following Table 3 presents the comparison of accuracy of the models that are implemented as part of this work.

6 Conclusion and Future Work

An in-depth study of why driver distraction is required is conducted. This includes the background and the motivation that is needed as part of the research question formulation. Once the research question has identified an ensemble of deep neural network models are formulated which involves vanilla CNN with three variations, Xception, ResNet50, MobileNet, and VGG16 are incorporated. Each of these methods is implemented and the results that they can obtain to the formulated train and test data are employed in predicting the activity of the driver with the input image. From the results and analysis conducted, optimized vanilla model has obtained the highest accuracy which is 97.7% and this can be considered as the best model for the current problem being considered. Coming to the future directions of the work is to improve the processing capability of the model as the accuracy is pretty satisfactory but the amount of time that is considered in formulating the results frame each of the models has a considerably huge amount of time due to the processing capability limitation even with the graphical processing unit i incorporated. With this, parallel processing can be one possible approach through which the processing time can be improved along with no compromise on the accuracy that can be obtained. For the future improvement in terms of detection, various weight initialization techniques can be used to obtain best performance of the deep learning models.

References

Alotaibi, M. and Alotaibi, B. (2019). Distracted driver classification using deep learning, *Signal, Image and Video Processing* pp. 1–8.

Baheti, B., Gajre, S. and Talbar, S. (2018). Detection of distracted driver using convolutional neural network, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1032–1038.

Cronje, J. and Engelbrecht, A. P. (2017). Training convolutional neural networks with class based data augmentation for detecting distracted drivers, *Proceedings of the 9th International Conference on Computer and Automation Engineering*, pp. 126–130.

Dong, Y., Hu, Z., Uchimura, K. and Murayama, N. (2010). Driver inattention monitoring system for intelligent vehicles: A review, *IEEE transactions on intelligent transportation systems* **12**(2): 596–614.

Dwivedi, K., Biswaranjan, K. and Sethi, A. (2014). Drowsy driver detection using representation learning, *2014 IEEE international advance computing conference (IACC)*, IEEE, pp. 995–999.

Fernández, A., Usamentiaga, R., Carús, J. L. and Casado, R. (2016). Driver distraction using visual-based sensors and algorithms, *Sensors* **16**(11): 1805.

Greenberg, J., Tijerina, L., Curry, R., Artz, B., Cathey, L., Kochhar, D., Kozak, K., Blommer, M. and Grant, P. (2003). Driver distraction: Evaluation with event detection paradigm, *Transportation Research Record* **1843**(1): 1–9.

Jo, J., Lee, S. J., Kim, J., Jung, H. G. and Park, K. R. (2011). Vision-based method for detecting driver drowsiness and distraction in driver monitoring system, *Optical Engineering* **50**(12): 127202.

King, L., Nguyen, H. T. and Lal, S. (2006). Early driver fatigue detection from electroencephalography signals using artificial neural networks, *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, pp. 2187–2190.

Koesdwiady, A., Bedawi, S. M., Ou, C. and Karray, F. (2017). End-to-end deep learning for driver distraction recognition, *International Conference Image Analysis and Recognition*, Springer, pp. 11–18.

Kutila, M., Jokela, M., Markkula, G. and Rué, M. R. (2007). Driver distraction detection with a camera vision system, *2007 IEEE International Conference on Image Processing*, Vol. 6, IEEE, pp. VI–201.

Liang, Y., Reyes, M. L. and Lee, J. D. (2007). Real-time detection of driver cognitive distraction using support vector machines, *IEEE transactions on intelligent transportation systems* **8**(2): 340–350.

Liao, Y., Li, S. E., Wang, W., Wang, Y., Li, G. and Cheng, B. (2016). Detection of driver cognitive distraction: A comparison study of stop-controlled intersection and speed-limited highway, *IEEE Transactions on Intelligent Transportation Systems* **17**(6): 1628–1637.

Liu, T., Yang, Y., Huang, G.-B., Yeo, Y. K. and Lin, Z. (2015). Driver distraction detection using semi-supervised machine learning, *IEEE transactions on intelligent transportation systems* **17**(4): 1108–1120.

Tran, D., Do, H. M., Sheng, W., Bai, H. and Chowdhary, G. (2018). Real-time detection of distracted driving based on deep learning, *IET Intelligent Transport Systems* **12**(10): 1210–1219.

Vicente, F., Huang, Z., Xiong, X., De la Torre, F., Zhang, W. and Levi, D. (2015). Driver gaze tracking and eyes off the road detection system, *IEEE Transactions on Intelligent Transportation Systems* **16**(4): 2014–2027.

Xing, Y., Lv, C., Zhang, Z., Wang, H., Na, X., Cao, D., Velenis, E. and Wang, F.-Y. (2017). Identification and analysis of driver postures for in-vehicle driving activities and secondary tasks recognition, *IEEE Transactions on Computational Social Systems* **5**(1): 95–108.

Yan, C., Jiang, H., Zhang, B. and Coenen, F. (2015). Recognizing driver inattention by convolutional neural networks, *2015 8th International Congress on Image and Signal Processing (CISP)*, IEEE, pp. 680–685.

M. Wollmer. (2011). Online Driver Distraction Detection Using Long Short-Term Memory,quot; in *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 574-582.

Mogelmose, A., Trivedi, M., amp; Moeslund, T. (2012). Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey. *IEEE Transactions on Intelligent Transportation Systems*, **13**(4), 1484-1497.

Muhlbacher-Karrer, S., Mosa, A., Faller, L., Ali, M., Hamid, R., Zangl, H., amp; Kyamakya, K. (2017).A Driver State Detection System—Combining a Capacitive Hand Detection Sensor with Physiological Sensors. *IEEE Transactions on Instrumentation and Measurement*, **66**(4), 624-636.

N. Das, E. Ohn-Bar, and M. M. Trivedi. (2015). On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics. *In 2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2953– 2958.

Sheng, Weihua amp; Tran, Duy amp; Do, Ha amp; Bai, he amp; Chowdhary, Girish. (2018). Real-time Detection of Distracted Driving based on Deep Learning. *IET Intelligent Transport Systems.* 12.

Tango, F., amp; Botta, M. (2013). Real-Time Detection System of Driver Distraction Using Machine Learning. IEEE Transactions on Intelligent Transportation Systems, **14**(2), 894-905.