Autonomous Car Using CNN Deep Learning Algorithm

Ilvico Sonata1,a), Yaya Heryadi1,b), Lukas1,c), Antoni Wibowo1,d)

1)Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University Jakarta, Indonesia 11480

a)ilvico@binus.ac.id, b)yayaheryadi@binus.edu, c)lukas@binus.ac.id, d)anwibowo@binus.edu

**Abstract**. Autonomous cars have become an interesting discussion in recent years. Several major automotive industries such as Tesla, GM and Nissan are trying to become pioneers in autonomous car technology. Giant technology companies such as Google Waymo, Baidu and Aptiv also developing autonomous car technology. Several technological approaches are carried out in implementing autonomous cars for recognizing surrounding situations, such as radar, lidar, sonar, GPS, and odometry. An automatic control system is used to control navigation based on the data obtained from these sensors. This paper will discuss the use of CNN deep learning algorithm for recognizing the surrounding environment in creating the automatic navigation required by autonomous cars. The system designed will create and learn the data set that will be taken in advance and the learning outcomes will be implemented in an open simulation system. This simulation shows high accuracy in learning to navigate the autonomous car by observing the surrounding environment.

1. Introduction

Applications of artificial intelligence and machine learning are increasingly widespread and penetrate all fields including in the field of autonomous cars. The development of autonomous cars which are currently being developed by many technology companies such as Waymo and Uber is starting to lead to the use of artificial intelligence and machine learning to replace the conventional systems which require very expensive equipment such as LRF, lidar, GPS for detecting the surrounding [4].

Many researches of autonomous cars are currently being progress to ensure the safety of autonomous cars before they are launched to the public. A car with a driver provides a sense of safety and comfort for passengers where the driver can control the car well and meet safety standards. The challenge of autonomous cars today is how to produce a model of driving behaviour that is the same as humans so that it still provides a sense of safety and comfort for passengers [1].

Artificial intelligence and machine learning approaches have proven successful in areas of image classification such as facial and image recognition systems [2][5]. This classification system approach will be used for the recognition of the surrounding environment in determining the direction and movement of autonomous cars. Machine learning with images as inputs is achieved by Convolutional Neural Networks (CNN), which has become dominant in various computer vision tasks including autonomous car. CNNs are one of the best deep learning algorithms for recognizing image content and have demonstrated good performance in image segmentation, classification, detection, and retrieval related tasks (Ciresan et al. 2012; Liu et al. 2019).

1. Proposed Method

CNN is a technique that is inspired by the biological visual perception. CNN always contains three basic operations, namely convolution, pooling and fully connected [2].

Convolution layer is a fundamental component of the CNN architecture that performs feature extraction using multiple filters. The convolution layer consists of neurons arranged to form a filter with length and height (pixels). Convolution process is done by transform the parts of certain size of image become small array of number called kernel, the system gets new representative information from the multiple parts of kernel and process it become feature map [5].



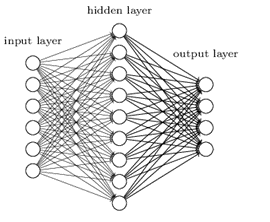
**Figure 1.** The original image from the image data becomes smaller images with the same convolution.

Each thumbnail of the convolution results is used as input to produce a feature map as shown in Figure 1. Feature map gives CNN the ability to recognize an object, wherever the object's position appears in an image. All parts of each thumbnail are processed using the same filter. Each part of the image will have the same multiplier factor called weights sharing. Every different of each image will be marked as object of interest. The whole thumbnail collection is arranged into a kernel.

Figure 2 shows the CNN architecture. The pooling layer performs a sub-sampling process to reduce input spatially (reduce the number of parameters) with a down-sampling operation. This is used to reduce the dimensionality of feature maps from the convolution operation. So that, even when reducing the number of parameters, the most important information from that section is still retrieved. Then using that small array, we can input it into other neural networks. The small array will be converted into a 1-dimensional array by the flatten process. The final neural network will decide whether the images match or not by the fully connected layer.

Backpropagation is a supervised learning algorithm and is usually used by multi-layered perceptrons to change the weights associated with neurons in the hidden layer. This algorithm uses an output error to change the weight values in the backward direction. The error is obtained from forward propagation process output (predicted value) compared with actual value. The weights will be updated by the calculation base on the error value. The process will be repeated until the error rate decrease.

The most important thing in CNN modelling is choosing how many types of pooling layers. This can benefit model performance (Lee, Gallagher, & Tu, 2015).



Flatten

Input Image

Actual Value

C

o

n

v

o

l

u

t

i

o

n

P

o

l

l

i

n

g

Compare

Error/

Adjustment

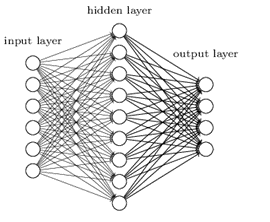
Fully Connected

Feature extraction

Feed Forward



|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |



Back Propagation

Error/

Adjustment

**Figure 2.** CNN Architecture.

Figure 3 shows a simplified block diagram of the autonomous car system for training data collection. Images from the three cameras and steering angle are fed into a CNN. CNN then computes a prediction model of steering command. The actual model of steering command is compared to the prediction command and the weights of the CNN are adjusted to bring the CNN output closer to the desired output. The back propagation is used to accomplish the weight adjustment.

After training process finish, the system generates steering model and this model will be used for driving command base on single centre camera as shown in figure 4.

In previous models, many autonomous cars were developed using some sensors such as LRF, GPS and LIDAR as described in Lattarulo et al (2017) and Du et al (2017). This model only uses one camera as a sensor for autonomous mode to reduce hardware usage and thus lower costs.

Centre Camera

Correction value

Prediction

value

Actual value

CNN

Left Camera

Right Camera

Steering angle

Error/

Adjustment

Back propagation Weight

**Figure 3.** Autonomous car training model.

Centre Camera

Trained CNN

(Driving model)

Steering command

**Figure 4.** Autonomous car driving model.

1. Learning Environment and Simulation Experimental Result

The simulation environment is using udacity self-driving car simulator and the network is based on NVIDIA model. Figure 5 show the learning environment created in the simulation study and figure 6 show sample image taken by left camera, centre camera and right camera in training mode.



**Figure 5.** Learning environmental.



Figure 6. sample image taken by left camera, centre camera and right camera.

Collected data is labelled with road type and driver activity such as in the lane or turning. The simulation sampling rate is 10 FPS and image resolution 320x160 pixels. Before the learning process, the image taken will be normalize by removing the sky and other unnecessary parts of picture for simplifying the learning process.

The training process using 5x5 kernel for the first three convolution layer and 3x3 kernel for the last three convolution layer and maximum epochs 20 with 2.000 sample for each epoch.

After the training finish after two or three laps, the simulator then switches to the autonomous mode and then base the model created by the CNN, the car will run autonomously and can predict for every road condition.

The simulator accesses recorded video footage along with synchronized steering commands performed at the time of learning. The simulator sends the first frame of the video, adjusted for each basic condition, to the trained CNN input. The CNN steering commands and the recorded training commands are sent to the driving model of the car for updating the position and orientation of the simulated car.



**Figure 7.** Autonomous mode in simulator.

1. Conclusion

The proposed method of autonomous car using CNN deep learning can run smoothly without error in simulation environmental using udacity self-driving car simulator.

CNN can learn road condition from three cameras (left, centre and right) and also make a model for driving in autonomous mode controlled by one centre camera.

To get the better performance, next experiment will include the obstacle in the simulation environment and rain simulation with additional noise and distortion at data image set.

References

1. Zhu, M., Wang, X., & Wang, Y. (2018). Human-like autonomous car-following model with deep reinforcement learning. Transportation research part C: emerging technologies, 97, 348-368.
2. Giusti, A., Ciresan, D., Masci, J., Gambardella, L., & Schmidhuber, J. (2013). Fas Image Scanning With Deep Max-PoolingConvolutional Neural Networks. IEEE International Conference on Image Processing, 4034-4038.
3. Yang, Y., Wu, Z., Xu, Q., & Yan, F. (2018). Deep Learning Technique-Based Steering of Autonomous Car. International Journal of Computational Intelligence and Applications, 17(02), 1850006.
4. Okuyama, T., Gonsalves, T., & Upadhay, J. (2018). Autonomous Driving System based on Deep Q Learning. International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 201-205.
5. Yamashita, R., Nishio, M., Do, R., & Togashi, K. (2018). Convolutional Neural Networks: An Overview and Application in Radiology. Insights Imaging, Springer, 611-629.
6. Masci J., Meier U., Cireşan D., Schmidhuber J. (2011) Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction. In: Honkela T., Duch W., Girolami M., Kaski S. (eds) Artificial Neural Networks and Machine Learning – ICANN 2011. ICANN 2011. Lecture Notes in Computer Science, vol 6791. Springer, Berlin.
7. Sallab, A., Abdou, M., Perot, E., Yogamani, S. (2017). Deep Reinforcement Learning framework for Autonomous Driving. IS&T International Symposium on Electronic Imaging, 70-76.
8. Du, X., Ang, M., & Rus, D. (2017). Car detection for autonomous vehicle: LIDAR and vision fusion approach through deep learning framework.IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 749-754.
9. Bechtel, M., Mcellhiney, E., Kim, M., & Yun, H. (2018). DeepPicar: A Low-Cost Deep Neural Network-Based Autonomous Car. IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Hakodate, 11-21.
10. Al-Qizwini, M., Barjasteh, I., Al-Qassab, H., & Radha, H. (2017). Deep Learning Algorithm for Autonomous Driving using GoogLeNet. IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA.
11. Lattarulo, R., Perez, J., & Dendaluce, M. (2017). A Complete Framework for Developing and Testing Automated Driving Controllers. ScienceDirect IFAC PapersOnline 50-1, 258-263.