**DRIVER RATING SYSTEM USING DEEP LEARNING** *(for the partial fulfillment of* Bachelor of Technology Degree in Computer Science & Engineering)

*Submitted by*

**Mr. Amit Gusain**

**Mr. Arvind Singh Rawat**

**Mr. Lalit**

**Mr. Himanshu Pal**

**Under the guidance of**

*Dr. Indrajeet kumar*

*Professor*



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GRAPHIC ERA HILL UNIVERSITY**

**May, 2023**

**CERTIFICATE**

This is to certify that the thesis titled **“Driver Rating System Using Deep Learning” submitted** by **Amit Gusain, Arvind Singh Rawat, Lalit, Himanshu Pal**, to Graphic Era Hill University for the award of the degree of **Bachelor**  **of Technology**, is a bona fide record of the research work done by him/her under our supervision. The contents of this project in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

**Dr. Indrajeet Kumar**  Project Guide

(Professor)

GEHU,Dehradun

Place: Dehradun

Date:

**TABLE OF CONTENTS**

**ACKNOWLEDGEMENT i**

**ABSTRACT ii**

**LIST OF TABLES iv**

**LIST OF FIGURES v**

# INTRODUCTION 1

## History of the Project…………………………….……………….................... 1

## Technology Involved………………………………………………………….. 2

## Motivation………………………………………………………………. . 2

## Dataset Used………………………………………………………………….. 3

# PROJECT DESIGN 4

## Basic CNN Architecture………………………………………………………. 4

## Algorithm/Model Used 6

# Result/Testing of Project/Software 8

## Model 1 9

## Model 2 11

## Model 3 13

## Comparison 14

# Conclusion And Future Scope 15

# APPENDIX 16

**REFERENCES 27**

**ACKNOWLEDGEMENT**

First of all we would like to thank almighty god Vishnu ji, the most Beneficient and the Most Merciful.

We would like to express our special thanks to our mentor Mr.Indrajeet Kumar Sir for his time and efforts that he provided throughout the course of this project. Your useful advice and suggestions were really helpful to us during the project’s completion. In this aspect, we are eternally grateful to you.

**Amit Gusain Arvind Singh Rawat**

Roll no. 1918203 Roll no. 1918263

**Lalit Himanshu Pal**

Roll no. 1918437 Roll no. 1914016

**ABSTRACT**

Driving is a complex task which requires full focus of the person driving the car. Even a small distraction of the driver can lead to big accidents. Distracted driver is an activity which takes away driver’s attention from the road. Thousands of people die every day just because of road accident where the lead cause of the accident is usually the distracted driver. The prevalence of driver distraction is on the rise due to the widespread adoption and complexity of in-vehicle technologies and portable devices. This trend poses a significant risk to road safety, as distractions and inattentiveness are major contributing factors to accidents. The continuous advancements in wireless communication, computer systems, and sensor technology have introduced a plethora of new distractions for drivers. In addition to managing traditional devices like cell phones, CD players, and navigation systems, drivers now find themselves increasingly involved in lengthy text message conversations and browsing through extensive music catalogs, often lasting more than 30 seconds.

So our goal in this project is to develop a software which can detect whether a driver is driving the car safely or performing an activity that may lead to an accident. In this project, The primary source of data for this project comprises images of drivers captured by an in-car camera, encompassing their face, arms, and hands. This dataset, which is openly accessible on Kaggle, is widely utilized for similar projects in the field.

In this project we will be using deep learning and machine learning techniques to classify from the images whether the driver is distracted or not. By using these techniques we will classify our input images into different categories of distraction in which our driver may be indulging in. As we have to classify our images into different categories we will likely be using classification techniques.

.

**LIST OF TABLES**

1. Table 1: In this table, we have the training result of model 1 after completing the training. It involves the training accuracy, loss and validation accuracy and validation loss of model 1 after completion of training.
2. Table 2: This table also involves the training data of model 2. It mainly has accuracy, loss, validation accuracy and validation loss of model 2.
3. Table 3: This table is also similar to the above tables. It also contains the training data of model 3 which include training accuracy, training loss , validation loss and validation accuracy for the model.
4. Table 4: This is a main performance indicator table for all our models. It compares the performance of each model when evaluated on test data. This ‘test data’ is the data which was not seen by the models during their training.

**LIST OF FIGURES**

Figure 1: Dataset used

Figure 2: Basic Convolution Layer mechanism

Figure 3: ReLU function and Graph

Figure 4: Max Pooling method

Figure 5: Softmax function

Figure 6: Model 1 Summary

Figure 7: Model 2 Summary

Figure 8: Model 3 Summary

Figure 9: Model 1 Accuracy

Figure 10: Model 1 Loss

Figure 11: Model 2 Accuracy:

Figure 12: Model 2 Loss

Figure 13: Model 3 Accuracy

Figure 14: Model 3 Loss

# CHAPTER 1

# INTRODUCTION

As per the WHO (World Health Organization), more than 1 million people die due to traffic accidents worldwide per year, making it one of top 10 leading causes of death and around 30 million people are injured from such accidents.

According to the National Crime Branch Research Bureau (NCRB), Govt. of India, the Indian roads account to the highest of fatalities around the world.

**1.1: History of the project:**

1. Authors in [13] propose a method to enhance the accuracy of a CNN-based Inception ResNet model by fine-tuning it using a dataset consisting of images depicting drivers' interruption actions. Their dataset comprises six classes, with each class containing 1000 images. The authors employed several preprocessing techniques such as cropping, adjustments, and flipping on both the training and test images. Subsequently, the preprocessed data was fed into the Inception ResNet model for training purposes. It's worth noting that the model was initially pretrained on the ILSVRC 2012 dataset. The experimental results demonstrated an impressive test accuracy of 83%.
2. Authors in [19] propose a deep learning architecture for detecting driver inactivity. Their approach leverages the features extracted from a pretrained VGG-19 model, which is fine-tuned on a publicly available dataset. The experimental results are remarkable, with a test accuracy of approximately 95% and an accuracy of around 80% on the validation data for each class.
3. In [20], the authors propose a classification approach using Support Vector Machines (SVM) to categorize different types of driver drowsiness based on the detected edges. The four classes include alertness, nodding, drowsiness with blinking, and yawning. It is important to note that the technique was evaluated using simulated data, where the signs of drowsiness were readily observable, enabling classification based on obvious drowsiness indicators. The achieved accuracy for this approach was 65.2%.
4. In their study [21], the authors introduce a CNN model called DarNet for recognizing inactive drivers. They utilized a dataset comprising six distinct categories associated with driver inactivity, including everyday driving, talking, texting, reading, makeup, and eating. The authors optimized the Inception V3 module, which was initially trained on the DAggle dataset for inactive driver recognition. Impressively, their proposed DarNet model achieved a precision rate of 87% in their experimental evaluation.
5. In their work [22], the author presents a CNN-based architecture that builds upon the VGG-16 network. The proposed architecture incorporates a transformation by replacing the ReLU activation function with the Leaky ReLU activation function. Additionally, different regularization strategies were applied to address the challenge of overfitting. Through these modifications, the system achieved an impressive precision score of 96% on the AUC Distracted Driver Dataset.

**1.2: What kind of technology was involved?**

In the previous work done on this project we could see two type of different approaches which included

➢ Machine Learning models -> In these types of projects we could see image dataset being taken as input and using various features extraction and feature decomposition techniques this dataset was passed as an input to various machine learning classification models and then these trained models were tested on the test dataset to achieve highest accuracy and precision score. The main technologies or frameworks used were scikit-learn for importing our classification models, OpenCV for image processing, numpy for handling image data as an array or 2D matrix.

➢ Deep Learning models -> Same as machine learning techniques they used image datasets as input. Then they build convolutional neural network to process this image dataset and get the desired output. The main technologies and framework used were tensorflow, pytorch, and keras for developing our deep learning model as well as image processing.

**1.3: Motivation**

Advancement in technology is making people more distracted not only in their homes but even while driving vehicles, such advancements may be in their vehicles itself or the devices they carry. Many people face either the financial lose or the lose of life by the accident caused by distracted drivers. Distracted drivers cause harm not only to themselves but also to others people or road. Millions of people suffer financial losses worth billions of dollars every year which makes their financial situations much worse. These accidents also lead to health complications which may be permanent. Traffic accidents almost cost each country 3% of their GDP all around the world. This is a major issue which cannot be seen lightly. So our project proposes a solution for this problem.

**1.4: Dataset Used:**

We have used StateFarm Dataset, download from Kaggle for our project. This dataset consists of 10 classes, depicting different forms of distractions during driving. Below are the given classes.



Figure : Dataset used

# CHAPTER 2

# project design

In this project we will be using Convolutional Neural Network to classify our images based on our dataset.

A deep CNN is a type of artificial neural network. Deep CNNs are motivated by the animal visual cortex. Currently, for several recent years, CNNs have demonstrated tremendous achievements in various applications, e.g., image classification, object and action detection, and natural language processing

**2.1:** **Basic CNN Architecture**:

1. Convolution: Convolve the filter with the input i.e. slide over the image spatially, computing dot products. As we slide the filter over our pre-processed images we do a dot product of our filter with the image matrix and set the result as a value in our activation map.

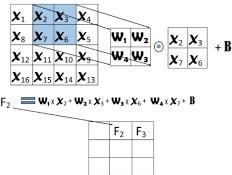


Figure : Basic Convolution Layer mechanism

2. Non-linearity: This refers to the activation function that we use for our layers to assign weights to our neurons. Some examples of activation functions are Sigmoid, Tanh, Relu.

For our project, we have used ReLU activation function.

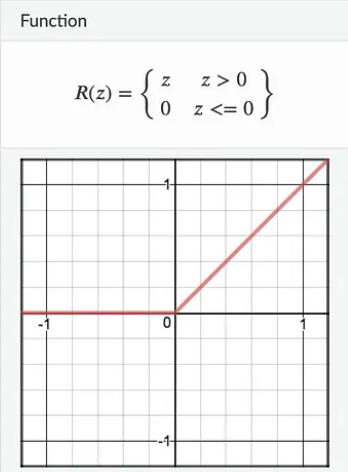


Figure : ReLU function and Graph

3. Spatial Pooling: This is used to solve the invariance by converting them into small transformations. It works the same as convolution but in this we can also define the strides. Their two most used ways of doing this which include max pooling and average pooling. This operates over each activation map independently.

A picture containing text, screenshot, line, font

Description automatically generated

Figure : Max Pooling method

4. Fully connected layers: These layers have a global receptive field which means that each fully connected layer looks at the entire image(or in this case the output provided by the previous layers).These layers are fully connected to all the activation of the previous layers.

5. Classifier: This is the final layer that finally classifies our input image to the categories defined by the user. Example Softmax classifier.

A picture containing text, font, receipt, screenshot

Description automatically generated

Figure : Softmax function

In a CNN we combine multiple layers of convolutions, non-linearity, and pooling. This process is called stacking.

## **2.2: Algorithms / Models Used**

In this project, we have made 3 CNN models, each model is different from other and has different layers. They are used to compare the results and choose out the best of three to proceed with the project.

1. First Model: In this model, we have used a Basic Cnn Model in which we have used 10 layers which include 2 convolutional layers, 2 max pool layer, 3 fully connected layers, 4 other layers. We have used ReLU function as our activation function. While optimizing our CNN architecture we have used ‘adam’ as our main optimizer. For calculating the losses for our training and validation data, we have used ‘sparse\_categorical\_crossentropy’ function. We have defined ‘accuracy’ as the main metric for our model comparison.

A picture containing text, font, graphic design, design

Description automatically generated

Figure : Model 1 Layer Architecture

2. Second Model: In this model, we have used an improved CNN Model in which we have used total 25 layers which include 7 convolutional layers, 3 max pool layer, 5 fully connected layers, 10 other layers. In this we have also used Batch Normalisation to increase the efficiency and reliability of our model. We have used ReLU function as our activation function. While optimizing our CNN architecture we have used ‘adam’ as our main optimizer. For calculating the losses for our training and validation data, we have used ‘sparse\_categorical\_crossentropy’ function. We have defined ‘accuracy’ as the main metric for our model comparison.

A picture containing text, line, parallel, rectangle

Description automatically generated

Figure : Model 2 Layer Architecture

3. Third Model: In this model, we have used an augmented CNN Model in which we have made improvement to our previous model, which was slightly overfitting. We have used ReLU function as our activation function. While optimizing our CNN architecture we have used ‘adam’ as our main optimizer. For calculating the losses for our training and validation data, we have used ‘sparse\_categorical\_crossentropy’ function. We have defined ‘accuracy’ as the main metric for our model comparison.

A picture containing text, red

Description automatically generated

Figure : Model 3 Layer Architecture

# CHAPTER 3

# Result/TESTING OF PROJECT / SOFTWARE

**3.1.** **Model 1**: Our first model was trained for 20 iterations(epochs), each iteration calculated training accuracy, training loss, validation accuracy, validation loss. On the last iteration we obtained the following results.

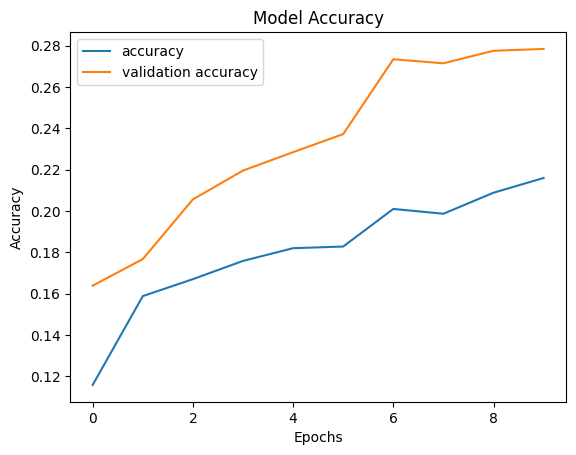


Figure : Model 1 Accuracy

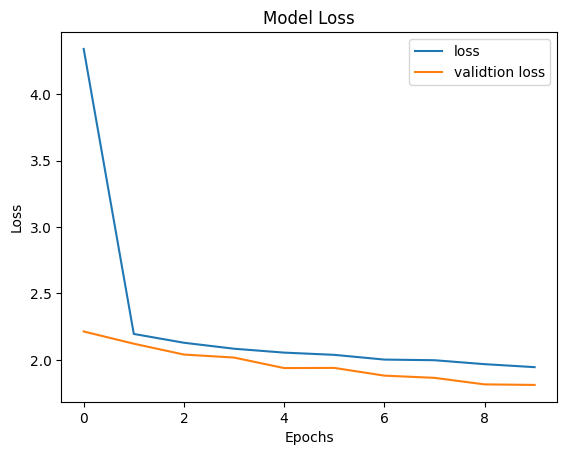


Figure : Model 1 Loss

|  |  |  |  |
| --- | --- | --- | --- |
| Training Accuracy | Training Loss | Validation Accuracy | Validation Loss |
| 0.216 | 1.9454 | 0.2785 | 1.8117 |

Table : Model 1 training result:

**3.2. Model 2**: Our second model was trained for 20 iterations, however since we used an Early Stopping function, this model was overfitting only after certain iterations so Early Stopping was triggered and this model was stopped training before completing all the iterations.

A picture containing text, line, screenshot, diagram

Description automatically generated

Figure : Model 2 Accuracy:

A picture containing text, screenshot, diagram, line

Description automatically generated

Figure : Model 2 Loss

|  |  |  |  |
| --- | --- | --- | --- |
| Training Accuracy | Training Loss | Validation Accuracy | Validation Loss |
| 0.9917 | 0.0311 | 0.9841 | 0.0563 |

Table : Model 2 Training Result

**3.3. Model 3**: Our third model was also trained for 20 iterations, as it was an improved version, it gave the best accuracy on both training data and validation data. Also, the accuracy for our test data was also highest in this model.

A picture containing text, screenshot, line, diagram

Description automatically generated

Figure : Model 3 Accuracy

A picture containing text, screenshot, line, diagram

Description automatically generated

Figure : Model 3 Loss

|  |  |  |  |
| --- | --- | --- | --- |
| Training Accuracy | Training Loss | Validation Accuracy | Validation Loss |
| 0.9885 | 0.0446 | 0.9871 | 0.0610 |

Table : Model 3 training result

**3.4: Comparison**: At the end we compared all our 3 models based on their training and validation accuracy and also their losses. In the diagram represented below, we can see the difference in output of all the different models. Also in the table represented below we have compared the test accuracy of our 3 models.

|  |  |  |  |
| --- | --- | --- | --- |
| Models | Model 1 | Model 2 | Model 3 |
| Test Data Accuracy | 28% | 98% | 98.58% |

Table : Models Result Comparison

# CHAPTER 4

# Conclusion AND fUTURE SCOpe

In conclusion, this project has given us successful results compared to related work done in this domain. We have created multiple models from scratch that have given us high level of accuracy and precison. Through the course of this project, we have explored various deep learning techniques, data preprocessing methods and various ways for fine tuning our deep learning models.

We have demonstrated that our model works on par with pre-existing and fine-tuned deep learning models. Additionally, we have identified key insights and trends in the data that can inform further research and decision-making. This field can be further be improved and researched upon as the their much to explore and this may soon be applied to vehicles for further advancement in driving system.

Overall, this project has highlighted how deep learning models are more efficient and more powerful in solving complex problems and its has also shown great potential for real-world applications. Future work can include fine tuning the model architecture, and evaluating the performance of the model in different contexts also applying this project details in real time data so that it can prevent many accidents from happening and also this project can also be used as a rating system for drivers to evaluate themselves.

This project has important real-world applications, as distracted driving is a major cause of road accidents and fatalities. By developing a robust and reliable model for detecting distracted drivers, we can potentially save lives and improve overall road safety.

Future work can include expanding the dataset to include more diverse scenarios and classes of distractions, as well as exploring additional techniques for improving model interpretability and explaining its decisions to end-users. Overall, this project highlights the potential of deep learning for addressing complex problems in the field of transportation safety.

**APPENDIX A**

**(Code)**

Code for app.py file:

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer program

Description automatically generated with medium confidence

Code for JupyterNotebook



A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot, font, line

Description automatically generated

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidenceA screen shot of a computer

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with low confidenceA screenshot of a computer

Description automatically generated with low confidenceA screenshot of a computer

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with medium confidence

**References**

[1] R.A. Berri, A.G. Silva, R.S. Parpinelli, E. Girardi, R. Arthur, A pattern recognitionsystem for detecting use of mobile phones while driving, in: International Conference on Computer Vision Theory and Applications, VISAPP, IEEE, 2014, pp. 411–418.

[2]https://ai.plainenglish.io/distracted-driver-detection-using-machine-and-deeplearning-techniques-1ba7e7ce0225

[3] C. Streiffer, R. Raghavendra, T. Benson, and M. Srivatsa, „„A deep learning solution for distracted driving detection,‟‟ in Proc. 18th ACM/IFIP/USENIX Middleware Conf., Ind. Track, 2017, pp. 22–28.

[4] E. Ohn-Bar, S. Martin, A. Tawari, M.M. Trivedi, Head, eye, and hand patterns for driver activity recognition, in: 22nd International Conference on Pattern Recognition, IEEE, 2014, pp. 660–665

[5] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, “Real-time distracted driver posture classification,” in Proc. NIPS Workshop Mach. Learn. Intell. Transp. Syst., 2018, pp. 1–8.

[6] C. Zhao, B. Zhang, J. He, J. Lian, Recognition of driving postures by contourlet transform and random forests, IET Intell. Transp. Syst. 6 (2) (2012) 161–168.

[7] C. Zhao, Y. Gao, J. He, J. Lian, Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier, Eng. Appl. Artif. Intell. 25 (8) (2012) 1677–1686.

[8] John Lee, Timothy Brown, Dary Fiorentino, James Fell, Eric Traube, and Eric Nadler. 2011. Using Vehicle-Based Sensors of Driver Behavior to Detect Alcohol Impairment. In 22nd International Technical Conference on the Enhanced Safety of Vehicles (ESV)..

[9] B. Baheti, S. Gajre, and S. Talbar, “Detection of distracted driver using convolutional neural network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2018, pp. 1032–1038.

[10] Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, F.-Y. Wang, Driver activity recognition for intelligent vehicles: A deep learning approach, IEEE Trans. Veh. Technol. 68 (6) (2019) 5379–5390.

SNAPSHOTS

A person holding a phone in front of a steering wheel

Description automatically generated with low confidence

A picture containing screenshot, car, text, person

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A close up of a hand

Description automatically generated with medium confidence

Here the output is correct. The given image is categorized as ‘safe driving’.