2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

End-to-End Steering Controller with CNN-based Closed-loop Feedback for Autonomous Vehicles

Junekyo Jhung[1], Il Bae[1], Jaeyoung Moon[1], Taewoo Kim[1], Jincheol Kim[2], and Shiho Kim[1]

Abstract— **Many significant research achievements in the past few decades have demonstrated that convolutional neural networks (CNNs) could be capable of steering wheel control which is the basic and essential maneuver of autonomous vehicles. We propose an end-to-end steering controller with CNN-based closed-loop feedback for autonomous vehicles that improves driving performance compared to traditional CNN-based approaches. This paper demonstrates that the proposed neural network, DAVE-2SKY, is able to learn to inference steering wheel angles for the lateral control of self-driving vehicles through initial supervised pre-training and subsequent reinforced closed-loop post-training with images from a camera mounted on the vehicle. We used the PreScan simulator and Caffe deep learning framework for training under diversified circumstances in a software in the loop (SIL) simulation environment. We used DRIVE™ PX2 computer to implement a self-driving car for an experimental validation of the proposed end-to-end controller. The performance of the proposed system has been investigated under simulations and on-road tests as well. This work shows that the CNN-based end-to-end controller performs robust steering control even under partially observable road conditions, which indicates the possibility of fully self-driving vehicles controlled by CNN-based end-to-end steering controllers.**

I. INTRODUCTION

Over the past few decades, there have been significant achievements in autonomous driving technology as a result of efforts from industrial institutes including automotive original equipment manufacturers (OEMs), related companies, research institutes, and universities. Moreover, significant advancements in machine learning have allowed innovative approaches to autonomous vehicles using the deep neural networks. In particular, convolutional neural network (CNNs) [1] have demonstrated the potential for implementing end-to-end learning for steering wheel control, which is the basic and essential maneuver of self-driving autonomous vehicles. In order to build a fully self-optimized learning system to maximize performance for trajectory tracking and driving safety, instead of using model-based intermediate control criteria, we developed a reinforced closed-loop feedback training and inferencing architecture to learn steering wheel angles from a front-facing on-vehicle camera.

In 1989, Carnegie Mellon University developed an autonomous vehicle called Autonomous Land Vehicle In a Neural Network (ALVINN) [2], which first demonstrated the possibility of camera-based end-to-end steering control for a self-driving car. Since ALVINN, noteworthy works have researched neural networks (NNs) and Defense Advanced Research Projects Agency (DARPA) challenges that have boosted NN growth. In 2004, the DARPA Autonomous Vehicle (DAVE) [3] project demonstrated training a radio controlled (RC) car on hours of human driving data captured by left and right cameras, and driving it on off-road environments. Although DAVE could not demonstrate a complete solution for complex driving environments, it inspired an advanced version called DAVE-2 [4]. NVIDIA developed DAVE-2 as a robust system for self-driving cars on public roads that steers an autonomous vehicle based on the data trained with images obtained by a front-facing camera.

In autonomous vehicles, lateral control, which involves the steering of the vehicle, is a fundamental function. Lane keeping is a representative maneuver of lateral control, keeping the vehicle in the center of a lane during self-driving. Although steering control is a basic function for self-driving vehicles, the CNN-based end-to-end controller still has difficulty implementing steering in autonomous vehicles [5].

Our motivation is to provide a frontier method for the steering controller of autonomous vehicles that overcomes the current limitations of traditional CNN-based end-to-end control approaches. We propose DAVE-2SKY (DAVE-2 modified by SK Telecom & Yonsei University), an end-to-end steering controller with a CNN-based closed-loop feedback architecture. The network is not explicitly taught, and learns the entire processing pipeline required for steering control to maintain its lane while following a front vehicle. The learning procedure is composed of two steps of training: supervised pre-training and reinforced closed-loop feedback post-training. Consequently, DAVE-2SKY produces proper steering wheel angles from camera image data for stable and complete lateral control of autonomous vehicles. Compared to traditional back-propagation training approaches as DAVE-2 [4], the proposed system is able to learn the driving task in shorter training time with robust, improved performance.

The remainder of this paper is structured as follows: In Section II, we provide an overview and details of the proposed system, DAVE-2SKY. The environment and results of the implementation of SIL using a simulator for our approach are

[1]Junekyo Jhung, Il Bae, Jaeyoung Moon, Taewoo Kim, and Shiho Kim are with the Seamless Transportation Lab(STL), School of Integrated Technology, and Yonsei Institute of Convergence Technology, Yonsei University, Incheon 21983, South Korea (e-mail: ejhung@yonsei.ac.kr; kakao@yonsei.ac.kr; ekuno90@yonsei.ac.kr; boratw@yonsei.ac.kr; shiho@yonsei.ac.kr).

[2]Jincheol Kim is with Vehicle Tech Lab, Network Technology R&D Center, Corporate R&D Center, SK Telecom, Seoul 04539, South Korea (e-mail: jincheol.b.kim@sk.com).

presented in Sections III and IV, before the experimental results in Section V. In Section VI, we discuss the results of the proposed system in the simulation, and finally, we conclude this paper in Section VII.

I. Proposed System

A. Network Architecture

Our network has an architecture originated from DAVE-2 [4], which trains deep NNs with images from an input camera mounted on an autonomous vehicle to compute steering wheel angle for lateral control. The proposed DAVE-2SKY shown in Figure 2 is implemented using the Caffe deep learning framework [6]. A Euclidean loss model is used to compute the sum of squared errors between ground truth and predicted steering wheel angles as in [7].

The NN consists of 10 layers, including five convolutional layers, three normalization layers, and two fully connected layers as Figure 1. As Table I, the first three convolutional layers have a 5 × 5 kernel and a 2 × 2 stride each, and the next two convolutional layers have a 2 × 2 kernel and a 1 × 1 stride each. Rectified linear unit (ReLU) activation is used in each convolutional layer. The input image is split into RGB planes and then passed to the network. The convolutional layers are designed as feature extractors and the fully connected layer is a controller for steering the vehicle. Normalization layers are added before convolutional layers 2, 3, and 4 to avoid the gradient vanishing/exploding problem and to increase training speed by stabilizing the training process [8].

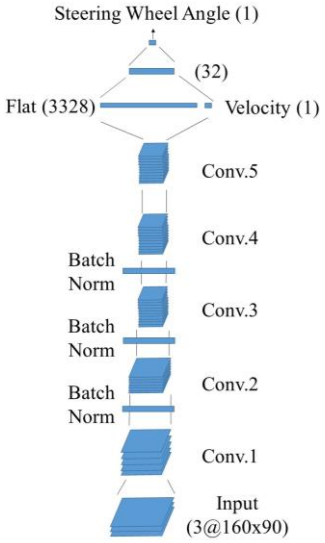

Figure 1. Architecture of proposed DAVE-2SKY CNN.

TABLE I. Convolutional Layers of DAVE-2SKY Network

	Kernel	Stride	Activation
Convolutional layer 1	5 × 5	2 × 2	ReLU
Convolutional layer 2	5 × 5	2 × 2	
Convolutional layer 3	5 × 5	2 × 2	
Convolutional layer 4	3 × 3	1 × 1	
Convolutional layer 5	3 × 3	1 × 1	

B. Reinforced Closed-loop Feedback Post-training System

The proposed system, DAVE-2SKY, based on DAVE-2 [4] can infer steering control command during the reinforced feedback loop. DAVE-2SKY can be trained by two steps consisting of supervised pre-training and reinforced closed-loop feedback post-training. The reason why two training steps are combined is explained below.

Supervised pre-training is the first step with the architecture shown in Figure 2(a). We implemented a SIL configuration similar to the traditional training system with the PreScan simulator running on a PC, and a CNN training architecture based on Caffe deep learning framework running on a DevBox computer. An image from a camera is fed into DAVE-2SKY to produce a steering wheel angle which is going to be compared to recorded feasible steering wheel angles to adjust weights of filters of the convolutional network using the back-propagation mechanism. We trained DAVE-2SKY in the supervised pre-training step for a limited number of iterations, less than 80,000 cycles in this study. The two separate computers, the PC for simulator and the DevBox, communicate via a user datagram protocol (UDP) network.

After the pre-training step, DAVE-2SKY is trained in the reinforced post-training step. For the purpose of the feedback loop, we formed a distinctive SIL closed-loop architecture composed of the PreScan simulator and MATLAB Simulink running on a PC and the Caffe deep learning framework running on a DevBox computer as shown in Figure 2(b).

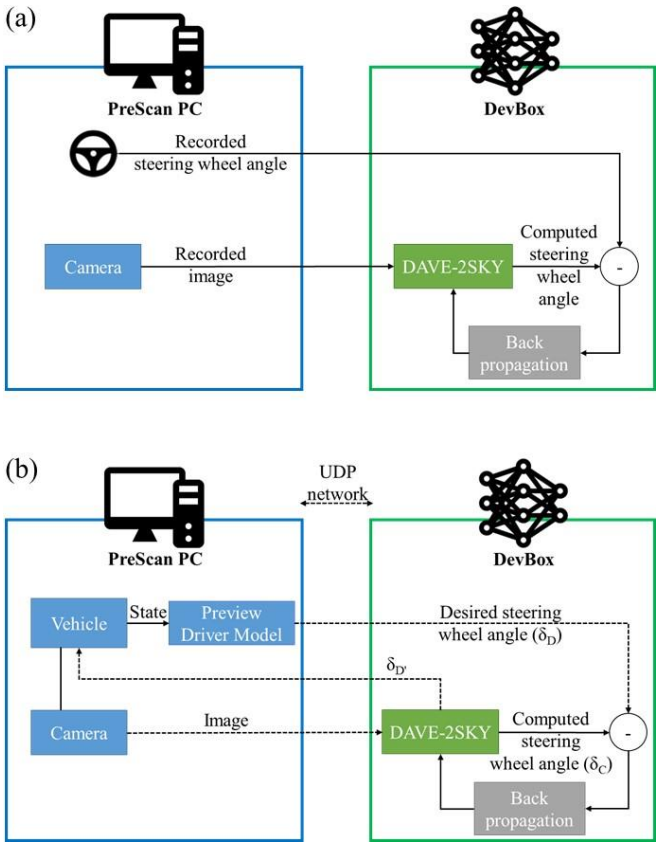

Figure 2. Block diagram of training the NN, DAVE-2SKY: (a) supervised pre-training; (b) reinforced closed-loop feedback post-training.

We used a virtual vehicle with a front-facing camera in the PreScan simulation. By using a built-in plug-in of MATLAB Simulink, we acquired the internal states of the vehicle and image data from the camera during the simulation. These state data represent the current state of the vehicle such as speed, position, heading, and yaw rate. The vehicle states are fed into the preview driver model (PDM), a built-in controller model configured by the simulator, to produce desired steering wheel angles (δ_D) using the algorithm provided by the author's group [9,10]. In this study, the steering outputs predicted by PDM were used as the ground truth. It is possible to replace PDM with a human driver using an interface device on the steering wheel such as the Logitech G27. This flexibility allowed us to transfer the trained DAVE-2SKY to implement a real autonomous vehicle for on-road experiments.

The reason why the proposed two training steps can enhance the efficiency of learning and performance of vehicle maneuvering compared to conventional supervised training can be explained as follows. Image data for the NN are acquired from the front-facing camera and sent to the DevBox. DAVE-2SKY is trained in a reinforced manner in the closed-loop feedback architecture in addition to the normal back-propagation mechanism of the conventional deep NN training routine. Similar to typical CNNs, DAVE-2SKY maps the acquired pixels to steering wheel angle (δ_C). The mechanism of back propagation [1] adjusts the weights of the CNN filters of the network to minimize the error between δ_D and δ_C. Then corrected steering wheel angle ($\delta_{D'}$), inferenced by the CNN network with the adjusted weights, is fed to the vehicle as a control input. Thus, there is a reinforced learning routine embedded in the proposed post-training configuration.

As mentioned previously, the trained NN DAVE-2SKY is a controller for a self-driving vehicle. The conventional simple supervised pre-training step may not be enough to learn the human-like end-to-end vision-based intelligence necessary for maneuvering vehicles. The post-training step involving closed-loop feedback allows reinforced learning in the SIL environment. If we replace PDM with a human driver in the driving simulator, DAVE-2SKY can learn heuristic driving experiences.

II. Environment of Simulation

The PreScan simulator allows to build a virtual validation environment with realistic configurations to acquire rich information using a virtual vehicle model. The training dataset was collected by using plug-ins of Simulink and PreScan.

A virtual vehicle (an Audi A6) with default physics model and dynamic configurations were used in the simulation. A single virtual front-facing camera was mounted to the virtual vehicle as shown in Figure 3. The controllable steering wheel angle range of the vehicle was from – 500° to + 500°. The sign of a steering wheel angle represents the direction of the steering wheel as negative for clockwise and positive for counterclockwise, and the steering ratio [11] of the vehicle was set to 20:1. The driving track was 1,492 m in length and consisted of two lanes (4 m wide) as shown in Figure 4. Lanes were separated by a yellow solid center line and sidewalks (2 m in height) were present on each side of the road. During a simulation of automatic cruise control with a lane keeping maneuver, we set the velocity of the vehicle at a constant speed of 5 m/s (18 km/h).

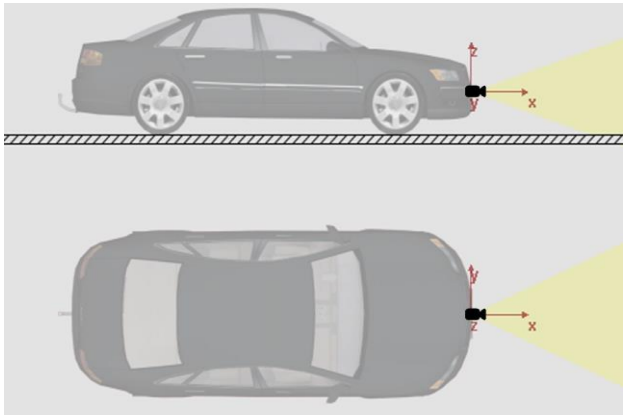

Figure 3. Front-facing camera sensor mounted on the virtual vehicle.

A. Data Collection and Preprocessing

We purposed to train the system to evaluate the capability for lateral control; therefore, only image and steering wheel angle data of the vehicle were needed. The data were extracted at a frequency of 10 Hz while the vehicle drove on the track (Figure 4) in counterclockwise direction. The vehicle states' data were synchronized with the frame number by a period of 0.1 s.

Image data were captured with 160 × 90 pixel frames, and then cropped to 160 × 40 pixels to eliminate unnecessary information of the upper pixels such as the sky, trees, or buildings far from the road.

Steering wheel angle data are acquired from the built-in PDM algorithm, which was applied as the vehicle controller model to obtain precise steering wheel angle control data. Because PDM produces precise real-time values based on the road environment and vehicle states, we should extend it to obtain a wider range of learning. We intentionally added a random interference ranging from – 50° to + 50° to the steering wheel angle output of the PDM. The purpose of the random interference was to determine how robust the proposed training loop is against random disturbances that may occur during driving.

B. Training

We used an NVIDIA DevBox for training, and the collected image and steering wheel angle data were frame-synchronized. As explained in Section II, the NN trains in two stages.

The supervised pre-training method trains the network with the data for 80,000 cycles of iterations. The number of iterations was found through experiments to determine the fewest number of iterations that allow the reinforced closed-loop feedback post-training to simultaneously train and simulate properly. After pre-training, DAVE-2SKY model undergoes post-training iterations. During the training cycles, we could monitor all state data and also visualize training and simulation environments as shown in Figure 5.

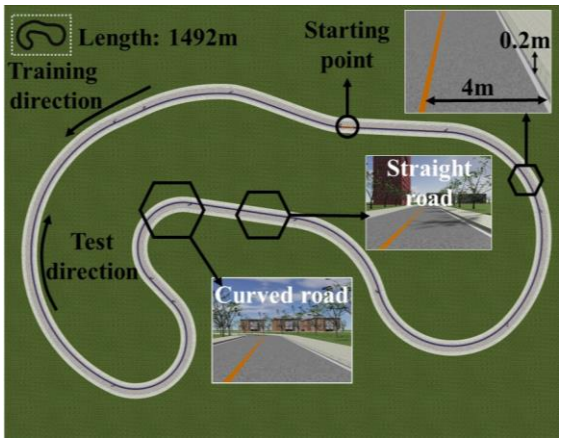

Figure 4. Overview of the track used for training and test drive simulation.

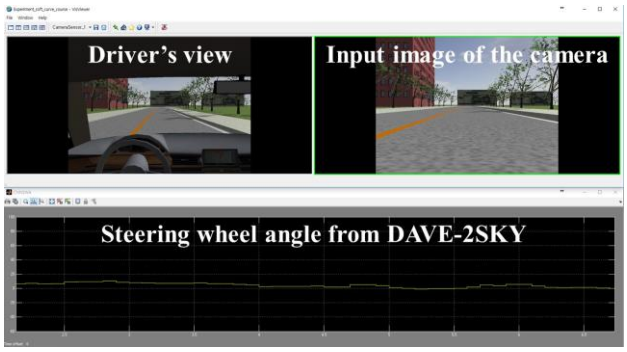

Figure 5. Screenshots of the visualized simulation environment.

III. SIMULATION RESULTS

With the simulation, we intended to investigate performances and capabilities of the proposed end-to-end controller by allowing the self-driving vehicle to cruise under diverse circumstances in the SIL environment. The vehicle drove in a clockwise direction during the test, which was opposite to the training direction. By alternating the route during the training and test simulations, we could easily provide different experiences to DAVE-2SKY network. We investigated the operational limit of the end-to-end controller by disturbing the camera's view range of the road ahead: we simulated the steering performance both in *fully observable* and *partially observable* circumstances. The various driving conditions including fully as well as partially observable cases caused by obstacles disturbing the full observation of the curved road are required to evaluate how the proposed CNN model with the reinforced feedback loop is capable of performing lateral control during lane keeping maneuvers. The fully observable case was implemented as shown in Figure 6(b) with the vehicle driving alone on the track. For the partially observable case, a front vehicle was added at various distances from the ego self-driving vehicle as shown in Figure 6(c) and 6(d). The parameter ***d*** is defined as the distance between the centers of the rear wheels of the vehicles as shown in Figure 6(a). The distance is a critical factor in determining the observable range of the front camera, and we varied the distance from 7 to 12 m during the test. The test vehicle cruised at a constant speed of 5 m/s during the simulations.

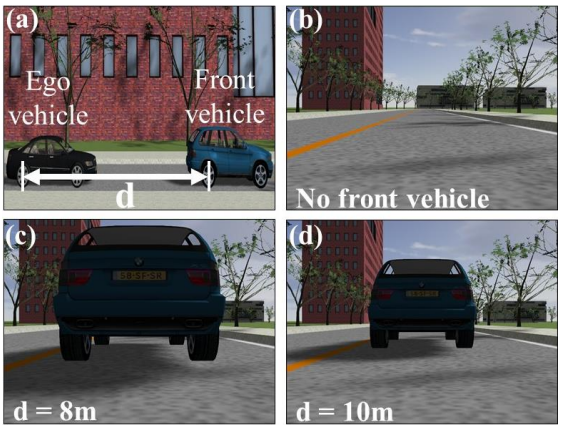

Figure 6. Input camera's view: (a) definition of the distance *d*; (b) fully observable (no front vehicle); (c) partially observable (d = 8 m); (d) partially observable (d = 10 m) cases.

We considered the fail-condition of lane keeping maneuver to occur if the vehicle deviated from the lane. The simulation is suspended if the fail-condition occurred.

All data from the controller and the front-camera were synchronized with the frame number of input images. The ground truth is represented by the steering wheel angles from the PDM. The traced error is the difference between a steering wheel angle and a ground truth at the n^{th} frame. The complete simulation concludes for a frame number of approximately 3,001 because the vehicle will return to the starting point of the 1,492 m track with constant cruise speed of 5 m/s with proper lane keeping maneuvers within 3,001 frames.

A. Fully Observable Case

Figure 7 shows the simulation results of the fully observable cases. The vehicle steered by DAVE-2SKY only pre-trained for 36 h successfully drove the vehicle until arriving at the sharp curved position of the track marked in Figure 4. The abrupt change in curvature of the track between frames 1,500 and 1,700 caused rapid change in the necessary steering control angle to maintain the lane. Consequently, the vehicle deviated from its lane shortly after the vehicle entered the sharp curved road section. Moreover, the DAVE-2SKY controller trained with the primary supervised pre-training step and subsequent reinforced post-training step successfully completed the full course of the track with a tolerable error distance from the center of the lane.

B. Partially Observable Case

In the partially observable case, every attempt to steer the vehicle by DAVE-2SKY with the pre-training step only was not successful in the simulation and could not steer the vehicle for lane keeping maneuvers if the camera could only partially observe the curved road.

The vehicle controlled by the fully trained DAVE-2SKY controller could keep its lane within a tolerable error range if the distance ***d*** was larger than a critical value. In the simulation environment of this paper, the critical value of ***d*** was 9 m. The CNN-based end-to-end controller trained with the proposed reinforced closed-loop training steps displayed much improved performance in partially observable cases.

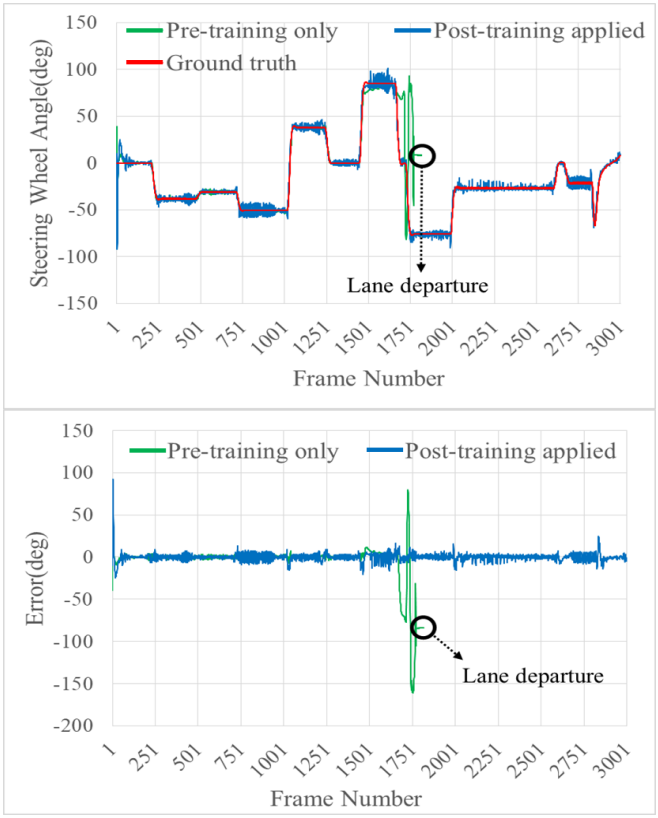

Figure 7. Simulated results for the fully observable case of DAVE-2SKY: (a) steering wheel angles (upper); (b) errors between the reference value of PDM (i.e. ground truth in the simulation) and inference output value (lower).

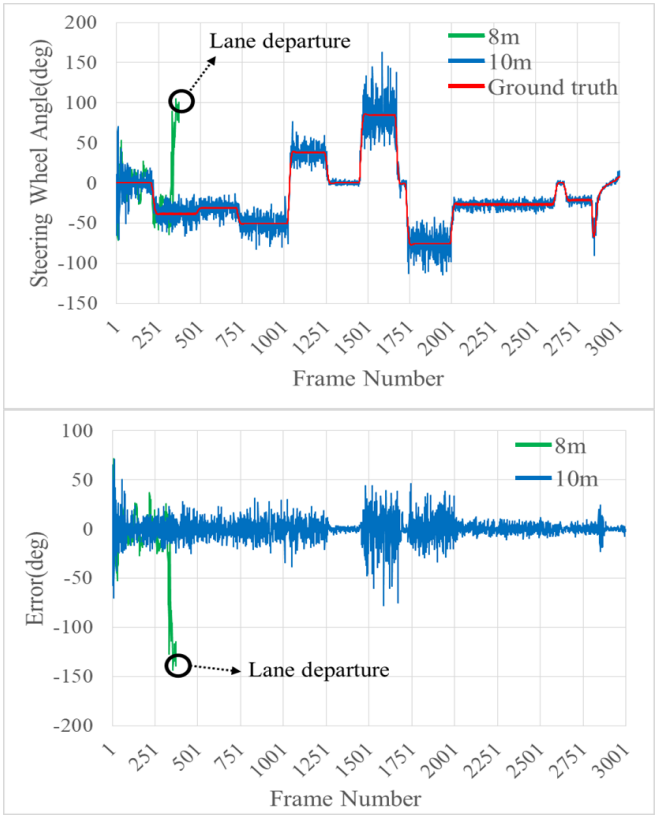

Figure 8. Simulated results for partially observable case of DAVE-2SKY: (a) steering wheel angles (upper); (b) errors between the reference value of PDM (i.e. ground truth in the simulation) and inference output value (lower).

The vehicle controlled by the fully trained DAVE-2SKY controller could keep its lane within a tolerable error range if the distance ***d*** was larger than a critical value. In the simulation environment of this paper, the critical value of ***d*** was 9 m. The CNN-based end-to-end controller trained with the proposed reinforced closed-loop training steps displayed much improved performance in partially observable cases.

When a front moving vehicle is too close to the ego vehicle, most input information of the curved road is screened by the front car. However, if the distance was greater than 9 m in out simulations, the vehicle could successfully cruise the track while keeping its lane within a tolerable error range. The simulated results presented in Figure 8 are somewhat noisy, but the vehicle drove itself successfully despite additional disturbances applied during the simulation.

IV. Experimental Results

Because several simulations have demonstrated acceptable performance for lane keeping tasks, DAVE-2SKY was integrated into a real autonomous vehicle using a DRIVE™ PX2 computer for an experimental real road autonomous driving test as shown in Figure 9 and Figure 10.

By applying the procedures described in Section III, image data of the real road were collected and preprocessed for training of the real vehicle. The collected datasets contained images over 2 h of driving on a track at Yonsei University International Campus (the environment is shown in Figure 10). During data collection, a skilled human driver drove the vehicle to maintain its lane without departure. The vehicle was trained with the proposed two consecutive training methods for 72 h. Then the trained model was transferred to the PX-2 computer mounted on the vehicle for experimental real road tests. During the test, we also evaluated a valet parking scenario with automatic parking algorithms based on prior articles [12,13]. The experiment verified that the aforementioned techniques applied to the DAVE-2SKY controller may be capable of longitudinal control of autonomous vehicles. The experimental videos are shown on YouTube [14,15]. Because of length restrictions, we omit the details of the experimental scenario and corresponding data in this article.

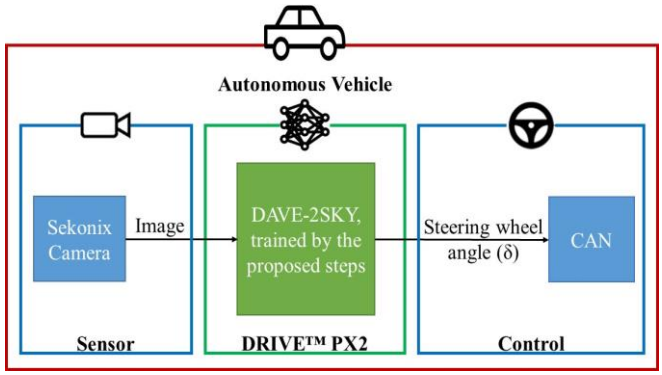

Figure 9. Test vehicle implementation for experimental autonomous real road driving test

Figure 10. Experimental routes for autonomous driving, and snapshot photos of the vehicle under road test

V. Discussion

The simulations for the fully and partially observable cases presented a distinctive capability for DAVE-2SKY trained with the proposed steps. The simulation results described in Section IV showed that the model self-taught only by the supervised pre-training, which has similar characteristics to traditional end-to-end CNN models, was unable to perform proper lateral control for the lane keeping task. In spite of a similar number of training cycles, however, the reinforced closed-loop feedback post-training literally reinforced and improved the performance of the steering control.

The partially observable case shows the robustness of DAVE-2SKY. Owing to the closed-loop feedback system in addition to the back propagation loop in the deep NN, the proposed DAVE-2SKY performs robust steering control even under partially observable cases in the lane keeping maneuvers. Simulation results show that DAVE-2SKY is able to recover from a small amount of disturbance during self-cruise control on a track.

The results support our assertion that a supervised pre-trained and subsequent post-trained step with the reinforced closed-loop feedback enables an end-to-end controller with complete lateral control for lane keeping tasks within a reasonable error range.

VI. Conclusions

We proposed an end-to-end steering controller with CNN-based closed-loop feedback for autonomous vehicles that improves performance compared to traditional CNN-based approaches. The proposed NN, DAVE-2SKY, is able to learn to control the steering wheel angle for the lateral control of self-driving vehicles through supervised pre-training and reinforced closed-loop post-training with images from a camera mounted on the vehicle. We used the PreScan simulator and Caffe deep learning framework for training under diversified circumstances in a SIL simulation environment. The performance of the proposed system has been investigated under simulations and on-road tests. We used a DRIVE™ PX2 computer to implement a self-driving car for an experimental validation of the proposed end-to-end controller. In conclusion, this work shows that the CNN-based end-to-end controller performs robust steering control even under partially observable circumstances, which indicates the possibility of fully intelligent self-driving vehicles controlled by a CNN-based end-to-end steering controller.

Acknowledgment

This work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Consilience Creative program (IITP-2017-2017-0-01015) supervised by the IITP (Institute for Information & communications Technology Promotion). The authors performed this work as a part of research projects of SKT-Yonsei Cooperative Autonomous Driving Research Center under the SKT-Yonsei Global Talent Fostering Program supported by the SK Telecom ICT R&D Center.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[2] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network." *Advances in neural information processing systems*. 1989.

[3] Net-Scale Technologies, Inc., "Autonomous off-road vehicle control using end-to-end learning," July 2004.
URL: http://net-scale.com/doc/net-scale-dave-report.pdf

[4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, and X. Zhang, "End to end learning for self-driving cars." *arXiv preprint arXiv*: 1604.07316 (2016).

[5] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars." *Intelligent Vehicles Symposium (IV), 2017 IEEE*, IEEE, 2017.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrel, "Caffe Convolutional architecture for fast feature embedding," *arXiv preprint arXiv: 1408.5093,* 2014.

[7] Berkeley Artificial Intelligence Research, "Caffe | Euclidean Loss Layer,"
URL: http://caffe.berkeleyvision.org/tutorial/layers/euclideanloss.html

[8] S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *International Conference on Machine Learning*. 2015.

[9] C. I. Chatzikomis and K. N. Spentzas. "A path-following driver model with longitudinal and lateral control of vehicle's motion." *Forschung im Ingenieurwesen* 73.4 (2009): 257–266.

[10] R.S. Sharp, D. Casanova, and P. Symonds, "A Mathematical Model for Driver Steering Control, with Design, Tuning and Performance Results," *Vehicle System Dynamics*, 33:5, 289–326.

[11] Wikipedia contributors, "Steering ratio," *Wikipedia, The Free Encyclopedia*. Wikipedia, Jan 2018.
URL:
https://en.wikipedia.org/w/index.php?title=Steering_ratio&oldid=818506438

[12] J. Moon, I. Bae, J. G. Cha, and S. Kim, "A trajectory planning method based on forward path generation and backward tracking algorithm for automatic parking systems." *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014.

[13] I. Bae, J. H. Kim, and S. Kim, "Steering rate controller based on curvature of trajectory for autonomous driving vehicles." *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013.

[14] Seamless Transportation Lab, Yonsei University, "Autonomous Valet Parking Demonstration 2017/ Seamless Transportation Lab, Yonsei University," *YouTube*, November 2017.
URL: https://youtu.be/kxaZvrhftuo

[15] Seamless Transportation Lab, Yonsei University, "Autonomous Valet Parking System with Surround View Camera (II)/ Seamless Transportation Lab, Yonsei University," *YouTube*, November 2017.
URL: https://youtu.be/9rSGsKXuat8