**Abstract**

Self driving cars are on the rise today and expected to become mainstream few years down the line. One of the key subsystems of such vehicles to become successful is that of a robust steering angle prediction to navigate the vehicle in an organized manner. Udacity has released a dataset of images along with steering angle while driving. We show an improvement on this prediction compared to several other solutions by using augmented data generated from a successful lane detector deep neural network.

1. **Introduction**

Self driving cars are expected to be one of the next big revolutions in the automobile industry. It is expected to generate enormous amounts of value and optimization of natural resources over a period of time. Multiple companies have invested aggressively in this area in order to develop the best such systems.

In the pursuit of solving the challenge, one of the key sub problem to be solved is that of creating a robust steering angle prediction model. Udacity, with the intention of building the world's first open source self driving car platform has released a set of online challenges that aim to solve different parts of the platform. We attempt to provide the best solution to the second challenge that was released which deals with task of predicting steering angles using deep learning.

As a part of the challenge, a dataset was released that contained images along with the correct steering angle captured while driving. There have been tons of submissions attempting to solve this better. However, the motivation for our attempt at this challenge is that the problem can be segmented into 2 parts : lane detection and steering angle prediction and that the latter is dependent on the former. We approach this problem to improve the predictions by augmenting data obtained from detecting lanes from an optimally trained lane detection module and use it on few of the best architectures and methods that have already been attempted.

The remainder of the paper is organized as follows: section 2 describes related works, section 3 describes the problem and the proposed method, section 4 presents the experimental results and section 5 describes the future works and conclusion

1. **Related work**

In 2016, Nvidia demonstrated that CNNs can be applied in prediction of steering angles. They steered an autonomous vehicle using CNNs in real time setting on highway lanes. Three cameras were mounted in front of car, however they were able to successfully steer the car using center camera only.

Udacity dataset also contain similar structure (left, center, right cameras are mounted) and driving information such as steering angle, speed, torque etc. are recorded. This open source challenge generated a lot of contribution and innovations.

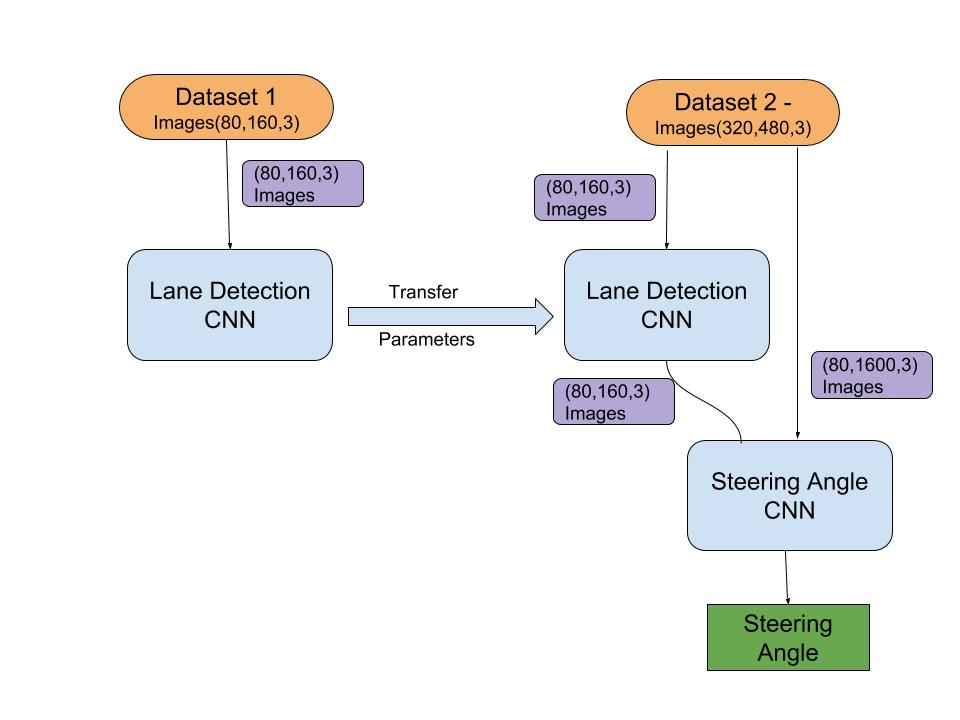
Some submissions tried to incorporate temporal information (time-series), building on top on Nvidia model which process single images. The ideas were based on papers ' Deep Learning for Video Classification and Captioning' by Wu et al., '3D Convolutional Neural Networks for Human Action Recognition' by Ji et al. and

'Large-scale Video Classification with Convolutional Neural Networks' by Karpathy et al.

Furthermore, Donahue et al.(https://arxiv.org/pdf/1411.4389.pdf) develop a novel recurrent convolutional architecture suitable for large-scale visual learning. This model was end-to-end trainable, and demonstrated good performance on benchmark video recognition tasks.

Wu et al.(https://arxiv.org/pdf/1509.06086.pdf) trained videos using multi-stream deep networks involving both CNN and LSTM. imonyan and Zisserman proposed Two-Stream Convolutional Networks for Action Recognition in Videos, fusing outputs of a spatial and motion stream CNN.

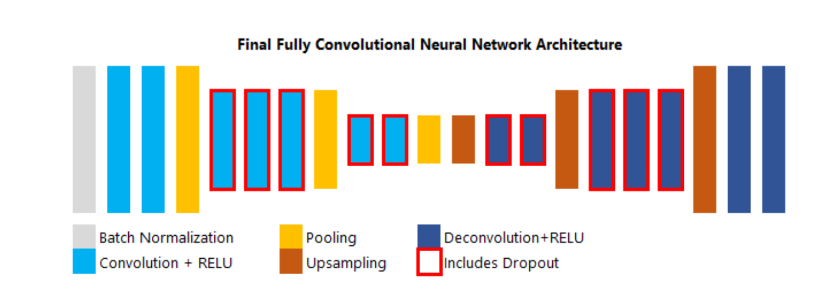
1. **Approach**

We divide our approach into two parts: one part of the approach deals with lane detection and the second part deals with steering angle prediction. The idea is to train lane on Dataset 1 and freeze the parameters. We will then use this model to predict lanes on Dataset 2. Then these outputs will be used as additional features to steering angle network, along with original images.

1. **Lane Detection**

The objective of this task is that given an image taken while driving and facing towards the road, detect the lane on the road on which the vehicle is travelling. For this, we have used a deep learning architecture inspired from : (https://github.com/mvirgo/MLND-Capstone) about which we describe below. We expect that this network will accurately be able to detect the lanes reliably.

The architecture is a standard one and uses a combination of convolutional layers with RELu, batch normalization, pooling, upsampling and dropout and can be seen in Figure.



Number of Convolutional Layers : 7 (Filter Size = (3,3))

Number of Pooling Layers : 3 (Max Pooling, Size = (2,2))

Number of Upsampling Layers: 3

Number of Deconvolutional Layers : 7 (Method: Bilinear Interpolation)

Total params: 181,693

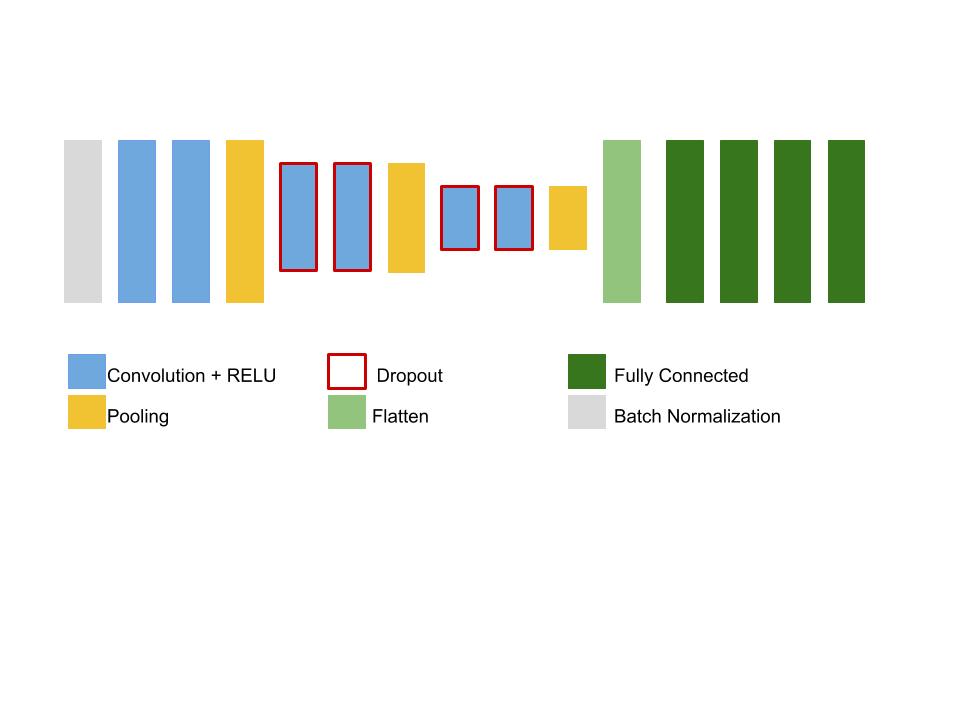
This is a regression problem, where each pixel can be either part of lane or not. Evaluation on this model can be done by measuring the Mean Squared Error (MSE) loss of the pixels values on the predicted output compared to the labeled images.

1. **Steering Angle Prediction**

The objective of this task is to be able to predict the steering angle given the image taken while driving acing towards the road. Apart from just using this image directly, we first detected the lane using the model mentioned above and append this detected lane image to the original one before passing it through the deep network that would predict the steering angle.

We combined lane detection module with 2 different architectures of Steering angle model.

1. Baseline – It is very similar to lane detector architecture, except that deconvolution layers are replace by fully-connected layers.
2. Nvidia Model – The filter size is (5,5) for first 3 layers and (3,3) for later layers, regularization is added in fully-connected layers.



Number of Convolutional Layers : 7 (Filter Size = (5,5))

Number of Pooling Layers : 3 (Max Pooling, Size = (2,2))

Fully connected Layers : 4

Total params: 11,502,851

1. **Datasets**
2. **Lane Detection Dataset (Dataset 1) :** This dataset (link) contains 12764 images of size (80,160,3). The output labels are images with lanes marked in green channel.

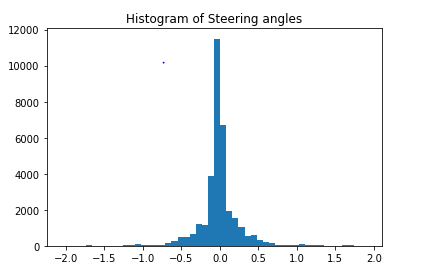
Some data statistics-

* 26.5% were straight or mostly straight roads, 30.2% were a mix or moderate curves, and 43.3% were very curvy roads
* 17.4% were clear night driving, 16.4% were rainy morning driving, and 66.2% were cloudy afternoon driving

1. **Steering angle Dataset (Dataset 2)** – The dataset was part of Udacity self-driving car : Challenge 2. Dataset statistics taken from Udacity Github repository:-

* Clip 1: 221 seconds, direct sunlight, many lighting changes. Good turns in beginning, discontinuous shoulder lines, ends in lane merge, divided highway
* Clip 2: 791 seconds, two lane road, shadows are prevalent, traffic signal (green), very tight turns where center camera can't see much of the road, direct sunlight, fast elevation changes leading to steep gains/losses over summit. Turns into divided highway around 350s, quickly returns to 2 lanes
* Clip 4: 99 seconds, divided highway segment of return trip over the summit
* Clip 5: 212 seconds, guardrail and two lane road, shadows in beginning may make training difficult, mostly normalizes towards the end
* Clip 6: 371 seconds, divided multi-lane highway with a fair amount of traffic

We plotted histogram of steering angles in the dataset, and found that a large number of labels were neutral angles (+- 0.050) . This affects the CNN prediction since the model is biased towards neutral angles. To overcome this, we removed frames within (+-0.010) with 40% probability.



1. **Experiments and Results**
2. Training

All the models were trained using Keras, using tensorflow as backend.

1. Lane detection network-

Adam optimizer was used for training, and learning rate was kept as 1e-3.

Mean square error (MSE) over the pixel values of predicted images and labeled images was used as loss function.

1. Steering angle network –

To validate our proposal, we trained steering angle networks separately, with lane detection and without lane detection (vanilla). This was done on 2 different architectures, so in total 4 models were trained. We call baseline architecture(section 3.2.1) as A1, and Nvidia architecture (section 3.2.2) as A2.

The images were downsized from (480,640) to (80,160) to match input layer of lane detection module.

Adam optimizer was used for training, and learning rate was kept as 1e-3.

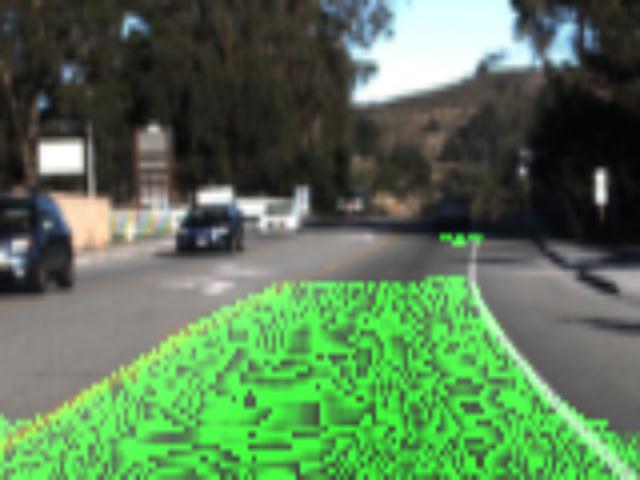
Mean squared error (MSE) over predicted steering angles and labels was used as loss function. However, to keep the scoring standard, RMSE error was used to compare to Udacity leaderboard. The test set was used for scoring purposes.

1. Results

The score are shown in table 1.

1. **Discussion**

The lane detection module was working decently well on dataset 2 although there was a domain shift due to transfer learning. (See fig.)



Our models rank in top 10 of Udacity challenge. For A1 architecture, steering angle network with lane detector works better than vanilla steering angle network. We see improved predictions both on validation set and test set. For A2 architecture, both model performed equally well on test set. However, on validation set, lane detection module improved the predictions.

In general, we expected Nvidia architecture (A2) to perform better than baseline architecture (A1). We can attribute this opposite result to difference in filter sizes used in these architectures. A2 architecture has (5,5) size while we downsized image from (480,640) to (80,160). So (3,3) filter from A1 is better for such small images.

Also, this problem of steering angle prediction cannot be totally accurate if one does not have GPS information. For the system to predict correctly at intersections and highway exits, we also need to input the destination information. This is because there are many possible lanes to follow at turns and intersections. These inputs could be complementary to images inputs.

1. **Conclusion and Future Work**

We were able to demonstrate that features from lane detection helped increase the accuracy of predictions on steering angle network. We tackled the challenge 2 by Udacity for self-driving cars and performed in league with top 10 models, just by using simple architectures.

In future work, this lane detection module can be combined with other models, like 3D convolution NN, or LSTMs. These models also exploit time-series information and temporal features, in addition to spatial feature of CNNs. This can be used when camera directly faces sunlight for few frames, therefore information from previous frames is used for prediction.

Also, dataset can be preprocessed extensively using CV techniques, to remove sunlight glares and improve night visibility.