INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

Data preprocessing and balancing to enhance end to end learning in self-driving vehicle

Syed Zishan Ali
zishan786s@gmail.com
Bhilai Institute of Technology, Raipur, Chhattisgarh

Rishabh Sharda
sharda.rishabh24@gmail.com
Bhilai Institute of Technology, Raipur, Chhattisgarh

Abhishek Dewangan
luc.dewangan@gmail.com
Bhilai Institute of Technology, Raipur, Chhattisgarh

Sourabh Chawda
sourabhchawda30@gmail.com
Bhilai Institute of Technology, Raipur, Chhattisgarh

Rahul Sharma
kukretirahul26@gmail.com
Bhilai Institute of Technology, Raipur, Chhattisgarh

ABSTRACT

Driving a vehicle has always been a demanding task be it any vehicle since robotics and artificial intelligence has progressed multi-folds in the last decade, this gave us the technological grounds to automate many processes which include driving. Developing autonomous vehicle is a current research area for many corporate like are Google, Tesla, Nvidia, and Uber. Several proposed methodology by them is Nvidia's Behavioural Cloning, CommaAI's OPENPILOT, Tesla's AUTOPILOT all of which uses the camera to process surrounding of the vehicle. In this paper, we discuss Nvidia's recent work (behavioral cloning) and incorporate their work with few techniques of our own like filtering the repeating data and augment the input data to reduce the amount of data collection required.

Keywords: *Convolutional Neural Network (CNN), Autonomous vehicle, Artificial Intelligence (AI), Machine learning (ML), Behavioral cloning, Simulated environment, Graphical processing unit(GPU), Model (complete CNN with other processing code).*

1. INTRODUCTION

An autonomous agent is an entity that can act independently and can take decision according to its environment since defining each and every case and situation that the vehicle may encounter in the real world is impractical (as there may be infinite possibilities in the real world). Some sort of learning must be implied which is when machine learning comes into play. It is a sub-domain of Artificial Intelligence which uses neural networks (Figure-1) for learning and computing. Unlike standard algorithm which follows only specific steps and usually solves the problems whose solution is known to us, on the other hand, neural networks learn by example. They cannot be programmed to perform a specific task [1]. Further, the network consists of 9 layers so for more general mappings we consider successive transformations [4].

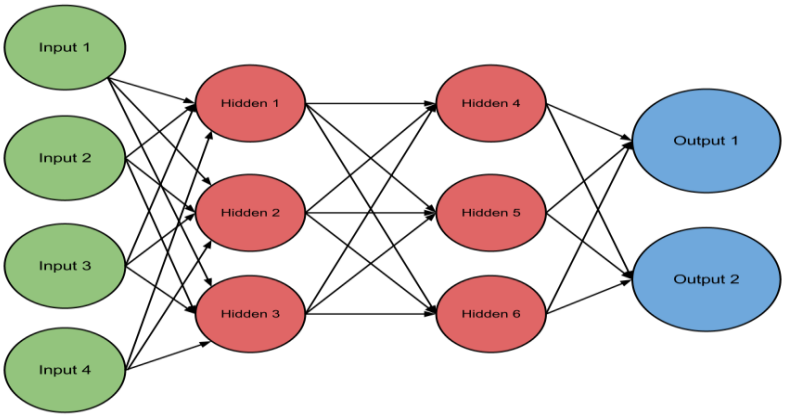

Fig-1: Neural Network

1.1 Convolutional Neural Network (CNN)

CNN is a deep, feed-forward neural network which is used to analyze visual images. In the Convolutional layer, a weight matrix is defined which stores the specific feature of an image.

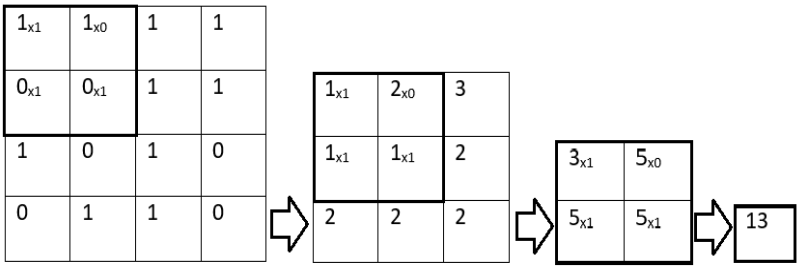

Fig-2: CNN processing

The input image is treated as pixel matrix computed against the weight matrix to produce the output information. In this model we use CNN (Figure-3) that learns entire driving process which was used in Nvidia's behavior cloning [3], grounds to which came from Defense Advanced Research Projects Agency (DARPA) from around 10 years back known as DARPA Autonomous Vehicle (DAVE), in which an RC car drove through an alley filled with junk. DAVE was trained by human driving data. Since end-to-end had shown potential further research and work started on it called DARPA Learning Applied to Ground Robot (LAGR) [5].

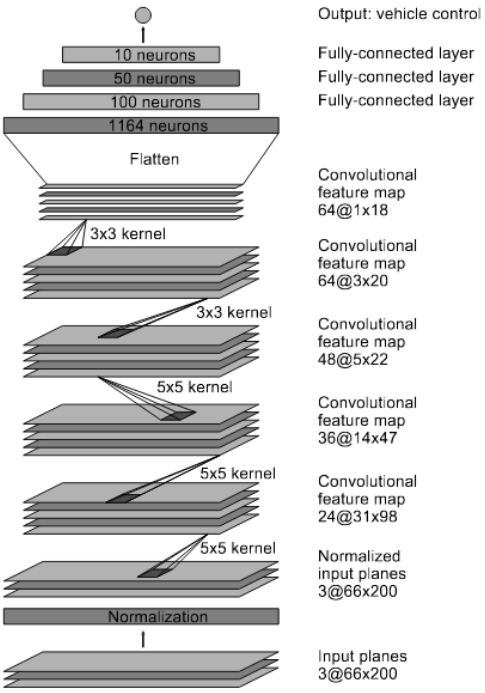

Fig-3: CNN Architecture

1.2 Behavioral-Cloning

Behavioral cloning involves training of neural networks by providing a dataset of images and control logs that is generated while the human driver operates the vehicle.

This deep learning method learns by example through processing the dataset in the convolutional neural network (CNN) and hence is called behavioral cloning.

The aim of this method is to train the model that mimics the behavior using the collected data. This reduces the complexity involved in training and testing the model and also the model is able to tackle new driving situations more efficiently.

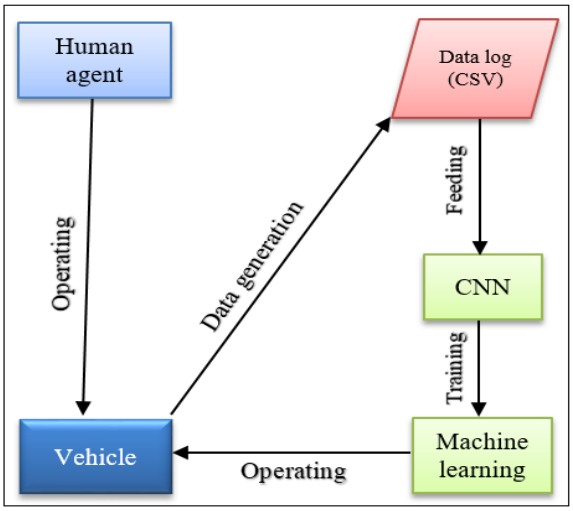

Fig-4: Cloning Life-Cycle

2. METHODOLOGY

The process to train the agent includes data generation by human driver, the collected raw data is then processed by filtering (data augmentation) only significant data from rest also there maybe instances where similar data is generated (data input is very large) so to avoid processing repetitive data we implement data balancing which generally divides the data into bins and using fixed amount of frames from each bin which provides us more balanced data.

Now feeding the data to CNN yield better-trained model which then operates the vehicle. The above process can be divided into three phases:

2.1 Data Generation

- The data is generated using the frames from the center, left, and right cameras, as well as it is associated with the information from different driving statistics (like throttle, brake, speed, and steering angle).
- This information is stored in the form of .csv format (Comma Separated Value format).
- Initially, this data is generated from a human agent (human driver) then the data is modified and used to autonomously drive the vehicle.

2.2 Training Mode (Behavioral cloning)

- The data generated is trained using the different layers of the convolutional network.
- We used some of the augmentation techniques like flipping, cropping [7] to increase the dataset by a factor of 2:
  - **Left and right cameras**: we get frames from 3 camera positions: left, center and right. We can use left and right cameras along with the center camera's data during training after applying steering angle correction.

```
views = ['left', 'center', 'right']
steering_correction = [.25, 0., -.25]
camera = np.random.randint(len(views))
image =mpimg.imread(data[views[camera]].values[i])
angle = data.steering.values[i] + steering_correction[camera]
```

  - **Horizontal flip**: For every set of data we flip half of the frames horizontally and change the sign of the steering angle.

```
flip= random.sample(range(x.shape[0]), int(x.shape[0] /2))
x[flip] = x[flip, :, ::-1, :]
```

```
y[flip] =-y[flip]
```

- **Vertical shift:** In each frame, we have insignificant portions of the sky which is of no use. So we can remove those portions during the pre-processing which will increase the ability of the model to generalize.

```
tp =int(random.uniform(.325, .425) * image.shape[0])
bm =int(random.uniform(.075, .175) * image.shape[0])
image = image[tp:-bm, :]
```

- **Random shadow**: By decreasing brightness of a frame slice, we have added a random "shadow" in the frame.

```
r, j= image.shape[0], image.shape[1]
[x1, x2] = np.random.choice(j, 2, replace=False)
k = r/ (x2 - x1)
b =- k * x1
for i inrange(r):
    c =int((i - b) / k)
    image[i, :c, :] = (image[i, :c, :] *.5).astype(np.int32)
```

- These processes are depicted in the following figures.

Fig-5: Original Frames

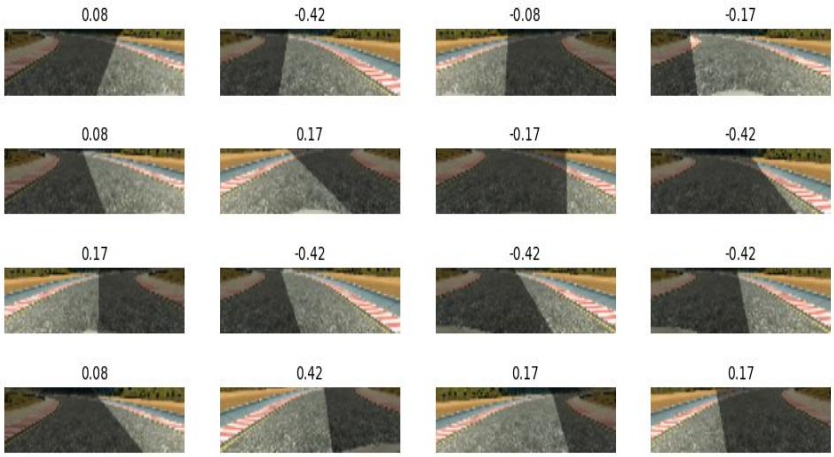

Fig-6: Augmented and Pre-processed Frames

2.3 Testing

For testing the model we use the UDACITY'S simulator for the self-driving car nano-degree program [8] which they open sourced recently. Using data augmentation, the model require less input data as it can produce more data from the given input and then by filtering similar data (data balancing) to be fed into the CNN Which then calculates a proposed steering command which is compared to the desired command and the weights of the neural network is adjusted so as to produce the expected output. It is a backpropagation neural network design consisting of fully interconnected layers in which sigmodium function play an essential role [6].

Fig-7: Testing in a different environment

3. CONCLUSION

In this paper, we proposed the method which enhances the result of end-to-end learning referred by the nvidia's proposed method. The model was equally skilled in driving with fewer amounts of data than used in the referred work.

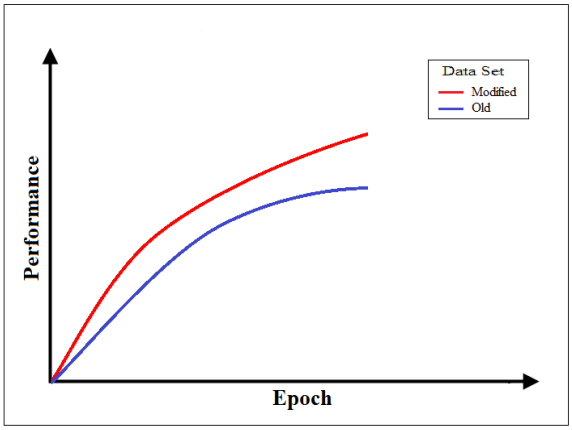

Fig-8: Performance graph

Even when the model is tested in new driving conditions, it is able to drive fairly well like expected (refer Figure-7). Hence it learns the behavior depicted to it and develops a sense to drive, Hence the method is a success.

4. FUTURE SCOPE

Further, the model can be used in a GPU (Graphical processing unit) can be implemented in the real world which would involve further work on the following area:

4.1 Software

Some of the software up gradation can be done from the following fields:

- Using the deep Convolutional network to restore diffractive image [2] for better performance in case of bad image case like camera noise, image saturation.
- Passenger state/behavior may be taken into account and based on that the route or the pace of the car may be decided for example - A casual commuter may be allotted a medium pace route while a medical emergency may be allotted higher pace and a shorter path, for this using internal cameras and deep CNN to judge the behavior and state [9].
- Along with the cloning method, training with specific labeled data can be done which will enhance its understanding with real-world entities. This will allow the model to properly function in complex situations and environments like construction areas or when an accident occurs on the road.

4.2 Hardware

In this method we augmented and balance the imagery data, so providing the CNN with more set of sensory input like:

- Ultra-Sound sensors
- Proximity sensor
- Electronic compass
- Light sensors
- sBluetooth

Using a variety of sensors will enable us to factor in various physical parameters while driving along with the ability to communicate with other autonomous vehicle making it a smart object, this would allow better road dynamics and traffic management [10].

5. REFERENCES

[1] Kumar, D. and Sagwan, H. (2015). RESEARCH PAPER ON BASIC OF ARTIFICIAL NEURAL NETWORK. *International Journal of Innovative Research in Technology 1*(12), 1086

[2] Li Xu,Jimmy SJ. Ren,Ce Liu,Jiaya Jia, "Deep Convolutional Neural Network for Image Deconvolution". NIPS 2014. URL: https://papers.nips.cc/paper/5485-deep-convolutional-neural-network-for-image-deconvolution.pdf

[3] Bojarski, M. *et al*.," End to End Learning for Self-Driving Cars". URL:https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf.

[4] Christopher M. Bishop (1995). *Neural network for pattern recognition.* United States of America

[5] Wikipedia.org, DARPA LAGR program. URL: https//en.wikipedia.org/wiki/DARPA_LAGR_Program

[6] Robert Hecht Nielson. *Theory of Backpropagation in neural network.* I-593

[7] Jia Shijie, Wang Ping, Jia peiyi & Hu Siping, "Reseach on data augmentation for image classification based on convolutional neural network", 2018 Jan, IEEE 10.1109/CAC.2017.8243510

[8] Udacity's self-driving car simulator. URL: https://github.com/udacity/self-driving-car-sim

[9] Shiyang Yan, Yuxuan Teng, Jeremy S. Smith & Bailing Zhang, "Driver behavior recognition based on deep convolutional neural networks", 2016 OCT, IEEE 10.1109/FSKD.2016.7603248

[10] Cristian González García, Daniel Meana-Llorián, B. Cristina Pelayo G-Bustelo & Juan Manuel Cueva Lovelle, "A review about Smart Objects, Sensors, and Actuators", 2017,International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, Nº3.

