IV Semester

MINOR PROJECT REPORT
ON
STEERING ANGLE PREDICTION IN AUTONOMOUS VEHICLES

BY

SURYA PRATAP SINGH RATHOR	191020454
ABHEESHTH MISHRA	191020401
MAHIMA SAHU	191020430

UNDER THE GUIDANCE OF
DR. RASHMI CHAUDHRY

DEPARTMENT of COMPUTER SCIENCE AND ENGINEERING DISCIPLINE
Dr. SPM INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY NAYA RAIPUR
ATAL NAGAR - 493661, INDIA

MAY 2021

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Dated: 10 MAY 2021

Surya Pratap Singh Rathor 191020454
Abheeshth Mishra 191020401
Mahima Sahu 191020432

PLAGIARISM REPORT

cloud

ORIGINALITY REPORT

25%	13%	23%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	link.springer.com Internet Source	2%
2	Rifaqat Ali, Arup Kumar Pal. "Three-Factor-Based Confidentiality-Preserving Remote User Authentication Scheme in Multi-server Environment", Arabian Journal for Science and Engineering, 2017 Publication	2%
3	Zezhong Zhang, Qingqing Qi. "An Efficient RFID Authentication Protocol to Enhance Patient Medication Safety Using Elliptic Curve Cryptography", Journal of Medical Systems, 2014 Publication	2%
4	Zhenguo Zhao. "A Secure RFID Authentication Protocol for Healthcare Environments Using Elliptic Curve Cryptosystem", Journal of Medical Systems, 2014 Publication	1%
5	Jin, Chunhua, Chunxiang Xu, Xiaojun Zhang,	

Certificate

This is to certify that the project titled STEERING ANGLE PREDICTION IN AUTONOMOUS VEHICLES by Surya Pratap Singh Rathor, Abheeshth Mishra, Mahima Sahu has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree/diploma.

Dated: 10 MAY 2021

Dr. Rashmi Chaudhry
Assistant Professor
Department of CSE
DSPM IIIT Naya Raipur-493661

Acknowledgments

At the outset, I would like to express my whole hearted and deep sense of gratitude to my guide Prof. Rashmi Chaudhry for her guidance, help and encouragement throughout my research work. I greatly admire his attitude towards research, creative thinking, hard work and dedication in work. I am highly grateful to her for patiently checking all my manuscripts and thesis. This thesis would not have been possible without his bounteous efforts. More than guide, she is my mentor for shaping my personal and professional life, without whom I would not have been where I am today. I owe my profound gratitude to Prof. Rashmi Chaudhry for her supports in all respects.

Surya Pratap Singh Rathor
Abheeshth Mishra
Mahima Sahu

ABSTRACT

These days, a great many Autonomous Vehicles (AVs), otherwise called self-driving vehicles, are on streets. AVs are expected to establish 25% of vehicles worldwide in 2035. Steering angle prediction is basic in the control of Autonomous Vehicles (AVs) and has pulled in the consideration of researchers, makers, and insurance agencies in the car business. In this project Convolutional Neural Network model has been applied to predict the steering angle of AVs in different situations. Different sorts of data are fed to the system of Autonomous Vehicles like Image Data (cameras [front side, rear side]), LIDAR [3-D geometry], RADAR, GPS, Ultrasonic sensors, etc. Outputs predicted are Steering wheel angle, acceleration, braking. In this project, a section of the Self-Driving Cars System is implemented using Convolutional Neural Networks. Dataset of the dashcam is taken to train CNN Model. The model will learn the steering wheel angle from the dataset as per the turns in the image and then predict the steering wheel angle. The predictions of the model will be tested in a simulation.

Keywords: Convolutional Neural Networks (CNN), Tensorflow, Autonomous Vehicles (AVs), Deep Learning (DL)

Contents

1 Introduction **1**

1.1 How Autonomous Vehicles are becoming a substitute for human drivers? 1

1.2 Benefits of steering angle prediction with Deep Learning approach . 1

1.3 Challenges of Deep Learning for steering wheel angle prediction in AVs 2

2 Literature Survey **3**

3 Methodology **5**

3.1 Dataset 5

3.2 Image Preprocessing 5

3.3 Model Architecture using CNN 6

3.3.1 Convolutional layer 6

3.3.2 Stride, Padding and Pooling 6

3.3.3 Activation Functions 7

3.3.4 Optimizer 8

4 Implementation and Results **10**

4.1 Environment 10

4.2 Implementation 10

4.3 Results 11

4.3.1 CNN Model 11

4.3.2 CNN Model + Transfer Learning 12

4.3.3 Comparison of Models 13

4.3.4 Code Implementation 13

5 Conclusion and Future Scope **16**

5.1 Conclusion 16

5.2 Future Scope 16

References **16**

List of Figures

3.1 **Images recorded from a car dashcam with labeled steering angles** 5
3.2 **Figure describing how dash cam images and their corresponding steering wheel angle(in radians) are feeded into CNN to predict Steering Wheel Angle** 8
3.3 **Overview of Model** 9

4.1 **CNN architecture of Proposed Model. The network has about 27 million connections and 250 thousand parameters. [5]** 11
4.2 **Loss is plotted on the vertical axis and visulaised in TensorBoard[16] with no. of steps in the horizontal axis(1 epoch = 100 steps)** . . . 12
4.3 **Loss is plotted on the vertical axis and Epochs in the horizontal axis** 13
4.4 **Model Predicting - 1** 14
4.5 **Model Predicting - 2** 14
4.6 **Model Predicting - 3** 15
4.7 **Model Predicting - 4** 15

List of Tables

2.2 Literature Survey Table 4

Chapter 1

Introduction

1.1 How Autonomous Vehicles are becoming a substitute for human drivers?

The National Motor Vehicle Crash Causation Survey (NMVCCS) [1], directed from 2005 to 2007, gathered data about the variables which led to street mishaps, and discovered that more than human mistakes are the significant explanation for more than 90% of the auto collisions and mechanical disappointments were the explanation in 2% of the accidents and other 1.3% involved natural elements (streets, environment, and so on). Autonomous Vehicles' control framework comprises longitudinal and lateral regulators. Steering control, changing pathway, and maintaining a strategic distance from crashes are constrained by lateral regulators while longitudinal regulators control the vehicle's cruise speed [4]. AVs have a wide extension matched with the tendency to commit fewer blunders than human drivers.

1.2 Benefits of steering angle prediction with Deep Learning approach

Deep Learning[15] has the capacity to inadequately handle unlabeled raw data. While examining a scene, DL is sensibly insensitive towards varieties of ecological conditions, yet requires an enormous measure of data to accomplish high accuracy. The benefits of steering angle prediction utilizing the DL approach are that it has the capacity to bear botches, the capacity to rapidly distinguish mistakes and better ability in overseeing flighty circumstances[2]. An autonomous vehicle is outfitted with a camera that envisions the street ahead and yields the angle of the steering wheel. The CNN is able to learn significant features of the road from a very sparse training signal. CNN is able to identify objects and other features like markings of the lane, edges of the roads, other cars/objects on the road, and typical vehicles that are additional features that would be difficult to be programmed manually by engineers[11].

1.3 Challenges of Deep Learning for steering wheel angle prediction in AVs

Scientists in this space vigorously depend on artificial datasets to explore the use of DL in predicting the steering angle of AVs. The utilization of artificial datasets has difficulties in directing tests with the point of deploying the outcomes in the real world [3]. Although simulated environment models the real-world scenario but unexpected situations can occur in the real world. CNN driving at higher speed is a big challenge, it is really slow while responding to unexpected scenarios.CNN requires high computational costs memory space [12]. The ideal presentation of controlling control by means of DL has not however been accomplished, and there is still an opportunity to get better to accomplish the ideal performance.

Chapter 2

Literature Survey

We have performed a literature survey to know about the existing methods which are already used for classification. One of the journal papers focused to quantify and analyze the factors responsible for the onset of wilt and its different stages.

[6] Akhil Agnihotri, Prathamesh Saraf Kriti Rajesh Bapnad prepared a CNN which takes in the extreme variable highlights from the dashcam camera, thus rechoirs no human mediation. Platforms comprised of an ultrasonic sensor for detecting obstacles and an RGBD camera for continuous position observing. and Raspberry Pito output the steering angle and then it is converted to angular velocity for steering.

[7] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski had prepared CNN to plan crude pixels from a solitary forward-looking camera straightforwardly to guiding orders. They contend that end to end framework upgrades all taking care of steps simultaneously appeared differently with explicit division, lane-marking, path planning and control.

[8] Junekyo Jhung, Jaeyoung Moon and other authors of this paper proposed an end-to-end steering angle controller with CNN-based closed-loop feedback for autonomous vehicles which improves performance compared to CNN-based approaches. Their work illustrates that the proposed model can learn to conclude steering wheel angles for the lateral control of self-driving vehicles through supervised pre-training and subsequent reinforced closed-loop feedback with images from a dashcam camera.

[9] Michael G. Bechtel and Elise McEllhiney despite platform which uses a convolutional neural network (CNN), which takes pictures from a forward-looking camera as information and produces vehicle directing points as yield. Utilizing their foundation, we break down the Pi 3's registering abilities to help start to finish profound learning-based ongoing control of self-sufficient vehicles. To guarantee the CNN duty, they surveyed top tier hold distributing memory move speed gagging

systems on the Pi 3.

[10] Ailec Wu, Abu Hasnat Mohammad Rubaiyat presented the preliminary results on developing a weighted N-version programming (NVP) plot for guaranteeing versatility of AI-based guiding control calculations. The proposed conspires planned dependent on the combination of yields from three excess Deep Neural Network (DNN) models.

Reference	**Methodology**	**Result**	**Limitations**
[6] A Convolutional Neural Network Approach Towards Self-Driving Cars	CNN	CNN learns the maximum variable features from the camera input, the steering angle, which is converted to angular velocity for steering	Work is needed to improve the robustness of the network.
[7] End to End Learning for Self-Driving Cars	CNN	Learns significant features from sparse training steering angle	Accuracy of Model
[8] End to End steering Controller with CNN based Closed loop Feedback for Autonomous Vehicles	CNN	Accomplished robust steering control even in partially observed scenarios	The simulation could not perform proper lateral control in lane keeping
[9] A low cost Deep neural Network-based Autonomous car	CNN Raspberry pi3	Out performed other models in speed and most cost efficient	Overfitting issue ,high power consumption
[10] Model fusion :Weighted N-version programming for resilient autonomous vehicle steering control	DNN,CNN and LSTM/RNN	Weighted model fusion on average achieved 40% accuracy as compared to separate algorithms	High rate of failure

Table 2.2: Literature Survey Table

Chapter 3

Methodology

3.1 Dataset

The dataset is taken from the sullychen [13] dataset which contains a total of45406 images *Figure*3.1. Images of the dataset are recorded from a car dashcam with labelled steering angles. It contains continuous images of the video. Dataset is later plotted into two categories: train and test. After splitting the dataset convolutional neural networks (CNN) is used to train the model.

Figure 3.1: Images recorded from a car dashcam with labeled steering angles

3.2 Image Preprocessing

Images of the dataset are divided into training and test. Images of both sets are cropped from the top and only lower (150 pixels) part is taken we are only con-

cerned with roads in images. Then resized from 455 X 150 to 200 X 66, after which normalized (dividing by 255) for easier computation.

3.3 Model Architecture using CNN

The CNN model architecture is discussed in detail and how it is used in predicting the Steering Wheel Angle as showin in *Figure*3.2 *Figure*3.3 .

3.3.1 Convolutional layer

t contains various filters; each filter extracts different kinds of features and gives an activation map and multiple activation maps are combined by stacking to form output volume so the CNN layer takes input a volume and produces an output volume of different shape. Convolution defines a process by which one real-valued function works on another real-valued feature to yield a new real-valued role. Let two real-valued function be $f(t)$ and $g(t)$, where t signifies consistent time. The convolution process is commonly indicated as:-

$$f(t) * g(t) = (f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{3.1}$$

Similarly, convolution is a commutative operation that implies, $(f * g)$ which is same as $(g * f)$,

$$(g * f)(t) = (f * g)(t) = \int_{-\infty}^{\infty} g(\tau)f(t-\tau)d\tau \tag{3.2}$$

The distinct partners of the two equations can be represented as $f(k)$ and $g(k)$, where k signifies a discrete example of time

$$f(k) * g(k) = (f * g)(k) = \sum_{\delta=-\infty}^{\infty} f(\delta)g(k-\delta) \tag{3.3}$$

and

$$(g * f)(k) = (f * g)(k) = \sum_{\delta=-\infty}^{\infty} g(\delta)f(k-\delta) \tag{3.4}$$

These definitions are fundamentally the same as the definition of connection that exists between two function[14].

3.3.2 Stride, Padding and Pooling

Stride: Filters can have different size as well as movement. Stride defines how a filter should move across the image. In other words, No. of pixels we skip each time is called stride.

Padding: The convolution operation we have seen reduces the height and weight of the original image but sometimes we want the output image to have the same size as the input image. So, we can achieve this by adding 0 value pixels(neurons) outside the original image, this is called Padding.

Pooling: Pooling is performed after Convolution Operation. There are two types of pooling layers - Average Pooling and Max Pooling.
Max-pooling layer: slides a window over the input and stores the max value of the window in the output. The max-pooling operation is expressed as

$$\mathrm{P4}^k = MaxPool(C3^k) P4^k_{i,j} = max \begin{pmatrix} C3^k_{(2i,2j)'} & C3^k_{(2i+1,2j)'} \\ C3^k_{(2i,2j+1)'} & C3^k_{(2i+1,2j+1)'} \end{pmatrix}$$

where (i, j) are indices of $k^t h$ feature map of the output, and k is the feature map index[15].
Average-pooling layer: slides a window over the input and stores the average value of the window in the output. It helps to reduce computation by discarding 75% of the neurons (assuming 2X2 filters with a stride of 2). And It makes feature detectors more robust. It consists of no parameters for learning, only hyperparameters such as filter size and type of pooling.

3.3.3 Activation Functions

The activation function is a mathematical "gate" in between the input feeding the current neuron and its output going to the next layer. It can be as simple as a step function that turns the neuron output on and off, depending on a rule or threshold. There are various activation functions example - Relu, sigmoid, tanh, etc. But in this model, Relu and tanh function is used. Relu stands for Rectified Linear Function.ReLu function takes the maximum value. So, it can be written as

$$f(x) = max(x, 0) \tag{3.5}$$

The rectifier function will help us to increase the non-linearity in our images. Tanh (Hyperbolic Tangent) is a non-linear activation function that compresses all its inputs to the range [-1, 1].

$$f(x) = \frac{e^x - e^- x}{e^x + e^- x} \tag{3.6}$$

$$f'(x) = \frac{\delta f(x)}{\delta x} = 1 - (f(x))^2 \tag{3.7}$$

The reason we want to do that is that images are naturally non-linear. Tanh function is zero-centred so that the gradients are not restricted to move in a certain directions[17]. When you look at any image, you'll find it contains a lot of non-linear features (e.g., the transition between pixels, the borders, the colours, etc.).

3.3.4 Optimizer

There is a lot of optimizers example - Adagrad, adadelta, Adam etc. In this project, Adam is used. Adam stands for Adaptive Moment Estimation. With this optimizer, one can train the neural network in less time and more efficiently. The intuition behind Adam is that just because we can jump over the local minima, we don't want to roll so fast. we want to decrease the velocity a little bit for a careful search. It is not only storing an exponentially decaying average of the past squared gradient but also stores an exponentially decaying average of the past gradient

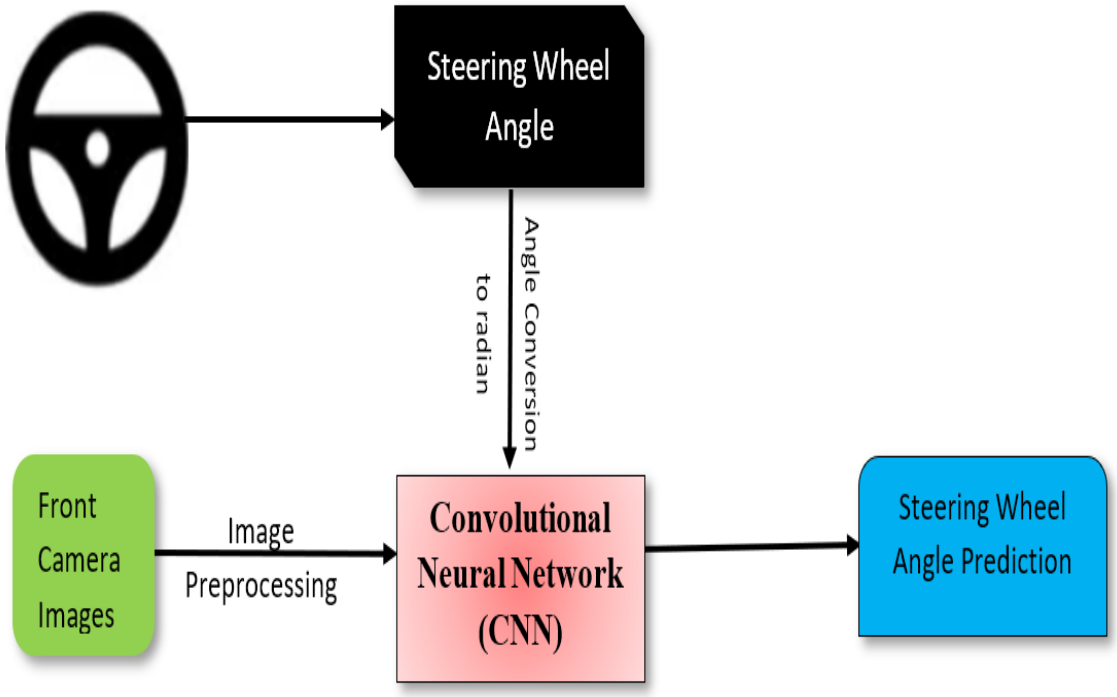

Figure 3.2: Figure describing how dash cam images and their corresponding steering wheel angle(in radians) are feeded into CNN to predict Steering Wheel Angle

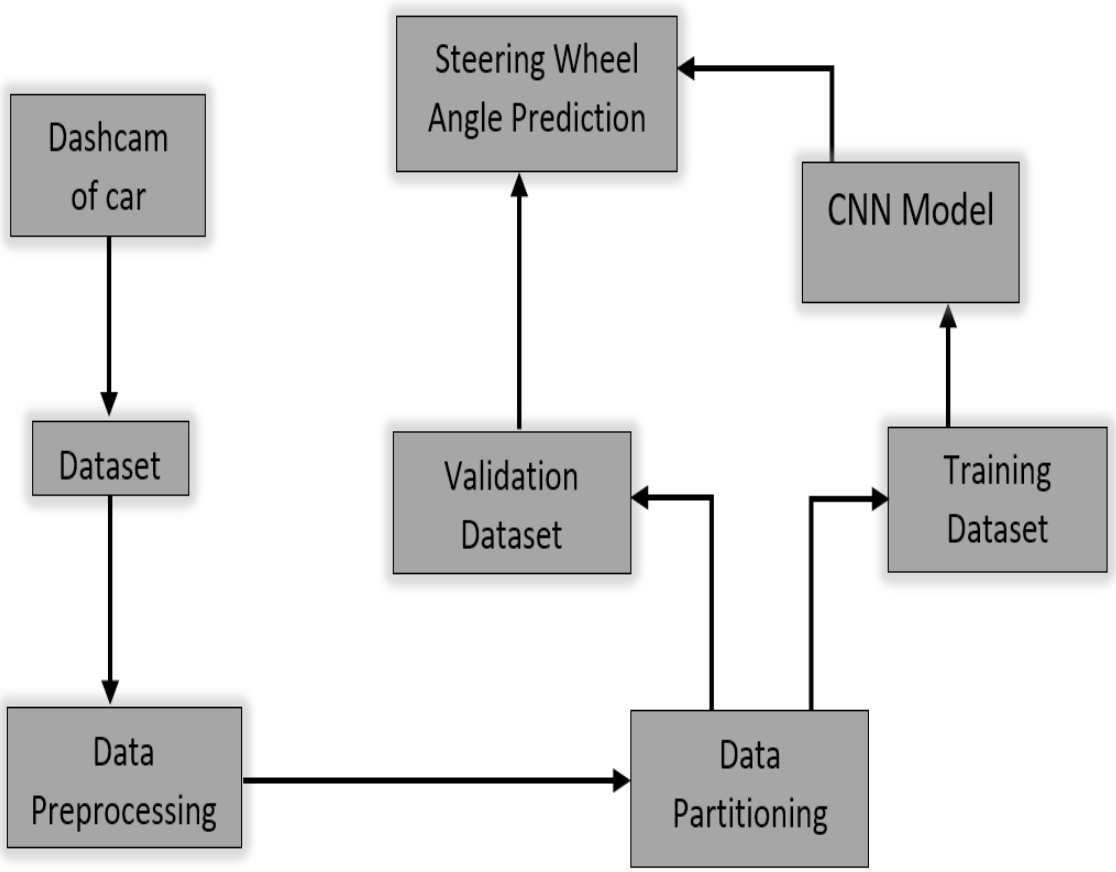

Figure 3.3: Overview of Model

Chapter 4

Implementation and Results

4.1 Environment

Code is implemented in Python. The environment has Tensorflow, Keras and some other commonly used libraries of python. TensorFlow[12] is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in DL and developers easily build and deploy DL powered applications. Keras is a high-level neural networks API that runs on top of TensorFlow.

4.2 Implementation

The dataset is used in this model contains a CSV file that contains angles of a steering wheel Angles in the CSV file is inverse of the turning radius but the steering wheel angle is proportional to the inverse of the turning radius so the steering wheel angle in radians used as the output. we split the dataset into train and validation set with an 80:20 ratio. After that dataset should be loaded in batches for better efficiency that's why data is loaded in batches of size 100. In this model Adam optimizer is used for training and The activation function is used in this model is tanh. CNN architecture *Figure*4.1 is used in this model. In this model, there are 5 Convolutional Layer with some stride and padding and 3 Fully Connected layers.

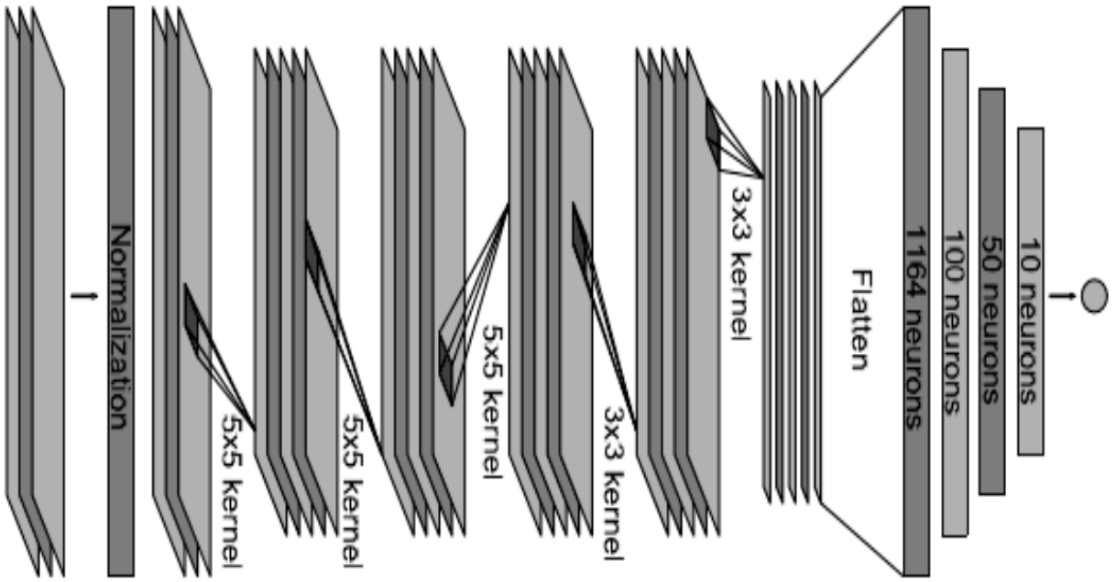

Figure 4.1: CNN architecture of Proposed Model. The network has about 27 million connections and 250 thousand parameters. [5]

4.3 Results

4.3.1 CNN Model

Loss *Figure*4.2 is decreased to 0.5%. According to the results, it can be said that The model was able to detect useful road features on its own. As seen in fig5.1 there is a stark difference in the validation Mean Square Error(MSE) loss of the model after 30 epochs. The model is able to clone the behaviour of human of the human driver very efficiently.

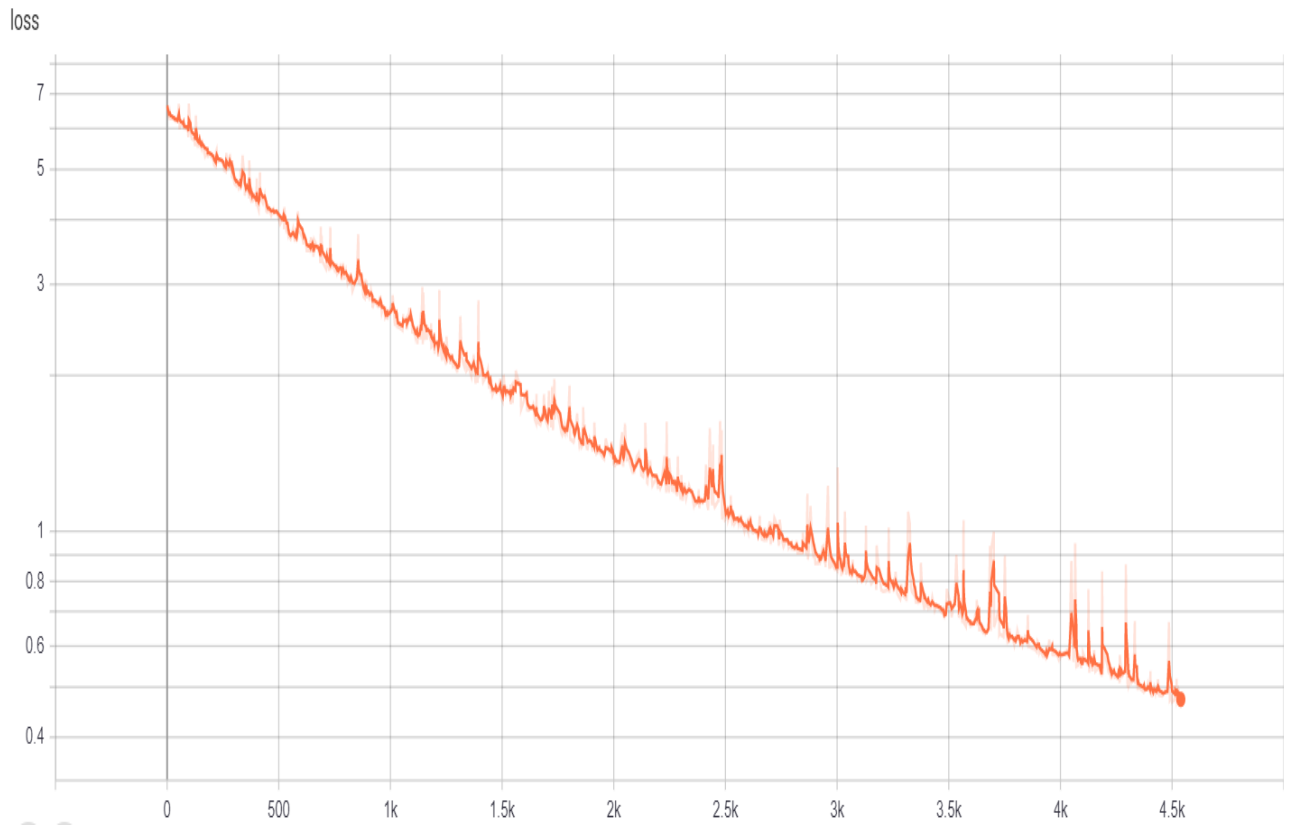

Figure 4.2: Loss is plotted on the vertical axis and visulaised in TensorBoard[16] with no. of steps in the horizontal axis(1 epoch = 100 steps)

4.3.2 CNN Model + Transfer Learning

The work of transfer learning[18] is simple it takes a model which was trained on a large dataset and transfers its weights and knowledge to a smaller dataset. In this, we train only the last few layers to make predictions after freezing early convolution layers of the network so that the previously trained model could be used in this model. But here a problem occurs When we add more layers to our deep neural networks, the performance becomes stagnant or starts to degrade. This happens due to the vanishing gradient problem[19]. Here Resnet50[20] model is used to train the model. Resnet50[20] is a 50 layer Residual Network.

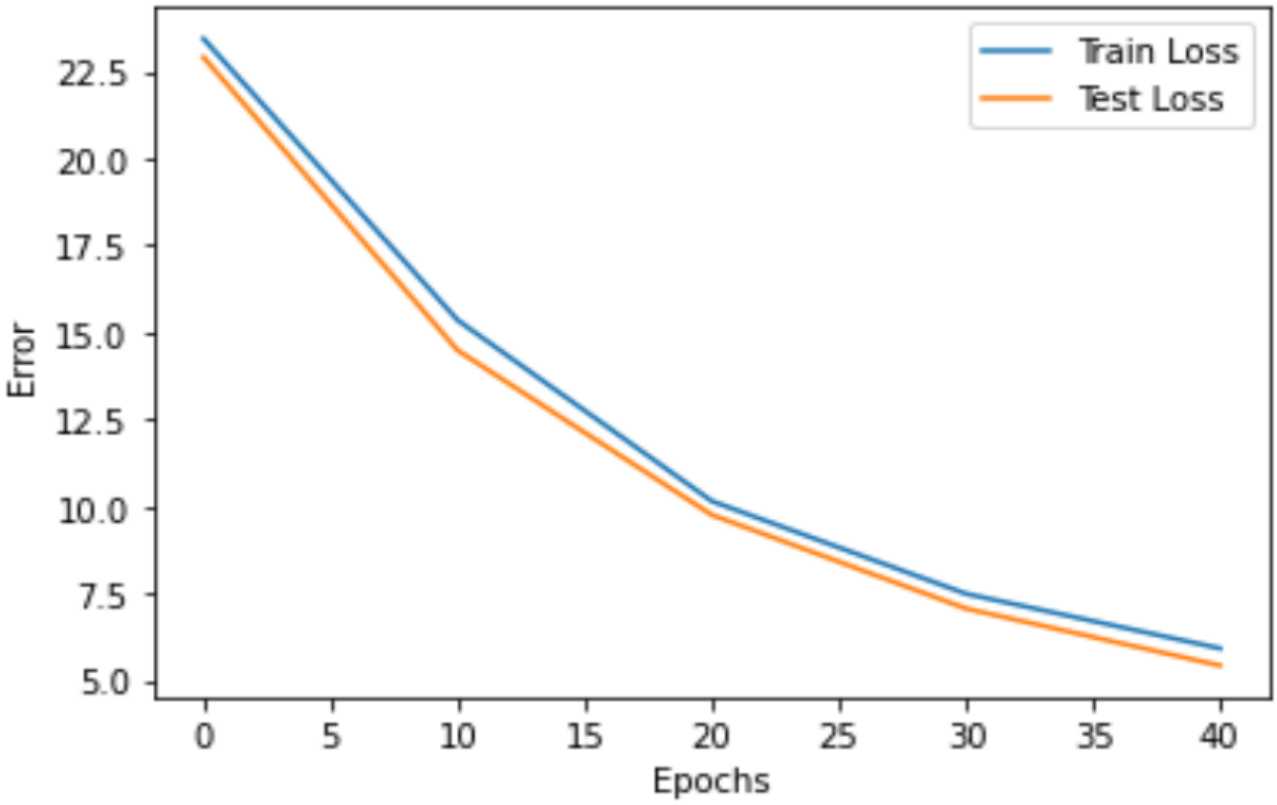

Figure 4.3: Loss is plotted on the vertical axis and Epochs in the horizontal axis

4.3.3 Comparison of Models

As seen in the figure first loss is decreased and if epochs will increase loss curve will be stagnant. This is due to the Vanishing gradient problem. The vanishing gradients problem may be manifest in a Multilayer Perceptron by a slow rate of improvement of a model during training and perhaps premature convergence e.g. continued training does not result in any further improvement. Inspecting the changes to the weights during training, we would see more change (i.e. more learning) occurring in the layers closer to the output layer and less change occurring in the layers close to the input layer. That why here we will prefer the simple CNN model over the CNN+transfer learning model (*Figure*4.3).

4.3.4 Code Implementation

The simulator takes pre-recorded videos from a dashcam of a human-driven car that generates images of our dataset. These were used in our dataset. These test videos are time-synchronized with recorded steering commands generated by the human driver. This demonstrates that the CNN learned to detect useful road features on its own, i.e., with only the human steering angle as a training signal. We never explicitly trained it to detect the outlines of roads. The predictions of the model are shown in the following figures *Figure*4.4, *Figure*4.5, *Figure*4.6, *Figure*4.7.

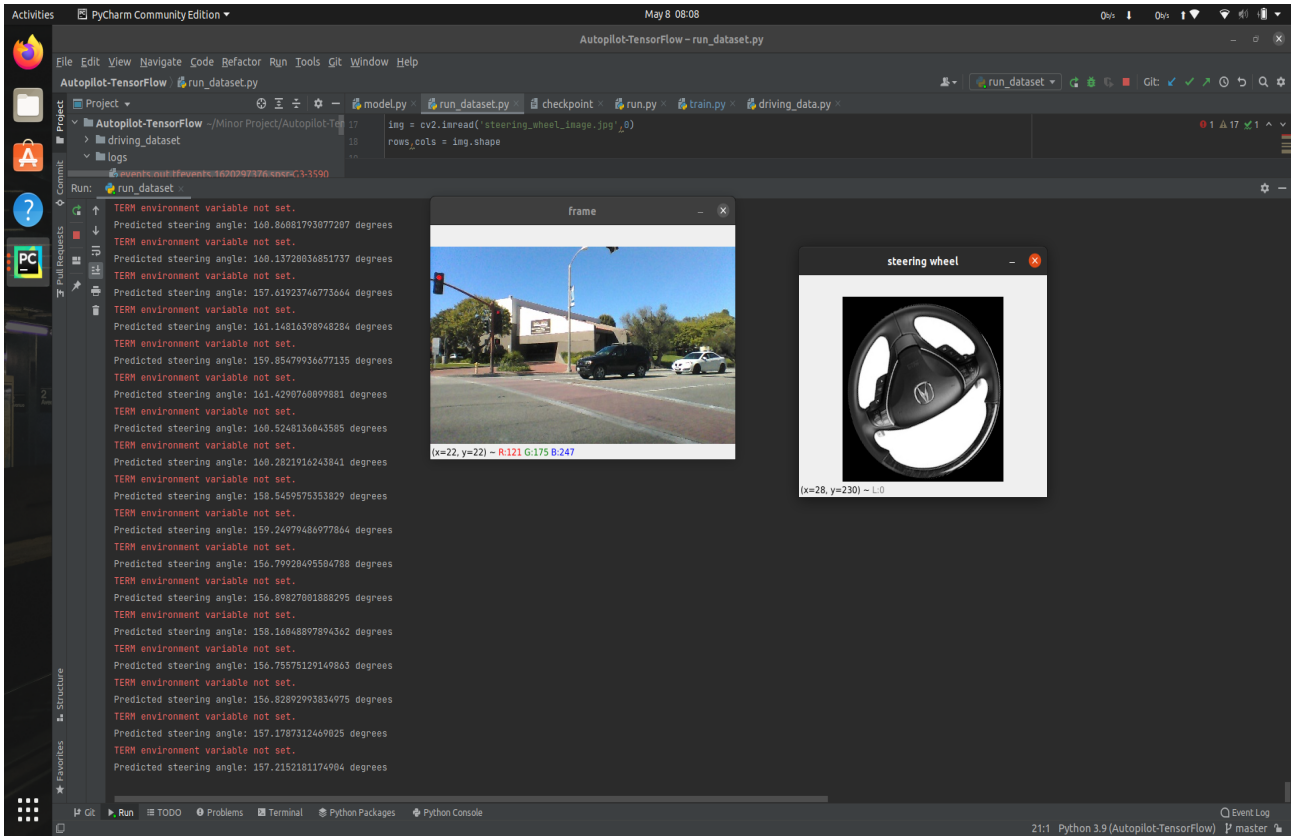

Figure 4.4: Model Predicting - 1

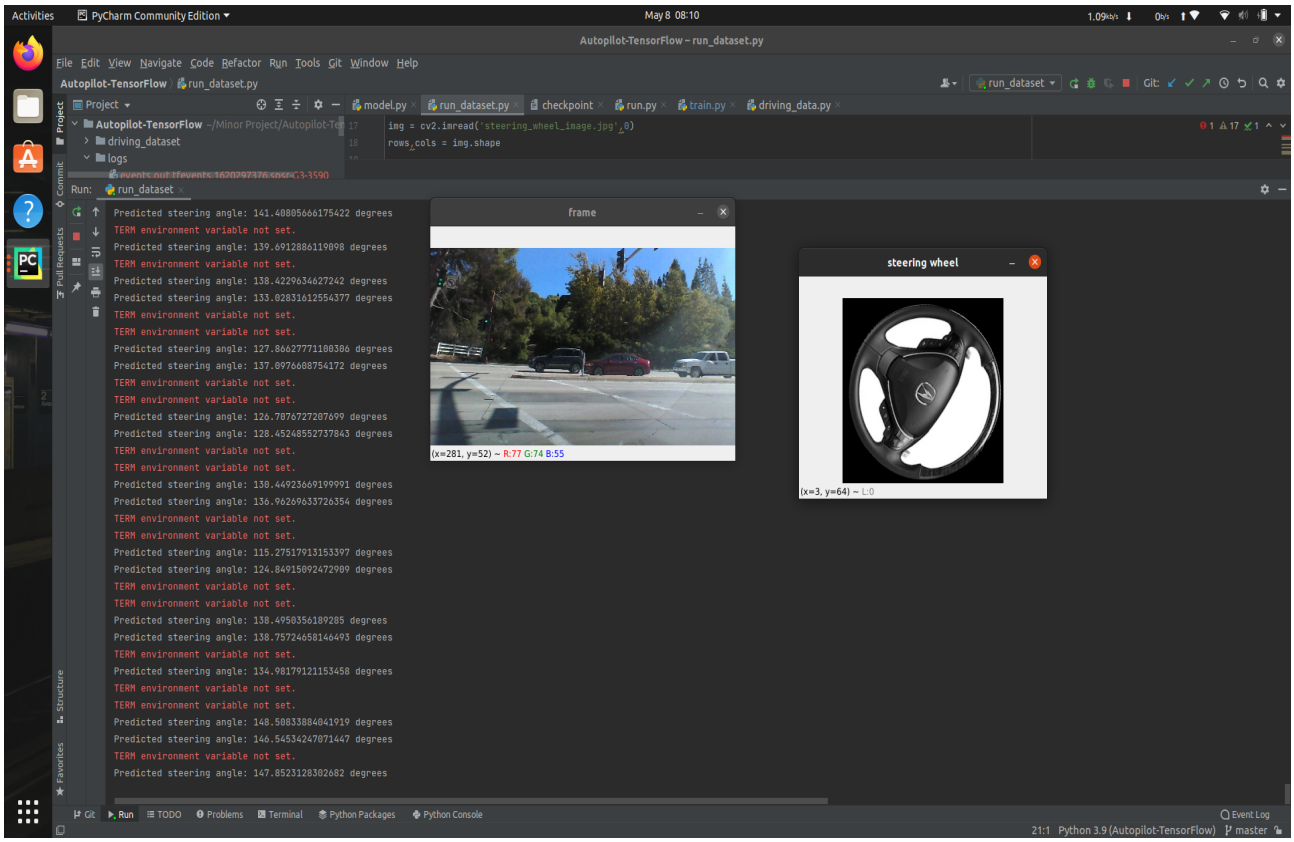

Figure 4.5: Model Predicting - 2

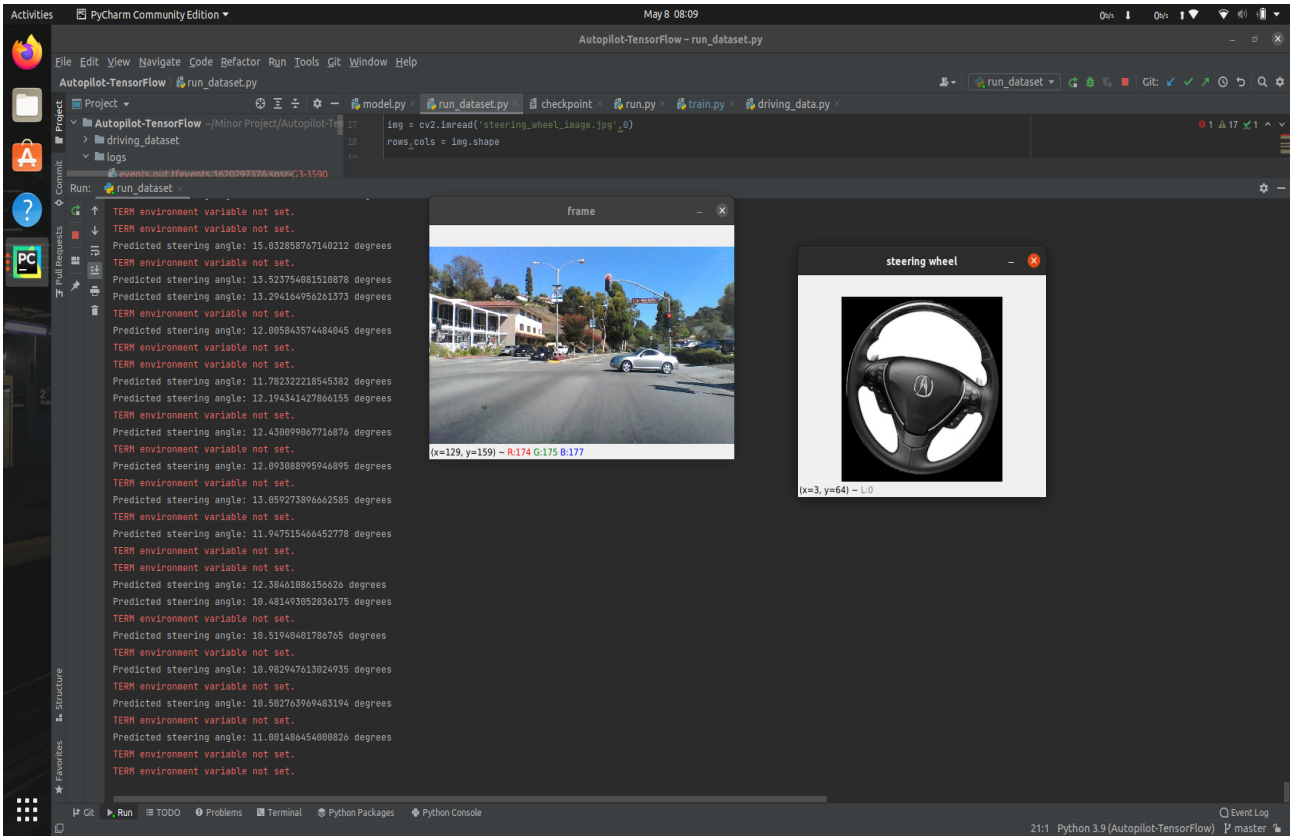

Figure 4.6: Model Predicting - 3

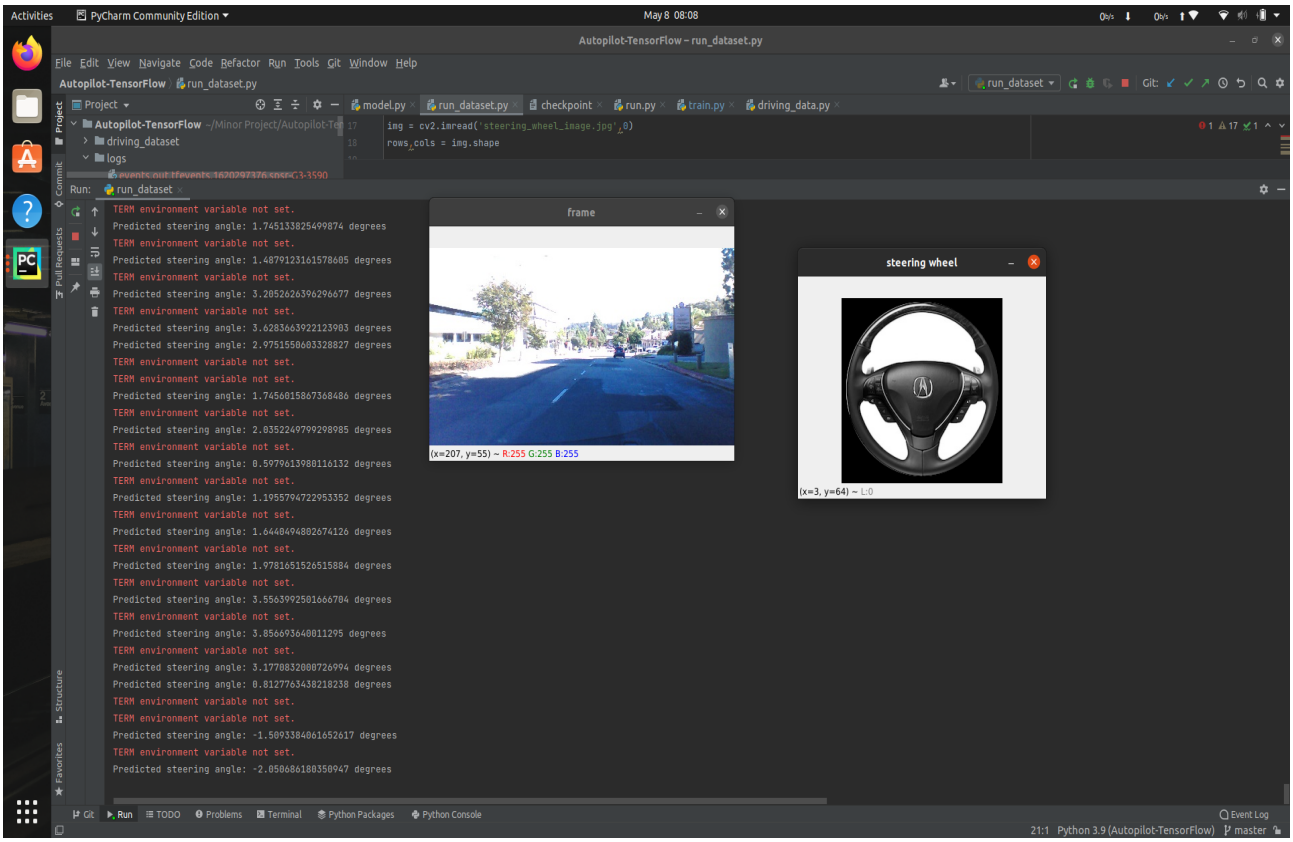

Figure 4.7: Model Predicting - 4

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

This work supports the notion that Autonomous Vehicles can work using convolutional neural networks trained model. This work exhibited that CNNs are able to learn without decomposition into road detection or lane detection, semantic abstraction, path planning, and control. The CNN is able to learn road and lane characteristics from a sparse training signal.

This project implemented a combination of computer vision techniques and a CNN model that can minimize the unsafe behaviour of Autonomous Vehicles. Much of implementation time was spent on setting up the environment. We can infer from the outcomes that deep learning is one of the safest and accurate approaches for autonomous vehicles, therefore we believe that future projects will profit incredibly from presenting a framework that uses the merits of computer vision-based techniques with the computational potential of the CNN model.

5.2 Future Scope

For future work, we would like to add other features such as break and acceleration and we will also add more training data with different scenarios so that our model could be more robust. Also, increasing and decreasing light conditions in images can be used to increase the dataset. Implementing the CNN in the physical car could have a great impact on the accuracy of the model and with the use of a hybrid model (CNN and RNN etc.) accuracy should be increased significantly.

References

[1] Santokh Singh ,"Critical reasons for crashes investigated in the national motor vehicle crash causation survey," Nat. Academies-Nat. Center Statist.

[2] M. Z. Islam, "Self-driving car: A step to the future of mobility," BRACUniv., Dhaka, Bangladesh, Tech. Rep. 10361/11027, 2018.

[3] J. Woo, C. Yu, and N. Kim, "Deep reinforcement learning-based controller for path following of an unmanned surface vehicle," Ocean Eng., vol. 183, pp. 155166, Jul. 2019.

[4] Denis Wolf , Carlos M. Massera ,Valdir Grassi Jr ,Fernando Osorio, "Longitudinal and lateral control for autonomous ground vehicles",Conference: 2014 IEEE Intelligent Vehicles Symposium At: Dearborn, MI, USA Volume: 588 - 593 June 2014.

[5] Mohamed A. A. Babiker, Mohamed A. O. Elawad,Azza H. M. Ahmed , "Convolutional Neural Network for a Self-Driving Car in a Virtual Environment" 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE19).

[6] Agnihotri, Akhil Saraf, Prathamesh Bapnad, Kriti. (2019). A Convolutional Neural Network Approach Towards Self-Driving Cars.

[7] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016, arXiv:1604.07316. [Online]. Available: `http://arxiv.org/abs/1604.07316`

[8] Jhung, I. Bae, J. Moon, T. Kim, J. Kim, and S. Kim, "End To-end steering controller with CNN-based closed-loop feedback for autonomous vehicles," in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2018, pp. 617–622.

[9] M. G. Bechtel, E. Mcellhiney, M. Kim, and H. Yun, "DeepPicar: A lowcost deep neural network-based autonomous car," in Proc. IEEE 24th Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA), Aug. 2018, pp. 11–21.

[10] A. Wu, A. H. M. Rubaiyat, C. Anton, and H. Alemzadeh, "Model fusion: Weighted N-version programming for resilient autonomous vehicle steering

control," in Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW), Oct. 2018, pp. 144–145.

[11] M. Abe and W. Manning, "Steering system and vehicle dynamics," in Vehicle Handling Dynamics, M. Abe andW. Manning, Eds. Oxford, U.K.: Butterworth-Heinemann, 2009, ch. 5, pp. 149164.

[12] TensorFlow https://www.tensorow.org/

[13] SullyChen/driving-datasets: A collection of labeled car driving datasets (github.com) https://github.com/SullyChen/driving-datasets

[14] A. V. Joshi, Machine Learning and Artificial Intelligence. Cham, Switzerland: Springer, 2019.

[15] M. A. Wani, Advances in Deep Learning, vol. 57. Cham, Switzerland: Springer, 2020.

[16] Tensorboard https://www.tensorflow.org/tensorboard

[17] Activation Functions and Optimizers for Deep Learning Models https://www.exxactcorp.com/blog/Deep-Learning/activation-functions-and-optimizers-for-deep-learning-models

[18] Deep Learning using Transfer Learning -Python Code for ResNet50 https://towardsdatascience.com/deep-learning-using-transfer-learning-python-code-for-resnet50-8acdfb

[19] Vanishing gradient problem https://en.wikipedia.org/wiki/Vanishing_gradient_problem

[20] ResNet50 https://towardsdatascience.com/deep-learning-using-transfer-learning-python-code-for-resnet50-8acdfb