Angle Detection of steering in a self-driving Car

Vipul Devnani
Computer Science and Engineering
Vivekanand Education Society's
Institute of Technology
Mumbai, India
2018vipul.devnani@ves.ac.in

Navin Kachhela
Computer Science and Engineering
Vivekanand Education Society's
Institute of Technology
Mumbai, India
2018navin.kachhela@ves.ac.in

Neeraj Gwalani
Vivekanand Education Society's
Institute of Technology
Mumbai, India
2018neeraj.gwalani@ves.ac.in

Chandan Panjwani
Vivekanand Education Society's
Institute of Technology
Mumbai, India
2018chandan.panjwani@ves.ac.in

Mrs. Abha Tewari
Vivekanand Education Society's
Institute of Technology
Mumbai, India
abha.tewari@ves.ac.in

***Abstract*—Self-learning and automation are not limited to human beings only. A new era which is very exciting is coming into focus, the age of self-driving cars. In this era humans have to no longer care about the obstacles and driving paths, no more stressful rush hour traffics, vehicles will take us to the destination fast and efficiently. Udacity has provided a dataset containing a set of images with steering wheel angle recorded during driving. With the help of the dataset provided by Udacity we can predict the steering wheel rotation angle.in order to predict the angle we have used computer vision and senor fusion which are the backbones of autonomous driving. In this project, a system is created which is ideally capable of automatically driving a vehicle on roads without any human input required. We have used machine learning models like CNN, OpenCV for certain application to detect an object, for vehicle detection, traffic sign detection, etc. The project will be deployed as a simulation software that will provide us the output in a frame. There are two frames one consisting of steering wheel and other will consist of road.**

***Keywords*— Autonomous Cars, CNN, Machine Learning, Computer Vision.**

I. Introduction

The concept of self-driving cars was into existence much back than today's new technology developed by google and NVidia. In fact, an exhibition was held in 1939 in New York Fair. General Motors created the exhibition to show the world about how in future the autonomous self-driving cars will look like in 20 years, and this system included highway system that would guide self-driving cars. Meanwhile, continuous work was carried out to construct a full-fledged autonomous vehicle. With the goal of making the car safer and simpler in coming years. Experiments have been conducted since 1920 and the trials were started in early 1950's first trail was conducted in 1920 on automated driving systems; researchers first presented a research paper for autonomous vehicles using neural networks. In year 1958, General Motors made their futuristic vision into reality. The model of car's front side was embedded with sensors called pick-up coils which could detect the current flowing on wires that were embedded on roads.

The motivation of the project is to remove the human input required to drive a car and to develop a system that learns how to drive by observing. Predicting steering angle which is important part of self-driving car and would allow us to explore the power and capability of neural networks. For example, using only steering wheel angle detection as training signal, deep neural networks can automatically extract features to help self-driving car to make the prediction for angle of rotation. desired rotation of a steering wheel angle is computed using CNN and the weight adjustment on the nodes to reduce the error rate according to the steering command the electronic system communicates with the model to achieve the goal. Different layers included in our network will help to reduce the error rate and various processing done by different layers carried out to provide desired results.

Udacity is on a mission to create an open-source self-driving car. In their mission they have released a dataset of images taken while driving car, and steering wheel angle and ancillary senor records for training set (left, right, and center cameras). The goal of this mission was to develop a model that given image taken while driving, will reduce the RMSE (root mean square error) between what the model predicts, in reality how the rotation of angle take place. In order to develop the system technologies like convolution neural network is used to achieve the end result.

II. Related Work

Using a neural network for autonomous vehicle navigation was given by Pomerleau (1989). He also built autonomous land vehicle in CNN. The model structure was simple and comprising a fully connected network which is small in today's standard. The network predicted actions from pixels inputs applied to simple driving scenarios with few obstacles. However, it demonstrated the potential of neural network for end-to-end autonomous navigation. The layout of the architecture was augmentation of the data collected of artificial shifts and rotations data was found to be important.

Sr No	**Previous Papers**		
	Authors	***Methodology***	***Inference***
1	Shuyang Du, Haoli Guo, Andrew Simpson	The have used 3D convolutional layers followed by recurrent layers using LSTM	The two models that will be discussed are a model that used 3Dconvolutional layers followed by recurrent layers using LSTM (long short-term memory).

2.	Md Najmus Saquib1 , Mr.Javed Ashraf2 and Col. (Dr) O.P.Malik3	They have used use 3d camera to manage steering wheel of car	By using GPS, it navigate across the road. The steering angle is controlled by stepper motor which is connected through a computer CPU and GPU are already trained through millions of images taken from real road transportation in all traffic and weather conditions.
3.	Praval Kumar, Shivam Shandilya, Tushar Sachan and Mr. Nizam Uddin Khan	Their paper focuses on algorithmic "brain" and powerful sensors "senses" using Deep Q-learning	In this, the Learning Model will learn by experimentation. The agent in an environment with specific states has a set of actions that it can perform and after performing those actions, it receives a reward which tells it how good that action was and stores these correct action in "Brain". We have to find when this car receives positive reward or negative reward using graph.

III. Methodology

We will use Convolutional Neural Network (CNNs) to map the raw pixels from a front-facing camera to the steering commands for a self-driving car. The images collected by the dashcam (Our dataset) will be passed to the Convolutional Neural Network. A CNN is basically a deep learning algorithm, which in the input can take an image, process that image in order to learn and assign importance(biases) to various aspects/objects in the image that can help us to differentiate these objects within the image from one another. This is the technique that will help us to evaluate the image that the car dashcam will provide to the neural network. The design of a neural network is inspired from the neurons present in the human brain and the organization of the visual cortex. Through relevant filters applied to the image, A CNN is successfully able to capture the Spatial/Temporal dependencies in the image

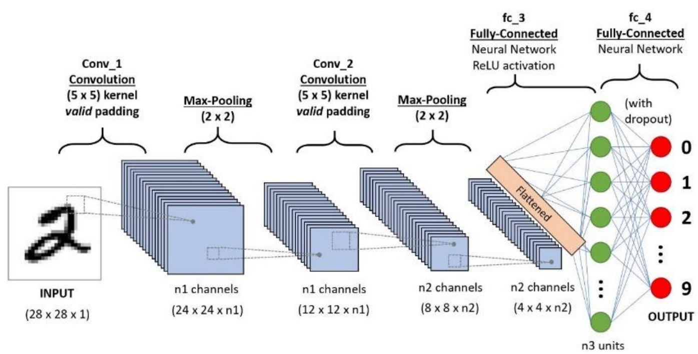
- Road Detection: - This is the first step in face processing. The main purpose of this step is to detect the road from the images from the dataset. In this step individual images are taken from the dataset, scanned and then verified whether the image contains a road or just only the background image. The road determination system determines if the input data (image) is a road. After this step the result is sent for pre-processing so that features can be extracted from the image.

- Feature Extraction: - This step is critical as it extracts the features using the applied feature extraction algorithm. The steps perform compression of image information, reducing irrelevant features. After, analysis of the features is done and then the model is trained, then the model is tested against a given input image

IV. Proposed System

A.Overview

The basic idea behind the implementation is to use the dataset that was collected by the Nvidia team using the Dave2 system and use it to train the Machine Learning Model using the CNN (Convolutional Neural Network) Deep Learning technique that will help us in getting the correct steering rotation angle based on the current road condition. Since this will be a simulation of the technique, we will be using a video of a car dashboard to simulate a driver's view and the steering will rotate based on the output received from the Driving System. The dataset is publicly made available by Udacity through open source.



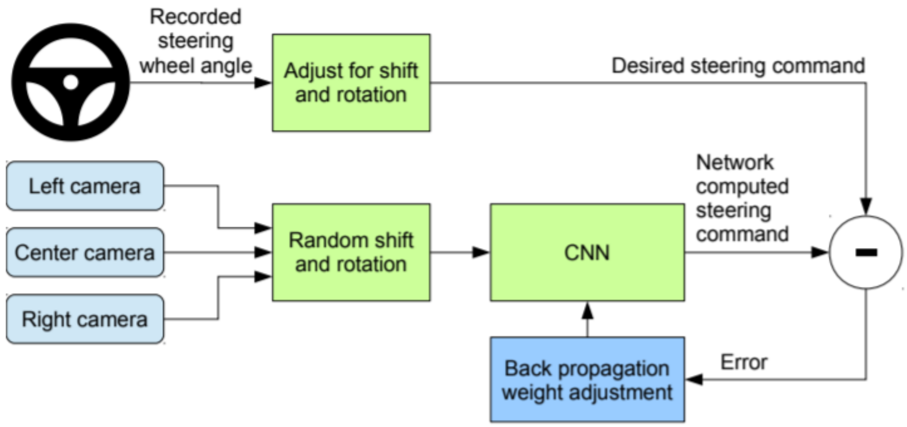


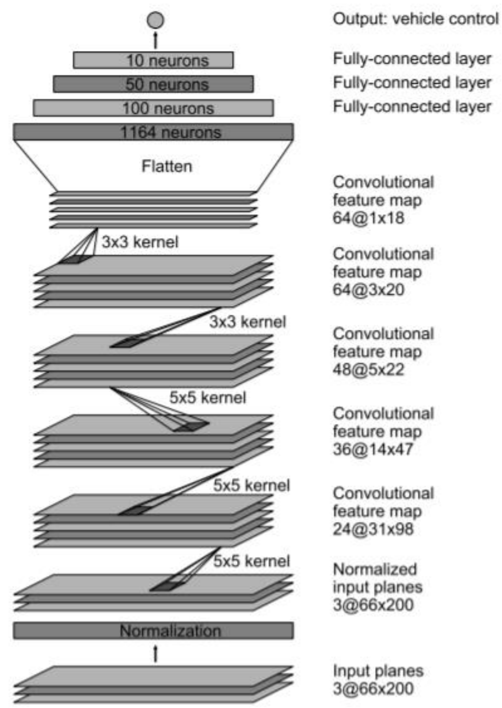
Fig. 1. Conceptual Architecture

B. Architecture

We train the weights of our network to reduce the mean squared error between the steering command output by the network and the human input, or the adjusting steering command for off-center and rotated images The architecture of our model is described in Fig. 1. The network consists of 9 layers which includes a normalization layer, 5 convolution layers, 3 fully connected layers. The input image is divided into YUV planes and passed through the network. The first layer of the network performs image normalization. The normalizer is not adjusted and just hard-coded for learning process. Performing normalization accelerated via GPU processing. The Convolution layers were designed to extract features and going through the series of experiments of the layered configurations. We use strided convolutions in the first three layers with 2x2 strided convolution with a 3x3 kernel size in the last two convolution layers. We have implemented 5 convolution layers and 3 fully connected layers leading to an output control value which is the inverse turning radius

C. Training Details

Data Selection: - The first step to training a neural network is selecting the frame to use. Our data is labelled with road type, weather condition, and the driver's activity. To train a CNN we only select data where the driver was staying in a lane and discard the rest. We then sample the video at 10 FPS. A high sampling rate would results in including images that are highly similar and thus not provide much useful information.



D. Kera's, NumPy, OpenCV

Kera's used in our project: Kera's is a big Machine Learning library created by Google. It is implemented in Python and is used for creating and training Neural Network Models. We have used keras in our project to accomplish various preprocessing tasks like adding the Lambda preprocessing layer, 2D Convolution layer, Max pooling operation for 2D Spatial Data.

For adding the Lambda layer, we have used the following snippet:

```
model.add(Lambda(lambda x: x / 127.5 - 1.,
input_shape=(image_x, image_y, 1)))
```

Similarly for adding the 2D Convolution layer , we have used the following snippet :

```
model.add(Conv2D(32, (3, 3), padding='same'))
```

NumPy used in our project:
NumPy is a library for the Python Programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. We have used NumPy to create arrays of features and labels present in the image. These arrays are basically portable, serialized representations of Python Objects.
For creating arrays for Features and Labels, we have used the following snippet:

```
features = np.array(pickle.load(f))
labels = np.array(pickle.load(f))
```

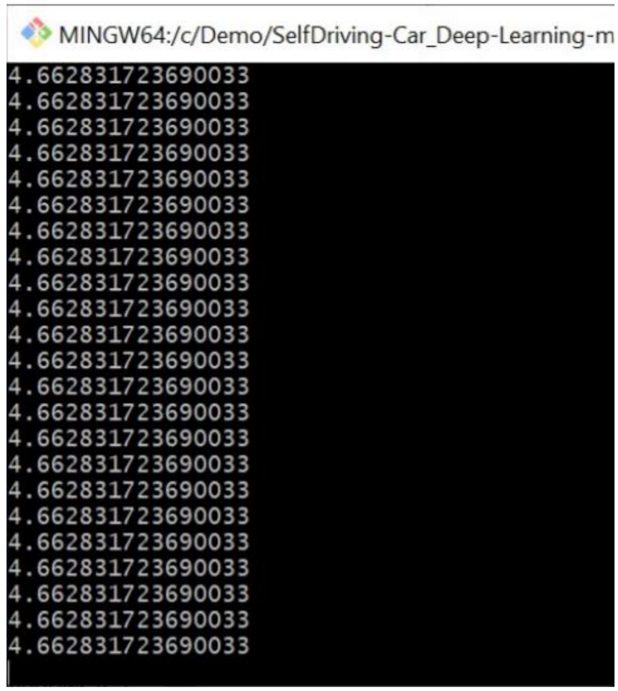
OpenCV in our project:
OpenCV is a large library for image processing, computer vision and machine learning. We read the frames from the input image and use them to do the preprocessing task of resizing the image. The resize function is used to resize the source image down to or up to the specified size, the initial destination and size are not taken into account.
We have used the following snippet to resize the image:

```
resized = cv2.resize((cv2.cvtColor(img,
cv2.COLOR_RGB2HSV))[:, :, 1], (100, 100))
```

V. Results

A. Demonstration

For demonstration of the project, the following are the implementation screenshots . There are 2 frames used in the project, one for the steering wheel and other for the video of the dashcam.



B. Result Evaluation

Evaluating our networks is done through simulation. In simulation we have the network provide steering commands in our simulator to an ensemble of pre-recorded video that consists of a road condition that is considered a normal situation i.e., in normal lighting conditions with lanes and adequate traffic in the city.

We estimate what percentage of the time the network could drive the car (autonomously). The metric is determined by counting simulated human interventions. These interventions occur when the simulated vehicle departs from the Centre line by more than one meter. We assume that in real life an actual intervention would require a total of six seconds: this is the time required for a human to retake control of the vehicle, re-center it, and then restart the self-steering mode. We calculate the percentage autonomy by counting the number of interventions, multiplying by 6 seconds, dividing by the elapsed time of the simulated test, and then subtracting the result from 1:

$$\text{autonomy} = \left(1 - \frac{(\text{number of interventions}) \cdot 6 \text{ seconds}}{\text{elapsed time [seconds]}}\right) \cdot 100$$

if we had 10 interventions in 600 seconds, we would have an autonomy value of

$$\left(1 - \frac{10 \cdot 6}{600}\right) \cdot 100 = 90\%$$

VI. Future Scope and Conclusion

This project demonstrates one of the most effective ways of creating a self-driving car system using the techniques of Machine Learning, Deep Learning, Convolutional NeuralNetworks.

It's one of the most basic systems for creating a self-driven car and is cost effective. The technique is simple and cost effective and doesn't require much pre-processing tasks for image processing which is very crucial while creating a system that's going to run in a constrained environment with respect to processing capabilities. Although this technique is workable, there is still scope for more improvement which will help us to get to the level 5 of autonomous cars. The current system guarantees level 4 of automation. Safety of the passengers in a car is an overarching concern. Many thousands of people die in motor vehicle crashes every year in the world (more than 1,51,417 in 2018); self-driving vehicles could, hypothetically, reduce that number—software could prove to be less error-prone than humans.

References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551, Winter 1989. URL: http://yann.lecun.org/exdb/publis/pdf/lecun-89e.pdf.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012. URL: http://papers.nips.cc/paper/ 4824-imagenet-classification-with-deep-convolutional-neural-networks. pdf.

[3] L. D. Jackel, D. Sharman, Stenard C. E., Strom B. I., , and D Zuckert. Optical character recognition for self-service banking. AT&T Technical Journal, 74(1):16–24, 1995.

[4] Large scale visual recognition challenge (ILSVRC). URL: http://www.image-net.org/ challenges/LSVRC/.

[5] Net-Scale Technologies, Inc. Autonomous off-road vehicle control using end-to-end learning, July 2004. Final technical report. URL: http://net-scale.com/doc/net-scale-dave-report.pdf.

[6] Dean A. Pomerleau. ALVINN, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, 1989. URL: http://repository.cmu.edu/cgi/viewcontent. cgi?article=2874&context=compsci.

[7] Wikipedia.org. DARPA LAGR program. http://en.wikipedia.org/wiki/DARPA_LAGR_ Program.

[8] Danwei Wang and Feng Qi. Trajectory planning for a four-wheel-steering vehicle. In Proceedings of the 2001 IEEE International Conference on Robotics & Automation, May 21–26 2001. URL: http: //www.ntu.edu.sg/home/edwwang/confpapers/wdwicar01.pdf.

[9] DAVE 2 driving a Lincoln. URL: https://drive.google.com/open?id=0B9raQzOpizn1TkRIa241ZnBEcjQ