Navigating Self-Driving Vehicles Using Convolutional Neural Network

Minh-Thien Duong, Truong-Dong Do and My-Ha Le*

Abstract - ***In this paper, a method for navigation of self-driving vehicles is proposed. Although the research for this problem has been performed for several years, we noticed that the elevated accuracy results have not been achieved yet. Therefore, the method using a convolutional neural network (CNN) for training and simulation of unmanned vehicle model on the UDACITY platform has been made. Details, we used three cameras mounted in front of a vehicle to follow three directions were left, right and center position to collect data. The data are the images that captured from three cameras. The number of samples image is 15504. In this research, the label with two parameters are the steering angle and speed from each image would also be created. After collecting the data, these parameters will be achieved by training CNN used to navigate the vehicle. With the combination of three cameras, the accuracy of this navigation task is improved significantly. When vehicle deviates to the left, we will compute the error of the steering angle value between the middle and left position. Afterward, the steering angle value will be adjusted to control the vehicle could run in the center of the lane. Similarly, in the case when vehicles deviate to the right. Based on the simulation platform of UDACITY, we simulated and obtained the result with accuracy was 98, 23%.***

Keywords: ***Steering Angle; Navigation; Convolutional Neural Network (CNN); Self-Driving Car; Autonomous Vehicle; Unmanned Vehicle Model.***

I. INTRODUCTION

Today, with the considerable development of technology and transportation, many vehicles are equipped with the self-driving mode to support the driver to maintain health while driving long distances as well as to reduce traffic accident risk.

Navigating trajectory for vehicles was one of the most important aspects of the development of the unmanned vehicle model. There were many methods to do this, but the way to get the best results and match with Industrial Revolution 4.0 is using an algorithm regarding Machine Learning field. Specifically, we made the convolutional neural network (CNN) algorithm to navigate for autonomous vehicles.

Deep learning is a subfield of machine learning that is inspired by an artificial neural network. A specific kind of such a deep network is the convolutional neural network, which is commonly referred to as CNN or ConvNet. The difference of CNN, in comparison to the traditional neural network, is the number of neural in a class may be reduced but the number of hidden layers is greater and called a deep network. They are trained by backpropagation strategy. So it can build the intelligent systems with high accuracy.

Minh-Thien Duong and Truong-Dong Do are students at Ho Chi Minh City University of Technology and Education, Vietnam.

My-Ha Le is with Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Vietnam. (*Corresponding author e-mail : halm@hcmute.edu.vn).

II. RELATED WORK

Formerly, there were many studies on navigation for vehicles, including methods lane detection such as Real-time lane detection for autonomous navigation [1], a lane tracking system for intelligent vehicle application [2], lane detection with moving vehicles in the traffic scenes [3] or the other papers regarding this method are mentioned in [4], [5]. Although this method gives convincing accuracy about lane detection there are several reasons make the unsuccessful detection. The first reason is subjective. After detecting two lines we need to calculate and draw a virtual line at the center then estimate the offset angle between the body of the vehicles and the virtual line to adjust steering of the car so that the vehicles are always in the middle of two lines under any circumstance. The calculations of steering angle that are mentioned above are complicated and can cause many errors. The second reason is objective. Several of the roads are lack of lane or the lane marking is blurred. Also, when the vehicles are running in the sloping street, the camera was mounted at previous will head to the sky and do not keep up with lane at the ahead. This can also lead to detection will be incorrect.

In addition, the GPS method is also applied for navigating self-driving vehicles. If GPS is used standalone, it will cause an error is quite high so apply this method on the model of unmanned vehicles will be very difficult and requires an adjustment of the error [6],[7].

Noticed with the restrictions mentioned above, we propose a method using a convolutional neural network to navigate the self-driving vehicles.

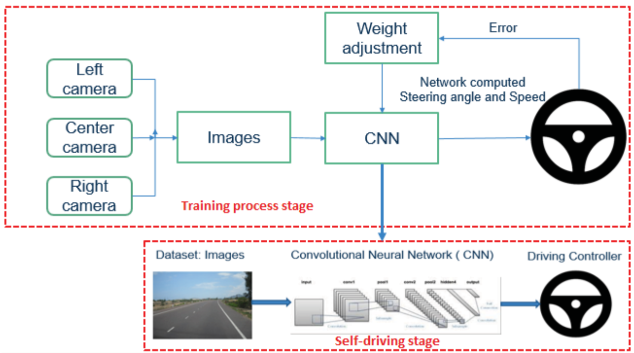

Fig. 1. *The proposed scheme for autonomous vehicles navigation*

In the work, we demonstrate autonomous driving in a simulation environment by predicting steering wheel angles and speed value from raw images which trained through CNN. Data was collected from three cameras later are preprocessed and fed into a CNN that then calculates value steering angle and speed. The proposed command is compared to the desired command for that image and the weight of the CNN are adjusted to obtain the better result (Figure 1).

978-1-5386-5126-1/18/$31.00 ©2018 IEEE

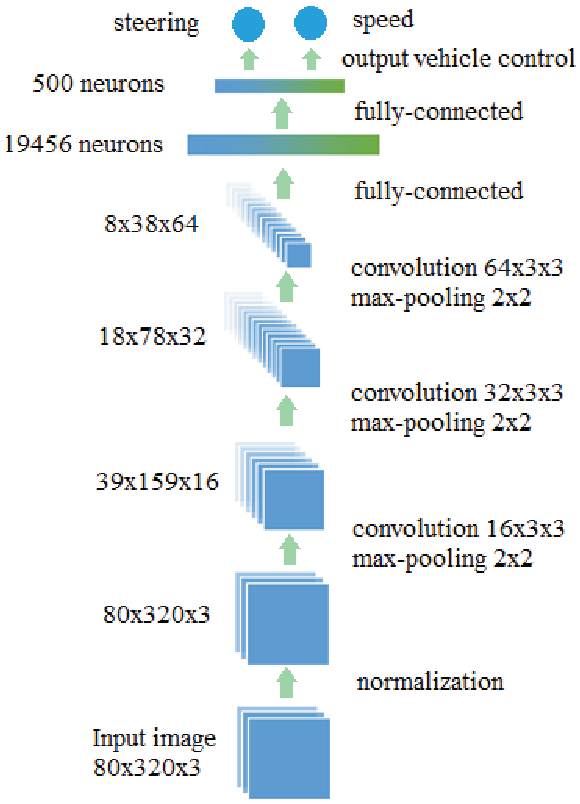

Fig. 2. *The network architecture*

III. NETWORK ARCHITECTURE

A convolutional neural network is used to training data. Authors train the weights of our network to minimize the mean squared error between the command of output vehicle control and the command of the human driver. The network architecture consists of 9 layers, including 1 normalized layer, 3 convolutional layers, 3 max-pooling layers and 2 fully connected layers. Our network architecture was described in Figure 2.

The first layer implements normalized all images have value pixels from -1 to 1. The convolutional layers were designed by experiment, we use kernels have size 5x5 with non-stride for the first layer, the others are 3x3 also non-stride. The respective depth of each layer is 16, 32 and 64.

The max-pooling layers were interleaved with the convolutional layers. Max-pooling layer is one of the powerful tools that usually used by CNN. This is a method that resizes large images but keeps the most important information about them. It involves sliding a small window pane along one image and getting the maximum value from the window at each step. After pooling, an image will have about a quarter of pixels compared to the beginning. This reduces the number of hyperparameters needing to calculate, thence decreasing the computational time and avoiding the overfitting. All max-pooling layers were chosen with kernel was 2x2 and non-stride.

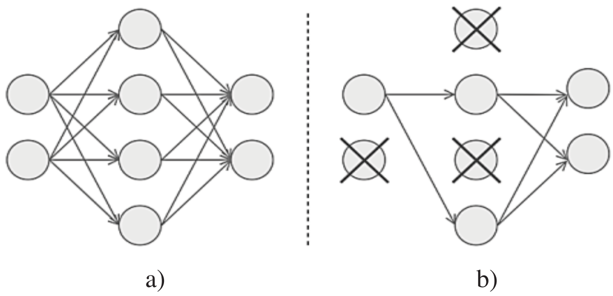

Fig. 3. *Dropout Neural Net Model. a) The standard neural network; b) The neural network after applying dropout technique*

The fully connected layers were designed with gradually reducing sizes: 19456 and 500. The output layer is 2 because our model predicts two values, one is the steering angle and another is speed.

Besides, The layers activation were ELUs (exponential linear units) following each convolutional layers to improve convergence are also used. Function ELUs try to make mean activations closer to zero which speeds up learning. It has been shown that ELUs can obtain higher classification accuracy than ReLUs [8]. The output of a ReLU class is the same size as the input, the difference is that all negative values of the image will be removed shortly afterward.

$$f(x) = \begin{cases} x; x \geq 0 \\ a(e^{x} - 1); x < 0 \end{cases} \quad (1)$$

With a was a hyper-parameter to be and $a \geq 0$ was a constraint.

For this paper, we used the mean-square-loss function. This function is common for regression problem that is simply the mean of the sum of the square difference between the actual and predicted results.

$$MSE = \frac{1}{n} \sum \left(y_i - \hat{y}_i \right)^2 \quad (2)$$

To optimize this loss, the Adam optimizer was used. This optimizer is usually chosen for deep learning application. We used the default parameters of Adam provide in Keras (learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e-8$ and decay = 0) [9].

Fig. 4: The input data

a) *b)*

Fig. 5. The augment data. a) Original Image, b) Flip Image

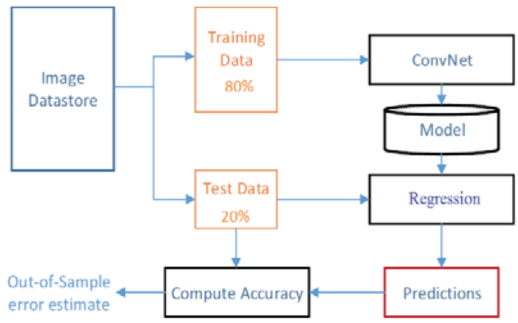

Fig. 6. *The block diagram for accuracy evaluation of proposed network*

Convolutional neural networks contain multiple non-linear hidden layers, this makes them become complication models and lead to complex relationships between their inputs and outputs. To make the architecture more robust and to prevent overfitting, dropout layers added to the network are implemented. Dropout disables neurons in the network by a given probability and prevents co-adaption of features. For this work, we have applied a dropout rate of 20% [10] (Figure 3).

IV. EXPERIMENTS

A. Training process, data and feature

Udacity provided the basic simulation framework for navigating the autonomous vehicle. The training data were collected by driving multiple turns on the road. In the training mode, the simulator records steering, speed and image with size is 160x320 from three cameras (Figure 4).

In addition to augmenting more data to improve accuracy by flipping for all the images previously captured is carried out (Figure 5). The collected data is not only pre-processed by converting from RGB to HSV color-space but also cropping the inconsequential frames and hold regions useful for learning trajectory. Therefore, the image size will be cropped to 80x320. The total samples are approximately 15500 images.

From the stored dataset, we divide them into two part separately. One is the training data and another is test data with proportion 80:20. A training diagram was shown in Figure 6. We take advantage of GPU NVIDIA GeForce GT 740M with memory was 2GB available to train the network with three times, each turn was 30, 50, 100 epoch (Table 2).

Total time that we implement training in conjunction with 100 epochs for 15500 samples was approximately 280 minutes. Therefore, the average one epoch takes 168 seconds to complete. Some images represent the output feature convolutional layers were shown in figure 7.

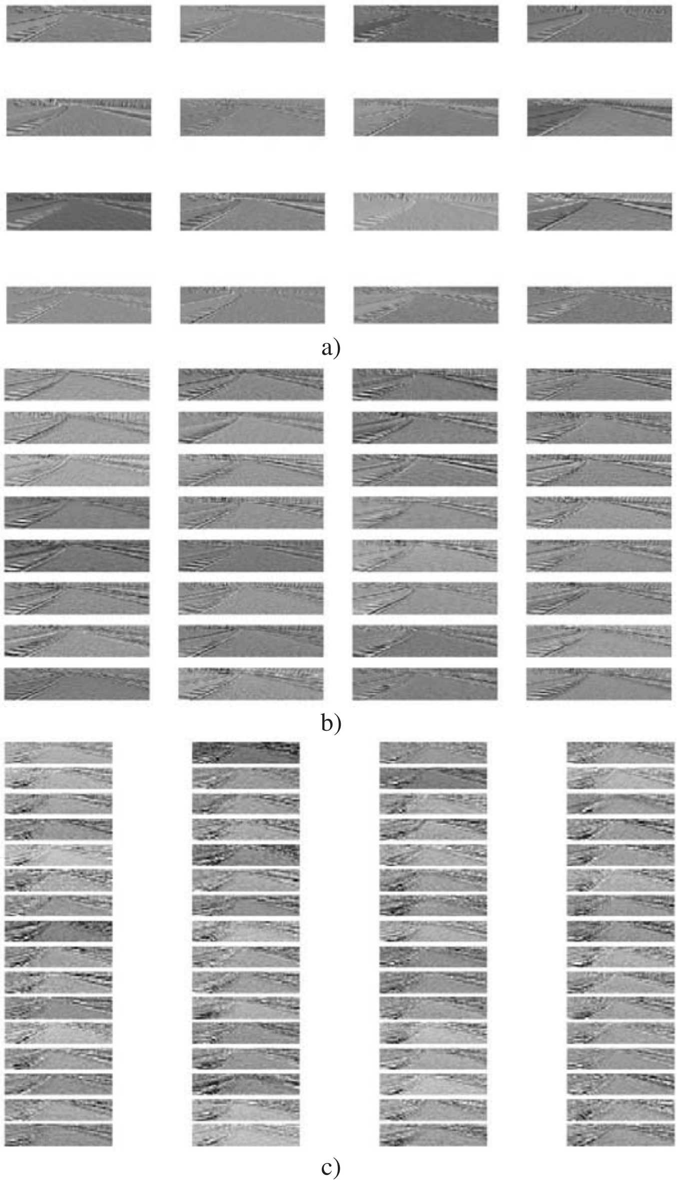

Fig. 7. *The visualization output of three convolutional layers. a) Convolutional layer 1, b) Convolutional layer 2, c) Convolutional layer 3*

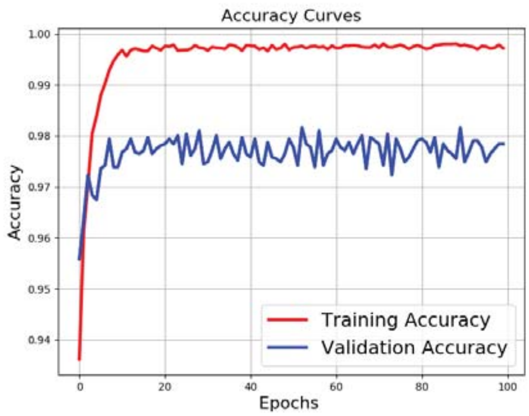

Fig. 8. The accuracy of the proposed deep neural network architecture

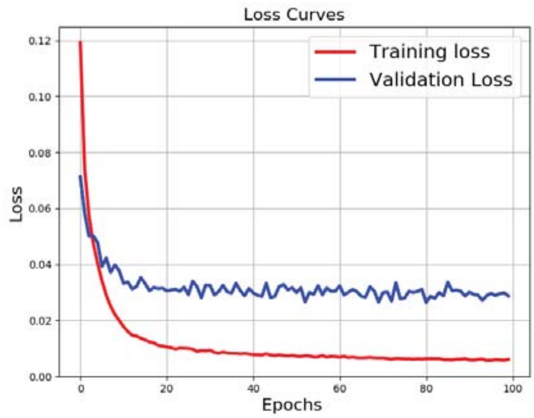

Fig. 9. *The loss of the proposed deep neural network architecture*

B. Autonomous vehicle navigation

We ended up with one virtual car that was able to drive autonomously on the road at simulation of UDACITY through a network had been trained in advance.

The model predicted steering angle and speed accomplish convincing accuracy, namely 98,23% (Figure 8). The result of this experiment on the simulation software is vehicle can drive automatically and follow the lane relatively stable and do not get out of the lane. Value steering and speed are predicted relatively accurately (Figure 10).

V. CONCLUSION

Advantages of this research are authors have improved and obtained a persuading outcome. Data are one of the most important matter lead to the accuracy of our model (Table 1). Collecting data through three cameras to calibration offset steering angle and ensure the vehicle always runs in the center of the lane, which is key to enhance the accuracy.

Moreover, increasing the epoch so that the model approaches the convergence position is a way to have a model with the persuading result (Table 2).

Fig. 10. *The result predicted steering angle and speed of car after training*

TABLE 1. THE EFFECT OF THE NUMBER OF SAMPLES TO THE ACCURACY OF THE MODEL

Number	Number of samples	Epoch	Time of training	Accuracy
1	10000	100	173 minutes	96,78%
2	12500	100	212 minutes	97,15%
3	15000	100	280 minutes	98,23%

TABLE 2. THE EFFECT OF THE EPOCH OF SAMPLES TO THE ACCURACY OF THE MODEL

Number	Number of samples	Epoch	Time of training	Accuracy
1	15000	30	92 minutes	97,08%
2	15000	50	155 minutes	97,77%
3	15000	100	280 minutes	98,23%

On the contrary, in this work has several drawbacks. Firstly, because the simulated environment is so ideal hence the noise from the outside environment is almost nonexistent. The second is the matter about the mechanical error of the vehicle is also ignored. With the restrictions mentioned above, authors will soon experiment real-time autonomous vehicle to prove robust of network architecture.

In short, authors have created a model which predicts the steering wheel angles and speed for a vehicle using a convolutional neural network. In spite of having obtained a satisfactory outcome but in the near future we are going to approach the several research following:

1) Real-time Self-Driving Car navigation using the deep neural network will be the matter that we interested.
2) Experiment with a more complicated landscape.
3) Take into consideration with the model rely on ResNets/VGG through transfer learning method.
4) Consider reinforcement learning as an alternate method to improve the driving ability.

REFERENCES

[1] S. G. Jeong, C. S. Kim, K. S. Yoon, J. N. Lee, J. I. Bae, and M. H. Lee, "Real-time lane detection for autonomous navigation," in Intelligent Transportation Systems, 2001 IEEE, 2001, pp. 508-513.

[2] K. A. Redmill, S. Upadhya, A. Krishnamurthy, and U. Ozguner, "A lane tracking system for intelligent vehicle applications," in Intelligent Transportation Systems, 2001 IEEE, 2001, pp. 273-279.

[3] H. Y. Cheng, B. S. Jeng, P. T. Tseng, and K.-C. Fan, "Lane detection with moving vehicles in the traffic scenes," IEEE Transactions on intelligent transportation systems, vol. 7, no. 4, pp. 571-582, 2006.

[4] K. Ghazali, R. Xiao, and J. Ma, "Road lane detection using H-maxima and improved hough transform," in Computational Intelligence, Modelling and Simulation (CIMSiM), 2012 Fourth International Conference on, 2012, pp. 205-208.

[5] M. Aly, "Real time detection of lane markers in urban streets," in Intelligent Vehicles Symposium, 2008 IEEE, 2008, pp. 7-12.

[6] S. Hong, M. H. Lee, S. H. Kwon, and H. H. Chun, "A car test for the estimation of GPS/INS alignment errors," IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 3, pp. 208-218, 2004.

[7] V. Milanés, J. E. Naranjo, C. González, J. Alonso, and T. de Pedro, "Autonomous vehicle based in cooperative GPS and inertial systems," Robotica, vol. 26, no. 5, pp. 627-633, 2008.

[8] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)", 2015.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", 2014.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014.