Prediction of Steering Angle for Autonomous Vehicles Using Pre-Trained Neural Network

Jonah Sokiprialu

Abstract — Autonomous driving is one promising research area that would not only revolutionize the transportation industry but would as well save thousands of lives. Accurate correct steering angle prediction plays a crucial role in the development of the autonomous vehicle. This research attempts to design a model that would be able to clone a driver's behavior using transfer learning from pretrained VGG16, the results showed that the model was able to use less training parameters and achieved a low mean squared error (MSE) of less than 2% without overfitting to the training set hence was able to drive on new road it was not trained on.

Index Terms — Autonomous vehicles, Convolution neural network, VGG16, Transfer Learning.

I. INTRODUCTION

Autonomous vehicles came about as an attempt to solve the number of car accidents which 90% have been a result of human errors [1]. Despite the advancement in the field of AV’s there are still numerous research which attempt to seek for better technologies for prediction of steering angle control. Autonomous vehicles have come to stay with some cities like California having deployed test AV’s on public roads many automobile companies showing strong promise to begin large scale production to meet up with commercial demands, some of the big companies investing in self-driving cars include tesla, google, Mercedes etc. [2]. Successful commercialization of Autonomous vehicles would help save millions of lives. Autonomous driving plays a pivotal role in intelligent transport systems, the safety and overall performance relies greatly on how accurate the car can predict the steering angle. The core of the control of autonomous vehicles is the machine learning algorithms which process the road images and adjust the steering angles based on the lane’s markers, traffic signs other road objects for obstacle avoidance [3]. The complexity in the design of autonomous vehicle is the prediction of when the vehicle should take a turn based on the curved nature of most road networks. This paper would attempt to develop a model using transfer learning from image net classification challenge to develop a model which can from a given road scene predict the model correct angle to steer. The model would be designed using Udacity simulator on a training lane and tested on the driving lane with a more complex road network the aim is to use a model with less parameters by incorporating transfer learning from a pretrained network and compare same with other models. Autonomous vehicle would drastically change the way people move, which in turn would have significant social and economic impact on our lives. The rest of this paper is divided into various sections related works, methodology, results, and conclusion.

II. RELATED WORKS

Navigation of self-driving cars dates as far back as 1989, when Pomerleau pioneered the first research in this area using neural network, the model was known as Autonomous Land Vehicle in a Neural Network (ALVINN) [4]. Another major advancement in AV’s is the DARPA Autonomous Vehicle (DAVE) this was part of the project of the Defense Advanced Research Projects Agency, this was one of the first to deploy the end-to-end neural network model [5]. There have been various methods for prediction of steering angles such as use of artificial intelligence, however the major challenge with this method is the use of hand-crafted features, these models tend to perform poorly due to the dynamic driven environment [6]. Several other research which have attempted to predict steering angle have been carried using Artificial Neural networks and support vector mechanisms these regular feedforward neural networks have been the focus of most AV research in the past. convolutional neural networks have been found to give good performance in image classification task [7] various research have attempted to use convolutional neural in prediction of steering angle used convolutional neural network on a Raspberry Pi and Arduino to model a self-driving car using single camera. [4] used convolutional network inspired by the Nvidia model [3]. The CNN was used to minimize the error between the predicted steering angle and the actual steering angle, which the car is to navigate through. While many researches have attempted to solve the automatic steering angle prediction using CNN. CNN which is one the most popular deep neural network for image processing because its ability to learn key features automatically and the accuracy with which it extracts these features however, the main disadvantage is it usually requires large dataset. Other researches have employed data augmentation to increase the number of available images [5]-[7]. The modern day autonomous vehicle research evolved from the pioneering works in behavioral cloning the model designed from NVIDIA the model consisted of 5 convolutional layers and 4 fully connected layers without any max pooling the model consisted of over 250 thousand parameters and 27 million connections [8]. The model was able to drive in areas such as highway lane and other less complex real world scenarios.

Many research have also attempted to utilize Recurrent Neural Network to analyze Recurrent Neural Network to analyze road images at different time...
stamps to predict what the next action that a pedestrian or driver from an incoming vehicle would most likely take, in order to steer in the appropriate direction. History of previous sensor data is used by Recurrent Neural Network to predict future behaviors [9]. used LSTM to predict human like steering angles in curved road from historical driving data. [10] applied long short-term Memory (LSTM) and Gated Recurrent Unit in predicting trajectory, from past trajectory, spatial-temporal features. [11], [12] also combined LSTM and CNN in steering angle prediction. Other models have also applied transfer learning where models applied in the image net classification challenge are reused as the starting point of the models for predicting the steering angle [13] combined transfer learning and LSTM in developing model for predicting steering angle. [14] compared the performance of 3D convolution and LSTM with a 2D convolutional layers on a pre-trained model [15]. Reinforcement Learning (RL) the autonomous vehicle acts as an intelligent agent which tries to maximize the cumulative rewards by making correct driving decision. [16], [17] employed reinforcement learning in prediction of Reinforcement learning. One main challenge faced by reinforcement learning is its inability to solve problem with sparse reward signal [18].

III. METHODOLOGY

The Udacity self-driving car is fitted with 3 front cameras which would take the images. The model will gather training data on a training lane using sampled images from the video during the test drive. The model would extract the steering angles via the vehicle’s controller area network (CAN) bus at different time stamp. The images would be saved on the SSD card of the Nvidia drive AGX which is the brain of the entire system, the steering angle and speed under different location would be used to accurately train the model using behavioral cloning, this would ensure the it would be able to stir and change speed based on test lanes which the car have never seen or drove on. Python flask would be used to link our model by creating a local host for our code to interact with the Udacity Simulator.

![Model for the steering angle control.](image1)

Behavioral cloning was applied because it would not be practically possible to hand code all possible drive condition, hence behavioral cloning is used to clone the behavior of a driver using Udacity self-driving car simulator to collect training data in order to enable our model to learn how to drive. During the training the steering angle, throttle, and speed, these data are extracted using three different cameras mounted in front shield of the car, they capture three different view of the road image in the test lane at every time stamp as shown in Fig. 3 represent different view of the car at same time, captured from the different angles of the three cameras, this method proposed by Nvidia [8] helps improves the training of the model to enable it generalize effectively.

![Simulator road structure.](image2)

![View from the left (a), center (b), and right (c) camera.](image3)

The data were extracted by making 3 complete driving laps in the left and right turn to obtain enough data set, these images are then split into training and validation data set in
In order to train our deep neural network (DNN) model, Fig. 4. shows histogram distribution for the data set obtained during training.

Data set is modified as shown in the Fig. 4, this is done to reduce the bias of the car always driving straight, threshold is set to reduce the data by extracting only the best 400 values in each steering direction this gives a more uniform data set, we then split our data set into training, validation which would be feed into our neural network.

### B. Data Augmentation

We augment the data to add variety to the training set this helps improve the amount of training images in our data this is a very import process which is used for model with less training set as proved by [19], this also improves the driving efficiency in real time on real lanes, some of the data augmentation applied includes flipping and rotation, zooming and variation of brightness.

### C. Flipping and Rotation

We add a random rotation range of -10, to +10 degrees to include any cases where the camera might have been rotated this also gives the training model more training sets to generalize on. The width of each frame is taken as the x while the height of each frame as y.

\[
\begin{pmatrix}
    x' \\
    y'
\end{pmatrix}
= \begin{pmatrix}
    \cos \beta & \sin \beta \\
    -\sin \beta & \cos \beta
\end{pmatrix}
\begin{pmatrix}
    x \\
    y
\end{pmatrix}
\]

(1)
D. Brightness

The brightness is also altered to account for poor weather condition, shadows etc., which were not included in the training data set, this would enable the car steer correctly under such conditions.

E. Zooming

This is also added to the training set to help generalize on features which may appear smaller or larger than their original size a ±10% zoom is added to the image training set.

F. Pretrained VGG16

The model used transfer learning from pretrained weights of VGG16, this model was chosen based on its superior performance on localization challenge and number of operations in the ImageNet challenge. VGG16 has already been trained on over 1.2 million high resolution images on expensive GPU than what is available to us, it would do well in extracting high level features, the head of the pretrained(final classification layer) is replaced with the Nvidia CNN model dense layers proven to be very effective in behavioral cloning [8] the model consist of pretrained VGG16, 3 dense layers and final output layer which predicts the steering angle, as shown in Fig. 10, this improve performance while also saving time used in training a full mode, also use lesser training parameters as only the dense layers would be trained making it less computationally expensive.

The Exponential Linear Unit (Elu) activation function is used to avoid dead neurons during the back-propagation process as it tends to converge cost faster, as shown equation 3, the Elu activation function does not converge to zero for negative values, $\alpha$ is a positive constant which must not be zero.

$$ f(s) = \begin{cases} 
-s, & s < 0 \\
\alpha(e^s - 1), & s \geq 0 
\end{cases} $$(2)

where
$S$= the steering angle;
$\alpha$ is a positive constant.
TABLE I: HYPER PARAMETERS FOR THE VGG16 MODEL

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization</td>
<td>Adam*</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.0001</td>
</tr>
<tr>
<td>Batches</td>
<td>40</td>
</tr>
<tr>
<td>Epochs</td>
<td>20</td>
</tr>
</tbody>
</table>

G. Loss Function

Steering angle prediction is a regression type problem, which can be analyzed using the mean squared error (MSE) as our loss function in predicting the actual target steering angle and predicted steering angle.

\[ E_r = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \]  

(3)

where \( y_i \) and \( \hat{y}_i \) are the predicted and actual steering angles, respectively.

MSE takes the squared difference hence would be very effected to punish any variation between the actual and predicted steering angle.

IV. RESULTS AND DISCUSSION

The use of data augmentation does not only increase the amount of samples however it add variations which would not be captured under actual driving the variations help improve the helps the model to learn better and minimize loss faster as shown in Fig. 11, the augmented images are artificially created and hence does not increase the trainable weights required hence does not increase the computational cost, which makes it very important.

The behavioral cloning algorithm performed better using the Elu activation function when compared to RELU as shown in Fig. 12, in the Nvidia model the relu activation function could not learn even after several epochs, we observed a dying relu , the relu dies during back propagation the relu derivative of the loss function becomes zero the loss function is the difference between the model prediction and the actual steering angle.

Hence when \( \frac{dl}{ds} \) is zero the new weight cannot be updated for Relu activation function, however using the Elu activation function derivative for the error function for \( s < 0 \) is not zero hence the new weight can be updated, and the model can learn by backpropagation.

\[ \frac{dl}{ds} = \begin{cases} ae^s, s < 0 \\ s, s > 0 \end{cases} \]  

(4)

\[ W_N = W_o \times a \frac{dl}{ds} \]  

(5)

where

\( W_N \) is the New weight;

\( W_o \) the old weight;

\( a \) is model learning rate, variation in learning rate.
The MSE for the validation loss in the pretrained VGG16 and the dense layers from the Nvidia Model performed better as its combined sections from two very good performing models hence outperforming the actual Nvidia model and ResNet50 which did better in ImageNet.

V. CONCLUSION

Finally, we were able to training our model using behavioral cloning to clone the behavior of a real driver by driving first on a test lane, and it was able to learn this features and was able to drive on test lane which it has not seen before and did very well, we also compare the performance of our cars steering angle using different Deep Neural Networks and we observed that transfer learning using VGG16 showed better performance when used with the Elu activation, based error metrics used. we also observed while our model has explicitly been trained on prediction of steering angle it was also able to detect lane markings, however due to limited data set and training time and less expensive GPU pretrained VGG16 showed improved performance compared to actual training set.

REFERENCE


Jonah Sokipriahla is a graduate assistant in electrical engineering department, at rivers state university port Harcourt, Nigeria. He received his B. Tech in Electrical Engineering and M.Sc. in Electrical and Electronics Engineering from Coventry University, United Kingdom. He is working on self-driving cars and computer vision applications.