National Conference on Engineering Innovations and Solutions (NCEIS – 2018)
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 6 | ISSN : 2456-3307

Simulation of Lane Switching in Self-Driving Automobiles using GTA-V

Varshini Reddy[1], Ajith Kumar V P[1], Akshay T Patti[1], Karthik H S[1], Dr Yuvaraju B N[2]

[1]Student VIII sem, BE, Computer Science and Engineering,The National Institute of Engineering, Mysuru, India

[2]Professor, Department of Computer Science and Engineering,The National Institute of Engineering, Mysuru, India

ABSTRACT

The key significance of a self-driving automobile is it is a mechanical contraption that can progress between objectives without human maneuvers, sounds exceptionally essential and clear yet, honestly, this scarcely covers the surface. For a self-driving automobile to come to affirmation, we require both gear fragments and programming packs that we compose and construct congruous with each other. In this paper, we exhibit the item points of view vital to producing a model that can make sense of how to drive an automobile in a to a great degree diverse plan of a virtual condition. To content with the software aspects of a self-driving vehicle, we make use of Convolutional Neural Networks (CNN) that works on the idea of regression at its crux. We further discuss the information outlines which shape the foundations of the proposed procedure. The process involves screen capturing by employing OpenCV while physically driving a vehicle in a PC amusement, GTA-V.

Keywords : Self-driving, CNN, Automobile, GTA-V, Simulation

I. INTRODUCTION

According to a survey by WHO more than 1.25 million people die in road traffic crashes and cause an additional 20-50 million people injuries or disability. 90% of these road crashes occur in low and centre pay nations. Road crashes cause tremendous monetary losses to individuals, their families and to the country as a whole. One strategy to tamper this global problem by a substantial amount is a self-driving automobile. Self-driving automobiles have the potential to significantly reduce the number of fatalities. They conjointly cut back the time period and fuel consumption, therefore truncating the extent of pollution.

An efficient self-driving automobile must hold the potential to eliminate or out or alleviate each one of the issues talked about beforehand. Self-driving automobiles are by and large composed of a variety of hardware equipment and software packages considered together. The key issues that a self-driving automobile designer must deal with are creating and maintaining maps for self-driving automobiles to tread, complex social interactions, change in climatic conditions and the driving approach to employ accordingly, regulations and political hindrances and cyber-security obstacles. A far more difficult hurdle, meanwhile, is from the fact that driving is an exceedingly communal process that as often as possible includes perplexing communications with different drivers, cyclists and pedestrians. In huge numbers of those circumstances, people depend on summed up insight and presence of mind that robots still particularly need. Fully self-driving automobiles will ultimately need to be adept at four key errands: 1) understanding the surrounding environment; 2) understanding why the general population they experience out and about are acting the way they are; 3) deciding how to respond and 4) communicating with other people. Out of the numerous choices that are to be made on the event snags that a automobile faces while conveying on roads, one major decision to be made is lane switching. An ideally working self-driving automobile must have the capacity to transit between lanes in a manner that abides by traffic rules, spare time and energy.

To stimulate lane switching in a self-driving automobile we use Convolutional Neural Networks (CNN), a class of deep, feedforward artificial neural networks in machine learning used to analyze images. They utilize varieties of multilayer perceptrons intended to require negligible preprocessing. A CNN comprises an information layer and a yield layer and in addition various concealed layers. Convolutional Neural Networks use relatively little pre-processing compared to other image classification algorithms[10]. Biological model of connectivity patterned between the neurons in the animal visual cortex stands as the ingenuity for the conceptualization of convolutional neural networks.

Organization of the rest of the paper is as follows. Segment II cities related work. In Section III, we depict our data set and the systems for formation of the prediction model. Section IV depicts various methodologies providing an aid to model creation. We then present the result and an analysis of the model in Section V. Finally, we conclude the paper with Section VI.

II. RELATED WORK

In the recent times, there is a lot of interest in studying in the field of automation and self-driving automobiles. Although there have been multiple studies on self-driving automobiles, due to the ceaseless changing nature of traffic patterns, human behaviour and rising congestion and tailbacks the designing of the system is an iterative process to meet with the new requirements. Jiman Kim et al [1] proposes a consecutive end-to-end exchange learning technique to assess left and the right sense of self-paths specifically and independently with no post-processing. This approach does not include post-processing and is in this manner adaptable to change of target space. A further study to improve the efficiency of lane switching uses the concept of convolutional neural network to map raw pixels from a single front-facing camera directly to steer commands put forward by Mariusz Bojarski et al [2], however this system performs well only in case of smaller systems because the internal components self-optimize to maximize overall system performance, instead of optimizing human-selected intermediate criteria.

Yet, another body of work has been undertaken to study the process of lane detection and switching, proposed by Lin Li et al [3] that presents a new driver demonstrate in view of human conduct elements for autonomous automobiles, which enables driverless automobiles to move suitably in heeding to the behavioral highlights of driver proprietors. Validation of the proposed model is made consummate by the hardware-in-loop simulator and real driving experiment. From the perspective of human dynamics, this paper introduces the theory of planned behaviour (TPB) into modelling driver for autonomous automobiles. Dissimilar to the conventional approach that physically decomposes the autonomous driving problem into specialized technical segments such as lane detection, path planning and steering control, the end-to-end model proposed by Zhilu Chen et al [4] can directly steer the vehicle from the front view camera information in the wake of preparing, yet with confinements caused because of shifting environmental components.
One of the most recent studies in this field proposed by Chenyi Chen et al [5] describes a direct perception approach to estimate the affordability for driving. The thought set forward is to delineate input image to few recognition markers indicators that specifically identify with the affordance of a road/traffic state for driving. Falling in the middle of the two extremes of interceded discernment and conduct reflexes, the immediate recognition portrayal proposed in this paper gives the correct level of reflection. The demonstration appears via preparing a profound Convolutional Neural Network, therefore, demonstrating that this model can function admirably to drive an automobile in an exceptionally differing set of virtual conditions. virtual conditions.

III. DATASET AND FRAMEWORK

This section describes the dataset used to train the model and the framework of the proposed system

Dataset

The data we use in training our model is self-engendered. The process of data generation involves screen capturing by employing OpenCV while physically driving a vehicle in a PC amusement, GTA-V. The data frames so obtained by screen capturing is of high graphics quality, thus taking up large volumes of space. To ensure low storage with inversely proportional volumes of data, we first convert the RGB pixelated images to grayscale, thus reducing the size by 3 times approximately. The data frames so obtained undergo noise filtering, smoothing and sharpening. Canny algorithm realizes these processes, which achieves it through 5 stages; smoothing to remove noise or remove it, finding gradients to determine the edge strength, non-maximum suppression converts blurred edges to sharp edges, double thresholding distinguishes noise and colour and finally, edge tracking which includes strong edges in the final image and weak edges if they are in connection with a strong one. At the end of the preprocessing stage of data, we obtain a dataset that consists of approximately 100000 frames, filtered and made proficient to act as an input to the prediction model that is portrayed in the accompanying section.

System Model

Our approach consists of the idea of regression at its crux that is set up with a prodigious image dataset that has been procured from different conditions delivered by physically driving an auto in a PC amusement. Based on the input given, the training ensures that the model as shown in fig[1] can determine the appropriate action to take without any human interference. The accuracy of its decision lies on its trained data-sets from an image that it receives, the quality of the image and the number of course changing factors taken into account. The aim of our approach is to convert frames per second into appropriate steering angle. We obtain the input data frames are by screen capturing utilizing OpenCV while manually driving a automobile on a GTA 5 game. The data set is further scaled down to an optimal resolution suitable for training. This scaled data set acts as the input to the neural network consisting of multiple hidden layers. Each layer applies a function to transform the input that has been broadcast to it into an output. CNN make use of filters to assess the frames which improve the efficiency of the system by removing the unwanted parts of the frame which are irrelevant for making driving decisions. This system doesn't require the presence of the exact road markings and signboards. CNN does not require decomposition of the process into several parts such as lane detection and steering control as it can directly steer the vehicle from the front view camera data after training.

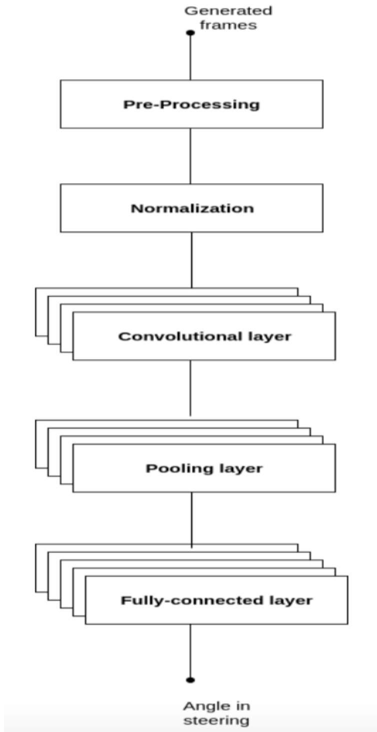

Figure 1. Network architecture

Pre-Processing: The preprocessing is a progression of tasks performed on the generated input frames. It essentially resizes and enhances the frame making it suitable for training. It fundamentally incorporates converting the RGB image frames into grayscale and resizing it to a lower resolution. Alternate activities incorporated into this stage noise filtering, smoothing and sharpening. This stage is imperative for influencing the preparation and forecast to process quicker.

Normalization: The second phase of the architecture performs image normalization. Normalization done prior to the training process is crucial to obtain good results as well as fasten significantly the calculations. This stage guarantees that every one of the data sources is at a tantamount range

Neural Network: This stage incorporates various convolutional layers designed to perform feature extraction. Convolution preserves the spatial relationship between pixels by learning image features utilizing bijou squares of input data. A weight matrix called filter slides over the input image to produce a feature map. The convolution of another filter (with the green outline), over the same image gives a different feature map. Convolutional Neural Network learns the values of these filters independently amid the preparation procedure.

Pooling: Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map while retaining the most important information. To rescale a large image, one natural approach is to aggregate statistics of these features at various locations. Spatial Pooling can be of different types: Max, Average, Sum etc. These summary statistics are much lower in dimension and improves results.

Max pooling: It registers the maximum value of a particular feature over a region of the image.
Mean pooling: It computes the mean value of a particular feature over a region of the image.

Fully connected layers: The frames then pass through a number of fully connected layers, leading to a final output control value which is the inverse-turning radius. The intent of the fully connected layers is to function as a controller for steering.

IV. METHODOLOGIES

TensorFlow is an open source programming library made by Google which is utilized to configure, construct and train profound learning models. The library of TensorFlow contains various powerful algorithms to do numerical computations, which in itself doesn't seem all too special, but achieve these computations with data flow graphs. In these graphs, edges depict the data while the nodes illustrate the mathematical operations, usually are multidimensional tensors and/or data arrays, that are conveyed between these edges. The operations which neural networks perform on multidimensional data arrays or tensors is literally a flow of tensors, hence the name "TensorFlow".

We utilize TensorFlow in our model to learn how to automatically spot a complex pattern or image. Depending on the images recognised the system takes the best possible decision independently. Further, we construct a computational graph that consists of nodes represents an operation and edges which represents multi-dimensional data arrays using TensorFlow. Subsequent to defining the operations we set up a TensorFlow session in order to perform calculations on the defined graph.

We employ Tensorflow TFlearn package to create an Alexnet, a type of CNN which contains five Convolutional layers and three fully connected layers as shown in fig.(2). The design of the convolutional layers is to perform feature extraction. It preserves the spatial relationship between the pixels by learning image features using small squares of input data. A weight matrix called filter slides over the input image produces a feature map. The network learns values of these filters on its own during the training process. The design of the fully connected layers is to function as a controller for steering where the image frames pass through these layers, leading to a final output control value which is an inverse turning radius. Rectified Linear Unit(ReLU) [6],[7] is applied after every convolutional and fully connected layer. ReLU is a function first introduced by Hahnloser et al. It was then stated by Nair et al that ReLU is an effective activation for use in neural network as well. ReLU function is given by:
f(x)=max(0,x)

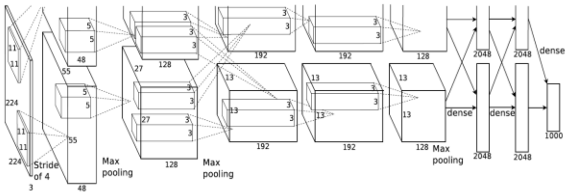

Figure 2. Alexnet Architecture

Dropout is applied before the first and the second fully connected layer. Dropout helps in removing complex co-adaption. Removal of complex co-adaption implies training node in a neural network with a randomly selected sample of other nodes. This makes the node more robust and drive it towards creating useful features, without relying much on other nodes. Overlap pooling is used to reduce the size of the network.

Further, we use TensorFlow object detection API to detect other vehicles on road and to determine the distance from them. They use pixels from the input matrix data set as predictors and predict which operation the vehicle has to perform (turn left/ turn right or / straight).

V. RESULTS AND DISCUSSION

In this section, we discuss the results of the prediction model. We have studied the effects of parameter variations in the tensor flow module. Our model works by providing an array of 3-values as the outcome, each of which is a floating number in (0,1). The array elements indicate the prediction of each direction namely; front left and right, made for that particular instance of time. Consequently, for every repetition of the same instance if the prediction value is above a threshold chosen then that action is performed accordingly. The threshold values we have appointed are 0.7 for the forward motion and 0.75 for both the turn (left and right) motion.

Making diligent alterations to the Alexnet parameters to obtain the most efficient model that results in the relatively high accuracy. The following graphs shown by fig (3), fig fig(4), fig(5) and fig(6) depict the accuracy of the prediction model. We have created proliferates as the size of the data frame input increases and on modification of parameters in the Alexnet. Thus, we can state with confidence that the prediction model has a high accuracy when the input frames are comparable to the size of the data set when the parameters are tuned to specific values.

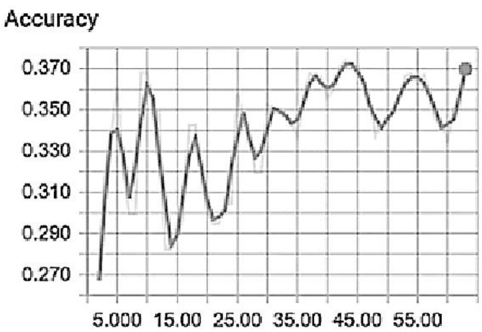

Figure 3. Accuracy plot with x-axis representing number of data frames (max value of x-axis : 55)

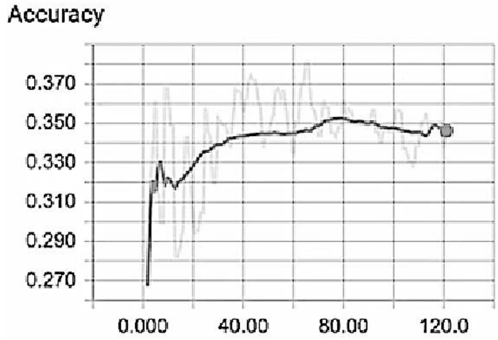

Figure 4. Accuracy plot with x-axis representing number of data frames (max value of x-axis : 120)

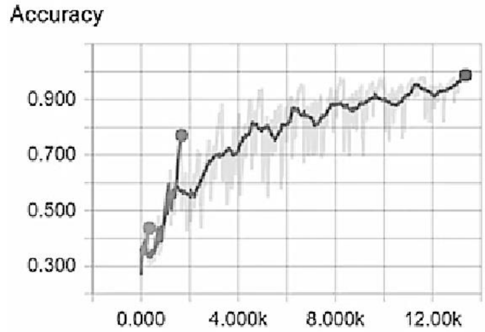

Figure 5. Accuracy plot with x-axis representing number of data frames (max value of x-axis : 12000)

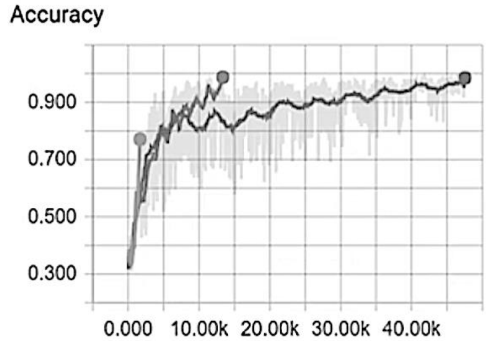

Figure 6. Accuracy plot with x-axis representing number of data frames (max value of x-axis : 40000)

VI. CONCLUSION AND FUTURE WORKS

In this paper, we have introduced the concept of lane switching in self-driving automobiles. We have clarified in detail Alexnet, a type of convolutional Neural Network which is most appropriate for this application. Put into words, the generation of input data sets using OpenCV, the system architecture and the methodologies employed for simulation. Further, a demonstration of the transition in system accuracy with the varying volumes of training data is portrayed through graphical depictions.

Robotization in automobiles is a fast developing field in the present world today. To stay aware of the changing scenario and requirements, the system must be updated frequently. Aside from the reenactment of lane switching in self-driving we have delineated through this paper, there are couple of more areas under this vast field that can be enhanced in future. One such feature would be the capacity of a self-driving vehicle to have the capacity to discover the way of slightest harm if there should arise an occurrence of unavoidable conditions. This would diminish the dangers caused by an extraordinary degree.

VII. REFERENCES

1. Jiman Kim, Chanjong Park, “End-to-End Ego Lane Estimation based on Sequential Transfer Learning for Self-Driving Car”, 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)
2. Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller,Jiakai Zhang, Xin Zhang, Jake Zhao, Karol Zieba, “End to End learning for Self-Driving Cars”, Computer Vision and Pattern Recognition
3. Lin Li, Yanheng Liu, Heekuck Oh, Jian Wang “Human dynamics based driver model for autonomous car”, IET Intelligent Transport Systems
4. Zhilu Chen, Xinming Huang, “End-to-End Learning for Lane Keeping of Self-Driving cars”, 2017 IEEE Intelligent Vehicles Symposium (IV)
5. Chenyi Chen, Ari Seff, Alain Kornhauser, Jianxiong Xiao, “Deep Driving- Learning Affordance for Direct Perception in Autonomous Driving”, 2015 IEEE International Conference on Computer Vision (ICCV)
6. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”
7. Wenling Shang, Kihyuk Sohn, Diogo Almeida, Honglak Lee, “Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units”, 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48
8. Dosovitskiy, A. and Brox, T, “Inverting convolutional networks with convolutional networks”, In CVPR, 2015
9. A Berg, J. Deng, and L. Fei-Fei, “Large scale visual recognition challenge” , 2010, www.imagenet.org/challenges. 2010
10. D. Cire¸san, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification”, Arxiv preprint arXiv:1202.2745 2012
11. Y. LeCun, K. Kavukcuoglu, and C. Farabet, “Convolutional networks and applications in vision”, In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pages 253–256. IEEE, 2010
12. H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”, In Proceedings of the 26th Annual International Conference on Machine Learning, pages 609–616. ACM, 2009
13. P.Y. Simard, D. Steinkraus, and J.C. Platt, “Best practices for convolutional neural networks applied to visual document analysis”, In Proceedings of the Seventh International Conference on Document Analysis and Recognition, volume 2, pages 958–962, 2003
14. Shin HC, Roth HR, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, Summers RM, “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning”, IEEE Trans Med Imaging. 2016 May