Article

A Study on an Enhanced Autonomous Driving Simulation Model Based on Reinforcement Learning Using a Collision Prevention Model

Jong-Hoon Kim ¹, Jun-Ho Huh ², Se-Hoon Jung ^{3,*} and Chun-Bo Sim ^{4,*}

¹ Interdisciplinary Program in IT-Bio Convergence System (BK 21 Plus), Sunchon National University, Suncheon 57922, Korea; kjh1004aos@snu.ac.kr

² Department of Data Science, (National) Korea Maritime and Ocean University, Busan 49112, Korea; 72networks@pukyong.ac.kr or 72networks@kmou.ac.kr

³ School of Creative Convergence, Andong National University, Andong 36729, Korea

⁴ Interdisciplinary Program in IT-Bio Convergence System and School of ICT Convergence Engineering, Sunchon National University, Suncheon 57922, Korea

* Correspondence: jungsh@anu.ac.kr (S.-H.J.); cbsim@snu.ac.kr (C.-B.S.)

Abstract: This paper set out to revise and improve existing autonomous driving models using reinforcement learning, thus proposing a reinforced autonomous driving prediction model. The paper conducted training for a reinforcement learning model using DQN, a reinforcement learning algorithm. The main aim of this paper was to reduce the time spent on training and improve self-driving performance. Rewards for reinforcement learning agents were developed to mimic human driving behavior as much as possible. High rewards were given for greater distance travelled within lanes and higher speed. Negative rewards were given when a vehicle crossed into other lanes or had a collision. Performance evaluation was carried out in urban environments without pedestrians. The performance test results show that the model with the collision prevention model exhibited faster performance improvement within the same time compared to when the model was not applied. However, vulnerabilities to factors such as pedestrians and vehicles approaching from the side were not addressed, and the lack of stability in the definition of compensation functions and limitations with respect to the excessive use of memory were shown.

Keywords: autonomous driving; collision prevention; reinforcement learning

Citation: Kim, J.-H.; Huh, J.-H.; Jung, S.-H.; Sim, C.-B. A Study on an Enhanced Autonomous Driving Simulation Model Based on Reinforcement Learning Using a Collision Prevention Model. *Electronics* **2021**, *10*, 2271. <https://doi.org/10.3390/electronics10182271>

Academic Editor:

Teresa Orłowska-Kowalska

Received: 6 August 2021

Accepted: 11 September 2021

Published: 16 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Autonomous driving refers to a vehicle being able to recognize its surrounding driving environment and assess the environmental elements that affect its driving, such as risk factors to itself, until arriving at a destination. Interest and research have been dedicated towards autonomous driving for many years, and it is a field expected to undergo explosive growth. Current commercialized autonomous driving technologies remain at the level of performing low-level functions or are restricted to road environments where driving is possible. However, high-level artificial intelligence technologies have emerged, and have resulted in remarkable developments in recent years, driving research efforts to overcome these limitations of autonomous driving. Interest in autonomous driving, which has received a great deal of attention for a long time, has even increased in recent years as a result of its fast development and higher levels of commercialization.

In general, there are six stages of autonomous driving, based on the technological levels defined by the Society of Automotive Engineers (SAE) [1]. Stage 0 (Figure 1-(1)) means that the driver operates a vehicle fully. At Stage 1 (Figure 1-(2)), the system assists the driver in acceleration/deceleration or steering and helps him or her remain in their lane. At Stage 2 (Figure 1-(3)), the autonomous driving system assists the driver on highways and offers remote smart parking assistance, taking a step further from Stage 1. At Stage 3

(Figure 1-(4)), the system assesses situations and drives the vehicle. Under conditional autonomous driving, it slows down during congestion and performs highway driving and lane changing. Today, autonomous driving has been commercialized at Stage 3. The system asks drivers to operate manually during autonomous driving. Even though autonomous driving is taking place, drivers are asked to intervene. Moving to Stage 4 of advanced automation is inevitable, shifting from the current conditional automation that requires driver intervention to a higher level of autonomous driving technology. Under advanced autonomous driving at Stage 4 (Figure 1-(5)), the system performs autonomous driving on prescribed roads and under prescribed conditions without any need for driver interventions in the process. Even though it is not completely autonomous, as the system enables fully autonomous driving only on prescribed roads, it is autonomous driving in the true sense, as it requires no driver intervention. The last of the five steps (Figure 1-(6)) refers autonomous driving on all roads and in any situation. It requires high technological development in that driver intervention is unnecessary in all driving situations and the vehicle can be driven without a driver.

Figure 1. Outline of SAE autonomous levels.

There is an absolute need for image processing [2,3] and AI in order to realize completely autonomous driving [4–8]. These technologies comprise the core of autonomous driving, across all of its stages. World-class corporations are devoting active R&D effort in these areas. IT giants such as Google, Naver, and NVIDIA are developing their own autonomous driving systems [9–11]; in addition, Microsoft and Amazon are making huge investments in autonomous driving [12,13]. It is, of course, automobile manufacturers that are taking the lead in the development and commercialization of autonomous driving technologies. There is a range of corporations, regardless of field, that sees AI as the core of autonomous driving and that are conducting research into it [14].

In autonomous driving, there is a need to assess various situations on the basis of massive amounts of data. Deep learning can offer solutions for this. However, high-quality datasets are necessary that define the meaning of information clearly in order to solve problems with common deep learning algorithms. The circumstances make it difficult to collect data from real driving environments while granting clear meanings to data, raising the need to improve our capacity to process vehicle driving data.

It is not general practice to embody all functions of autonomous driving only on the basis of reinforcement learning. Research has been conducted to create local routes, predict collisions, and brake for autonomous driving vehicles on the basis of reinforcement learning, but most of this research has remained at the level of exploring optimal routes through reinforcement learning algorithms [15–27]. In previous studies, autonomous

driving models based on reinforcement learning have been employed to supplement the system's shortcomings to some degree. These autonomous driving models were able to train under many different environments and conduct driving training and testing in various environments. As only driving images collected from the front cameras were used, they required a comparatively smaller amount of hardware resources and less time. In addition to these advantages, the models improved the performance with respect to preventing collision situations, and reduced the time allocated to training by adding an algorithm that prevented unnecessary collisions in episodes of reinforcement learning and autonomous driving processes.

Previous studies have been dependent on small amounts of sensor data and have had issues related to possible performance deterioration under the influence of unpredictable road environment elements. As can be seen from Table 1, autonomous driving performance has been lower on city roads than on uncrowded roads and in mountainous environments. In an attempt to solve this issue on the basis of actual human driving methods using a reward function or segmenting driving control, the present study proposes an autonomous driving agent by applying a reinforcement learning algorithm and deep learning to automatically process driving data on the basis of the image analysis of an autonomous driving system, in order to embody stability and optimize agent actions using continuous data. Simulations were carried out using AirSim [14], a virtual simulation platform, in order to improve the performance of the autonomous driving system based on reinforcement learning algorithms that comprised the core of the present study: DQN (Deep-Q-Learning), which is less influenced by data preprocessing, and a deep learning algorithm for classifying preprocessed images at a low level, were combined to create a fusion model to efficiently solve old issues related to real-time learning of continuous data [28,29]. The aim of this study is to reduce learning time by creating a collision prevention model and applying it to reinforcement learning training, since learning time is one problem of reinforcement learning training. The study sets out to propose the optimization of an autonomous driving model on the basis of reinforcement learning.

Table 1. Collision conditions by driving environment.

Item	Pedestrian	Driving Environment
City Road 1	Yes	City Road
City Road 2	No	City Road
Mountain Road	No	Mountain Road

The present study consists of five chapters: Section 2 defines autonomous driving in terms of its basic theory, and reviews previous studies; Section 3 offers explanations regarding simulations aiming to improve autonomous driving performance, providing an overall overview of proposed simulations, and descriptions of the collection of driving data, the replay memory buffer, reward functions, and collision prevention algorithms; Section 4 compares the experimental results using the proposed simulations of performance improvement with the results of autonomous driving with no performance improvements on the basis of objective indicators, in order to assess them in performance; and Section 5 presents the conclusions.

2. Related Work

In a study by I. Kang [30], an autonomous driving system was proposed using a supervised learning approach based on the demonstrations of human drivers. The study performed artificial neural network learning with low-cost parts and proposed autonomous driving with low dependence on hardware. This autonomous driving model was able to perceive and avoid obstacles in response to driving images obtained from the front cameras rather than simply following the driving routes. An experiment was conducted using a server computer with a small embedded system and four GPUs, as well as a compact simulated vehicle in the real world. The combination of multiple sensors provided better

performance with respect to image classification than a single sensor, but the experimental results demonstrated that the classification performance of an image-based neural network with cameras played the most significant role in performance improvement in real environments. On this basis, the study carried out reinforcement learning training based on driving images from the front cameras.

S. Park et al. [31] used a convolutional neural network to adjust vehicle steering values. In this study, the steering values of an autonomous driving vehicle were controlled using the CNN inputs of images from a black box. The CNN structure included three layers, including convolutional and pooling layers, in accordance with a common CNN composition, and an algorithm was proposed with two fully connected layers. In this CNN, the steering angles were estimated as output values and were used geometrically with accurate vanishing point coordinates obtained from images. It is difficult to detect stable vanishing points due to the prediction errors of the steering angles resulting from various factors including the driving environment and lane disappearances, but the CNN obtained prediction errors of steering angles within an allowable scope and provided significant results.

X. Pan et al. [32] performed autonomous driving in reality using virtual environments and reinforcement learning. In this study, a virtual driving environment similar to real roads was created to perform reinforcement learning training, and this was then compared with results on real roads. Three major methods of reinforcement learning were used, including the standard A3C [33], the reinforcement learning method proposed in the study, and the random domains method. The study carried out autonomous driving training using reinforcement learning in a virtual environment by means of these three methods and compared them with respect to accuracy. The method proposed in the study recorded an accuracy of 43.4%, which was a better performance overall than training a reinforcement training agent in a virtual environment without any real data, which recorded an accuracy of 28.33%. The SuperVision (SV) method recorded the highest accuracy, at 53.6%, but it required more training sessions than the other models due to its structure, which means that it did not demonstrate a superior performance.

Abdur R. Fayjie et al. [34] performed autonomous driving in an urban environment using reinforcement learning. A model was established that applied autonomous driving and obstacle avoidance learning based on DQN to a simulation vehicle in an urban environment in order to assess its performance. Two types of sensor data were used as input data. An autonomous driving agent was proposed with driving images from cameras and distance data from obstacles from laser sensors attached to the front. In this study, a reinforcement learning algorithm was applied with DQN in order to develop a prototype of an autonomous driving vehicle. Performance evaluation in a virtual environment showed that the accumulation of rewards from over 100,000 episodes tended to increase over time. The range of increase decreased along with the time required. Performance evaluation was conducted on the basis of experiments in which 100,000 episodes were carried out in a virtual environment using the prototype. Compensation values were provided continuously to achieve autonomous driving training using DQN, which demonstrates that a DQN-based agent can be an effective method for autonomous driving, and motivated the selection of DQN as a reinforcement learning algorithm in the study. There were, of course, limitations to this prototype as well. According to the results of change in compensation, the compensation values gradually decreased toward the end of training, resulting in a decreasing pattern of increases in accumulated compensation values. This means that towards the end of the learning, the compensation values provided were decreased. The present study attempts to tackle this reduction issue with respect to compensation values.

In the study reported by Li et al. [35], a steering system is proposed based on the assessment of driving images using a “recognition module” and a “control module”, unlike other studies, which had employed an end-to-end method. Based on a multiple classification neural network, the “recognition module” was fed inputs of images from the driver’s viewpoint for the purpose of predicting driving direction. The “control module” made

control decisions on the basis of reinforcement learning. In the process of implementing this autonomous driving model, a prototype agent was proposed, along with reinforcement learning and deep learning. On the basis of the performance evaluation results in the virtual environment, a recognition module was used to predict lane characteristics in a stable and accurate manner, placing the vehicle within the predicted lanes. The control module generally demonstrated excellent performance in a variety of driving environments and represents a case in which a different AI algorithm was applied to the reinforcement learning training process.

In a study by H. Yi et al. [36] studying autonomous driving using DDPG instead of DQN with a reinforcement learning algorithm, a multi-agent approach was adopted with 16 agents used for learning simultaneously. Performance evaluation was conducted using the average compensation value obtained during the learning process. The initial values were negative, meaning that there was little learning. As the number of learning sessions increased, however, the compensation increased, nearing an average of two, indicating that autonomous driving was taking place in a relatively stable manner. However, compensation did not increase steadily. Furthermore, there were many intermittent sharp declines in compensation values, with periods of negative compensation values despite sufficient learning having taken place, which was attributed to improper compensation value decisions for brake values. The present study addresses this issue by means of a collision prevention model.

In a study by Yu A et al. [37] that controlled an autonomous driving vehicle in a virtual environment, an experiment was conducted that applied compensation values for various parameters in a reinforcement learning training process. The experiment induced gradation updating rules and other hyperparameters for the model by creating an autonomous driving agent on the basis of Double A-Learning and CNN. In this experiment, the changes in average compensation values were examined with respect to various learning rates. The experiment performed autonomous driving in urban areas with no other vehicles and was conducted many times using different learning rates. Changes to compensation values show that the compensation values granted tended to decrease during training when the learning rate was 0.001. This meant that relatively high learning rates did not help the compensation function. When the learning rate was set at the lowest level of 0.00005, the training speed was too slow to produce satisfying results or offer significant benefits with respect to compensation values. The same results were obtained at a learning rate of 0.0001. A learning rate of 0.005 demonstrated good performance; however, the compensation value decreased as the learning progressed. The compensation values continued to increase when the learning rate was set at 0.0025, providing the best outcomes in terms of compensation values. These results were consulted for adjusting the hyperparameters of the autonomous driving model proposed in the present study.

3. Proposed Simulation for the Improvement of Autonomous Driving Performance

3.1. Overall Structure of the Proposed Simulation

This paper develops a driving and training environment for autonomous models using the AirSim simulator to support autonomous driving in vehicles. AirSim is a virtual simulator made by Microsoft in the United States. An environment was established using AirSim for training using reinforcement learning based on roads. The steering wheel, accelerator, and brake pedals were controlled during driving, which were then used to control driving in a virtual environment developed using the Python Library in AirSim. For the training of the reinforcement learning agent, the input data included vehicle driving data, distance from obstacles based on LiDAR sensors, and vehicle driving videos taken from the front camera. Figure 2 shows the overall flow chart of the proposed simulation.

Figure 2. Overall flow chart of the proposed simulation.

A reinforcement learning agent trains the autonomous driving model on the basis of the obtained driving and sensor data using the AirSim simulator. Driving and sensor data are entered as “behavior” and “state” parameters, respectively, for the purpose of reinforcement learning. The parameters generated during this process are entered into the Deep-Q-Learning algorithm to determine ideal driving control values and to perform optimization.

The driving data of the vehicle and the images obtained during collisions are used as input data to train the collision avoidance model. A model was designed and updated to increase the brake values, and thereby prevent collision, when driving data are received that possess a high degree of similarity to those obtained during collision situations. This prevents unnecessary collisions being included in the reinforcement learning training process. One of the advantages of this method is the savings with respect to time and money during the reinforcement learning training process, as a result of the inclusion of fewer redundant states during the training process.

Once training has taken place for these two models using a reinforcement learning agent, the vehicle control method is optimized accordingly. Following the customary procedure, a normal vehicle drives in accordance with a reinforcement learning-based control algorithm. When images with a high degree of similarity to a collision situation are received via the front camera, the brake value is increased in the reinforcement learning training process.

3.2. Simulation Configuration

Training and testing for the vehicle driving simulator were carried out in a virtual environment using the AirSim simulator by Microsoft. Since the simulator had remarkable compatibility with the Unreal Engine, environments appropriate for autonomous driv-

ing was developed using Unreal Editor. Three autonomous driving environments were designed in order to assess autonomous driving performance under various conditions. These included an urban environment with pedestrians, an urban environment with no pedestrians, and a mountainous road environment for training and performance testing. Only image data from the front cameras was used as input data, with the aim of minimizing the necessary hardware resources and reducing the volume of input data. A reinforcement learning agent requires driving data that include information on vehicle collision, distance from lane markings, and speed in each episode of the reinforcement learning training process, and these data were collected using AirSim simulator. Figure 3 shows the training environment using AirSim.

Figure 3. The training environment using AirSim (city road and mountain road). (a) city road; (b) mountain road.

3.3. Collection of Driving Data

The autonomous driving method proposed in this paper only uses as input data the images obtained from the front cameras while driving and the current driving data of the vehicle. The front images consist of first-person-view images that are collected through the API of the AirSim simulator. The front images and driving data underwent a preprocessing stage and were used in the learning model, which was based on a convolutional neural network.

In the preprocessing process of the front driving images, images with dimensions of 128×128 pixels were used, which were cut from front images with dimensions of 144×256 pixels. Figure 4 shows an example. Forward driving images were not used in their original collection states, but cut out during the preprocessing process, as there were hardly any elements influencing the outcomes of vehicle driving that were found above the vehicle.

Figure 4. The training environment using AirSim (city road and mountain road).

In some cases, a flying object or a falling object from a building might have a negative impact on driving and constitute a risk element, but such cases are exceptional. The training of such special situations is not significant in a training process for reinforcement learning that is repeated many times. Thus, these situations were excluded from the process in this study. The amount of computation can be reduced, thus enhancing the training efficiency, by removing unnecessary information from the images, such as objects above a vehicle, and presenting track angles in the images. CNN results obtained on the basis of the characteristics of the input images are delivered as the state during reinforcement learning. The agent takes action and learns on the basis of state and reward. The action space of an agent accommodates eight discontinuous actions, which include driving straight, turning left at 45 degrees, turning right at 45 degrees, turning left at 20 degrees, turning right at 20 degrees, and stopping. Road environments were taken into consideration, as well as the driving directions of vehicles. Driving data—including vehicle speed, accelerator values, brake values, steering values, and driving outcomes—were collected using replay memory, as described below.

3.4. Replay Memory Buffer

Replay memory was used to implement the experience replay in the DQN algorithm. In DQN, replay memory saves the actions of an autonomous driving vehicle on the basis of reinforcement learning and their states; thus, it reduces correlations among data. The performance improvement model proposed in this study does not save new values in a replay memory update when actions were taken in the same state as before in response to an image in the actor network.

The state, action, reward, and next state parameters— S , A , R and s' , respectively—were updated in the memory for each episode in the reinforcement learning training process. Here, driving outcome data were also saved in the same file. An index was added to the driving data to be used as the file name for the driving outcome data, which were applied in a collision prevention algorithm designed to improve the performance of the reinforcement learning model. Driving data were saved in memory in multiples every 0.95 s. When additional data could not be saved in the current memory buffer due to prolonged training, the earliest driving experience data in the memory buffer were saved in the file. The driving data saved in the file were then deleted from the memory buffer to make room for new data. When updating the CNN neural network with the driving images and data as inputs, the driving data saved in the replay memory were extracted with a degree of randomization and used to calculate the driving reward function. When certain multiples were extracted at random from the replay memory and used in a CNN update, the continuous relationship among the input data was reduced to prevent a local minimum occurring. Since the old driving data were saved during learning, they can be used again for the purpose of training under the same or similar driving situations. DQN-based reinforcement learning has the limitation of being dependent on episode order. When experience replay is not used, the input data of the CNN are placed in order of occurrence, resulting in a local minimum, and making it difficult to ensure efficient learning.

3.5. Design of the Reward Function

A reward function in reinforcement learning is determined on the basis of the current driving states and the outcomes of the vehicle. Equation (1) presents the reward function to be granted to the agent proposed in this study. During a normal driving state, positive reward values are given with a prescribed reward function. When difficulties arise in driving the vehicle due to collision, lane departure, and stopping, F , representing flag bits, is set to a value of 1, with negative reward values granted appropriate to the situation before moving to a new episode.

$$R = F_1 \cdot A1(\alpha + \beta) - F_2 \cdot A2(-0.5t) - F_3 \cdot A3(-3) + F_4 \cdot A4(-3) \quad (1)$$

A1 refers to regular driving; A2 to a collision; A3 to lane departure; and A4 to car stopping. Under regular conditions, the compensation values are determined according to the distance from the lane (α) and vehicle speed (β). The distances are determined between the center of the vehicle and the outer edge of the lane. For the compensation values with respect to lanes, higher compensation values were given corresponding to greater distances from the center of the lane. Basically, the vehicle should never cross the central line. When the threshold value for distance from the center of the lane is reached, the vehicle is considered to have a collision. The distance between two lanes was obtained using a virtual simulation. $Dist_1$ denotes the distance between the center of the vehicle and the center of the lane, and $Dist_2$ denotes the distance between the center of the vehicle and the opposite lane. The distance between two lanes was normalized to within a range of 0~2 for $Dist_1 + Dist_2$.

$$\beta = 2 - |Dist_1 - Dist_2| \quad (2)$$

Partial compensation values for lanes were determined as shown in Equation (2). When the distance between the vehicle and two lanes is the same, it is standing in the middle of the road, and receives high compensations with values within the range 0.1~2. In short, the compensation values for lanes are determined on the basis of convex combinations of lanes and vehicles. Higher compensation values are given at higher vehicle speed. The initial vehicle speed was in the range of 5–13 m/s, meaning that the vehicle is never driving too fast according to urban standards, even when running at maximum speed. In addition, the minimum and maximum speed were determined and normalized to within a range of 0~1 to determine partial compensation values. Added to these are compensation values (α) determined on the basis of the distance between a vehicle and a lane. Normalized values of 0~1 are added depending on vehicle speed. When the three compensation values are simply added together, however, the characteristics of the individual elements become dull. Moreover, the probability of collision increases when vehicles quickly move closer to another lane. This problem was solved by limiting the compensation values under normal driving to a range of 0.1~3, with compensation values increasing with speed only at a distance of more than 0.4 m from both neighboring lanes. When the distance between the vehicle and a collision body becomes small while driving, -0.5 is added in order to reduce the compensation values. The compensation value is determined as (compensation value under normal driving $-0.5t$) at each time step of t seconds (0.95 s). When a vehicle touches the lane boundary, departs from a lane, or stops for three seconds or longer, a compensation value of -3 is added before moving to the next episode.

In a CNN structure, reward values are determined as follows: the images from the front cameras of the autonomous driving vehicle are altered and used as input data in the input layer in the way explained above. The hidden layer consisted of three CNN layers and flatten layers. Convolutional layers of [8, 8, 16], [4, 4, 32], and [3, 3, 32] are used for each of the three CNN layers, respectively, and ReLu functions are used for active functions. After altering the data in one dimension with the flatten layers, training is carried out to determine six behaviors through the dense layers. A neural network is trained to receive six behaviors as prescribed compensations. The network uses a mini batch of 32 training data at a learning rate of 0.00025. The depreciation rate, which is a parameter required for the reinforcement learning process, was set at 0.95, and epsilon decay was set at 0.1 to adjust exploratory fluidity. Figure 5 shows the composition of the proposed CNN network.

Figure 5. CNN architecture.

3.6. Design of the Collision Prevention Algorithm

In reinforcement learning, it is necessary for the vehicle repeat driving episodes. It is difficult to expect efficient training when implementing repeated training sessions as a result of the unnecessary repetition of collision situations. In this study, an algorithm to prevent front collisions was designed and applied to the training stages of the reinforcement learning episodes in order to increase the efficiency of the reinforcement learning training process. An algorithm for collision prevention predicts possible collision situations on the basis of images obtained from the front cameras as well as the driving data from collision situations in previous reinforcement learning episodes.

Algorithm 1 presents the pseudo-code for collision prevention and braking. Data are collected regarding collision situations during the reinforcement learning training process. Images at the time of collision and the corresponding driving data, saved in the replay memory, are used. The previously collected front images and corresponding driving data are used for the time of collision during an episode.

Algorithm 1. Collision Prevention Model training

Input: Img_{coll} , $D_{coll} = (Action_{accel}, Action_{brake}, Action_{steering})$

Output: Collision List

Collision time step t_{coll} in Img_{coll}

for $i = 1$ to 50 **do**

$Img_{coll*} = Img_{coll} - Img_{collision}(Action_{brake} \geq 0.3)$

$Img_{coll*}(\forall t_{coll} - 5 \sim) \leftarrow$ collision label

$Img_{coll*}(\forall \sim t_{coll} - 5) \leftarrow$ collision Warning label

end for

Multiple Classification Neural Networks model π

Train classifier: π on Img_{RL}

Return Collision List

- Img_{coll} : 10 timestep images before collision
 - D_{coll} : Driving data at Img_{coll}
-

Images for timestep t_{coll} , including ten frames before the time of collision, are saved as a set. Img_{coll} , an image set of collected collision situations, represents ten image sets and includes a driving time of 8~9 s right before a collision. Driving data D_{coll} are collected at the same time as the image set being created. D_{coll} includes the following driving data for the image sets: $Action_{accel}$, $Action_{brake}$, and $Action_{steering}$. Data filtering begins when enough image sets and driving data are collected for 50 collisions. When $Action_{brake}$ for braking right before collision is higher than the threshold, a collision occurs despite braking, meaning that the data are not fit input data for the collision prevention model. Img_{coll*} is created from the filtered data after the exclusion of such data. Image labeling is carried out for Img_{coll*} after training preparation. Images at 5t before the time of collision are very close to the time of collision and have a high degree of similarity to the collision, and are labeled as "collision". The remaining images in which a collision cannot be avoided are labeled as "caution for collision". Img_{coll*} , the image set of 50 collision situations for which labeling has been completed, is entered into a multiple classification neural network in order to create the classification model π . Img_{RL} , the driving images obtained during a reinforcement learning episode, is classified according to the classification model in order to tune the braking values on the basis of the collision list results. If the classification results are [0, 1], indicating a collision, the braking values are tuned to maximum. If they were [1, 0], indicating caution for a collision, the braking values are increased gradually in order to increase the time until the collision and secure the action space to avoid the collision. When the results are [0, 0], indicating no risk, the braking values are not tuned on the basis of the collision prevention model under the normal state.

Labeled images are used as input data in the CNN-based learning model to classify collisions. The CNN model uses the Tensorflow library, and the activation function uses Softmax for multiple classification. The loss function uses a CEE (Cross Entropy Error), and the optimizer uses Adam, a gradient descent optimizer. The collision prevention model conducts training using the input of labeled images, and the weights and bias values of the learning model are saved in file format. The resulting collision judgment model is applied during the episode stage of reinforcement learning. The driving images continuously collected from the front cameras during episode training are extracted and used as input data in the collision prevention model. The input images produces outcomes of “collision” and “caution for collision”. When images are assessed as belonging to the “caution for collision” class, braking values are increased to cause the vehicle to decelerate. Even with deceleration, the vehicle moves forward some distance due to inertial force, during which process the collection of image data continues, along with the prediction of collisions. The agent revises the steering values of the vehicle even at the decelerated speed, during which process the collected driving images may change, decreasing or increasing the collision prediction values. In the case of a decrease in these values, the braking values are reduced and the accelerator values are increased, causing the vehicle the drive at a higher speed again. In the case of the “collision” class following “caution for collision”, the reward values for a collision are granted on the basis of a judgment that it is a collision-like situation, despite there being no actual collision, and the episode is ended and the next one is started. In the reinforcement learning training process, the training of the collision prevention algorithm continues simultaneously. The driving data and images of saved collisions are extracted every 50 collisions to update the collision prevention algorithm. Figure 6 shows the flow chart of collision prevention and braking.

Figure 6. Flowchart of training for episodes of reinforcement learning.

4. Simulation Evaluation and Result

4.1. Simulation Evaluation Environment

The training and testing environments for the reinforcement learning training process were created using the Unreal Engine by Epic Games. There were three types of driving environment: urban environments with pedestrians, urban environments without pedestrians, and mountainous environments without pedestrians. Each of these environments had approximately 50,000 episode training sessions. Table 2 shows the learning and performance evaluation environments of the implemented autonomous driving model.

Table 2. Environments for autonomous driving model learning.

Parameter	Value
Car speed	5~13 m/s
Radius of safety (collision)	2 m
Lane violation recognition distance	0.2 m
Data collection and control time (ΔT)	0.95 s
Time to recognize a vehicle stop as a collision	1 s
Handle steering value	$\pm 45^\circ$

Learning happened on the basis of a total of 50,000 repeated simulations. Training sessions with an episode time of less than 5 s were not included in the training process. The existing steering values included five stages, and these were expanded to seven in the present study to prevent the vehicle from shaking as a result of extreme changes in steering values.

For the testing of the implemented autonomous driving model, environments were created that were similar to the three training environments. The road compositions and the orders were different from those in the training environments, while retaining the objects on the roads, the lighting, and the pedestrians in order to check for overfitting of the training environments. Each of the three testing environments had two maps, each of which had 8~11 sections for testing. Ten tests were conducted on 20 roads in each environment, resulting in 200 tests in total.

4.2. Simulation Results

Tests were conducted using a simulation environment. Figure 7 shows autonomous driving outcome screens obtained during the simulations. Tests were conducted in three driving environments. Vehicle driving was disrupted to the following extents due to the termination conditions of the simulation episodes. In City Road 1 with pedestrians, a collision or lane departure occurred in 9% of the 200 tests.

Figure 7. Three stop conditions of simulation episodes. (a) Too close to the lane. (b) Collision occurred (c) Stop the car.

In the City Road 1 environment, there may be pedestrians crossing the road ahead of the vehicle, or other vehicles approaching from the sides, such as at an intersection. The implemented autonomous driving model used only the front cameras, so it was vulnerable to these elements. In the reinforcement learning training process, there were more pedestrians not crossing the road than there were crossing it, meaning that it is difficult to conclude whether or not pedestrians had a significant effect on the training results. On city roads without pedestrians, there are no objects approaching the driving vehicle, as there were in the previous case, meaning that there were no collisions or lane departures, or that the percentages were low.

On mountain roads, however, 7.5% of vehicles stopped as a result of shadows cast on the road by trees or high stone walls in the mountainous environments not being recognized in the driving images. The actual cases of collision or stopping indicate that it was difficult to distinguish roads due to a shadow.

The driving results in Table 3 demonstrate that the proposed autonomous driving model enabled very stable driving in environments with no approaching objects or in the first lane, even though unstable driving performance was demonstrated due to collisions with objects in driving environments with objects approaching the vehicle. These data were used to analyze which elements were factors in performance changes in the proposed autonomous driving model in road environments. City Road 2 was chosen as a driving environment to ensure efficient learning for a reinforcement learning agent. The performance evaluation of the reinforcement learning process was carried out using City Road 2.

Table 3. Comparison of collisions by driving environment between the existing method and the proposed algorithm.

Item	Pedestrian	Driving Environment	Rate of Collision and Lane Departure	Rate of Car Stopping
City Road 1	O	City Road 1	9%	11%
City Road 2	X	City Road 2	0%	0%
Mountain Road	X	Mountain Road	4%	7.5%

4.3. Evaluation of Autonomus Driving Efficiency Using Reinforcement Learning

Figure 8 shows the results obtained over 100 cases of time until collision during training in an urban environment with no pedestrians using the collision prevention algorithm proposed in this study. The driving time increased dramatically at certain points, reflecting the points at which the collision prevention algorithm was updated. The time until collision increased with every application of the algorithm during training, leading to an increase in the training time for each episode.

Figure 8. Time until collision results when using the collision prevention algorithm.

Meanwhile, Figure 9 shows 100 units of autonomous driving time occurring in episodes during the reinforcement learning training process when applying the collision prevention algorithm. This summarizes the training process from Session 1 to Session 10,000, showing the results as units of 100 in order to better visualize performance. A comparison is drawn between models with the collision prevention algorithm applied and models in which it was not applied. The increase in the normal driving time in each episode of reinforcement learning indicates that the autonomous driving model was trained well through the reinforcement learning training process.

Figure 9. The results of models with and without the application of the collision protection algorithm. (Mean and standard deviation of 5 training simulations).

The driving time for each episode without the application of the collision prevention algorithm was examined. In the early phase, training efficiency was good, leading to a gradual increase of driving time. After approximately 1000 episodes, however, the driving time of the vehicle exhibited only a small increase. Even with the addition of training, learning efficiency dropped. Driving time increased as a result of continuous training in repeated episodes, but the increase was not as large as that during the early phase of training. The models in which the collision prevention algorithm was applied did not exhibit a drop in training efficiency as in the case of the models in which the collision prevention algorithm was not applied. During the early phase of training, there were similar patterns between them. After approximately 2000 training sessions, the gap in driving time began to widen between them. The probability of avoiding collision increased as a result of the implementation of the collision prevention algorithm, leading to an increase in the maximum driving time. During the early phase of episode learning, there were still various collision situations that had received no training, meaning that the collision prevention algorithm also received little training with similar patterns. As the episodes were repeated, however, the algorithm learned various collision environments, leading to an increase in maximum driving time and a gradual increase of total driving time. The models in which the collision prevention algorithm was applied exhibited a greater increase in driving time than those in which the collision prevention algorithm was not applied; this occurred because the point at which the agent arrived grew more distant in correspondence with the increased driving time. New potential collision situations would occur at distant points of arrival, and the collision prevention algorithm was able to learn each new situation. The driving time increased for each new learning, and the repetition of this process resulted in a relatively larger increase or decrease in driving time.

Figure 10 shows a graph of the average compensation values received by the agent after 10,000 episodes of training. Observation and measurement took place every ten episodes. Unlike previous studies [2,3], a certain range of 0~3 was shown for the average compensation values. This is because existing reinforcement learning-based autonomous driving models assign compensation values on the basis of driving distance or distance from destination; thus, the increase in compensation values is reduced toward the end of training. The autonomous driving model proposed in the study, however, assigns compensation values on the basis of real-time driving videos and data entered according to time steps, and enables the learning of certain compensation values without taking into consideration continuous situations with respect to the compensation values.

During the early phase of training, the average compensation values were closer to -3 , as a result of there being more collisions during the driving time. Once training passed a certain point, however, the average compensation values moved away from -3 , corresponding to increased driving time, and there being more regular driving than there were collisions.

Figure 10. The average reward for 10,000 episodes.

We also assessed the changes in the reward values with respect to autonomous driving models implemented using the existing DQN and clash algorithms in the early stage of learning in order to set optimized threshold values for reward values for simulations. In this way, we were able to assess the efficiency of stable reward values. Figure 11 presents a performance comparison between the results obtained using a DQN-based autonomous driving model [38], and the model proposed in this paper. As in our study, the previous study proposed autonomous driving using an improved DQN model. The previous study also conducted an experiment in environments under the same conditions as used for our autonomous driving model, including acceleration, deceleration, braking, and steering angle control in the action space. Ref. [38] carried out training and performance evaluation for the autonomous driving model in an urban environment with pedestrians, and set the number of steps for controlling action per episode to 1000. For performance evaluation, the study measured reward values for a total of 200 episodes. In this study, we created the same environment as that described in [38] and carried out performance evaluation for the purpose of comparison. The driving environment was an urban environment with pedestrians, and the number of steps per episode was set to 1000. Since the clash prevention model had no significant impact on the early stage of learning, we applied a model that carried out learning in advance in order to compare performance in the early stage of

learning. Five rounds of performance evaluation were carried out, and the stability of the reward values for the model were examined on the basis of mean and standard deviation. As seen in Figure 11, Ref. [38] exhibited a gradual increase of rewards as the agent learned to drive in the early stage of training. There was, however, a point at which the reward level plunged, when the agent learned to move (after 25 episodes). In the study of autonomous driving based on DQN, there was a huge range of reward values: from -800 to 315 . These issues relating to huge gaps and plunge were the result of repeating clashes in the early stage. The model proposed in this study, however, maintained a consistent and stable range of reward values of -3.0 – 0 and displayed no reward plunge due to unnecessary clashes in the early stage of learning, thus superseding [38] as a result of its issues regarding optimal standards of reward values throughout the learning sections.

Figure 11. The analysis of average reward values for behavioral control by episode.

5. Conclusions

An autonomous driving system rapidly assesses road situations in real time on the basis of numerous sensors and camera images. It should be able to recognize rapidly changing road situations by processing the collected data quickly and making a quick judgments with respect to actions such as acceleration, deceleration, stopping, and avoidance. The essence of data processing in autonomous driving is addressing the continuous values of action outcomes on the basis of the continuous values of input images or driving data. In this paper, a reinforcement learning algorithm was applied to build an autonomous driving system for vehicles. An improved autonomous driving model based on reinforcement learning was also proposed based on simulations performed in the driving simulator AirSim.

In this paper, an autonomous driving model was proposed using the driving data of autonomous driving vehicles applied to DQN. Replay memory buffers were employed to increase training efficiency, as well as to improve the reward functions. The simulations were based on CNN, which is usually used to process images or image data, with images collected from the front cameras of autonomous driving vehicles as the input data. It was found that training based on reinforcement learning alone would be prolonged as a result of the unnecessary repetition of collision situations in the training process, thus presenting the issue of lower efficiency. Trying to solve this, the present study employed a collision prevention algorithm and applied it to the reinforcement learning training process. The purpose of this algorithm was to reduce the time spent on the old reinforcement learning training process and to improve autonomous driving functions.

The autonomous driving model consisting of DQN and CNN was similar to the episode environment, but an environment that was not exactly the same was built using the Unreal Engine to conduct test driving and test the proposed model in performance.

This autonomous driving model had outstanding collision performance on roads with no pedestrians or in mountainous environments. Collisions occurred during the test when driving images could not be collected normally from cameras due to shadows. There were abnormal vehicle stops on mountain roads. The autonomous driving model that was implemented based on the driving simulation results showed unstable driving performance, with collision objects approaching the vehicle. In environments with no approaching objects or on first lanes, however, the model showed very stable driving performance. Moreover, the model improved in driving performance with fewer training sessions, and was thus demonstrated to be a lighter model in the given environments, requiring fewer resources with respect to hardware and time than existing models.

A future study is planned with the aim of narrowing the scope of driving errors by means of sensor fusion. The preprocessing of images collected only from the front cameras presents clear limitations in addressing the issue whereby vehicles are not able to recognize road environments due to shadows. It is thus necessary to use other sensors to assess road environments. The virtual simulator AirSim supports the functions of remote LiDAR sensors, which examine objects within a certain radius by laser, analyze the reflected light, and measure the distance accordingly. In a future study, LiDAR sensors will be used and activated to detect elements approaching or colliding to the side and rear, as well as in front of the vehicle, and take viable actions so that the vehicle will be able to avoid collision. The dependence on LiDAR sensors will increase when vehicles drive in dark driving environments, as it will be necessary to recognize the collision prediction elements detected by the LiDAR sensors in the front and carry out actions such as changing the stopping and steering values. This study then attempted to improve the performance of behavioral control by enhancing the reward functions. The reward function of the autonomous driving model proposed in this study has the limitation of not being able to drive like a human being that is distant from their destination. A recent study by Churamani, N. et al. [39] addressed the topic of learning emotions and human compensations. The study extracted the characteristics of photos of human facial expressions using CNN and used these as the input for SOM (self-organizing map) training, capturing a cluster of emotion information. The BMUs (best matching units) of SOM were entered into an MLP (multilayer perceptron) along with feasible behavior (facial expression) to generate compensation values. Then, information was sent by the user regarding differences between the performed task and the intended one to an agent, who in turn adjusted the BMUs in the direction necessary to produce a shorter distance on the basis of this information. The core of this study was the approximation of an optimal reward function with the aim of estimating the user's reward for each piece of emotional expression data. This could be used to improve the reward function, which is a limitation of the present study, bringing it closer to a real person's driving intention. In this study, another issue is related to the excessive memory sizes required in the training process. This is critical, as it clearly lowers the training efficiency. This can be solved by reducing memory use. In a study by Cruz, F. et al. [40] on a robotic system based on reinforcement learning, an introspection-based method of reinforcement learning was proposed rather than a memory based one. In the study, the probabilities of training success were calculated directly using Q-values on the basis of internal numerical conversion, and this method was reported to use fewer memory resources and provide more diverse alternatives for reinforcement learning issues than memory-based reinforcement learning operations. These will be used to ameliorate the dependency on memorize size present in this study. In addition, we will conduct further studies using human feedback for learning reinforcement [41], and we will conduct further studies to solve issues related to learning time and memory consumption [42].

Author Contributions: Conceptualization, J.-H.K., S.-H.J. and C.-B.S.; data curation, J.-H.K. and J.-H.H.; formal analysis, J.-H.K., S.-H.J. and C.-B.S.; funding acquisition, C.-B.S.; investigation, J.-H.K., S.-H.J. and C.-B.S.; methodology, J.-H.K., J.-H.H., S.-H.J. and C.-B.S.; project administration, J.-H.H., S.-H.J. and C.-B.S.; resources, J.-H.K. and C.-B.S.; software, J.-H.K., S.-H.J. and C.-B.S.; supervision, J.-H.H., S.-H.J. and C.-B.S.; validation, S.-H.J. and C.-B.S.; visualization, J.-H.K. and S.-H.J.; writing—original draft, J.-H.K., J.-H.H., S.-H.J. and C.-B.S.; writing—review and editing, S.-H.J. and C.-B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2020-0-01489) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation) and this research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2020R111A3054843) and this research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2021R111A3050843).

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

References

1. SAE. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*; SAE J3016 Technology Report; SAE: Warrendale, PA, USA, 2016.
2. Kim, J.; Kim, J.; Park, J.; Jung, S.; Sim, C. Autonomous driving model using reinforcement learning. In Proceedings of the Fall Conference of the Korea Multimedia Society, Yeo-su, Korea, 19–20 November 2020; pp. 143–146.
3. Szőke, L.; Aradi, S.; Bécsi, T.; Gáspár, P. Driving on Highway by Using Reinforcement Learning with CNN and LSTM Networks. In Proceedings of the 2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES), Reykjavík, Iceland, 8–10 July 2020.
4. Kritika, R.; Kaur, P. Simulation based vehicle movement tracking using Kalman Filter algorithm for Autonomous vehicles. In Proceedings of the 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 28–30 April 2021.
5. Milani, S.; Khayyam, H.; Marzbani, H.; Melek, W.; Azad, N.L.; Jazar, R.N. Smart autodriver algorithm for real-time autonomous vehicle trajectory control. *IEEE Trans. Intell. Transp. Syst.* **2020**, 1–12. [\[CrossRef\]](#)
6. Lio, D.; Dona, R.; Papini, G.P.R.; Biral, F.; Svensson, H. A mental simulation approach for learning neural network predictive control (in self-driving cars). *IEEE Access* **2020**, *8*, 192041–192064. [\[CrossRef\]](#)
7. Jelena, K.; Jovičić, N.; Drndarević, V. An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms. *Sensors* **2019**, *19*, 2064.
8. Hu, B.; Li, J.; Yang, J.; Bai, H.; Li, S.; Sun, Y.; Yang, X. Reinforcement learning approach to design practical adaptive control for a smallscale intelligent vehicle. *Symmetry* **2019**, *11*, 1139. [\[CrossRef\]](#)
9. Bansal, M.; Krizhevsky, A.; Ogale, A. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. *arXiv* **2018**, arXiv:1812.03079.
10. Jeong, S.; Kim, J.; Kim, S.; Min, J. End-to-end Learning of Image based Lane-Change Decision. *IEEE Intell. Veh. Symp.* **2017**, *5*, 11–14.
11. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Zieba, K. End to End Learning for Self-Driving Cars. *arXiv* **2016**, arXiv:1604.07316.
12. Valdes-Dapena, P. Microsoft joins in a new \$2 billion investment in GM's self-driving car company. *CNN Business*, 19 January 2021.
13. Balaji, B.; Mallya, S.; Genc, S.; Gupta, S.; Dirac, L.; Khare, V.; Karuppasamy, D. DeepRacer: Educational Autonomous Racing Platform for Experimentation with Sim2Real Reinforcement Learning. *arXiv* **2019**, arXiv:1911.01562.
14. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. *arXiv* **2017**, arXiv:1705.05065.
15. Kim, M.; Choi, G.; Oh, B.; Yang, J. Local Path Generation Method for Unmanned Autonomous Vehicles Using Reinforcement Learning. *KIPS Trans. Softw. Data Eng.* **2009**, *3*, 369–374. [\[CrossRef\]](#)
16. Aradi, S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, 1–20. [\[CrossRef\]](#)
17. Bálint, K.; Hegedüs, F.; Bécsi, T. Design of a Reinforcement Learning-Based Lane Keeping Planning Agent for Automated Vehicles. *Appl. Sci.* **2020**, *10*, 7171.
18. Huang, C.; Huang, H.; Hang, P.; Gao, H.; Wu, J.; Huang, Z.; Lv, C. Personalized Trajectory Planning and Control of Lane-Change Maneuvers for Autonomous Driving. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5511–5523. [\[CrossRef\]](#)
19. Szoke, L.; Aradi, S.; Bécsi, T.; Gaspar, P. Vehicle Control in Highway Traffic by Using Reinforcement Learning and Microscopic Traffic Simulation. In Proceedings of the 2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 17–19 September 2020.

20. Matthias, N.; Lobig, T.; Köster, F. Case Study on Gap Selection for Automated Vehicles Based on Deep Q-Learning. In Proceedings of the 2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST), Dalian, China, 28–30 June 2021.
21. Masmoudi, M.; Frijj, H.; Ghazzai, H.; Massoud, Y. A Reinforcement Learning Framework for Video Frame-based Autonomous Car-following. *IEEE Open J. Intell. Transp. Syst.* **2021**, *2*, 111–127. [\[CrossRef\]](#)
22. Ye, F.; Zhang, S.; Wang, P.; Chan, C.-Y. A Survey of Deep Reinforcement Learning Algorithms for Motion Planning and Control of Autonomous Vehicles. *arXiv* **2021**, arXiv:2105.14218.
23. Zhou, M.; Yu, Y.; Qu, X. Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 433–443. [\[CrossRef\]](#)
24. Gao, H.; Shi, G.; Xie, G.; Cheng, B. Car-following method based on inverse reinforcement learning for autonomous vehicle decisionmaking. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1–11. [\[CrossRef\]](#)
25. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A.A.; Yogamani, S.; Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–18. [\[CrossRef\]](#)
26. Osiński, B.; Jakubowski, A.; Zięcina, P.; Miłoś, P.; Galias, C.; Homoceanu, S.; Michalewski, H. Simulation-based reinforcement learning for real-world autonomous driving. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020.
27. Zhu, M.; Wang, Y.; Pu, Z.; Hu, J.; Wang, X.; Ke, R. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transp. Res. Part C Emerg. Technol.* **2020**, *117*, 102662. [\[CrossRef\]](#)
28. García Cuenca, L.; Puertas, E.; Fernandez Andres, J.; Aliane, N. Autonomous driving in roundabout maneuvers using reinforcement learning with Q-learning. *Electronics* **2019**, *8*, 1536. [\[CrossRef\]](#)
29. Wang, X.; Wu, C.; Xue, J.; Chen, Z. A method of personalized driving decision for smart car based on deep reinforcement learning. *Information* **2020**, *11*, 295. [\[CrossRef\]](#)
30. Kang, I. Obstacle Avoidance and Autonomous Driving by Embedded Deep Neural Networks. Master's Thesis, Hanyang University, Seoul, Korea, 2020.
31. Park, S.; Hwang, K.; Park, H.; Choi, Y.; Park, J. Application of CNN for steering control of autonomous vehicle. In Proceedings of the Spring Conference of the Korea Institute of Information and Communication Sciences, Yeo-su, Korea, 20–22 May 2018; pp. 468–469.
32. Pan, X.; You, Y.; Wang, Z.; Lu, G. Virtual to Real Reinforcement Learning for Autonomous Driving. *arXiv* **2017**, arXiv:1704.03952.
33. Mirchevska, B.; Blum, M.; Louis, L.; Boedecker, J.; Werling, M. Reinforcement learning for autonomous maneuvering in highway scenarios. *Workshop Driv. Assist. Syst. Auton. Driv.* **2017**, 32–41.
34. Fayjie, A.R.; Hossain, S.; Oualid, D.; Lee, D. Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment. In Proceedings of the 2018 15th International Conference on Ubiquitous Robots (UR), Jeju, Korea, 28 June–1 July 2018; pp. 896–901.
35. Li, D.; Zhao, D.; Zhang, Q.; Chen, Y. Reinforcement learning and deep learning based lateral control for autonomous driving. *IEEE Comput. Intell. Mag.* **2019**, *14*, 83–98. [\[CrossRef\]](#)
36. Yi, H.; Park, E.; Kim, S. Multi-agent Deep Reinforcement Learning for Autonomous Driving. *J.-Form. Sci. Comput.* **2018**, *24*, 670–674.
37. Yu, A.; Palefsky-Smith, R.; Bedi, R. *Deep Reinforcement Learning for Simulated Autonomous Vehicle Control*; Stanford University: Stanford, CA, USA, 2019.
38. Churamani, N.; Cruz, F.; Griffiths, S.; Barros, P. iCub: Learning emotion expressions using human reward. *arXiv* **2020**, arXiv:2003.13483.
39. Cruz, F.; Dazeley, R.; Vamplew, P. Explainable robotic systems: Understanding goal-driven actions in a reinforcement learning scenario. *arXiv* **2020**, arXiv:2006.13615.
40. Millán, C.; Fernandes, B.J.; Cruz, F. Human feedback in continuous actor-critic reinforcement learning. In Proceedings of the European Symposium on Artificial Neural Networks Computational Intelligence and Machine Learning, Bruges, Belgium, 2–4 October 2019; pp. 24–26.
41. Cruz, F.; Parisi, G.I.; Wermter, S. Learning contextual affordances with an associative neural architecture. In Proceedings of the European Symposium on Artificial Neural Networks Computational Intelligence and Machine Learning, Bruges, Belgium, 2–4 October 2019; pp. 27–29.
42. Deshpande, N.; Vaufreydaz, D.; Spalanzani, A. Behavioral decision-making for urban autonomous driving in the presence of pedestrians using Deep Recurrent Q-Network. *arXiv* **2020**, arXiv:2010.13407.