Small Scaled Autonomous Vehicle

Saumendra Kumar Mohanty, Badri Narayan Sahoo

Abstract— This paper presents a self-driving car using low cost convolutional neural network based platform. The project replicates the real autonomous vehicle/car into a small scaled remote controlled (RC) car. The CNN (convolutional neural network) provides a high accuracy prediction. A small camera is mounted on the RC car for capturing real time video feed to firstly train and then to predict the direction. The camera provides video feed to the input layer of the CNN that has hidden layers, that analyzes and compute the data to predict the output from an output layer that tells rc car to move in predicted direction.

Keywords: GPU, convolutional neural network, grey scale

I. INTRODUCTION

Convolutional Neural Networks(s) [1] revolutionized the way of pattern recognition and detection [2]. Before adopting and wide spreading of the CNN(s), the most common pattern recognition operations were operated by using a first stage of extraction of the feature by hand crafted feature extractor and after that a classifier. CNN(s) breakthrough is the automatically learning of the feature from training set of data/examples. The CNN(s) method for image recognition is very powerful due to the fact that it captures and analyzes the images in 2D. And also by using kernels from convolutional network to scan and analyze the full image by using only few parameters rather than scanning full image by using relatively more parameters. CNN(s) have been in the commercial and government use for long time with learned features for more than 20 years [3][4]. These are used for learning have been efficiently implemented on very large and parallel graphics processing unit i.e. GPU(s) that accelerates training and validation of the data sets in very significant manner [5].

Self-driving cars i.e. autonomous cars are now mostly interested topic for everyone as it has been advancing in recent years by many companies, most of the companies are developing new hardware and software for making fully autonomous and highest safety rated vehicle in the market that requires no intervention from any human being. CNN(s) is also developing with an increasingly interest from most companies and governments that is successfully used and applied to various fields and operations for different controls tasks. They are used in combination with other algorithms for developing autonomous vehicle.

For deeply understanding the type of hardware and software needed for an autonomous vehicle, we need to test both hardware and software for a realistic result. Using a real car for testing all hardware and software would not be ideal solution because of many safety and cost concerns. That's why there is need for a method for testing and analyzing results with more safety and more cost effective way.

Therefore in this paper a low cost RC (Remote Control) car is used for testing as a solution for above mentioned problem. An affordable small computer i.e. Raspberry pi is mounted on the frame of RC car along with a power bank to supply power to the pi. One camera mounted in front of the RC car for capturing live video feed and transferring it to another computer via wireless transmission. A motor driver connected between RC car and raspberry pi for controlling motor of the car as per the instructions received from the raspberry pi. One different computer with high specifications of CPU and GPU for training and validating data sets through a convolutional neural network received from the raspberry pi.

II. OVERVIEW OF THE CNN

The convolutional neural network used in this project has input layer, a hidden layer and an output layer. The input layer consists 320x120 inputs for inserting data value of image of resolution 320x120 that is the total input in the input layer is 38400. Hidden layer consists 9 layers for extracting and analyzing features from an image. The output layer consists 4 outputs i.e. forward, backward, left and right direction.

The convolutional neural network receives an image of resolution 320x120 with a high frame rate from the camera mounted on the front of the vehicle and generates a prediction of the vehicle steering and forward direction in real time. The CNN has 32 layers, about 27 million connections and 250 thousand parameters (weights). The CNN architecture is identical to that of NVIDIA's real-sized self-driving car, called DAVE-2 [5], which drove on public roads without human driver's intervention while only using the CNN.

Revised Manuscript Received on September 10, 2019.

Saumendra Kumar Mohanty, Dept. of Electronics Communication Engineering, Siksha O Anusandhan Deemed to be University, Odisha, India.
(E-mail: saumendramohanty@soa.ac.in)
Badri Narayan Sahoo, Dept. of Electronics Communication Engineering, Siksha O Anusandhan Deemed to be University, Odisha, India.
(E-mail: badrinarayansahu@soa.ac.in)

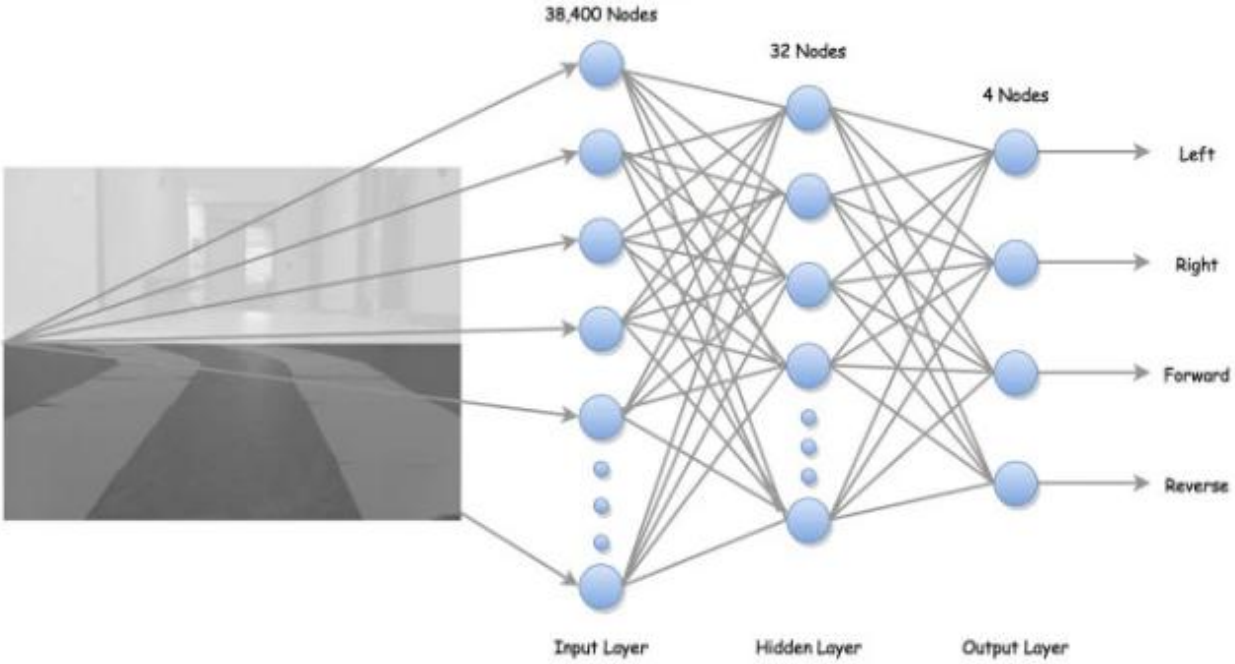

Figure 1

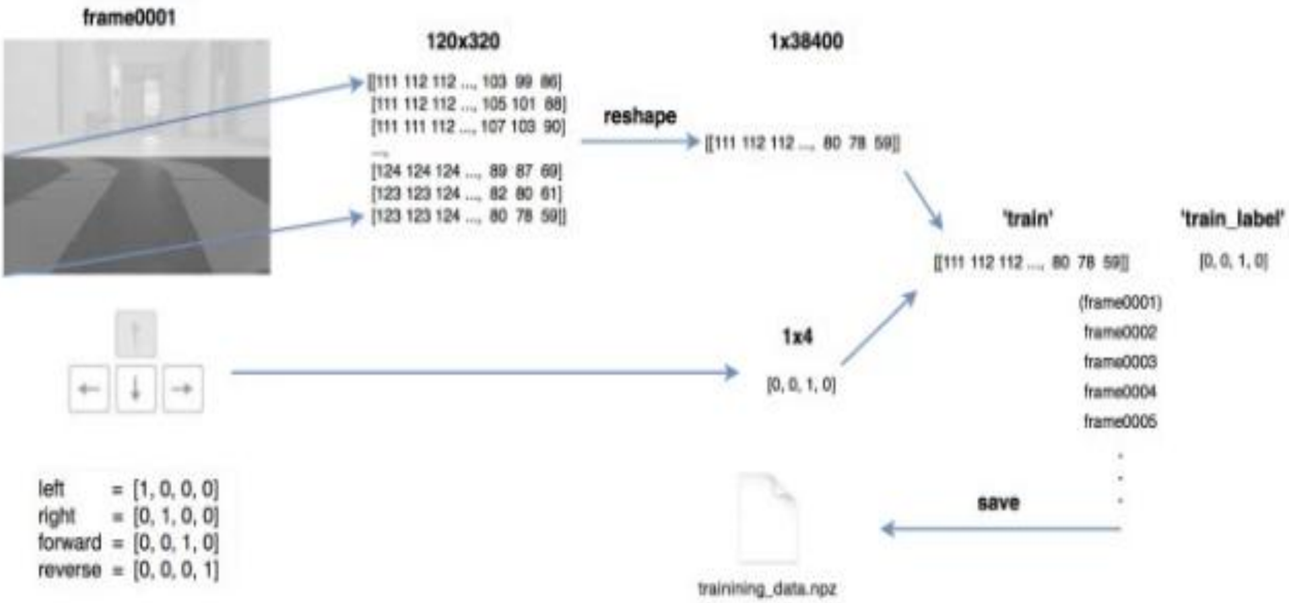

Figure 2

Overview of the DAVE-2 System

The figure 3 illustrates the working diagram of the system that is used for training the data that is linked with DAVE 2. There are 3 cameras that are mounted on vehicle for covering all directions through these cameras. These cameras are used to capture images and videos and for data acquisition. The training data is acquired by human driver steering the vehicle and the cameras capturing images.

The steering wheel records every turn and movement made by driver and then stores it to the hard disk for further computation of data. A GPU is connected between them for training and driving purpose.

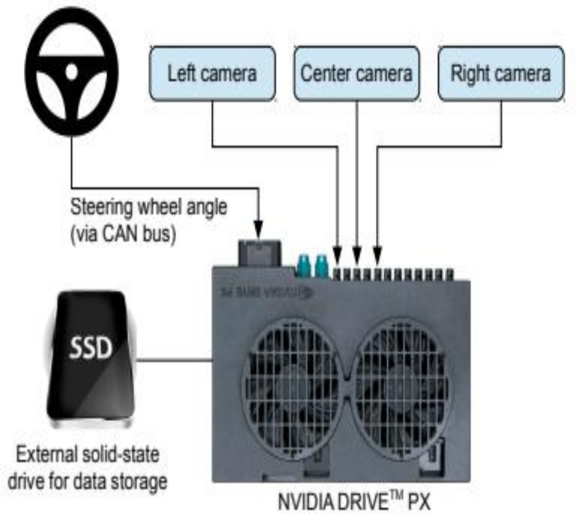

Figure 3

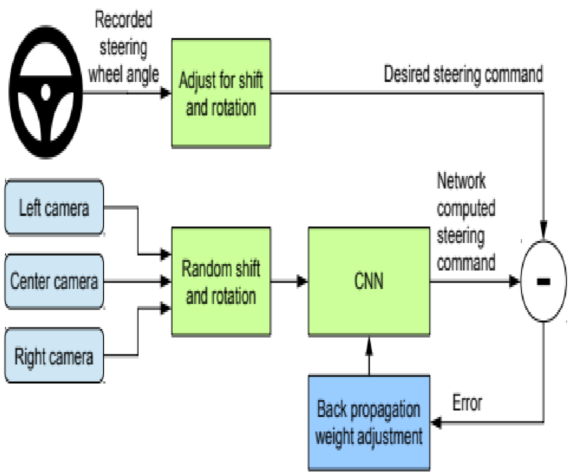

Figure 4

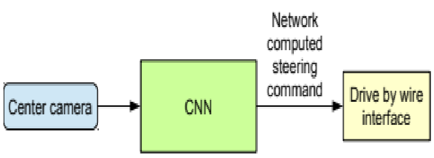

Figure 5

Data collection

The training data is collected by running the RC car on the track shown in figure 6 by controlling the car through the computer, the camera connected to raspberry pi captures multiple images to form a video stream, the captured collected images are then sorted on the basis of the direction of the moving car on the track as shown in figure 6. These collected and sorted images are then resizes to 320x160 pixels for reducing size that results to less computation and hence fast processing. The collected images are in RGB format, meaning the image contains three layers that corresponds to red, green and blue layer that defines all different colors, these images are converted to grayscale [6] for reducing size and increasing computation speed that results to less time consumption.

Figure 6

Training

Collected data is transmitted to the input of the convolutional neural network and labelled output direction is fed to the neural network. The network analyses given images in accordance with the given output and trains itself.

The network uses back propagation method to train data through the neural network.

III. RESULT

We have thoroughly analyzed and demonstrated that convolutional neural network learns pattern recognition and are able to distinguish between different images and act accordingly to give different results for different lanes and tracks. Large amount of training data is required i.e. training data by driving the RC car. The system learns by the recorded data and works accordingly.

IV. REFERENCE

1. Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, 2008.
2. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *ImageNet Classification with Deep Convolutional Neural Networks*, 2012.
3. L. D. Jackel, D. Sharman, C. E. Stenard, B. I. Strom, and D. Zuckert, "Optical Character Recognition for Self-Service Banking," *AT&T Tech. J.*, 1995.
4. Imagenet, "Large Scale Visual Recognition Challenge (ILSVRC)," *ImageNet Large Scale Vis. Recognit. Chall.*, 2015.
5. Z. Chen and X. Huang, "End-To-end learning for lane keeping of self-driving cars," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2017.
6. M. G. Bechtel, E. McEllhiney, M. Kim, and H. Yun, "DeepPicar: A low-cost deep neural network-based autonomous car," in *Proceedings - 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2018*, 2019.