Comparative Analysis of Steering Angle Prediction For Automated Object Using Deep Neural Network

Md Khairul Islam

*Information and Communication
Technology, Islamic University
Kushtia, Bangladesh.
mdkito51@gmail.com*

Most. Nilufa Yeasmin

*Information and Communication
Technology, Islamic University
Kushtia, Bangladesh.
nilufayeasmin299@gmail.com*

Chetna Kaushal

*Chitkara University Institute of Engineering
and Technology, Chitkara University
Punjab, India.
chetnakaushal3558@gmail.com*

Md Al Amin

*Computer Science and Engineering
Prime University
Dhaka, Bangladesh
ahmedalamin2357@gmail.com*

Md Rakibul Islam

*Information and Communication
Technology, Islamic University
Kushtia, Bangladesh
mrpislam@gmail.com*

Md Imran Hossain Showrov

*Institute of Computer Science
Bangladesh Atomic Energy Commission
Dhaka, Bangladesh
showrov@baec.gov.bd*

Abstract—Deep learning’s rapid gains in automation are making it more popular in a variety of complex jobs. The self-driving object is an emerging technology that has the potential to transform the entire planet. The steering control of an automated item is critical to ensuring a safe and secure voyage. Consequently, in this study, we developed a methodology for predicting the steering angle only by looking at the front images of a vehicle. In addition, we used an Internet of Things-based system for collecting front images and steering angles. A Raspberry Pi (RP) camera is used in conjunction with a Raspberry Pi (RP) processing unit to capture images from vehicles, and the RP processing unit is used to collect the angles associated with each image. Apart from that, we’ve made use of deep learning-based algorithms such as VGG16, ResNet-152, DenseNet-201, and Nvidia’s models, all of which were trained using labeled training data. Our models are End-to-End CNN models, which do not require extracting elements from data such as roads, lanes, or other objects before predicting steering angle. As a result of our comparative investigation, we can conclude that the Nvidia model’s performance was satisfactory, with a Mean Squared Error (MSE) value of 0.3521. But the Nvidia model outperforms the other pre-trained models, even though other models work well.

Index Terms—Steering Angle Prediction; Deep Neural Networks; Autonomous Vehicles, Convolutional Neural Networks; Self Driving car; End-to-End

I. INTRODUCTION

The car accidents are happened due to the fault of mankind as well as autonomous cars are 90% and 2% respectively (1). For the reason, the number of Autonomous Vehicles(AV) are increasing day by day in human life. The implementation of AV saves the lives of mankind and make their life more easier. Artificial Intelligence (AI) will play a vital role to transfer the work of explicitly formulating rules to automated system which is able to learn all of rules. Scientist try to solve various problems related to vehicles(steering angle prediction, path planning, traffic recognition, other vehicles recognition, etc.) that are used in daily basis. If researchers want to solve those problems, then they need a huge amount

of sensor data for processing (2). An enormous advantages can be gained in further time using autonomous car like as f better fuel economy, decrease air pollution, car sharing, more productivity, improved traffic flow (3)

Considering the needs and comfort of the people, many companies are continuing their research work on automatic vehicles very fast and around 1400 automatic vehicles are in testing phase in only USA. Self-driving cars and other automatic robotic complexes had made historical progress in recent years. The main elements of vehicle motion control are in autonomous car steering control, lateral and longitudinal motions establishment. Steering of AV can take literal actions like as easily control the position of others vehicles in the lane, control lateral motion, for avoiding major accident it can take different route. A self-driving car has wide range of scopes with less errors compare to human drivers.

A function of artificial intelligence(AI) that follows the working principle of the human brain function for processing NLP, take important decision, speech recognition, voice recognition and so many works is called Deep Learning(DL). It also known as deep neural learning or deep neural network, can be learned highly nonlinear functions for its own complex structure. Convolutional Neural Network(CNN) is one of the most popular deep neural network used for classification of images, image segmentation, NLP, face recognition, self diagnoses of medical problems, and understanding climate change. CNN model can provide best performance in large data set than traditional machine learning models. The working principal of CNN in self driving is learning all important features to solve specific problem automatically.

DL techniques recently introduced a golden framework named End-to-End process for searching control policy (4). End-to-end process in deep learning technique provides very good accuracy score in automatic tasks like steering angle prediction, grasping and so on. Deep neural network can extract features automatically and help to predict the position

of road using only steering angle as the training signal. Deep learning based end-to-end process is more determined to observe the control actions of explicit mapping where the traditional models could not be determined.

The aim of our work is to illustrate end-to end approach in deep neural network for predicting steering angle. We use a single framework for our work where previously used several works like as steering control ,lane detection, object detection, and path planning. In this paper, we intend to provide an overview of deep learning based end-to-end approach and describe several deep learning models including VGG16, ResNet-152, DenseNet-201 as well as Nvidia that are successfully predict steering angle in autonomous car. We collect our data using IoT based system to train our selected models. Furthermore, we analyzed and compared our proposed work into previous works and proved that our model is the best to predict steering angle.

Figure 1: IoT based data collection system that collects images of roads and corresponding steering angle with frame rate 30.

In short, the main contributions of our paper are following:

- 1) Collected data using IoT base system
- 2) Proposed a deep learning based end-to-end approach for predicting steering angle to autonomous vehicles.
- 3) Presented a comparison between existing approach and proposed approach.
- 4) Evaluated performance metrics on that models are employed for predicting steering angle.

II. LITERATURE REVIEW

There are several papers about self-driven vehicles. Some of the literature are discussed in this section.

In (5), investigated two alternative models for high-quality steering angle prediction based on pictures utilizing deep learning approaches like as Transfer Learning, 3D CNN, LSTM, and ResNet. The drawback of the paper is that only minimal data augmentation has proven to be useful for these modes. Steering angle regression issue is posed as a classification problem with a spatial connection imposed between the output layer neurons. A CNN and LSTM network has been applied in the project. However the steering performance is 87% (6). Another study (7), the learning model is constructed

using a deep belief network (DBN), and the training data is obtained from real-world drivers. Though the prediction accuracy is not so good. Maqueda et al. (8) offers a deep neural network approach for releasing the potential of event cameras for a challenging motion estimation task: steering angle prediction. The paper employs cutting-edge convolutional architecture.

In recent study (9), examines the possibility of utilizing the emulator's pictures to train deep neural networks for steering angle prediction. In the paper their evaluated accuracy is 78.5%. Another group of reseracher (10) are applied CNN, LSTM, FC Image sharing models. Both LSTM and CNN were utilized to calculate a regression value based on the interdependence of successive frames. A most recent approach in 2020 (11), used the convolutional neural network method, the suggested system is built on Nvidia architecture. In the paper experiment result of the model is 95%. Only Nvidia's model is used in the project and didn't represent any comparative analysis with other existed models.

Figure 2: Preprocessing of collected data in order to remove noise and act perfectly when encounter unseen data. Shift, rotation of input images.

III. METHODOLOGY

A. IoT base Data Collection System

There will be two Raspberry Pi boards, one with a camera and wifi module for photo capturing and the other with a wifi module for algorithm processing. A Raspberry Pi camera will be installed in the car to collect photographs with frame rate 30 as shown in figure 1.

Following the gathering of photographs, the images will be sent from this raspberry pi board to another raspberry pi central processing unit through wifi modules included with raspberry pi boards. The steering wheel angle will be taken by the central processing unit, and the steering wheel angle

Figure 3: Basic CNN Architecture that used 5*5 kernel to extract all the required features to predict steering angle in End-to-End process.

Figure 4: Difference between specific features extracted methodology and End to End process.

and image of the road will be delivered to the SSD card for storage.

B. Data Collection

“Autopilot-TensorFlow” dataset (12) has been used in this research. Despite the fact that it has 44,401 images, it only covers 25 minutes of driving time and the datasets frame rate was 30. Data on how to drive in various weather and lighting situations was gathered by driving about in a number of different locations and on various roadways. Residential roads with parked cars, tunnels, and dirt roads are all among the types of roads that may be found on the road. Data was collected throughout the day and at night in a variety of weather circumstances, including clear, gloomy, foggy, snowy, and drizzly conditions.

C. Data Pre-processing

We teach the neural network how to recover from a bad location or orientation using preprocessing techniques, including adding false shifts and rotations. Disruptions of this size are picked from a normal distribution at random. It is twice as far from the mean as the variability we found in previous human driver data, and the allocation seems to have a zero mean. As the size of the data is increased, artificially enhancing it introduces unwanted artifacts. In figure 2 represents the preprocessing of collected data in order to remove noise and act perfectly when encounter unseen data

D. End-to-End CNN Model

In previous approaches, in order to predict steering angle, one needs to extract features such as road, lane, and other objects manually on which deep learning models are trained to predict angles. But manually extract features may include noise to our dataset that the error significantly increased. As well as it's hard to apply manual extraction on big datasets. So, as shown in figure 4, End-to-End approach introduces where

Figure 5: The architecture of VGG16 – Convolutional Network for Classification and Detection (13)

image data is input in a deep learning model and it extracts features automatically to predict the angles. The End-to-End process is impressive as the deep learning models learns easily different features on huge labeled data as shown in figure 3. After end-to-end model, we have attached a regression layer as steering angles is a continuous value.

We have used 4 different model such as VGG16, ResNet-152, DenseNet-201 and Nvidia's proposed model.

1) *Baseline Model*: In the baseline model, the projected steering angle is given a value of 0. Due to the fact that the majority of the roads in the dataset are straight, even angle 0 was assigned a high mean square error.

2) *VGG-16*: A convolutional neural network model has 16 layers deep is called VGG-16 which invented by K. Simonyan and A. Zisserman in 2014 (14). The VGG-16 model gain better accuracy in top-5 test around 92.7% on **ImageNet** (15) which is a image database has 14 million image data with 1000 categories. The input to the architecture of images of fixed size 224×224 and have RGB channels. A stack of convolutional layer is used for passing through a image has 3 very small receptive field with stride 1 for each convolutional layer. The model uses column and row padding to control

spatial resolution after convolution. It has 13 convolutional layer and the spatial pooling has five max-pooling layers with the size of 2 max pooling window and stride 2. Although every convolutional layer is not followed by max pooling layer, follow some of convolutional layer. The three fully connected layers(FC) which each of first FC two layers have 4096 channels and the last FC layer 1000 channels followed by a stack of convolutional layer. Soft-max layer is the last layer in VGG-16 model with 1000 channels, one channel is used for each category of images in above imagenet dataset. All hidden layers have ReLU activation function.

Although VGG16 is slow to train and take over memory space, VGG16 significantly outperforms compare than the previous models named ILSVRC-2012 and ILSVRC-2013. However it has eminent building block for Iraning purpose as well as easy to implement. It also provides better performance in many deep learning techniques for image classification.

3) *ResNet-152*: Residual Network (ResNet) is one of the well known model in deep learning techniques that first introduced in 2015 by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang for image recognition (16). The very first thing of ‘Deep Residual Learning for Image Recognition’ paper that there is a direct connection that skips some layers of the ResNet model and called ‘skip connection’. due to this skip connection, the output will not same. Then the skip connection is converted into the architecture of residual network. The architecture of ResNet is inspired by VGG-19 that has 34-layer plain network. There were several ResNet version in keras applications including: ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2.

ResNet-152 is one important ResNet version in keras applications. A convolutional neural network which has 152 layers deep is called ResNet-152 proposed by Microsoft Research Asia in 2015 (17). The structure of ResNet-152 is a feed-forward network which output is added to its input and using Rectified Linear Unit (ReLU) for passing out. Thus the previous layer carried information into next layer. The main benefit of this type of ResNet is provided better accuracy without any complexity of model and used for classification, detection, and localization with low error rate about 3.6%. Another advantage is that requires less parameters as well as need less time for computation.

4) *DenseNet-201*: Neural network introduced a new model for visual object recognition is known DenseNet also called Densely Connected Convolutional Network. With some basic differences, it has same architecture of ResNet model. For instance, DenseNet can concatenate () with the output of previous layer and the next layer where ResNet follow additive (+) process for these two layers. It was mainly developed for improving accuracy with less complexity. There are different version of DenseNet that are DenseNet-121, DenseNet-169, Densenet-201, and densenet-264.

A type of CNN model which has 201 deep layer neural networks is known as Densenet-201. The pre-trained version of the DenseNet network is trained on more than one million images **ImageNet** dataset and categorised with 10000 classes.

Figure 6: CNN architecture of Nvidia’s developed model to predict steering angles .

Thus the network can learn to train efficient features from a wide range of images. In traditional CNN model has L layers which has L connections but the newly invented CNN architecture(DenseNet-201) has $L(L+1)/2$ direct connections. In DenseNet-201, the previous layer is used as input and the all of the inputs are used in a feature maps and finally used it in next layer. Unlike ResNet, it requires less memory space and computation to achieve high performance.

5) *Nvidia's Model*: In (18), Nvidia predicted a model which gives a significant output in case of steering angle predication. It has a normalization layer, 5 convolutional layer, 5 dense layer and a flatten layer where almost 27 millions internal connection and 250 thousands various parameters as shown in figure 6. In its first 3 convolutional layer they introduced 24, 36 and 48's 5*5 kernels respectively. The later 2 convolutional layer contains sixty fours 3*3 kernels each. Then there are 5 dense layer with output space as 1164, 100, 50, 10 and 1.

Its weight initialized with random value using global variables initializer and optimizer to get minimum loss in mean squared error.

IV. EXPERIMENTAL SETUP AND RESULTS

This section our experimentation grouped into 2 subsections: Experiments and Evaluation methods.

A. Experimental Setup

In the experimental section, we will go over the numerous computational settings that we used to establish our methodology in the first place. We have employed the Python programming language, as well as the necessary development tools and a variety of important machine learning or deep learning libraries to accomplish our goals. Python is a good choice for building models, as it's an excellent general-purpose programming language for the purpose. For loading the training dataset, we construct a batch generator with a total of 100 batches. We have used Adam as the optimizer. Before putting the images into the neural network, we scaled them with a 66*200 resolution. The dataset is divided into three parts: the train set, the test set, and the validation set. The test set is used to ensure that each model performs as expected. Each epoch includes an image augmentation technique that is used before the data is fed into the neural network. Our model is trained across the dataset for a total of 50 iterations (epochs).

B. Evaluation Metrics

The mean squared error (MSE) is the most frequently encountered loss function in the context of a regression base problem. The loss is calculated as mean square differences between true and predicted values over all collected data, or as follows in mathematical form:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N (y - \hat{y}_i)^2 \quad (1)$$

Table I: Various Models loss for train, test and validation set of data.

Classification Model	Train Loss	Validation Loss	Test Loss
Baseline Model	0.645	0.464	0.552
VGG16	0.512	0.514	0.732
ResNet-152	0.791	0.464	0.743
DenseNet-201	0.294	0.375	1.148
Nvidia's Model	0.278513	0.31613	0.3521

C. Result and Discussion

In this study, we have deployed deep learning algorithms on road images using End-to-End process to train an autonomous steering angles predictor. We have also proposed a data gathering system based on Internet of Things (IoT) for future use. In this work, we have trained various models on a dataset named "Autopilot Tensorflow". Firstly, we have conducted image preprocessing steps such as shifts, rotations that are necessary to eliminate noise from the image. After that, we passed the images through various neural networks, where after 50 successful iterations (epoch), we obtained the expected trained model. In our approach, we have used the VGG16, ResNet-152, DenseNet-201, and Nvidia's anticipated models. In case of Nvidia's model, the weight was initialized randomly; however, for the other models, the weight was initialized using the "imagenet" weight initializer. In table 1 summarizes the performance of each model.

Our preferred loss measured method was MSE (mean squared error).

The Baseline model for this study has a training loss of 0.645, a validation loss of 0.464, and a test loss of 0.552 after 50 successful epochs. Although baseline model sometimes perform better than other models, it is useless as it constantly predict zero for each image. After the same amount of epochs for VGG16, training loss was 0.512, validation loss was 0.514, and test loss was 0.732. The test loss for the Vgg16 model is higher than the baseline model. Whereas for ResNet-152 the train, validation and test loss were 0.791, 0.464 and 0.743 respectively. The test loss of ResNet-152 is similar to that of VGG16. DenseNet-201's performance was slightly better than previous three models in terms of train loss (0.294) and validation loss (0.375), though its test loss (1.148) was highest among all the models. On the first epoch, Nvidia's model loss was 6.1285; however, after 50 epoch, the model loss has been reduced to 0.13. Because we didn't expressly extract any features for training, this is an extremely low score in terms of an End-to-End deep learning strategy from start to finish. Figure 7a and 7b depict the output in a clear and understandable manner for Nvidia's models. Finally, after evaluating all of the models and comparing their performance, our Nvidia model is the best among all.

D. Limitation

We cannot detect non-road images correctly, nor can we detect traffic conditions, and government laws are useless. In

Figure 7: Output visualization on random video from youtube..

order to function properly, the camera must be mounted in the right location. The only thing we can predict with this model is steering angles; all other objectives of an autonomous car will be impossible to achieve with this model. In order to generate more precise angles predictions, datasets need to be vast, yet Datasets contains a comparably small number of images

V. CONCLUSION AND FUTURE WORKS

The goal of this study is to determine the steering angle based on front images only. The Internet of Things (IoT)-based data collection system is introduced in order to collect labeled data for more robust model training. We will require a large amount of data in order to achieve a satisfactory outcome using this strategy, which is not currently available. As a result, our Internet of Things base collection system will assist us in collecting adequate images for future work. The results of our experiment demonstrate that deep learning-based techniques are rapidly evolving to the point where they can extract features by themselves, a process known as the End-to-End process. The model developed by Nvidia outperformed all of the deep learning algorithms tested in our research. Even so, the model can deliver the same results in a variety of weather and road situations. In the end, the time spent manually extracting features is saved for a valid purpose. In

the future, we aim to be able to further integrate the following characteristics in the future by constructing a model that can adhere to traffic regulations by accelerating or breaking the driving object as necessary.

REFERENCES

- [1] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical report, 2015.
- [2] Viktor Rausch, Andreas Hansen, Eugen Solowjow, Chang Liu, Edwin Kreuzer, and J Karl Hedrick. Learning a deep neural net policy for end-to-end control of autonomous vehicles. In *2017 American Control Conference (ACC)*, pages 4914–4919. IEEE, 2017.
- [3] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, and Saber Fallah. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):712–733, 2020.
- [4] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.
- [5] Shuyang Du, Haoli Guo, and Andrew Simpson. Self-driving car steering angle prediction based on image recognition. *arXiv preprint arXiv:1912.05440*, 2019.
- [6] Hesham M Eraqi, Mohamed N Moustafa, and Jens Honer. End-to-end deep learning for steering autonomous vehicles considering temporal dependencies. *arXiv preprint arXiv:1710.03804*, 2017.
- [7] Chunqing Zhao, Jianwei Gong, Chao Lu, Guangming Xiong, and Weijie Mei. Speed and steering angle prediction for intelligent vehicles based on deep belief network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 301–306. IEEE, 2017.
- [8] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018.
- [9] MV Smolyakov, AI Frolov, VN Volkov, and IV Stelmashchuk. Self-driving car steering angle prediction based on deep neural network an example of carnd udacity simulator. In *2018 IEEE 12th international conference on application of information and communication technologies (AICT)*, pages 1–5. IEEE, 2018.
- [10] Rodolfo Valiente, Mahdi Zaman, Sedat Ozer, and Yaser P Fallah. Controlling steering angle for cooperative self-driving vehicles utilizing cnn and lstm-based deep networks. In *2019 IEEE intelligent vehicles symposium (IV)*, pages 2423–2428. IEEE, 2019.
- [11] Mr Aires Da Conceicao and Sheshang Degadwala. Steering angle prediction based on road direction using convolution neural network (cnn). 2020.

- [12] Sully Chen. Autopilot-tensorflow, 2016. URL <https://github.com/SullyChen/Autopilot-TensorFlow>, 1(5):6, 2016.
- [13] Vgg-16 cnn model. www.geeksforgeeks.org.
- [14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [18] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseen Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.