Steering Wheel Angle Prediction in Autonomous Driving

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Aditya Sanjay Nagulpelli (112115010)

Semester: 4

Name of Department: Department of Computer Science and Engineering

Indian Institute of Information and Technology, Pune

(An Institute of National Importance by an Act of Parliament)

APRIL 2023

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"Steering Wheel Angle Prediction in Autonomous Driving"** submitted by **Aditya Sanjay Nagulpelli** bearing the **MIS No: 112115010**, in completion of his project work is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in the **Department of Computer Science and Engineering**, Indian Institute of Information Technology, Pune (IIIT Pune), during the academic year **2022-23**.

Dr. Sanjeev Sharma

Head of Department

Assistant Professor

Department of Computer Science Engineering

IIIT Pune

Project Viva-voce held on – 25/04/2023

Undertaking for Plagiarism

I **Aditya Sanjay Nagulpelli** solemnly declare that research work presented in the **report/dissertation** titled **"Steering Wheel Angle Prediction in Autonomous Driving"** is solely **my** research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete report/dissertation has been written by **me**. I understand the zero-tolerance policy of **Indian Institute of Information Technology Pune** towards plagiarism. Therefore, **I** declare that no portion of my **report/dissertation** has been plagiarized and any material used as reference is properly referred/cited. I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of the degree, the Institute reserves the rights to withdraw/revoke my **B.Tech** degree.

Aditya Sanjay Nagulpelli

Signature

Date

Conflict of Interest

Manuscript title: “Steering Wheel Angle Prediction in Autonomous Driving”

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers’ bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Aditya Sanjay Nagulpelli

Signature

Date

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all the individuals who have supported me throughout the completion of this project.

Firstly, I would like to extend my sincere appreciation to my teachers who provided me with valuable guidance and encouragement throughout the project. Their wisdom, knowledge, and expertise have been invaluable in helping me to develop my ideas and refine my approach.

I would also like to thank my friends who offered their time and expertise in providing feedback and constructive criticism that was instrumental in improving the quality of my work. Their willingness to lend a hand and offer their support was greatly appreciated.

Lastly, I would like to express my heartfelt gratitude to my parents, whose unwavering support, encouragement, and belief in me have been a constant source of strength throughout this project. Their love and dedication have been the driving force behind my success, and I am forever grateful for their guidance and support.

Thank you all for your contributions, your patience, and your support throughout this project. I could not have done it without you.

Abstract

Autonomous driving technologies have gained significant attention in recent years as a promising solution to address transportation challenges, including increased road congestion, accidents, and environmental impact. The development of self-driving cars has been a long-standing goal, with the Defense Advanced Research Projects Agency (DARPA) launching the Grand Challenge in 2004, aiming to develop autonomous ground vehicles that could navigate without human intervention. The advent of self-driving cars has the potential to revolutionize the automotive industry and enhance mobility, safety, and efficiency.

The DAVE-2 system by NVIDIA is a prominent example of a self-driving car technology that utilizes the Convolutional Neural Network (CNN) architecture for perception and decision-making tasks. However, the performance of these systems heavily relies on the quality of the CNN architecture used. This project aims to investigate the performance of various CNN architecture optimize the architecture for improved performance, leading to better autonomous driving technologies and safer vehicles.

A new Convolutional LSTM model is proposed in this study with end-to-end training. This model was implemented on 300MB dataset provides data images and labeled semantic segmentations captured via Udacity self-driving car simulator. The data consist of 11,790 images and has been imported from Kaggle. Mean Squared Error was used as the performance measurement of the study. The experimental works reveal that CNN with LSTM shows potential in the autonomous driving based on input from front cameras. The results of which showed the LSTM based CNN models to be quite efficient when compared to the established base models on only front cameras images.

However, these promises are yet to be fully realized due to various technical and societal limitations. The paper discusses the technological limitations of autonomous driving, including challenges in perception and security.

The paper concludes by highlighting the need for further research and development to overcome these limitations and realize the full potential of autonomous driving.

Keywords: Autonomous Driving, CNN, LSTM, Steering Angle

TABLE OF CONTENTS

List of Figures / Symbols/ Nomenclature

LSTM - Long short-term memory is an artificial neural network.

VGG16 - Visual Geometry Group is a Convolutional Neural Network with 16 layers.

CNN – Convolutional Neural Network.

MLP – The multi-layer perceptron (MLP) is another artificial neural network process containing a number of layers.

SSD - A solid-state drive (SSD) is a new generation of storage device used in computers.

Relu - Rectified linear activation unit.

Elu - Exponential Linear Unit

Chapter 1
Introduction

1.1 Overview of Work

In this project we have discussed a new approach towards the problem using Convolutional LSTM.
We have also compared existing models like VGG19, ResNet50 using transfer learning. A CNN architecture was also developed by NVIDIA in this subject, often regarded as PilotNet, has also been compared. Our model has been partially inspired by this model but has proven to give better results on Udacity self-driving car simulator dataset.
However, this model is far from perfect and there is substantial research that still needs to be done on the subject before models like these can be deployed widely to transport the public.

1.2 Motivation of the Work

The motivation for research in the field of autonomous driving is primarily due to interest in the field of autonomous driving and a desire to contribute to improve the safety and reliability of self-driving cars and advance the technology towards widespread adoption. Despite significant progress in recent years, autonomous driving technologies still face significant challenges, and there is a need to address these limitations to realize the full potential of self-driving cars.

Predicting the steering angle is a crucial aspect of the approach, known as end-to-end learning. By training deep neural networks to predict the steering angle using only this information, the networks can automatically identify and extract relevant features from the environment, such as the position of the road, enabling the car to make accurate predictions.

By focusing on the performance of the CNN architecture used for perception and decision-making tasks, this research aims to contribute to the development of better autonomous driving technologies and improve the safety and reliability of self-driving cars. By optimizing these systems, this research can help make autonomous driving systems more effective, efficient, and responsive to unexpected scenarios.

The outcomes of this research can have significant implications for the future of transportation and can help improve the safety and sustainability of our cities. By addressing the performance of autonomous driving technologies, this research can pave the way for the widespread adoption of self-driving cars, which can have significant benefits in terms of reducing traffic congestion, improving road safety, and decreasing carbon emissions.

1.3 Literature Review

Using neural network for autonomous vehicle navigation was pioneered by Pomerleau (1989) who built the Autonomous Land Vehicle in a Neural Network (ALVINN) system. The model structure was relatively simple, comprising a fully-connected network, which is tiny by todays standard. The network predicted actions from pixel inputs applied to simple driving scenarios with few obstacles. However, it demonstrated the potential of neural networks for end-to-end autonomous navigation.

Last year, NVIDIA released a paper regarding a similar idea that benefited from ALVINN. In the paper [6], the authors used a relatively basic CNN architecture to extract features from the driving frames. The layout of the architecture can be seen below

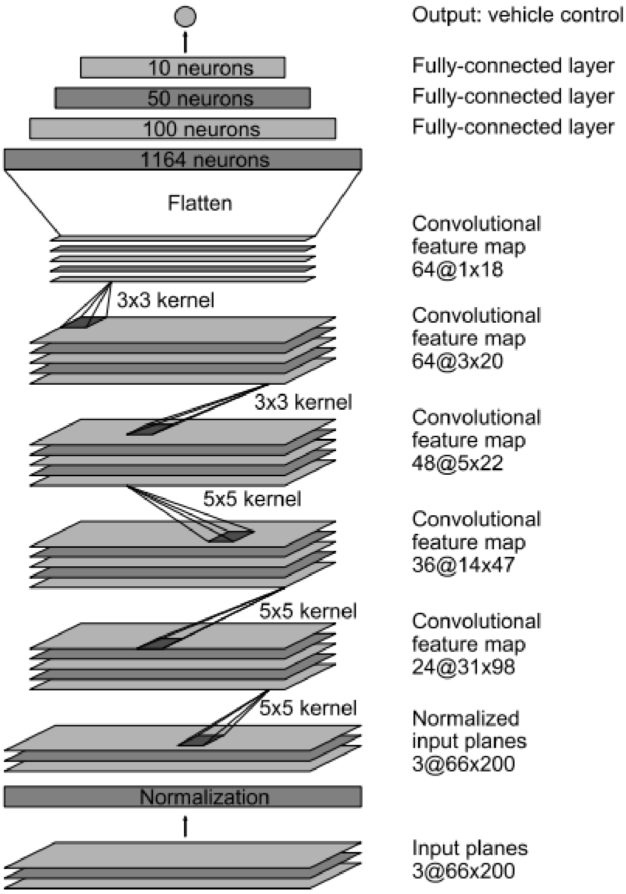


Figure 4: CNN architecture. The network has about 27 million connections and 250 thousand parameters.

[6]

Recently, more attempts on using deep CNNs and RNNs to tackle the challenge. Another model proposed in [2] is,

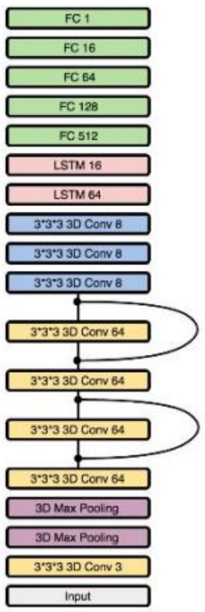


Figure 2. 3D convolutional model with residual connections and recurrent LSTM layers

The model consists of a few initial layers to shrink the size followed by ResNet like blocks of 3D convolutions with spatial batch normalization (only two of these in the trained model). Due to computational restraints, shrink layers are added to make the input to the LSTM layers much smaller. Only two levels of recurrent layers are used due to the speed of computation on these layers being much slower due to parts that must be done in a serial manner. The output of the recurrent layers is fed into a fully connected stack that ends with the angle prediction. All of these layers use rectified linear units, ReLUs, as their activation except the LSTM layers Spatial batch normalization is used on the convolutional layers. The LSTM layers used the hyperbolic tangent function as their activation, which is common to use in these types of layers.

[7]

Another CNN LSTM based network is shown below

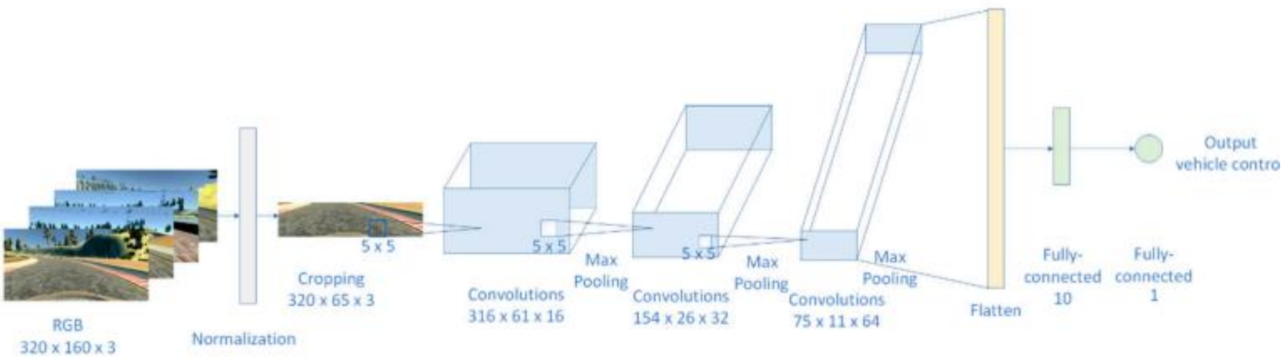


Figure 12. Architecture of proposed end-to-end deep neural network, J-Net, used for autonomous driving. The network has three convolutional layers with 16, 32, and 64 feature maps, one flattened layer, and two fully-connected (dense) layers. Max pooling is placed after every convolutional layer.

[8]

The above architecture uses three convolution layers of kernel size (5,5) each with 16, 32, 64 feature maps respectively. It then uses Max pooling with (2,2) with the intention of downsampling. However, this downsampling of an image may cause the loss of some important features since it removes a lot of information. The goal of the proposed deep neural network was to downsized the needs for real-time inference hardware in terms of computational power, cost, and size. Hence, they have cropped the image significantly. We do not find this useful as there maybe traffic sign and many other things we may miss due to this.

1.4 Research Gap

While there has been significant development in quality and number of architectures proposed taking camera image as input for predicting steering angle there is still a lot to be explored. This proposed model proves that there is wide scope for improvement in the subject. Our model is primarily LSTM where images are given as input and various operation along with convolution are performed in LSTM cell. As we will se further that this model has proved to perform better than conventional NVIDIA (PilotNet) model. The PilotNet architecture has proven to be significantly better than existing models. But driving can be considered a task with sequential input from sensors (camera) and hence we propose the use of LSTM to improve the performance of the model. Some progress has been made in the same direction while much it to be done in order to find the optimal structure.

The results obtained are at the initial stage of implementation of the idea and much work is to be done on the model. Due to limitation in resources the true potential the architecture remains to be discovered in future.

Chapter 2

Problem Statement- Optimizing Convolutional Neural Network Architecture for Autonomous Driving

While significant advancements have been made in autonomous driving technologies, the effectiveness of these systems still heavily relies on the quality of the Convolutional Neural Network (CNN) architecture used for perception and decision-making tasks. The aim of this research study is to investigate the performance of different CNN architectures for autonomous driving, and identify ways to optimize the architecture for improved performance using techniques such as transfer learning and LSTM network.

2.1. Research Objectives

The development of autonomous driving technologies has seen significant advancements in recent years, with the DAVE-2 system by NVIDIA being a prominent example. However, the performance of these systems is heavily reliant on the quality of the Convolutional Neural Network (CNN) architecture used for perception and decision-making tasks. The effectiveness of the CNN architecture is impacted by various factors, including the choice of model, the size and quality of the training data, and the parameters used during training. This project aims to explore the performance of different CNN architectures and identify ways to optimize the architecture for improved performance. The project will investigate the use of transfer learning and other techniques to enhance the performance of the CNN architecture and ultimately improve the effectiveness of the DAVE-2 system for autonomous driving.
The PilotNet architecture has proven to be significantly better than existing models. But driving can be considered a task with sequential input from sensors (camera) and hence we propose the use of LSTM to improve the performance of the model. Some progress has been made in the same direction while much it to be done in order to find the optimal structure.

2.2. Methodology of the Work

Dataset pre-processing was performed to improved usability of our dataset. After performing Exploratory Data Analysis, we found our data was imbalanced and was modified accordingly. Data Augmentation was also performed in order to increase size of our dataset and introduce variation to the same.

We initially attempted to implement LSTM to the output of CNN layers, which resulted in high loss. Therefore, we modified our approach by converting the 3D data into 5D during image pre-processing and giving this as input to the ConvLSTM cell. The next step was to determine the number of neurons, strides, and other parameters. Due to time constraints, we attempted to stick to the parameters proposed in PilotNet. However, we found that this approach did not provide us with satisfactory results, so we reduced layer 3 and 5 and retrained the model. The new architecture consisted of 3 ConvLSTM layers and 4 MLP layers, which yielded promising results. We then attempted to increase the number of ConvLSTM layers by putting layer 5 back, which required us to train the model for a longer period of time. We observed that increasing the number of layers proportionally increased the amount of data required to train the model. Although we trained the model for 100 epochs, the training and validation error converged without any sign of overfitting.

After training our model, we established a baseline for error by implementing VGG19, ResNet50, and PilotNet, which are well-established models. We utilized transfer learning to train VGG19 and ResNet50 by importing pretrained weights from the ImageNet dataset. These networks were initially developed for object detection, hence we attempted to use a model that was developed to operate autonomously for driving. Accordingly, we included PilotNet as well.

However, hyper-parameter tuning for the proposed model was not performed due to time constraints. We trained the model for 100 epochs with a batch size of 200 and validation size of 100, which took approximately 5 hours using GPU T4 X2.

All the training and testing was done on 300MB dataset which was generated on Udacity car simulator. We could have used larger data but that was decided to be a part of future scope as training such amount of data requires extensive and powerful resources.

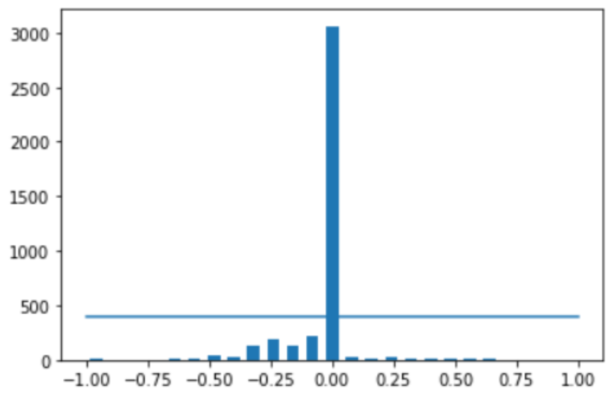
Chapter 3
Analysis and Design

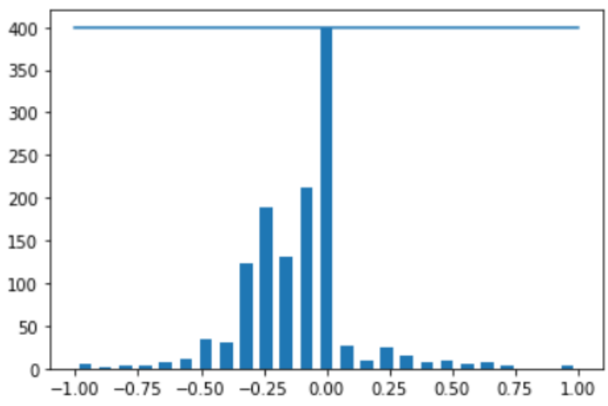
3.1) About Dataset

The data is recorded using Udacity's self-driving car simulator. It is recorded on lake track by driving 3-5 laps. The Udacity self-driving car is fitted with 3 front cameras which would take the images. The images would be saved on the SSD. The model would extract the steering angles via the vehicle's controller area network (CAN) bus at different time stamp, where it's value ranges from -1 to +1. This will be stored in a csv file along with other attributes. The attributes in our data are (center, left, right, steering angle, throttle, reverse, speed) where center, left and right columns consist of network path of images.

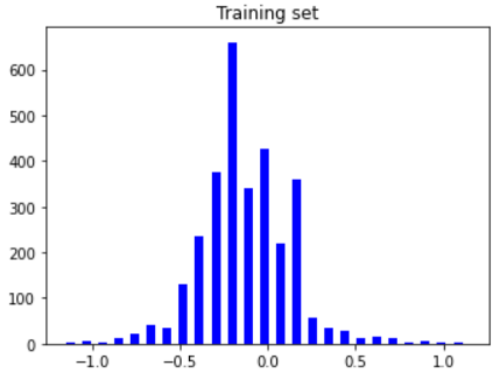
3.2) Data Prepossessing

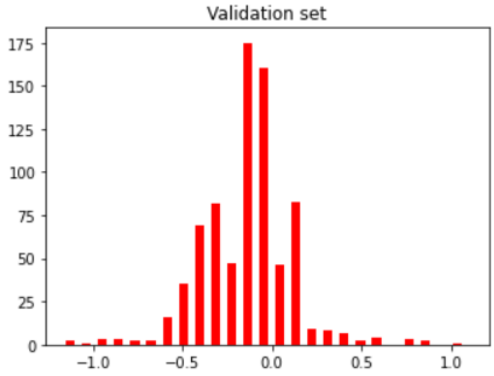
Since we drive straight most of the time the steering angle is accordingly distributed. We need to have a fair distribution in order to eliminate the bias to drive straight all the time. Hence, we have modified the data such that there are significant amount of left steering angle and right steering angle. The below histogram shows us the steering angles before and after modifying the dataset. We have also resized the data to (200,66,3) and normalize it before converting our 3D data to 5D after augmentation in order to give appropriate input to the layer.





So, our training and test dataset looks like





3.3) Data Augmentation

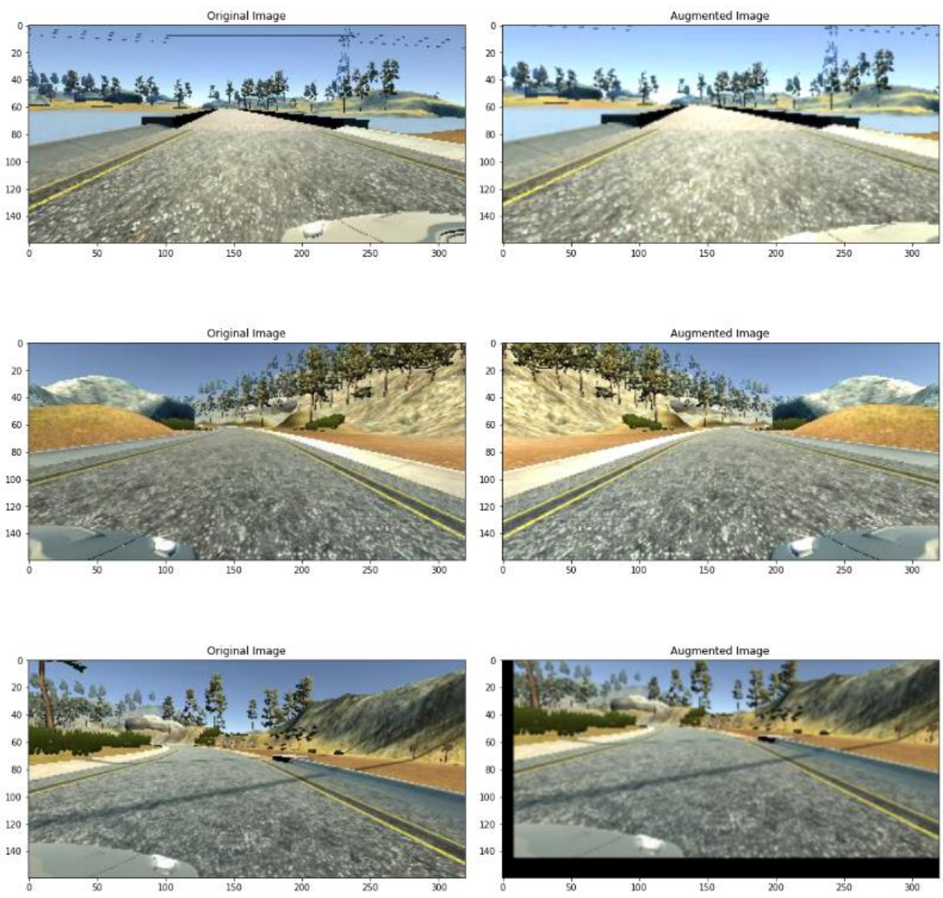
The amount of data required to train a self-driving car model for every track is simply too great to handle in practice. Additionally, it is not feasible to collect data for all weather and road conditions. Therefore, a concept for generalizing the behavior across several tracks must be developed. To introduce randomness and variation into our dataset we have augmented images based on 0.5 probability for each type of augmentation

Zoom - Relevant features are present in the lowest portion of the photos in the dataset where the road can be seen. The area of the image that is above the external environment will never be used to determine the output; it can therefore be clipped. In the training set, about 30% of the image's top third is trimmed and passed. The image below shows how an image is transformed after being cropped and resized to its original size.

Brightness - The brightness enhancement can be quite beneficial when applied generally to weather circumstances with bright sunny days or gloomy, lowlight conditions. Below is a screenshot showing the code excerpt and brightness increase. In a similar vein, I reduced the brightness for various circumstances.

Flip - The image is horizontally inverted (i.e., the dataset receives the mirror image of the original image). This serves to teach the model for turns of a similar nature on opposite sides as well.

Shift – The image is shifted horizontally and vertically by 10%.



3.4) Proposed Model

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv_lst_m2d_12 (ConvLSTM2D) (None, None, 31, 98, 24)  64896
_________________________________________________________________
conv_lst_m2d_13 (ConvLSTM2D) (None, None, 14, 47, 36)  216144
_________________________________________________________________
conv_lst_m2d_14 (ConvLSTM2D) (None, None, 12, 45, 64)  230656
_________________________________________________________________
conv_lst_m2d_15 (ConvLSTM2D) (None, None, 10, 43, 64)  295168
_________________________________________________________________
time_distributed_3 (TimeDist (None, None, 27520)       0
_________________________________________________________________
dense_12 (Dense)             (None, None, 100)         2752100
_________________________________________________________________
dropout_9 (Dropout)          (None, None, 100)         0
_________________________________________________________________
dense_13 (Dense)             (None, None, 50)          5050
_________________________________________________________________
dropout_10 (Dropout)         (None, None, 50)          0
_________________________________________________________________
dense_14 (Dense)             (None, None, 10)          510
_________________________________________________________________
dropout_11 (Dropout)         (None, None, 10)          0
_________________________________________________________________
dense_15 (Dense)             (None, None, 1)           11
=================================================================
Total params: 3,564,535
Trainable params: 3,564,535
Non-trainable params: 0
_________________________________________________________________
```

In the first layer of input, we have 24 filters with kernel size of (5,5) and stride (2,2).We have used activation function “elu” over “relu” in order to avoid dead relu, this is when a node in neural network essentially dies and only feeds a value of zero to nodes which follows it.
All the Convlstm layers have kernel initializing as “glorot_uniform”.
In the second layer we have 36 filters with kernel size of (5,5) and stride (2,2) with activation as elu.
In third and fourth layers we have 64 filters with kernel size (3,3) and strides (1,1).
Output from the fourth layer is flattened using time distribution to flatten for each timesteps.
This is now fed to a dense mlp layer of 100 neurons with elu activation and 0.5 dropout.
Next layer consists of 50 neurons with elu activation and 0.5 dropout.
Similarly,We have a layer with 10 neurons and dropout whose output finally goes to out final layer with 1 neuron (As we only predict one attribute i.e. Steering Angle). As we are performing regression, we want our output to be naturally linear.

We have named the model as larva for future reference.

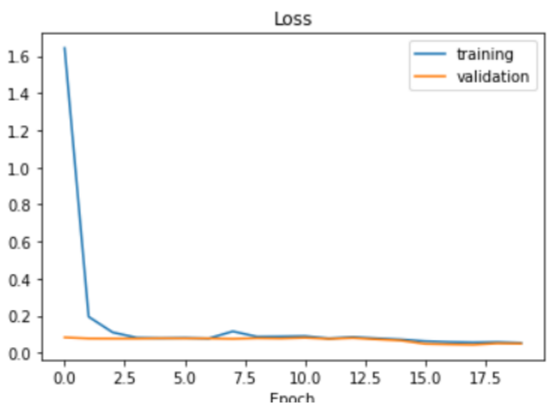
Chapter 4
Results and Discussion

Loss Function – We have used mean square error (MSE) to measure the error across all our models. It was used as it can be used interchangeably with rmse (root of mse) also and is a robust way of measuring total error. [11]

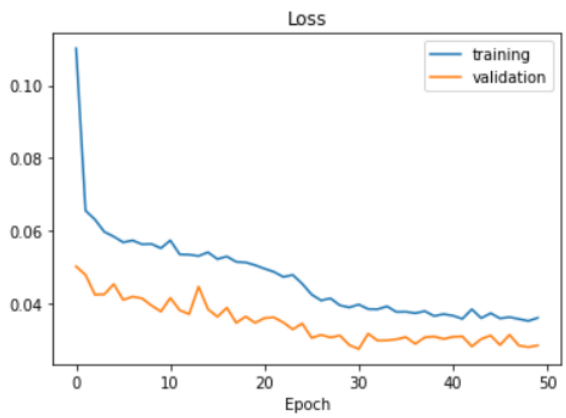
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Optimizer – The results of the Adam optimizer are generally better than every other optimization algorithm, have faster computation time, and require fewer parameters for tuning. Because of all that, Adam has been used for our network.

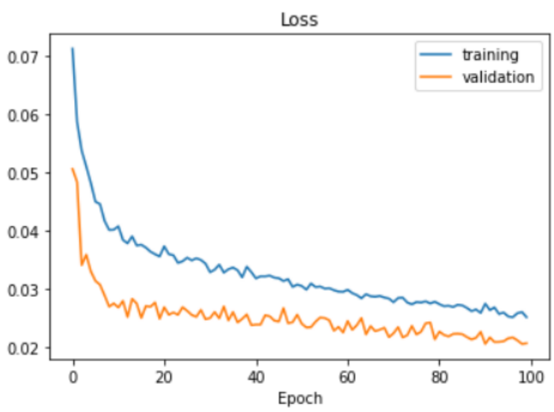
Models – We have trained and tested pretrained vgg16 (Imagenet weights), pretrained resnet50 (Imagenet weights), pilotnet (NVIDIA model) and our own model named larva. We optained best accuracy by larva (mse= 0.0198) , while pilotnet stood to be next in line (mse(best) = 0.0281). Resnet50 (mse=0.0450) and VGG16 (0.0742) were not even close to both models.



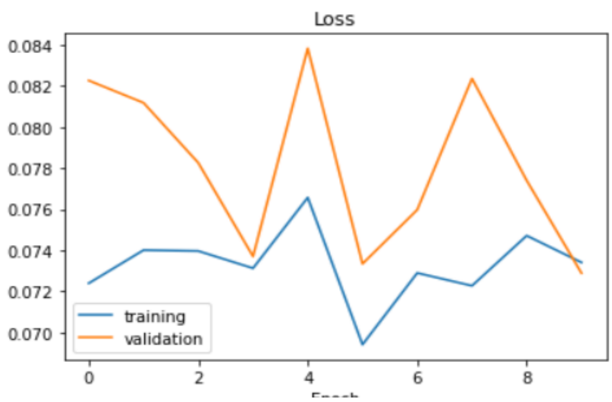
Pre-Trained Resnet50



PilotNet (NVIDIA)



Larva (proposed model)



Pre-Trained VGG16

Chapter 5
Conclusion and Future Scope

These results were produced solely from the models without any external smoothing function. We have shown that both, conventional models and a more advanced architecture have promise in the field of autonomous vehicles. The ConvLSTM model was limited by computational resources, but overall, it still provided a good result from a novel architecture. In future, the larva model's architecture could be expanded by having a larger and deeper layers, which may produce better results. Also, hyperparameter tuning and trying different variation like max-pooling or subsampling could enhance the performance. These models are far from perfect and there is substantial research that still needs to be done on the subject before models like these can be deployed widely to transport the public. These models may benefit from a wider range of training data. For a production system, a model would have to be able to handle the environment in snowy conditions. Generate adversarial models, GANs, could be used to transform a summer training set into a winter one. Additionally, GANs could be used to generate more scenes with sharp angles. Additionally, a high-quality simulator could be used with deep reinforcement learning. A potential reward function could be getting from one point to another while minimizing time, maximizing smoothness of the ride, staying in the correct lane/following the rules of the road, and not hitting objects. The ConvLSTM cell is a complex component of the model that acts as a black box. We would like to better understand how our model operates to increase transparency and gain insights into its behavior. Mass implementation of a black box model could lead to unexpected outcomes in unforeseen conditions. Therefore, it is important to increase transparency and understand the model's behavior.

References

[1] M. G. Usman, C. Haruna, A. Nahla, A. Saidu, S. I. Popoola and A. A.-G. Mohammed, "A Survey on Deep Learning for Steering Angle Prediction in Autonomous Vehicles," IEEEAccess, vol. 8, pp. 163797-163817, 2020.

[2] S. Vaibhav, G. Snehal, A. Rohit, D. Pritam and K. Urmila, "Steering Angle Prediction in Autonomous Vehicles Using Deep Learning," in 2019 5th International Conference On Computing, Communication, Control And Automation, Pune, India, 2019.

[3] Rodolfo Valiente∗, Mahdi Zaman∗, Sedat Ozert, Yaser P. Fallah, Center for Research in Electric Autonomous Transport (CREAT), Orlando, FL University of Central Florida, Orlando, FL
Controlling Steering Angle for Cooperative Self-driving Vehicles utilizing CNN and LSTM-based Deep Networks,2019

[4] Dhruv Choudhary and Gaurav Bansal. Convolutional Architectures for Self-Driving Cars. Technical report, 2017.

[5] Jonah Sokipriala. Prediction of Steering Angle for Autonomous Vehicles Using Pre-Trained Neural Network. 2017

[6] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016.

[7] Shuyang Du, Haoli Guo, and Andrew Simpson. Self-Driving Car Steering Angle Prediction Based on Image Recognition. Technical report, 2017.

[8] Jelena Koci´c, Nenad Joviˇci´c and Vujo Drndarevi´c . Article An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms.

[9] https://keras.io/api/ - Keras API Documentation

[10] https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7 - Blog for Introduction to ConvLSTM.

[11] https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/ - reference for image of MSE function