

Article

Development of Deep Learning Based Human-Centered Threat Assessment for Application to Automated Driving Vehicle

Donghoon Shin ¹ , Hyun-geun Kim ², Kang-moon Park ^{3,*}  and Kyongsu Yi ^{4,*}

¹ Department of Mechanical Systems Engineering, College of Engineering, Sookmyung Women's University, Seoul 04310, Korea; dhshin@sookmyung.ac.kr

² Department of Computer Science, Korea Aerospace University, Goyang-si 10540, Korea; rgdkdle4_doe@kau.kr

³ Department of Computer Science, College of Natural Science, Republic of Korea Naval Academy, Changwon-si 51704, Korea

⁴ Department of Mechanical and Aerospace Engineering, College of Engineering, Seoul National University, Seoul 08826, Korea

* Correspondence: kmun422@naver.com (K.-m.P.); kyi@snu.ac.kr (K.Y.); Tel.: +82-55-954-5251 (K.-m.P.); +82-2-880-1941 (K.Y.)

Received: 24 November 2019; Accepted: 25 December 2019; Published: 28 December 2019



Featured Application: Risk assessment, deep learning, network architecture search, recurrent neural network, automated driving vehicle.

Abstract: This paper describes the development of deep learning based human-centered threat assessment for application to automated driving vehicle. To achieve naturalistic driver model that would feel natural while safe to a human driver, manual driving characteristics are investigated through real-world driving test data. A probabilistic threat assessment with predicted collision time and collision probability is conducted to evaluate driving situations. On the basis of collision risk analysis, two kinds of deep learning have been implemented to reflect human driving characteristics for automated driving. A deep neural network (DNN) and recurrent neural network (RNN) are designed by neural architecture search (NAS), and by learning from the sequential data, respectively. The NAS is used to automatically design the individual driver's neural network for efficient and effortless design process while ensuring training performance. Sequential trends in the host vehicle's state can be incorporated through hand-made RNN. It has been shown from human-centered risk assessment simulations that two successfully designed deep learning driver models can provide conservative and progressive driving behavior similar to a manual human driver in both acceleration and deceleration situations by preventing collision.

Keywords: risk assessment; deep learning; neural architecture search; recurrent neural network; automated driving vehicle

1. Introduction

Autonomous vehicles are designed to take human error out of driving actions, which should help make self-driving vehicles safer, thus, dramatically reducing the number of road accidents. Nowadays, high-level automation projects such as General Motors' Cruise Automation [1], Waymo from Google [2], and the highly market penetrated Autopilot from Tesla [3] are the main examples of the leading commercial autonomous vehicle industry. Waymo's robot taxis, although there are still some notable

complaints about driver acceptance, according to [4] because of shaky driving experience, as well as lack of trust in the autopilot is one of the critical reasons for low acceptance of the system [5].

Recently, Hyundai Motor Group develops the world's first machine learning based smart cruise control (SCC-ML) technology [6]. SCC-ML combines AI and SCC into a system that learns the driver's patterns and habits on its own. Through machine learning, SCC autonomously drives in an identical pattern as that of the driver. Lefèvre et al. [7] proposed a method for personalizing the driving style of an autonomous vehicle. The driving styles are learned from the human driver in order to benefit from the experienced skill sets of different situations. According to [8], a confusion matrix method based on natural driving data sets was used to tune control parameters in the proposed adaptive cruise control (ACC) system.

Although conventional learning framework is beneficial for mimicking human driver behavior, however, it consumes a large amount of time and error-prone process due to the fact that it is developed by human experts [9]. As deep learning has scaled up to more challenging tasks, the architectures have become difficult to design by hand [10]. For these reasons, various studies have been carried out to generate the network architecture automatically [11]. Structure learning is a very useful instrument that is able to automatically find an appropriate artificial neural network (ANN) architecture [12]. For this reason, the structure learning algorithm is used to generate the topology of ANN in this study.

Nevertheless, NAS based DNN is still questionable due to a lack of sequential data. Zoph and Le [13] use a recurrent neural network (RNN) policy to sequentially sample a string that in turn encodes the neural architecture. The main advantage of RNN over ANN is that RNN can model a sequence of data (i.e., time series) so that each sample can be assumed to be dependent on previous ones. In order to express the sequential data in both acceleration and deceleration situations of human driving, hand-made RNN is implemented in this paper.

With a well-trained neural network, the automated driving function can be achieved by guaranteeing the safety at any driving speed. Thus, human-centered risk assessment approach [14] has been incorporated to demonstrate the naturalistic and safe driving performance for a vehicle longitudinal control situation. There are many different risk assessment methods that have been investigated previously through many scholars [15–17]. The current stages of risk metrics are mainly explained by the predicted time when some known hazardous events are encountered. The typical predicted time based indices include the time to collision (TTC) or time to impact [18]. Polychronopoulos et al. proposed the predicted time to minimum distance with sensor-fusion method [19]. Time to react which means the last point to decelerate or accelerate or steer to avoid the predicted collision has been proposed and analyzed [17]. Berthelot et al. presented an algorithm to compute the probability distribution of TTC induced by an uncertain system input and thus allows to use TTC as a more robust and reliable probabilistic activation condition [20].

This paper is organized as follows. The next section accounts for human driving data based threat assessment by investigating manual driving data. In Section 3, a design of deep neural network for learning human driver is demonstrated. The two deep learning driver model based risk assessment algorithms with manual human driving data are evaluated and validated via simulation studies in Section 4. Finally, conclusions are provided in Section 5.

2. Human-Centered Threat Assessment

A human driver's driving behavior is analyzed using real-world driving data. Driving data were collected using a test vehicle. Figure 1a shows the test vehicle used for a human-centered risk assessment in this study. We modified the 2014 Kia K7 with market-released sensor setup as follows: A multilayer laser scanner was added for monitoring static obstacles with increased precision. For lane detection, an additional monocular vision system was mounted on the windshield. In addition, a low-cost-GPS was equipped for a rough-precision host-localization which is used for initial corridor decision. The complete sensor setup is shown in Figure 1a. Yellow is a monocular vision system for lane detection. The front radar system is depicted in blue and two rear-side radars are depicted

in green. A multilayer laser scanner for obstacle monitoring is shown in red and a low-cost GPS is depicted in purple signal. Vehicle speed, engine RPM, turbine speed of the torque converter, throttle position, and gear status are obtained from the engine control unit (ECU) and each sensor through the controller area network (CAN). With 125 drivers, the measured data set for a normal driving is 1809, and for an emergency deceleration case is 62.

Figure 1b shows an architecture of human-centered threat assessment algorithm with sensors. In order to process the data from the sensor input, an environment description algorithm has 10 Hz of sampling rate [21].

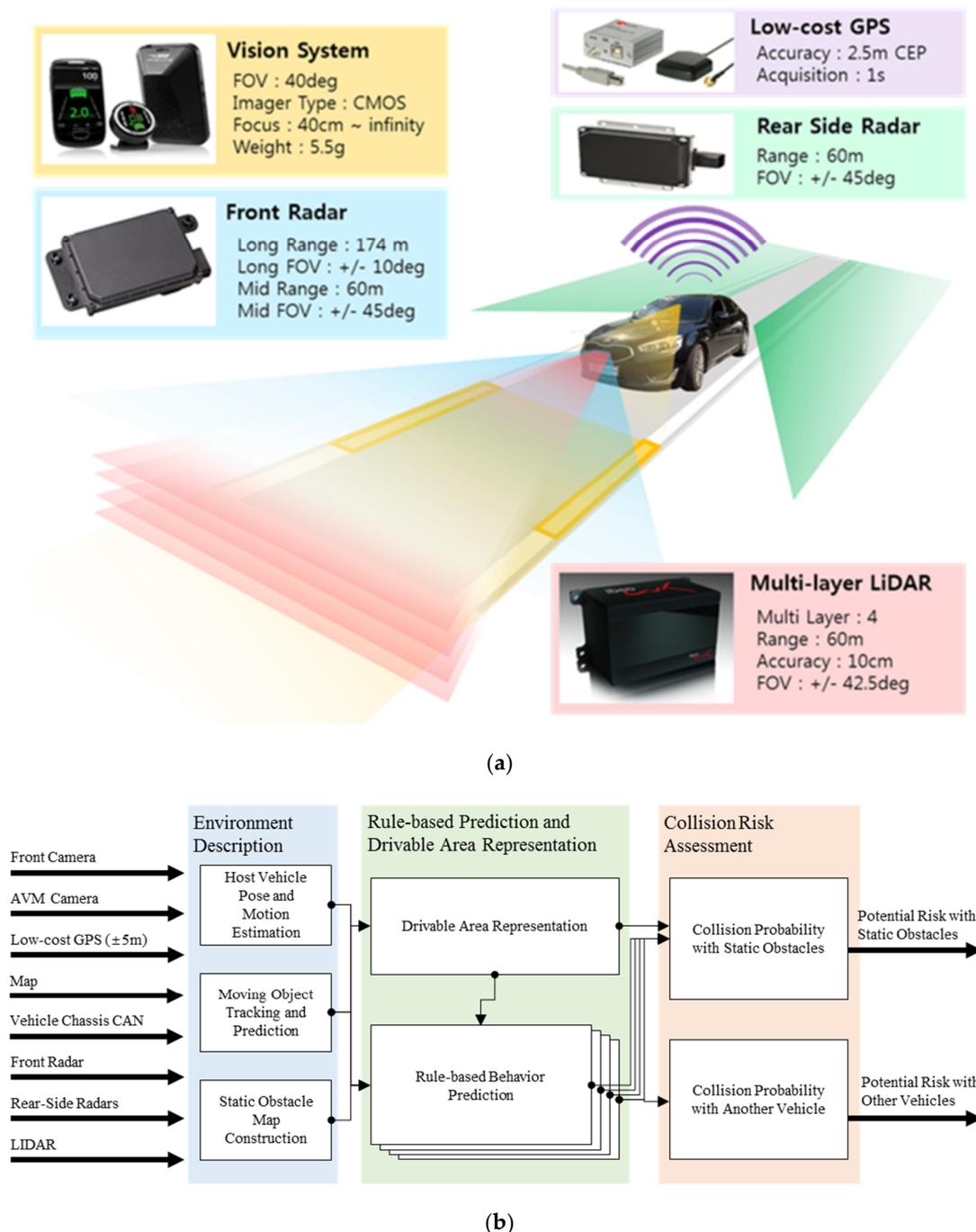


Figure 1. The experimental vehicle with sensor setup and architecture of a human-centered risk assessment. (a) Sensor configuration of test vehicle; (b) Architecture of human-centered threat assessment.

2.1. Driving Characteristics of Human Drivers

It is well known that the threat assessment performance can be significantly enhanced by human-centered risk assessment in [14]. The human-centered risk assessment algorithm monitors the risk and determines an active safety control intervention point based on collision probability. In order to analyze the driving pattern of the human driver, the collision probability is used to compute the threat vehicle ahead using real-road test driving data. The test had been performed on the highway and urban traffic situations. It consists of 40 to 1800 s of driving duration with the speed of 0 to 120 kph. With 125 drivers, the measured data set for a normal driving is 1809, and for an emergency deceleration case is 62. The collision probability, $maxCp$ is accumulated as seen in Figure 2. With the computation of collision probability, most probable predicted collision time, Tp_{maxCp} , is derived according to $maxCp$ real time. The driving data can be analyzed by contouring it based on the range of general acceleration pattern of the driver. To minimize the effect of outlier detection from sensors, accumulated average longitudinal acceleration is used by removing the outliers. Figure 3 successfully demonstrates the human driver’s driving pattern due to the fact that collision risk increases as the driver decelerates the vehicle. The driver’s normal driving pattern is not generally expressed on the “A” since the collision risk grows really fast with the short duration. Additionally, the collision probability cannot be defined on “B”.

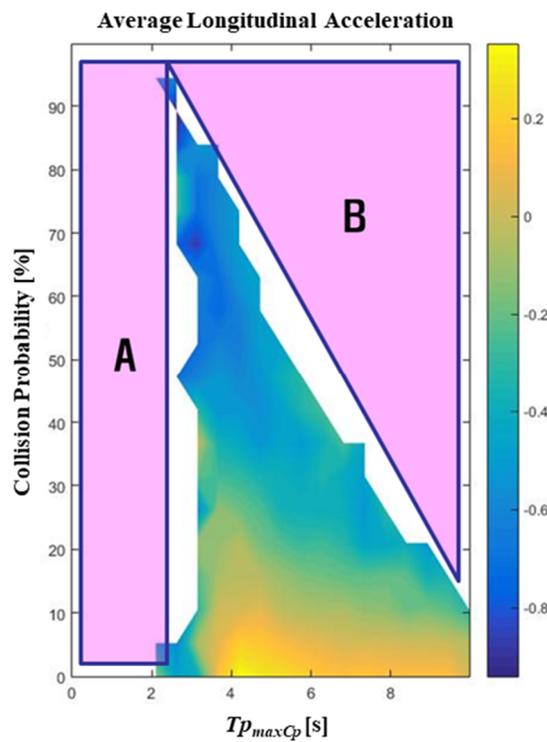


Figure 2. Human driver’s driving pattern with acceleration.

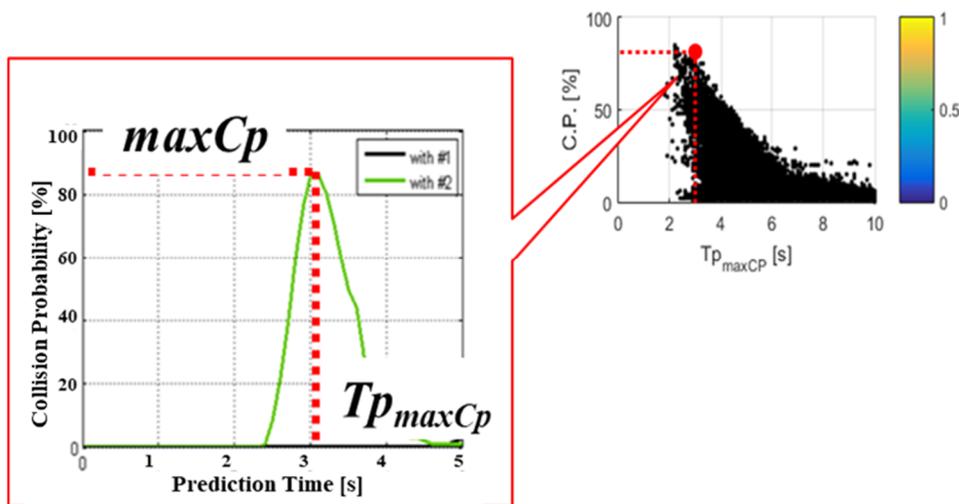


Figure 3. Accumulation of peak collision probability.

2.2. Risk Management

The test driving data usually constitutes four phases of risk management pattern of the human driver as shown in Table 1. Phase 0 is normal driving situation. Phase 1 is alert situation because of human perception failure. The human perception failure can be triggered by distracted or incautious driving. Phase 2 is hazard detection with control for the collision avoidance. Phase 3 is a recovery stage for a safety distance with respect to the preceding vehicle. In order to verify the threat assessment and risk management strategy, the vehicle test had been performed on the real road as seen in Figure 4. The host vehicle (denoted by green vehicle) drives in a straight lane with an initial speed of 50 km/h. The participant 1 (denoted by gray vehicle) drives with an initial speed of 40 km/h and participant 2 is the stop vehicle. Participant 1 suddenly starts to cut-out when the host-vehicle’s clearance to participant 2 is only just 35 m. As meeting the unexpected stop vehicle, the host vehicle suddenly brakes to assure a safe distance using the peak deceleration at about 4 m/s². Figure 5 shows the three stages of human driver’s risk management pattern as explained above according to the described test scenario.

Table 1. Risk management and driving phase [21].

Driving Phase	Risk Management
Phase 0	Normal Driving
Phase 1	Collision Risk Increase
Phase 2	Safety Control
Phase 3	Safety Spacing

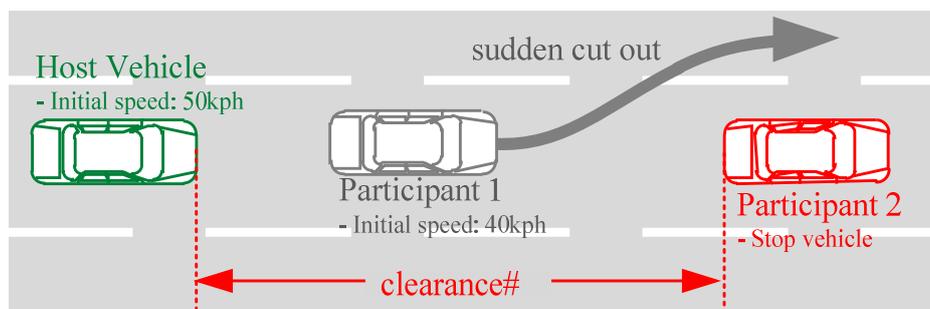


Figure 4. Test scenario: Sudden stop due to unexpected stop-vehicle.

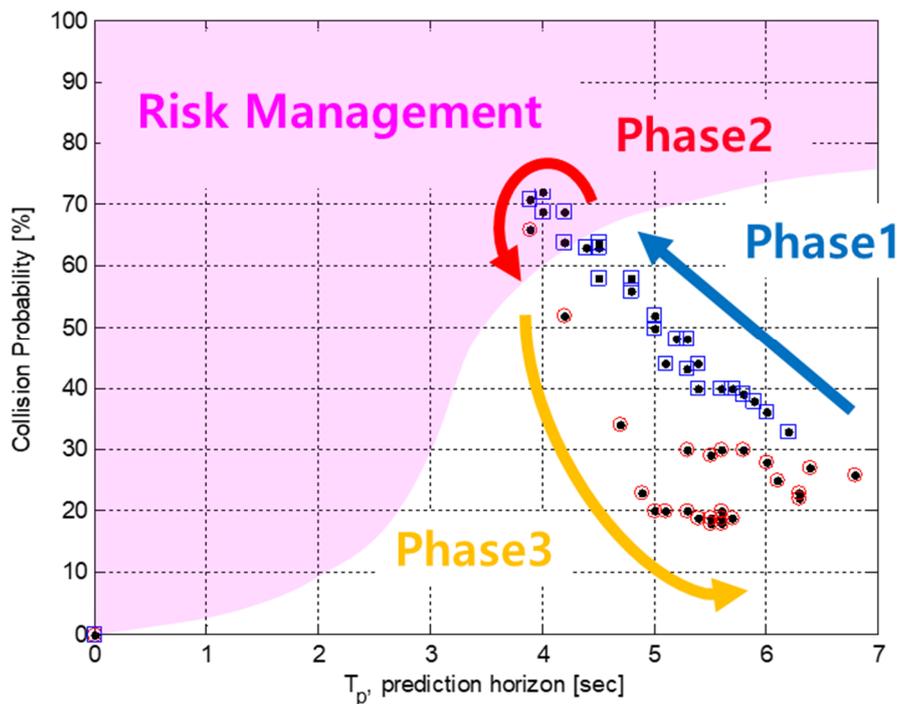


Figure 5. Risk management strategies.

3. Design of Deep Neural Network for Learning Human Driver

Deep neural network (DNN) is a multi-layer perceptron (MLP) with many hidden layers [22]. It consists of many neurons. Neurons which consist of MLP are called perceptron. Perceptron is an algorithm for supervised learning of binary classification which imitate neurons of human driver's driving pattern by mimicking its brain. It is trained with the error backpropagation algorithm [23].

There are various types of DNN and each has an appropriate application. For example, convolutional neural network (CNN) is the most famous structure of DNN. CNN is specialized in image data processing which can be regarded as a two-dimension (2D) grid of pixels [24]. To learn human driving data, other forms of DNN need to be considered.

3.1. Deep Neural Network Based on Neural Architecture Search

Neural architecture search (NAS) is a method for automatic design of DNN architecture. In general, designing high-performance DNN architectures require extensive research efforts by human experts through lots of trial and error [25]. It is a time consuming and error-prone process [26]. However, the NAS method allows researchers to find proper architecture easily.

The NAS method can design individual DNN architecture automatically according to every single driver dataset. Due to this, NAS techniques are beneficial for effortless and efficient design of tailor made DNN for intelligent driver assistance systems [27].

3.2. Recurrent Neural Network (RNN) Using Sequential Data

RNNs are DNN architectures for processing sequential data such as text or speech [28–30]. At each time step, the current output presents the network through a layer of input units [29]. In other words, output data of the previous step is used for current input data. It is possible for RNNs to remember previous states and to learn sequential data through the recursive process. RNNs are suitable for learning driving patterns due to the fact that driving patterns can be represented by time series sequential data [31]. Thus, RNNs are expected to show good results of human driving pattern compared to other DNNs.

3.3. Learning Driving Data of Adaptive Cruise Control

We used both DNN generated by NAS and general RNN architecture to learn human driving data as described in Section 2. Both NAS based DNN and RNN have two inputs as shown in Figure 6: A longitudinal acceleration (A_x) and the most probable predicted collision time (Tp_{maxCp}). Additionally, the output is collision probability, $maxCp$.

As shown in Figure 6, NAS methodology with an auto-design DNN architecture which uses evolutionary algorithm (EA) with variable chromosome genetic algorithm (VCGA) has been implemented [18]. VCGA makes ANN variable with variable chromosome and chromosome attachments, which makes it suitable for generating simple ANN.

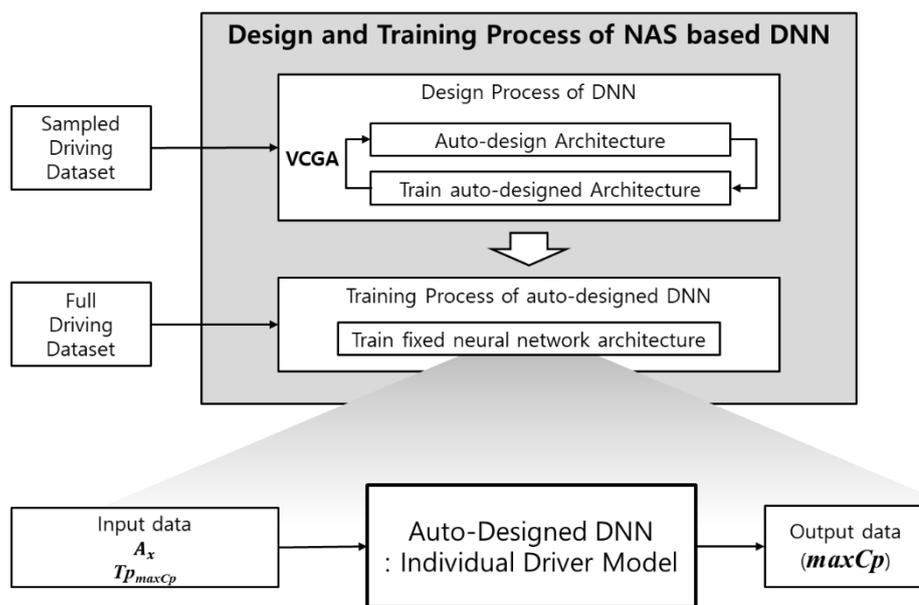


Figure 6. Auto-design process of DNN using NAS.

4. Simulation Studies

To validate the design of DNN and its applications for human-centered threat assessment of automated driving vehicle, simulations are conducted on the C/C++ and MATLAB/SIMULINK, respectively.

Design validations for the two DNN are performed using real world test driving data as described in Section 2. Using sampled data, NAS based DNN is generated. Trade-off between a loss rate and calculation speed have been investigated to make sure whether NAS makes the correct structure. The loss rate is the discrepancy of the manual driver's data and output data of trained model.

In order to investigate the effectiveness of sequential data characteristics of RNN, we compare auto-designed (NAS based) DNN with hand-made RNN by presenting the average loss rate of two.

Human-centered threat assessment has been implemented to represent driving behavior of two deep learning models. The two trained output have been analyzed and demonstrated through a collision risk framework compared with manual driving data as described in Section 2. According to the type of neural network, the driving styles are expressed.

4.1. Deep Neural Network Design Validation

We evaluated the proposed two ANN (auto-designed DNN and RNN) on driving dataset of ACC. The size of this dataset is 146,565. First, we sampled eight data for the process of auto-design DNN. Figure 7 shows the auto-design process of DNN using NAS. It has eight batch sizes and 5000 epochs with a 0.2 learning rate. The number of neurons including DNN has increased by 14 in early stage, and

it has an accuracy lower than 0.8. However, the number of neurons has decreased by six and it shows that oscillatory increased up to 10. When it has increased to 10, accuracy of this DNN is about 0.9. After 10 neurons, the number of neurons has decreased gradually. On the other hand, accuracy has settled down within about 0.9. The number of neurons has decreased and settled down to four or six.

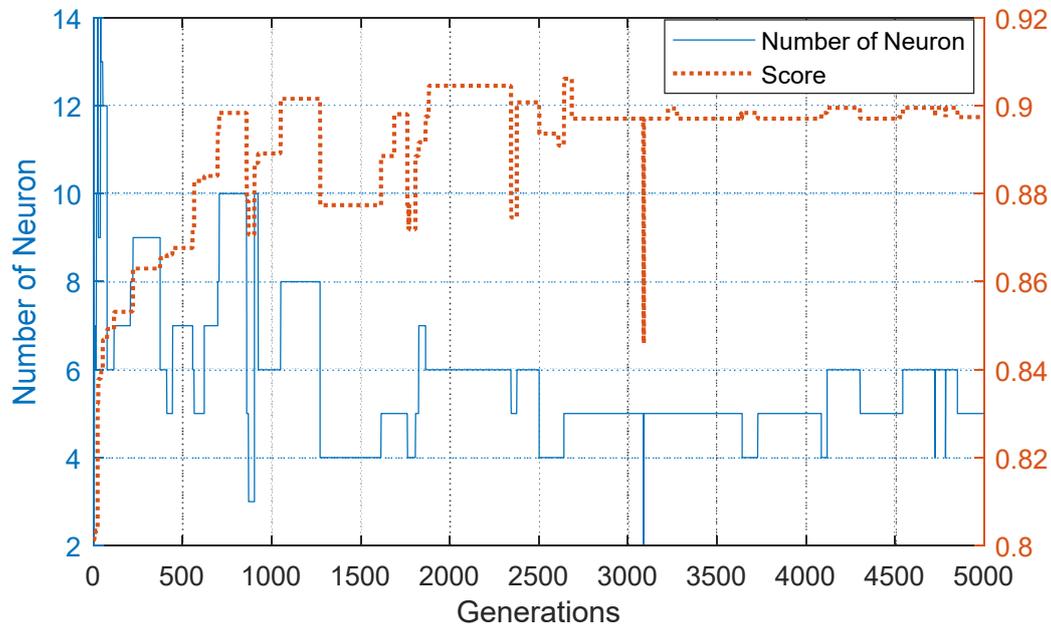


Figure 7. Auto-design process of DNN using NAS.

The auto-designed DNN architecture shown in Figure 7 has four or six neurons and a fully connected architecture with one hidden layer. Then, we simulated some DNN architecture trained with full dataset to validate this result. Figure 8 shows results of loss rate with respect to the number of neurons. The architecture with five or six neurons shows the minimum loss, meanwhile it has a similar loss rate. The architecture that has lower number of neurons is a much better architecture since the number of neurons affect the calculation speed. Figure 9 shows the loss of auto-designed DNN with five neurons. The average loss rate of this architecture is 0.016811.

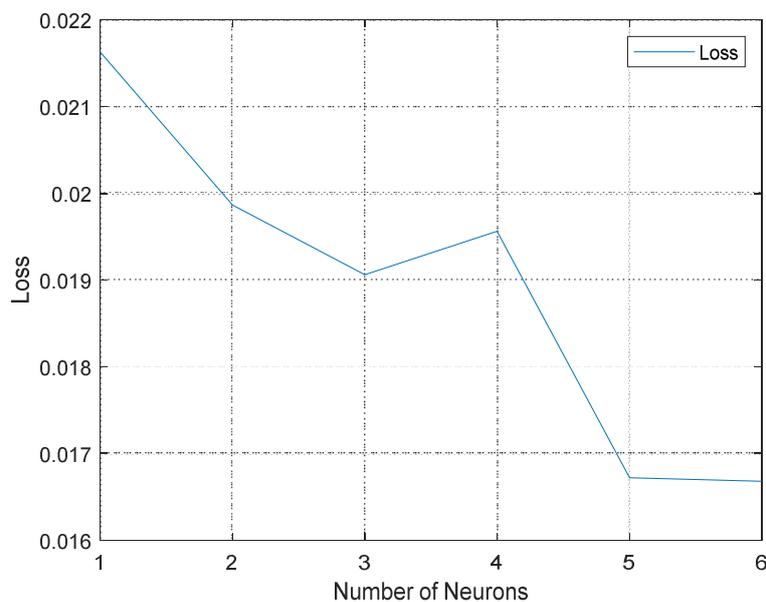


Figure 8. Loss rate with respect to the number of neurons.

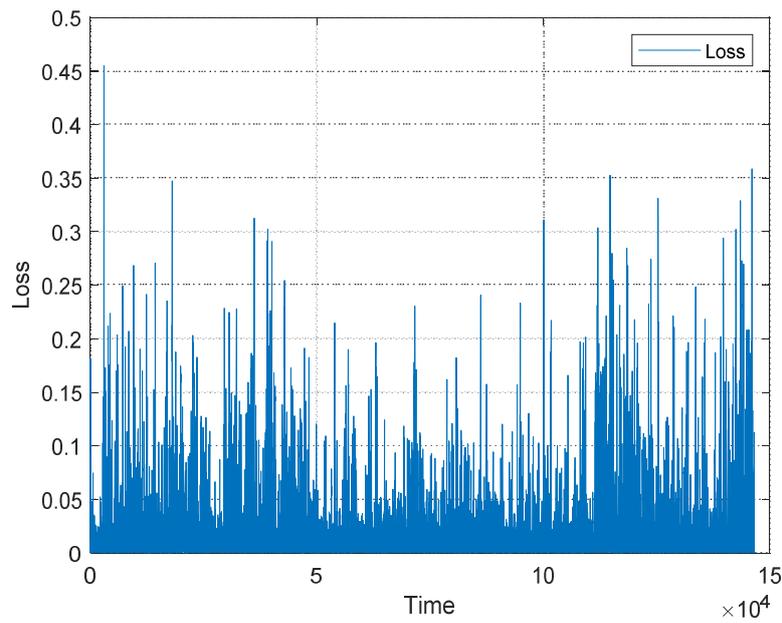


Figure 9. Loss of auto-designed DNN.

Since the human driving dataset has sequential data, we compared auto-designed DNN with RNN. The RNN architecture with a five sequence length is implemented. Figure 10 shows loss of RNN. The average loss rate of this architecture is 0.029620.

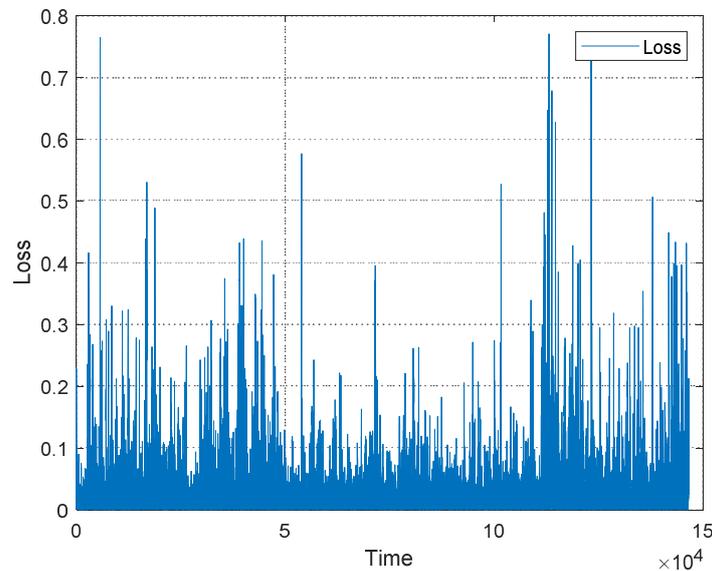


Figure 10. Loss of RNN.

Table 2 presents average loss rate of each of architectures. DNNs with 5 neurons and 6 neurons have almost same loss rate. However, DNN with 5 neurons is optimal architecture because the number of neuron affects calculation speed. And loss rate of RNN is lower than auto-designed DNN, it is regarded as discrete data like sudden brake.

Table 2. Average loss rate of each of the architectures.

Architecture	DNN with 1 Neuron	DNN with 2 Neurons	DNN with 3 Neurons	DNN with 4 Neurons	DNN with 5 Neurons	DNN with 6 Neurons	RNN with 5 Sequences
Loss	0.021684	0.019931	0.019142	0.019627	0.016811	0.016753	0.02962

4.2. Two Deep Learning Driver Models Based on Human-Centered Threat Assessment

The individual driver model has been made using NAS and RNN. In the case of NAS, the fixed neural network architecture has been trained by a full data set of human manual driving in order to design or mimic the individual driver model. With the series of sequential data, hand-made RNN is used to learn human driving pattern. Figure 11a–c show the human-centered threat assessment results using manual driving data, NAS and RNN, respectively. As discussed in Section 2, the accumulated collision risk reflects the acceleration and deceleration pattern of the human driver.

As shown in Figure 11b, the driving pattern of the auto-designed NAS is quite aggressive or progressive by expanding its collision risk to 97.53%. On the other hand, with the sequential data in Figure 11c, the driver model using RNN demonstrates conservative driving behavior by dramatically reducing its maximum collision risk to 60% and illustrates a more naturalistic driving behavior compared to the auto-designed DNN. It is shown that the two proposed deep learning driver models guarantee safety by maintaining a maximum collision risk under 100%.

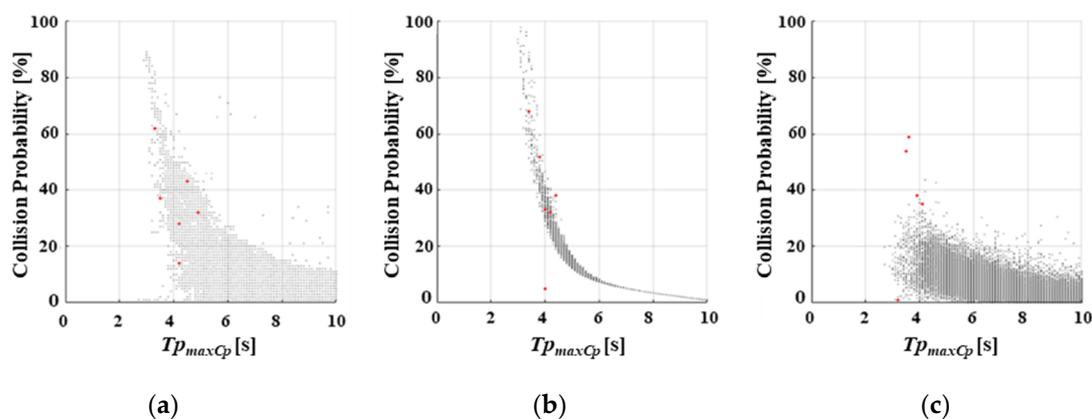


Figure 11. Human-centered threat assessment results. (a) Manual driving data; (b) NAS; (c) RNN.

5. Conclusions

A development of deep learning based human-centered threat assessment for application to automated driving vehicle has been presented. A NAS is used to design the individual driver model because of its effortless and efficient automatic design process. In order to design a correct structure of NAS based DNN, the loss ratio and the number of neurons which reflects calculation speed have been optimized to 0.016 and five, respectively. Hand-made RNN is used to design the driver model by incorporating a series of sequential human driving data. It was shown from simulation studies that the two deep learning driver models using NAS and hand-made RNN can provide both conservative and progressive driving behaviors similar to the manual human driver in both acceleration and deceleration situations while ensuring safety of the vehicle. The effects of sequential data series for deep learning, RNN, have been investigated through the human-centered threat assessment algorithm and this shows a dramatically reduced collision risk by 37.53%, and illustrates a more naturalistic driving behavior. In order to better represent increased risk instances of the human driver, the host vehicle's pitching motion due to sudden acceleration or deceleration can be additionally considered in the future. Meanwhile, implementation, validation, and evaluation of the auto-designed RNN for application to human-centered risk assessment of an automated driving vehicle are the topic of our future research.

Author Contributions: Conceptualization, D.S. and K.-m.P.; Methodology, K.-m.P.; Software, K.-m.P.; Validation, H.-g.K. and K.-m.P.; Formal Analysis, D.S.; Investigation, H.-g.K. and K.-m.P.; Resources, D.S. and K.Y.; Data Curation, H.-g.K.; Writing—Original Draft Preparation, D.S. and K.-m.P.; Writing—Review and Editing, D.S.; Visualization, D.S.; Supervision, K.-m.P.; Project Administration, D.S. and K.Y.; Funding Acquisition, K.Y. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This research was supported by the National Research Foundation of Korea (NRF-2016R1E1A1A01943543), by the technology innovation program, by the Technology Innovation Program (10079730, Development and Evaluation of Automated Driving Systems for Motorway and City Road and driving environment) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea), and by the Institute of Advanced Machines and Design (IAMD), Seoul National University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. GM's Cruise Rolls Back Its Target for Self-Driving Cars|WIRED. Available online: <https://www.wired.com/story/gms-cruise-rolls-back-target-self-driving-cars/> (accessed on 22 November 2019).
2. Technology. Available online: <https://waymo.com/tech/> (accessed on 22 November 2019).
3. Autopilot and Full Self-Driving Capability. Available online: <https://www.tesla.com/support/autopilot> (accessed on 22 November 2019).
4. Archive, P.; Twitter, F.A. Waymo's Backseat Drivers: Confidential Data Reveals Self-Driving Taxi Hurdles. Available online: <https://www.theinformation.com/articles/waymos-backseat-drivers-confidential-data-reveals-self-driving-taxi-hurdles> (accessed on 22 November 2019).
5. Van Huysduynen, H.H.; Terken, J.; Eggen, B. Why Disable the Autopilot? In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*; ACM: Toronto, ON, Canada, 2018; pp. 247–257.
6. Hyundai Motor Group Develops World's First Machine Learning based Smart Cruise Control (SCC-ML) Technology. Available online: <https://news.hyundaimotorgroup.com/MediaCenter/News/Press-Releases/hyundai-motor-group-develops-world-s-first-machine-learning-based-smart-cruise-control-scc-ml-technology> (accessed on 22 November 2019).
7. Lefèvre, S.; Carvalho, A.; Borrelli, F. A learning-based framework for velocity control in autonomous driving. *IEEE Trans. Autom. Sci. Eng.* **2015**, *13*, 32–42. [[CrossRef](#)]
8. Moon, S.; Yi, K. Human driving data-based design of a vehicle adaptive cruise control algorithm. *Veh. Syst. Dyn.* **2008**, *46*, 661–690. [[CrossRef](#)]
9. Elsken, T.; Metzen, J.H.; Hutter, F. Neural architecture search: A survey. *arXiv* **2018**, arXiv:1808.05377.
10. Miikkulainen, R.; Liang, J.; Meyerson, E.; Rawal, A.; Fink, D.; Francon, O.; Raju, B.; Shahrzad, H.; Navruzyan, A.; Duffy, N. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 293–312.
11. Larrañaga, P.; Poza, M.; Yurramendi, Y.; Murga, R.H.; Kuijpers, C.M.H. Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 912–926. [[CrossRef](#)]
12. Kwok, T.-Y.; Yeung, D.-Y. Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Trans. Neural Netw.* **1997**, *8*, 630–645. [[CrossRef](#)] [[PubMed](#)]
13. Zoph, B.; Le, Q.V. Neural architecture search with reinforcement learning. *arXiv* **2016**, arXiv:1611.01578.
14. Shin, D.; Kim, B.; Yi, K.; Carvalho, A.; Borrelli, F. Human-Centered Risk Assessment of an Automated Vehicle Using Vehicular Wireless Communication. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 667–681. [[CrossRef](#)]
15. Althoff, M.; Mergel, A. Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1237–1247. [[CrossRef](#)]
16. Ghazel, M. Using stochastic Petri nets for level-crossing collision risk assessment. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 668–677. [[CrossRef](#)]
17. Hillenbrand, J.; Spieker, A.M.; Kroschel, K. A multilevel collision mitigation approach—Its situation assessment, decision making, and performance tradeoffs. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 528–540. [[CrossRef](#)]
18. Labayrade, R.; Royere, C.; Aubert, D. Experimental assessment of the rescue collision-mitigation system. *IEEE Trans. Veh. Technol.* **2007**, *56*, 89–102. [[CrossRef](#)]
19. Polychronopoulos, A.; Tsogas, M.; Amditis, A.J.; Andreone, L. Sensor fusion for predicting vehicles' path for collision avoidance systems. *IEEE Trans. Intell. Transp. Syst.* **2007**, *8*, 549–562. [[CrossRef](#)]
20. Berthelot, A.; Tamke, A.; Dang, T.; Breuel, G. A novel approach for the probabilistic computation of time-to-collision. In *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Alcalá de Henares, Spain, 3–7 June 2012*; pp. 1173–1178.

21. Kim, B.; Yi, K.; Yoo, H.-J.; Chong, H.-J.; Ko, B. An IMM/EKF approach for enhanced multitarget state estimation for application to integrated risk management system. *IEEE Trans. Veh. Technol.* **2015**, *64*, 876–889. [[CrossRef](#)]
22. Miao, Y.; Metze, F. Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training. In Proceedings of the 14th Annual Conference of the International Speech Communication Association, Lyon, France, 25–29 August 2013; Volume 13, pp. 2237–2241.
23. Rojas, R. The backpropagation algorithm. In *Neural Networks*; Springer: Berlin, Germany, 1996; pp. 149–182.
24. Wang, D.; Zhang, M.; Li, Z.; Li, J.; Fu, M.; Cui, Y.; Chen, X. Modulation format recognition and OSNR estimation using CNN-based deep learning. *IEEE Photonics Technol. Lett.* **2017**, *29*, 1667–1670. [[CrossRef](#)]
25. Liu, H.; Simonyan, K.; Vinyals, O.; Fernando, C.; Kavukcuoglu, K. Hierarchical representations for efficient architecture search. *arXiv* **2017**, arXiv:1711.00436.
26. Park, K.; Shin, D.; Chi, S. Variable Chromosome Genetic Algorithm for Structure Learning in Neural Networks to Imitate Human Brain. *Appl. Sci.* **2019**, *9*, 3176. [[CrossRef](#)]
27. Wei, X.-S.; Wu, J.; Cui, Q. Deep Learning for Fine-Grained Image Analysis: A Survey. *arXiv* **2019**, arXiv:1907.03069.
28. Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A.C.; Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in Information Systems Development*; Springer US: New York, NY, USA, 2015; pp. 2980–2988.
29. Pasa, L.; Testolin, A.; Sperduti, A. A HMM-based pre-training approach for sequential data. In Proceedings of the 22nd European Symposium on Artificial Neural Networks, Bruges, Belgium, 23–25 April 2014.
30. Gao, J.; Murphey, Y.L.; Zhu, H. Multivariate time series prediction of lane changing behavior using deep neural network. *Appl. Intell.* **2018**, *48*, 3523–3537. [[CrossRef](#)]
31. Matsushita, A.; Sunda, T.; Nakamura, Y.; Takano, W.; Hashimoto, J. Driving Assistance System and Driving Assistance Method. U.S. Patent US10161754B2, 25 December 2018.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).