International Journal of Internet, Broadcasting and Communication Vol.13 No.1 210-218 (2021)
http://dx.doi.org/10.7236/IJIBC.2021.13.1.210

IJIBC 21-1-25

Implementation of Low-cost Autonomous Car for Lane Recognition and Keeping based on Deep Neural Network model

Mi-Hwa Song

Assistant Professor, School of Information and Communication Science, Semyung University, Jecheon, Korea
mhsong@semyung.ac.kr

Abstract

CNN (Convolutional Neural Network), a type of deep learning algorithm, is a type of artificial neural network used to analyze visual images. In deep learning, it is classified as a deep neural network and is most commonly used for visual image analysis. Accordingly, an AI autonomous driving model was constructed through real-time image processing, and a crosswalk image of a road was used as an obstacle. In this paper, we proposed a low-cost model that can actually implement autonomous driving based on the CNN model. The most well-known deep neural network technique for autonomous driving is investigated and an end-to-end model is applied. In particular, it was shown that training and self-driving on a simulated road is possible through a practical approach to realizing lane detection and keeping

Keywords: *Deep Neural Network, Low cost, Autonomous car, Lane detection, Lane Keeping*

1. Introduction

Recently, it has been applied to image recognition, natural language processing, etc. using the structure of the neural network of deep learning, and has shown high performance. Especially in the field of image recognition, much research is being carried out because CNN (Convolutional Neural Network) shows superior performance compared to conventional methods. Among the fields of much research, there are autonomous vehicles. Since it is necessary to collect and judge information inside and outside the autonomous vehicle, it is necessary to install various sensors and cameras to recognize the surrounding environment and quickly judge the response to acceleration, deceleration, and stop. Further, in the data processing in autonomous driving, since the input video or the input data has a continuous value, it is possible to solve this promptly as a core element.

In this paper, we apply a deep learning algorithm to build an autonomous driving system for automobiles that has been activated as recently, and try to study the autonomous driving system by driving it on a model road. The deep learning algorithm, which is an important part of this training, can efficiently handle the construction of autonomous driving systems by using the CNN algorithm, which has an excellent effect on image recognition, among the existing deep learning related algorithms.

Manuscript Received: January. 7, 2021 / Revised: January. 13, 2021 / Accepted: January. 15, 2021
Corresponding Author: mhsong@semyung.ac.kr
Tel: +82-43-649-1645, Fax: +82-43-649-1747
Assistant Professor, School of Information and Communication Science, Semyung University, Jecheon, Korea

The rest of the paper is organized as follows: Section II provides a background for the application of deep neural networks in autonomous driving. Section III describes the architecture design of the proposed prototype system and presents autonomous vehicle subsystems and their functions. Section IV represents the implementation, and Section V draws conclusions with some points for future work.

2. Theoretical Background

2.1 Deep Learning Method for Self-Driving Cars

The convolutional artificial neural network (CNN) is a type of deep neural network that uses convolutional coupling as its main operation, and was first developed by LeCun et al. in 1988 [1]. The structure of CNN is a deep feed forward neural network and has the following features [2]. 1) Hidden neurons between the two layers remain "local" with the two neurons not connected to each other for each layer. A neuron in the upper layer receives input from a neuron in the lower layer within a rectangular region close to the neuron. This area is known as the receptive field for neurons. 2) Share the weights for the "local" connection with other neurons in the same layer. Due to the nature of this method of utilizing the immutability of visual data, the number of parameters required for CNN models is reduced. In computer vision, the characteristic of CNN is treated as 'implicit prior knowledge', and through this characteristic, CNN has become a powerful model for solving computer vision problems. When AlexNet [3] won the 2012 ImageNet [4] image classification competition, CNN was introduced quickly, and the latest algorithms based on CNN were quickly introduced. As a result, CNN has become the key to the recognition of autonomous driving.

Recurrent Neural Network (RNN) is a deep learning algorithm that is effective in processing data mainly dealing with temporal sequences[5]. It shows good performance in natural language processing and video stream processing. Most of the implementation of autonomous driving is key to informing the perception of continuous changes in the surrounding environment. Therefore, more accurate expressions of the surrounding environment are acquired by storing and tracking information related to driving that has been acquired in the past.

Reinforcement learning is formalized through a model called the Markov Decision Process. For example, a robot recognizes the current environmental state through various sensors and receives a reward from the environment by performing one action, and the environment transitions to the next state. Reinforcement learning learns one optimal policy through this trial and error process. The policy is a function that returns the action to be performed in each state, and the optimal policy is a policy that maximizes the cumulative reward among these policies. Figure 1 shows the conceptual reinforcement learning structure. Reinforcement learning technology is also applied in various ways in self-driving cars.

Deep reinforcement learning is a technology that combines reinforcement learning and deep learning, and aims to find a policy of action that obtains the greatest benefit through the agent's continuous actions in a given environment. It consists of 5 elements such as Action, Reward, State, Agent, and Environment. Agent is a subject that performs reinforcement learning and takes action in the environment. When Agent takes Action, Reward is received, and State means its own state recognized by Agent. In this way, the agent interacts with the environment and reinforcement learning proceeds [6-9].

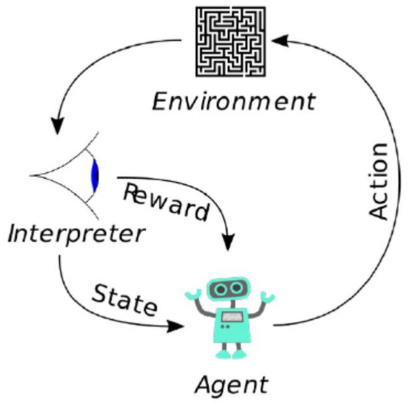

Figure 1. Structure of Reinforcement Learning [10]

2.2 Recognition of autonomous driving through deep learning

Figure 2 shows two pipelines to solve the problem of self-driving cars based on deep learning [11]. (a) is a standard robotics control architecture that is a sequential recognition-planning-execution pipeline, or as shown in (b), it is implemented as an end-to-end system. In the case of a sequential pipeline, a deep learning approach can be used or a non-learning approach can be designed in the same way as the existing one. In the case of an end-to-end learning system, the deep learning method is mainly used.

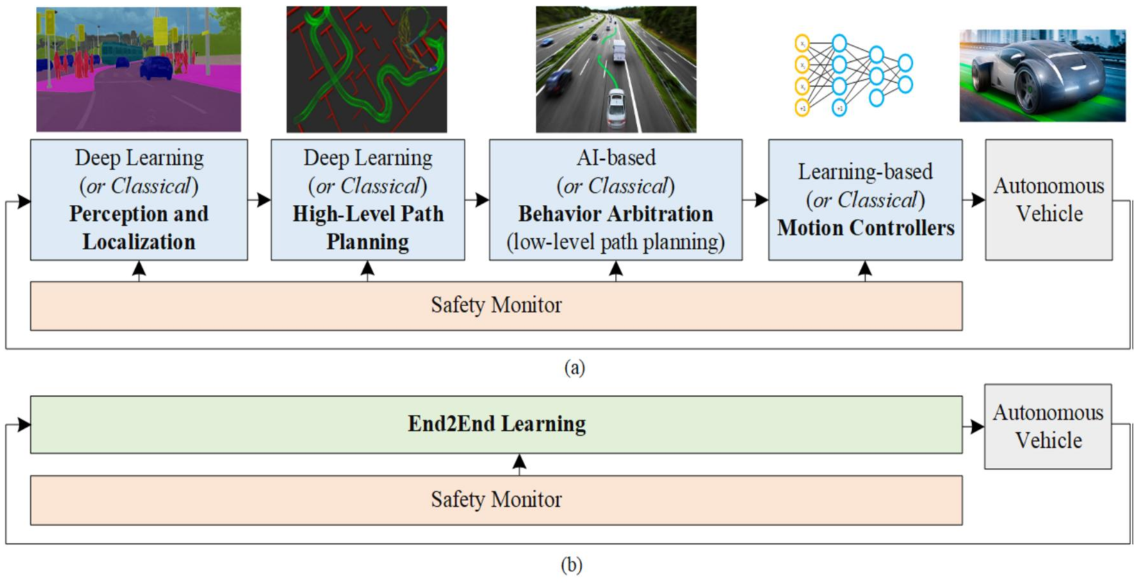
Deep Learning (or Classical) Perception and Localization
Deep Learning (or Classical) High-Level Path Planning
AI-based (or Classical) Behavior Arbitration (low-level path planning)
Learning-based (or Classical) Motion Controllers
Autonomous Vehicle
Safety Monitor
(a)
End2End Learning
Autonomous Vehicle
Safety Monitor
(b)

Figure 2. Deep Learning based self-driving car [11]

Creating control outputs directly from sensors based on deep neural networks simplifies the pipeline. Such studies have recently been reported [12]. A study on the application of neural networks for end-to-end control of autonomous vehicles was first proven in the late 1980s and was a layer 3 neural network [13]. And in the early 2000s, the DARPA autonomous vehicle (DAVE) project using a six-layer convolutional neural network (CNN) [14]; Most recently, it's in NVIDIA's DAVE-2 project [15] using a 9-layer CNN. In this case studies, the neural network model directly generates steering control commands using raw image pixels as input. All intermediate steps used in the traditional sequential pipeline approach are skipped.

According to NVIDIA research, when a driver drives along the road, a camera installed in the vehicle takes pictures of the road at 30 frames per second. And the handle value at the time each picture was taken (how many degrees the handle was turned to the right or left) was also collected [15].

In supervised learning, a cause value and a result value are required in pairs. In the case of lane keeping artificial intelligence, the image of the road becomes the cause value (X), and the handle value at that time becomes the result value (Y). In order to implement the lane keeping function, it is impossible with a simple linear equation and an artificial neural network must be used. In other words, instead of straight lines, artificial neural networks are constructed as predictive models. As before, when the road image is input into the artificial neural network, the predicted handle value is output by the artificial neural network as a result. The difference between this output, the predicted value of the steering wheel and the value of the actually driven steering wheel, is the error, and the artificial neural network is trained to produce the least sum of these errors [16].

3. Architecture Design for Low-cost Autonomous car

3.1 Data Collection & Preprocessing

In order to collect training data for autonomous vehicles, the road must first be prepared. The road was made and used so that it was too narrow or wide. The width of the road should be 1.5 to 2 times the width of the car. If the road is too narrow, it will be difficult to drive and collect good training data. Conversely, if the road is too wide, the camera may not capture the lines on either side of the left and right roads at once, resulting in poor learning. As much as possible, roads should be created using gentle curves and straight lines. Unlike real car driving environments, low-cost prototype research takes advantage of the difference in rotational speed between the left and right motors, making it difficult to deal with sharply curved roads. be able to. Also, even on a straight road, it is difficult to drive correctly along the road with a terrible bend angle. When a person drives directly, he / she will be able to drive like a person even if artificial intelligence drives in the road conditions where it is easy to drive. The road model produced for the autonomous driving prototype is shown in Figure 3.

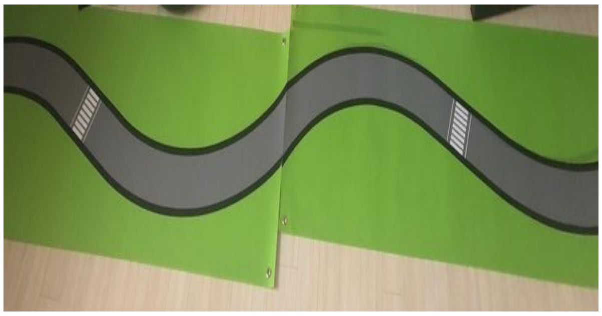

Figure 3. Road model built for self-driving prototypes

In order to train a classification model efficiently, the data proportions must be equalized so that the classification category proportions are similar. There are cases where there is much more straight forward data than left/right turn data, for example, if the ratio is 14%, 76%, and 20% for left turn, straight forward and right turn respectively. In this case, running decalcom.py will change the left turn, straight forward, right turn ratio to 18%, 76%, 18% data rate. The left/right ratio can be adjusted, but the straight ratio is still too high, which makes learning to turn left/right difficult. There are two ways to scale the data when the data is uneven. Down Sampling method is a method to make the number of left/right turn and data equal to the number of left or right turns by sampling the number of left or right turns from the entire data of straight forward when the ratio of left turn, straight turn, and right turn is 18%, 76%, 18%. In other words, it is a way to fit the ratio by reducing the amount of categories with a small number of categories from a category with a lot of data. The downside is that the total training data will be reduced, and if the data missing during the sampling process is important data, there is a possibility that it will be an artificial intelligence with poor performance. The Up Sampling technique is a method to make the number and ratio of straight data by repeatedly copying the left/right turn data when the ratio of left turn, straight turn, and right turn is 18%, 76%, and 18%. In other words, it is a method to create a data volume similar to a category with a large amount of data by repeatedly copying and using the data of a category with a small data rate. While it has the advantage that the total amount of data can be increased and the data rate can be easily adjusted, it has the disadvantage that overfitting learning can be performed on repeatedly used data. In this study, the data rate is adjusted through up sampling.

3.2 Training of Self-driving car prototype

Figure 4 shows the structure of System hardware. Image classification through Tensorflow is dealt with in detail in later Section. When the first user presses the learning start button and then starts driving, the photo and the arrow keys corresponding to the photo continue to be saved in the Orange Pie lite internal folder. After collecting the training data in this way, it is a system that controls the speed of the DC motor and runs by using the classification of images after deep running.

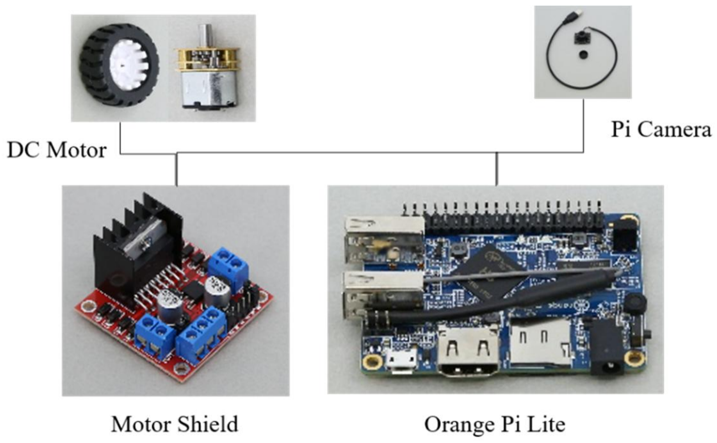

Figure 4. Overall System Structure

The model used in this paper used a classification model. The method of dividing the result into several categories and predicting any one of them is called classification. The flow of the whole system is as follows. First, after connecting to Orange-pi remotely through VNC Viewer, the user controls with the keyboard to collect the learning data and saves the learning data as an image. When training is completed after collecting the training data, a classification model is created. The process of classifying through this model is shown in Figure 5. When the image of the road is transmitted to the input layer of the artificial neural network, a specific value is predicted and expressed in the output layer of the artificial neural network composed of four neurons (stop, go straight, turn right, turn left). Also, if you recognize a crosswalk on the road, it stops. According to the image of the road at every moment, predicted values for four categories are displayed as probability. Among them, the value with relatively high probability becomes the output value suggested by artificial intelligence.

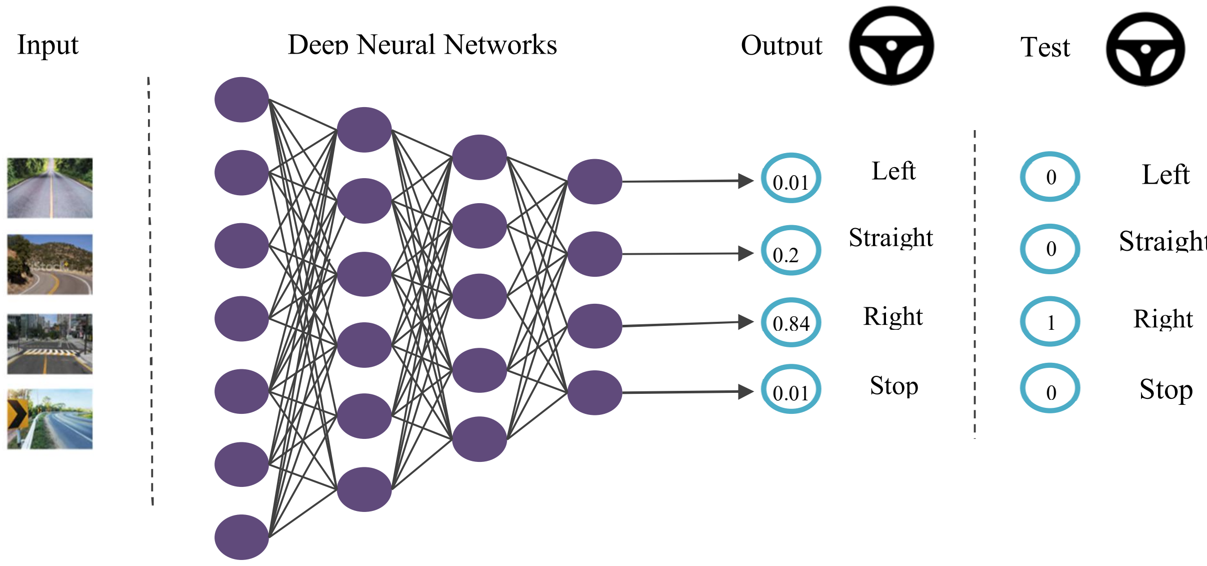

Figure 5. Classification of driving directions through images

3.3 Network Architecture

The overall structure of the learning algorithm used in this paper is composed of a total of 4 convolutional layers and 3 fully connected layers as shown in Figure 6. Also, the internal structure of the convolutional layer and the fully connected layer is shown in Figure 7. The convolution layer proceeds in the order of convolution and activation functions as shown in Figure 7, and a 3x3 kernel and a 5x5 kernel are used. The convolutional layer more robustly detects the features of an image as the number of steps increases, but requires a long learning time. However, if it goes beyond a certain level, the image is excessively implied, so the proper features may not be captured and performance may deteriorate. Conversely, if the level of the convolutional layer is small, there is a limit to derivation of features, resulting in poor accuracy.

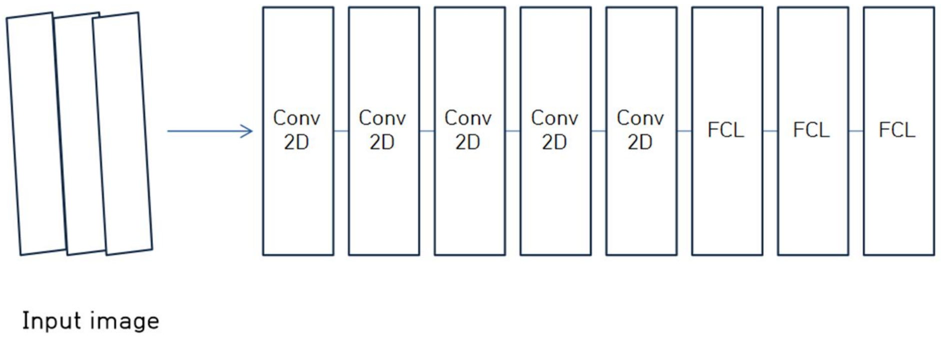

Figure 6. CNN Network structure

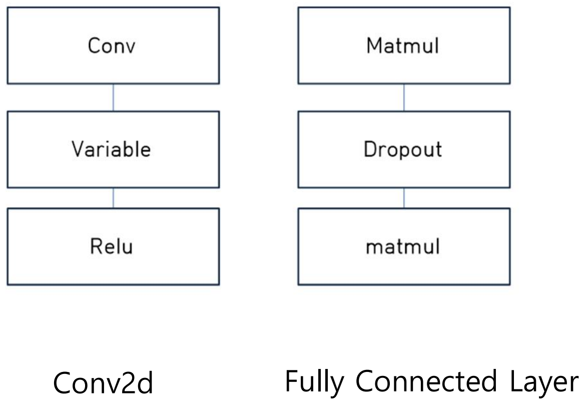

Figure 7. Convolutional layer, fully connected layer structure

The kernel is used to derive the features of the image, and it plays a role of checking whether the feature to be detected exists in the region based on the learned feature. In the case of the fully connected layer, three layers were used in this paper. In the case of the first fully connected layer, after flattening with 1D data, dropout is performed to prevent over-fitting. In the case of the second layer, dropout is performed again after using the Relu activation function. Relu and Dropout are also used in the last layer. After that, in the last layer, the images are classified by connecting them to the same four layers as the image types used in the experiment.

4. Implementation

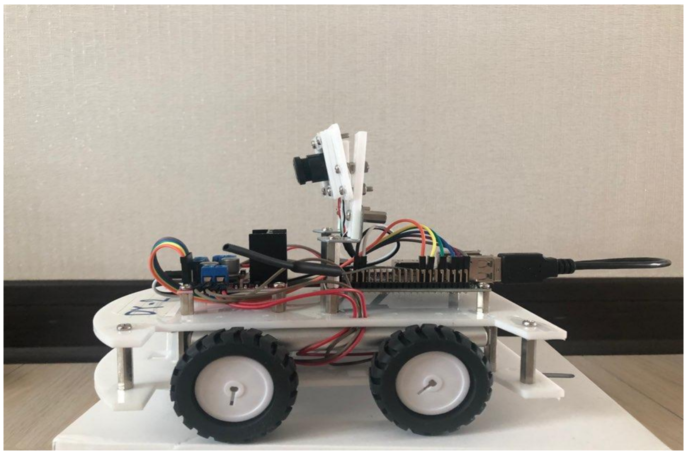

Figure 8. Autonomous car prototype

The whole system was implemented using Python, and Tensorflow and Otangepi were used. Orangepi is equipped with H3 Quad-core Cortex-A7 H.265/HEVC 4K CPU and Mali 400MP2 GPU @600MHz GPU, and the GPU supports OpenGL ES 2.0. Figure 8 shows a prototype for Autonomous car. The hareware components for implementing autonomous car prototype are shown in Table 1.

Table 1. Hardware components

Item	Component	Specification
OrangePi Lite	CPU	H3 Quad-core Cortext A7 H.265/HEVC 4K
	GPU	Mali400MP2 GPU @600MHz Supports OpenGL ES 2.0
	Memory	512MB DDR3
	Onboard Storage	32GB
Motor shield		
Motor	DC Motor	
External battery		

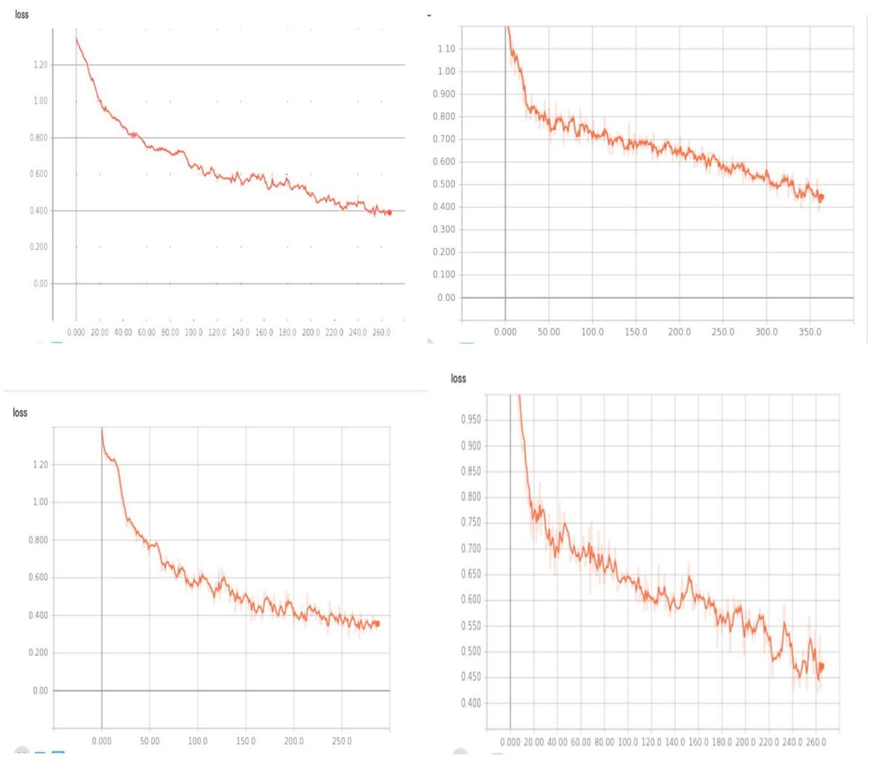

Figure 9. Self-driving prediction accuracy (Loss Value) for each experiment

Figure 9 is a table showing prediction rate, respectively, while learning. The final learning model performed tertiary learning with the highest prediction rate of 87%. As for the change in the loss value of the tertiary learning model, as the learning progresses in Figure 9, it can be seen that the loss value decreases. When the learning is completed like this, the completed model is loaded to perform autonomous driving. The image entered through the camera of the toy car prototype in Figure 8 is classified by the model and classified into straight forward, left turn, right turn, and stop. The overall accuracy is 87%, but the accuracy of the left turn is 100% and the accuracy of the right turn is 99%, so at least you will not go off the road. The higher the left/right turn accuracy, the more you will not get off the road. If the accuracy of going straight is too low, it will not go off the road because it is driven only by turning left/right, but it does not look good because it is a zigzag type operation. If the accuracy of the straight line is around or above 50%, the straight line is mixed appropriately and the operation becomes much smoother.

5. Conclusion

In this paper, a deep learning algorithm CNN was applied to construct the currently active vehicle autonomous driving system, and it was driven on a driving road to study low-cost autonomous driving through image classification among several approaches to the autonomous driving system. The model used in this paper used a deep learning algorithm CNN to determine the meaning of the image input to the autonomous car, receive real-time images, classify, and determine the direction in which to perform autonomous driving. For future research, various performances can be evaluated by experimenting in an actual autonomous driving environment through the construction of a low-cost test bed. In addition, it can be extended to research on object recognition and destination search in autonomous driving by specifying the functions of other subsystems.

Acknowledgement

This work was supported by the Semyung University Research Grant of 2020.

References

[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
[2] Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256). JMLR Workshop and Conference Proceedings.
[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.
[4] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.
[5] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.
[6] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. Vol. 1. No. 1. Cambridge: MIT Press, 1998
[7] Baird, Leemon. "Residual algorithms: Reinforcement learning with function approximation." Proceedings of the twelfth international conference on machine learning. 1995.
[8] Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018, April). Deep reinforcement learning that matters. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).
[9] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, 34(6), 26-38.
[10] Reinforcement learning - Wikipedia https://en.wikipedia.org/wiki/Reinforcement_learning
[11] Grigorescu, S., Trasnea, B., Cocias, T., & Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. Journal of Field Robotics, 37(3), 362-386.
[12] Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. The Journal of Machine Learning Research, 17(1), 1334-1373.
[13] Pomerleau, D. A. (1989). Alvinn: An autonomous land vehicle in a neural network. CARNEGIE-MELLON UNIV PITTSBURGH PA ARTIFICIAL INTELLIGENCE AND PSYCHOLOGY PROJECT.
[14] Lecun, Y., Cosatto, E., Ben, J., Muller, U., & Flepp, B. (2004). Dave: Autonomous off-road vehicle control using end-to-end learning. DARPA-IPTO Final Report.
[15] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zieba, K. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.
[16] NVIDIA. GPU-Based Deep Learning Inference : A Performance and Power Analysis. Technical Report November, 2015.