

Design of Obstacle Avoidance for Autonomous Vehicle using Deep Q-Network and CARLA Simulator

Wasinee Terapaptommakol ^{1,†,‡} , Danai Phaoharuhansa ^{2,†*}, Pramote Koowattanasuchat ^{‡*} and Jartuwat Rajruangrabin ^{‡*}

¹ wasinee.t@mail.kmutt.ac.th

² danai.pha@kmutt.ac.th

* Correspondence: jartuwat.raj@nstda.or.th, pramote.koo@nstda.or.th, danai.pha@kmutt.ac.th

† Department of Mechanical Engineering, King Mongkut's University of Technology Thonburi, Bangkok Thailand

‡ Rail and Modern Transports Research Center, National Science and Technology Development Agency, Pathum Thani, Thailand

Abstract: This paper proposes a deep Q-network (DQN) method for developing an autonomous vehicle control system to create the trajectory to destination and achieve collision avoidance from obstacles on the road on virtual environment. The intention of this work is to try to simulate the case scenario and train the DQN algorithm in the virtual environment before testing the real road for safety and reduce costs. CARLA simulator is used to demonstrate the motion of the autonomous vehicle in virtual environment that the obstacle vehicle is parked on the road and the autonomous vehicle is riding on the road. The target position, real-time position, velocity, and Lidar point cloud information are taken as input and the action setting such as acceleration pedal, brake pedal, and steering wheel are taken as an output. The actions are sent to torque control in transmission system of vehicle. The reward has been created by continuous function equations which have been designed especially ¹which are for this case to imitate the driver's behavior. The result shows that the proposed method can navigate the way to the destination without collision the obstacle by using the brake method and dodging method and shows the trend of DQN behavior when the training episode is increased, it will give a better result. This method which uses a non-global path planning method is successfully ²applied on a virtual environment platform ³which can be present that this method has an advantage in the autonomous vehicle area to allow to test in simulation and apply with the experiment in future work.

Keywords: Obstacle avoidance; Autonomous vehicle; Deep Q-network; CARLA simulator

Citation: Terapaptommakol, W.; Phaoharuhansa, D.; Rajruangrabin, J.; Koowattanasuchat, P. Title. *World Electr. Veh. J.* **2022**, *1*, 0.
<https://doi.org/>

Received:

Accepted:

Published:

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

Copyright: © 2022 by the authors. Submitted to *World Electr. Veh. J.* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the automotive industry has still grown, road traffic injuries in Thailand are increasing every year which is over 13,000 deaths and a million injuries each year [1]. Most of road traffic accidents are caused by human error. Autonomous vehicles have already been sold in the public market, which can assist drivers such as adaptive cruise control, automatic parking control and lane-keep assist system.

Thailand is one of the countries that have the most injuries and fatalities in car accidents in the world. The main cause is that the driver cannot avoid the obstacle in time and about 20,000 people die in road accidents each year [1,2]. The research ⁴papers [3] present the ⁵static of collision of a car with an obstacle in the front that will cause the risk of injury up to almost 100% of accident cases, and it causes death by road accident. Due to human error that leads to injury and loss, the autonomous vehicle was developed to solve this problem.

To improve autonomous control system, one of the important issues is about the safety of the driver and passengers. The conventional control system is based on a math model [4],

Summary of Comments on Design of Obstacle Avoidance for Autonomous Vehicle using Deep Q-Network and CARLA Simulator

Page: 1

 Number: 1 Author: Reviewer Subject: Cross-Out Date: 12/10/2022 09:51:17

 Number: 2 Author: Reviewer Subject: Highlight Date: 12/10/2022 10:20:42
what do you mean by "implied"?
Do you mean "implemented"?

 Number: 3 Author: Reviewer Subject: Highlight Date: 12/10/2022 09:55:15
grammar

 Number: 4 Author: Reviewer Subject: Highlight Date: 12/10/2022 10:32:42
single or plural?

 Number: 5 Author: Reviewer Subject: Highlight Date: 12/10/2022 10:34:45
statistics

but it can perform vehicle with limit situation. So that, the machine learning algorithm has been applied in autonomous systems [5,6] in order to control vehicle in various situations.

To experiment, they have to test on real traffic roads, which obtains a lot of costs to prepare the experiment such as environment control and measurement system. So that, the simulation software was greatly developed to simulate the vehicle motion in virtual environment before experimenting with actual conditions.

Thus, the main motivation of this study is to reduce traffic fatality rates and accidents on the road by testing out the proposed method in the virtual environment to validate the accuracy of the technique to improve the autonomous driving system through the simulation program and then apply it in onsite experiments in the further work.

2. Relate work

This section briefly reviews the reference works of the literature related to this project such as object detection using LiDAR point cloud, simulator program, and fusion of them. In the past few years, autonomous vehicle has been tested using the Deep Q learning technique [7]. The advantage of this technique can be learned through their training in the algorithm and used in huge sample spaces and complex problems. Due to the research paper [8], using the camera for detecting the distance still has problems that are uncertainties of information, and the error is more than information from the LiDAR sensor.

Referring to the research papers [9,10], both are working on autonomous vehicles which provide algorithms based on Deep Q-Network (DQN) for the control layer that simulates in the CARLA simulator to avoid any risk during testing. The results are shown that the image data had an effect on the calculation process in the DQN algorithm because it has a low frequency for collecting information. Due to this purpose, the lidar point is limited to detecting the object more than the image which is reasonable to apply in this project. The boundary of LiDAR point cloud information was scoped in detection space that demonstrated the distance between vehicle and obstacle which is enough to avoid the collision.

DQN algorithm will use this information to decide due to the reward changing [11]. As mention above, the purpose of this project is used LiDAR sensor for detecting the object and DQN algorithm is applied as avoidance control system in order to avoid the obstacle. Both research papers focus on adjusting the algorithm to succeed in the same mission with different models. Therefor, this paper focuses on adjusting the reward condition for the DQN algorithm to succeed in the mission with smooth motion.

to demonstrate the vehicle motion by visualization, Gazebo and Carla simulator are an open-source software, which are used to demonstrate the autonomous vehicle by 3D visualization. In the research paper [12], Gazabo was used to estimate the performance of the software before real track tests because of the limitations in graphics quality and physics engine. Carla software is introduced in several papers [9,10,13] that it may be better than Gazebo because it is prepared for testing the autonomous vehicle which has a physics engine and sensors that can adjust any noise. It has been developed from the ground up to support the development, training, and validation of autonomous driving systems by simulating the response of vehicle in virtual situation. The advantages are that the program obtains the physic engine, and real-world map, which have many assets to create the experiment condition such as any type of vehicle, traffic attribute, onboard sensor, and environment that prepare for an autonomous system. It supports flexible specification of sensor suites, virtual environmental conditions, full control of all static and dynamic actors, maps generation, and much more by using python code for controlling.

3. Autonomous vehicle

Page: 2

	Number: 1	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:24:46
---	-----------	--------------	--------------------	---------------------------

wrong word choice

	Number: 2	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:25:53
---	-----------	--------------	--------------------	---------------------------

wrong grammar

	Number: 3	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:26:47
---	-----------	--------------	--------------------	---------------------------

what distance? from where to where?

	Number: 4	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:28:24
---	-----------	--------------	--------------------	---------------------------

suitable

	Number: 5	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:29:33
---	-----------	--------------	--------------------	---------------------------

detect

	Number: 6	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:31:04
---	-----------	--------------	--------------------	---------------------------

sounds awkward. Need to revise it.

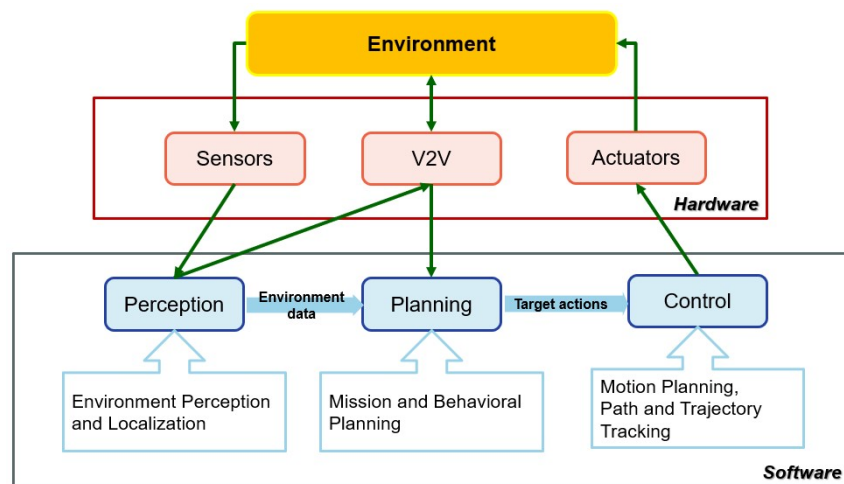


Figure 1. Architecture of autonomous driving system.

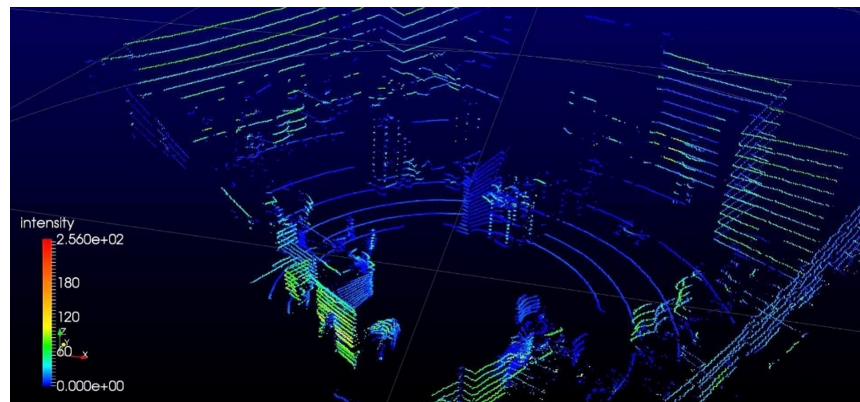


Figure 2. Example of point cloud output from a LiDAR.

Autonomous vehicle technology is evolving rapidly with the continuous improvement of artificial intelligence technology. It can perform on highways with the exact route path [14]. Autonomous driving systems consists of 2 main parts [15] as shown in Figure 1 that which consists of hardware and software[16]. The hardware part obtains the mechanical components such as sensors, Vehicle to Vehicle (V2V) hardware, and actuators. Sensors are used to observe the environment. V2V hardware is set up on each vehicle for communication and sharing data between the vehicles, and actuators are used to power the subsystem in the vehicle such as powertrain, steer wheel, and brake. The software models contain perception, path planning, and control system [15]. The perception module collects data from the sensors and builds 3D mapping of environment around the vehicle that makes cars understand the environment as shown in Figure 2 then, path planning design based on the user's command. Finally, the control system will order action to vehicle hardware by observing the actual environment.

The onboard sensor consists of several sensors, which is famous device for autonomous vehicles application such as camera, radar, and LiDAR. Camera sensors in automated driving systems have 2 concern points that sampling rate of image and visibility conditions.

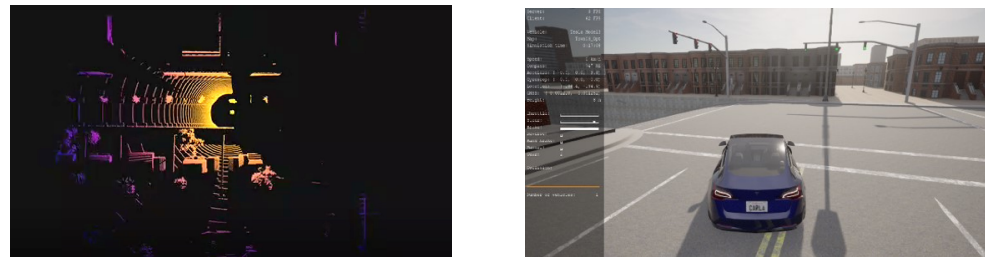
Radar technology uses high-frequency electromagnetic waves to measure the distance to objects based on the round-trip time principle. It is independence with light and weather conditions. It can measure up to 250 m in very adverse conditions.

Light Detection and Ranging (LiDAR) is an active-ranging technology that calculates the distance to objects by measuring the round-trip time of a laser light pulse [17]. Laser beams have a low divergence for reducing power decay and the measured distance is up

83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104

This page contains no comments

to 200 m under direct sunlight. The 3D point cloud is shown in Figure 2. It represents 3D visualization of environment that it is generated by the information from LiDAR [18].



(a) The LIDAR information

(b) Third person visualization

Figure 3. CARLA simulator presents by LiDAR information and 3D visualization.

4. Carla simulator

CARLA (Car Learning to Act) [19,20] is an open-source software for urban driving environment that it has been developed in order to simulate the vehicle and vehicle control system, which supports training, prototyping, and validation of autonomous driving model [21]. The simulator includes the visualization relative to vision of sensor and vision of third-person as shown in Figure 3a and 3b, respectively. It can be separated the process to apply with autonomous vehicle control simulation as 3 sections as shown in Figure 4 and it consists of virtual environment, agent, and control system.

4.1. Virtual environment

The virtual environment is presented as 3D models, which consists of static objects and dynamic objects. Static objects are such as road, buildings, traffic signs, and infrastructure, and dynamic objects denote the moving object in the environment such as pedestrians and other vehicles. All models are designed to reconcile visual quality and rendering speed. All environment models can be defined the dimension as same as real objects. They are presented in 3D visualization as shown in Figure 3.

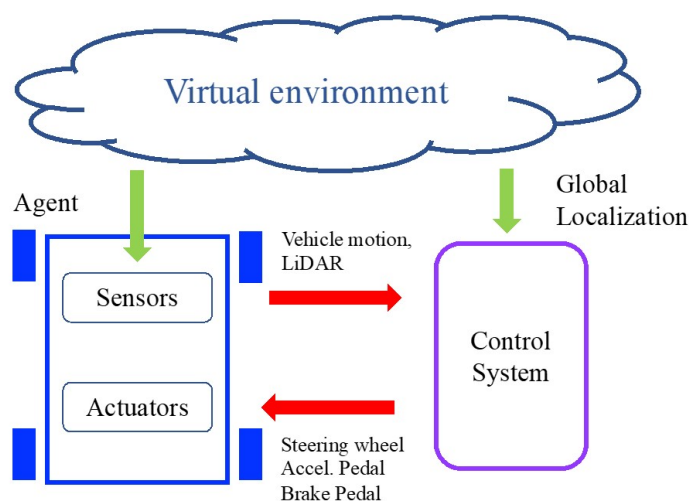


Figure 4. CARLA architecture for autonomous vehicle simulation

This page contains no comments

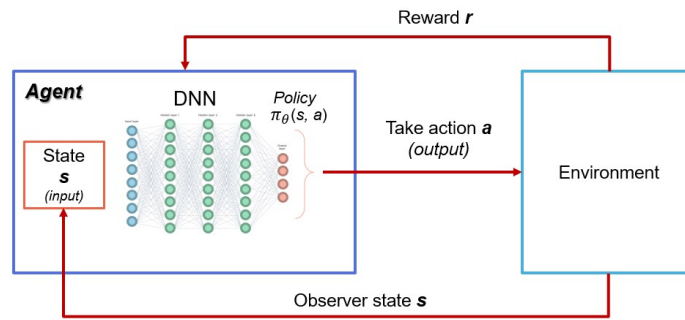


Figure 5. Deep Q-Learning process

4.2. Agent

The agent means that the object acts relative with control system such as pedestrian and vehicle. The agent is determined as vehicle and Tesla model 3 is represented instead of autonomous vehicle as shown in Figure 3b, dynamical model of which is included in CARLA. The agent consists of sensors and actuators in the vehicle. The sensor in the program obtains several kinds of sensor such as GPS, speedometer, IMU, LiDAR, camera, etc. The output of sensor can be applied the noise instead of the actual noise such as fog, rain, and smoke. The actuators denote steering wheel brake and traction motor.

In this study, LiDAR is added with conventional sensors and it is used to observe the obstacles and the environment. The attributes can be set the sensor parameters as Velodyne HDL-32E [19,22] and it obtains 32 channels and the sensor is mount in front of the vehicle. The range of detection is set at 40 meters. The vehicle motion is performed by steering wheel, acceleration pedal, and brake pedal. The steering wheel can control the steering angle, which is between ± 70 degrees. The acceleration and brake pedals can be assigned from 0 to 100 percent.

4.3. Control system

The control system can be used for feedback control or others. This study is designed based on DQN algorithms which is a kind of reinforcement learning. It consists of path planning, obstacle detection and DQN algorithm. The path planning is determined on the straight road. DQN algorithm received sensor data and it will determine the action to control the actuators.

5. Deep Q-network control

Deep Q-network is a kind of reinforcement learning, which applies to control the agent vehicle and it will be used to assign actuators such as acceleration pedal, brake pedal, and steering angle. It requires to define DQN model, training, and reward. Deep Q-learning can work well with complex decisions and a huge of data such as the large state input, the elaborate environment, and the delicate action.

Figure 5 presents that the process of Deep Q-learning that uses a neural network to approximate the Q-value function. The state is given as the input, and possible actions are generated as the output. DQN model will compute the reward to choose the action for a given state and store its experiences in a memory buffer. Then, the model will use the experiences from the memory buffer to train the target model, which calculates the best way to get more reward.

To train the DQN algorithm, the best reward and the state are collected in memory buffer as shown in Figure 6. Then, this section will be separated 4 parts such as model setting, action setting, reward setting, and hypermeter setting. These four parts are very important in enabling the DQN model to learn correctly and appropriately for the experiments. The details of designing the control system are listed as below.

 Number: 1	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:34:02
--	--------------	--------------------	---------------------------

separated into

 Number: 2	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:34:33
--	--------------	--------------------	---------------------------

wrong grammar

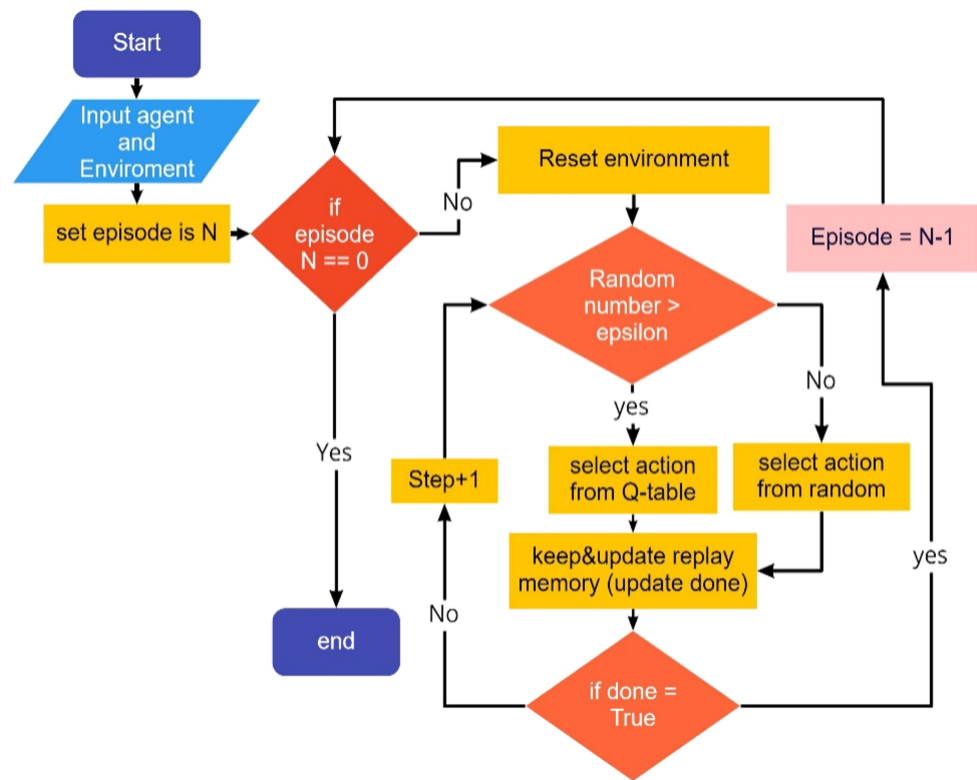


Figure 6. Process to select the action in DQN model

5.1. Model setting

DQN model consists of dense ¹nd layer, which has nodes in the neural network form. Each layer obtains the activation function, which matches the type of data that has been designed by the user. If the user doesn't config it, the activation function will be designated as the linear function by default. The activation function setting must be considered the problem of vanishing and the gradient explosion.

The dense layers that ²se in this project are 32 layers, 32 dense per layer, and 32 nodes per dense which have been ³and new trained in all the algorithms in the beginning. The Relu activation function has been applied in the hidden layer. They are defined as shown in Table 1. The information such as distance from start to destination which using the real-time position for calculating in the X-Y coordinate, velocity in the X-Y coordinate, and distance from the vehicle to the obstacle which come from the Lidar sensor in the X-Y coordinate is taken as input data to the DQN algorithm. The Lidar point cloud which already scopes the area where the vehicle will collide has been used for finding the nearest point and used for calculating the penalty reward later.

5.2. Action setting

In the action setting, ⁴his study defines ⁵actions such as straight forward and brake. The steering wheel are bounded at ± 35 degrees. The acceleration and brake pedals are acted relative to DQN algorithm consideration.

5.3. Reward setting

Reward setting consists of 3 rewards such as main reward, penalty reward, and extra reward. They should be set as continuous function in order to make the algorithm learn

Page: 6

 Number: 1	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:35:29
what layer?			
 Number: 2	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:36:46
is used			
 Number: 3	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:37:43
wrong word choice			
 Number: 4	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:40:54
If you only defined two actions, straight forward and brake, what's the point to have a steering wheel what are bounded at +/- 35 degree?			
 Number: 5	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:39:06
two			

in the correct and smooth direction. The hyperbolic function has been used in different weights for each reward condition. If the agent can success a good solution the reward will be given as double. If the agent did a bad solution the reward will be minus. The function of each reward condition effect to training period. Then, total reward (R_T) is given as

$$R_T = R_m + R_p + R_{ex} \quad (1)$$

Main reward (R_m) is designed relative to the distance from the vehicle to the destination (d_m) as shown in Equation (2). This reward is designed based on the quadratic polynomial which determines the farthest position is a minus value and the nearest position is the highest value. Thus, the agent will have the passion to arrive at the destination.

$$R_m = -a(d_m - b)^2 + c \quad (2)$$

$$d_m = \sqrt{(x_d - x_v)^2 + (y_d - y_v)^2} \quad (3)$$

where, R_m denotes main reward. d_m denotes distance from vehicle to destination. x_d and y_d denotes desired position in X and Y axis. x_v and y_v denotes vehicle's position in X and Y axis. The coefficient of a, b and c are -2.00, -98.00, and 3,000, subsequently.

Penalty reward (R_p) is computed by distance from LiDAR as shown in Equation (4). It depends on the minimum of vehicle and obstacle when the obstacle can be detected within the area that distance ± 3 meters on the lateral axis, distance from the road to 1.3 meters on the vertical axis, and distance 0 to 70 meters on the longitudinal axis. Thus, decreasing the size of LiDAR point cloud information will affect the process of calculating to make faster decisions and enough for detecting obstacles. A penalty reward has been created to restrain the motion to the destination and avoids obstacle. The equation is designed as a fourth-degree polynomial equation that is imitated from human behavior. If the LiDAR detected the obstacle at the farthest point, the value of the penalty reward would be a little bit positive. On the contrary, the value lower than 50 meters is taken as the minus value in the penalty reward and decreases rapidly when it is lower than 15 meters and approaches 0 meters which is the critical point that affects the algorithm to change decisions when compared with the main reward. The critical areas were calculated by the distance that was used for a stop at a speed of 40 km/h.

$$R_p = a_1 d_l^4 + a_2 d_l^3 + a_3 d_l^2 + a_4 d_l + a_5 \quad (4)$$

$$d_l = \min \sqrt{x_o^2 + y_o^2 + z_o^2} \quad (5)$$

where, x_o , y_o , and z_o denote nearest detected position, which scalar of distance vector from vehicle to obstacle. The coefficient of a_1 , a_2 , a_3 , a_4 and a_5 are -0.0011, 0.15, -6.60, 145 and -3,040, subsequently.

Extra reward (R_{ex}) is the special reward, which is added to the total reward when the agent can avoid the collision within the episode. This reward will be added when the vehicle can brake within 10 meters from the obstacle. The design of the reward was concerned with driver behavior. The value of the rewards has been effective with the choice for making decisions. The reward is given as

$$R_{ex} = b_1 + b_2 \quad (6)$$

where, the coefficient of b_1 and b_2 are -200.00 and 2,200, subsequently.

Page: 7

	Number: 1	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:42:54
For each symbol, you have to define/describe it before you first use it or describe/define it immediately after the equation. What are Rp and Rex?				
	Number: 2	Author: llxs	Subject: Cross-Out	Date: 13/10/2022 11:43:20
	Number: 3	Author: llxs	Subject: Cross-Out	Date: 13/10/2022 11:44:59
You have defined Rm in the first sentence of this paragraph. No need to define it again.				
	Number: 4	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:46:32
You need to explain why these values are used? By experiment/empirical, or randomly selected?				
	Number: 5	Author: llxs	Subject: Cross-Out	Date: 13/10/2022 11:46:47
	Number: 6	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:47:23
What minimum or minimum of what?				
	Number: 7	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:47:52
wrong grammar				
	Number: 8	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:49:20
which value or what value?				
	Number: 9	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:51:18
the nearest				
	Number: 10	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:51:30
positions				
	Number: 11	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:53:49
are scalars				
	Number: 12	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:55:49
How/why do you decide these values?				
	Number: 13	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:55:24
use plural please				
	Number: 14	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:57:04
what is xl?				
	Number: 15	Author: llxs	Subject: Highlight	Date: 13/10/2022 11:56:27
the				

Table 1. List of hyperparameters and their values.

Parameters	Values	Descriptions
Number of hidden layers and dense	32x32x32	1 the number of nodes in the neuron network. In this case, 2 the 3 hidden layers and 32 dense 3 each.
Activation function	ReLu	Function which defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.
Replay memory size	1,000,000	SGD updates are sampled from this number of most recent steps.
Target network update frequency	400	The frequency (measured in the number of parameter updates) with which the target network is updated.
Discount factor	0.95	Discount factor gamma used in the Q-learning update.
Learning rate	0.001	Tuning parameter that determines the step size at each iteration while moving toward a minimum of a loss function.
Initial exploration	1	Value of ϵ in ϵ -greedy exploration.
Final exploration	0.0001	Data
Final exploration step	1,840 episodes (600,000 steps)	The number of steps over which the initial value of ϵ is linearly annealed to its final value.
Replay start size	4,000	A uniform random policy is run for this number of steps before learning starts and the resulting experience is used to populate the replay memory.
Input data size	6	The number of the data that given as input in Q-network
Output data size	4	The number of the action that given as output in Q-network

5.4. Hyperparameter setting

The hyperparameter setting is separated into 2 parts relative to network structure and training algorithm. They are defined as 4 shown in Table 1 and the hyperparameters for network structure are the number of hidden layers and units, dropout, network weight initialization, and activation function. The hyperparameters for training algorithms are learning rate, number of epochs, and batch size. ϵ -greedy exploration is an exploration strategy in reinforcement learning that takes an exploratory action with probability and a greedy action [11]. This project uses the initial exploration value 5 1 and the Final exploration value is 0.0001. The method to decrease the epsilon value is to decay by 0.995 in every episode. 6 Every 2000 episodes, the DQN model that got the highest reward 7 have been selected and retrained with a new initial exploration value that is 0.7 to protect convergence resulting in an incomplete collection of values.

6. Case scenario

Page: 8

 Number: 1	Author: Ilxs	Subject: Highlight	Date: 13/10/2022 11:59:20
Need to have a clear line/space between rows so the description can be readable/separated.			
 Number: 2	Author: Ilxs	Subject: Highlight	Date: 13/10/2022 11:57:37
there are			
 Number: 3	Author: Ilxs	Subject: Highlight	Date: 13/10/2022 11:57:51
in each layer			
 Number: 4	Author: Ilxs	Subject: Cross-Out	Date: 13/10/2022 11:59:56
 Number: 5	Author: Ilxs	Subject: Highlight	Date: 13/10/2022 12:00:50
of			
 Number: 6	Author: Ilxs	Subject: Highlight	Date: 13/10/2022 12:01:33
For every			
 Number: 7	Author: Ilxs	Subject: Highlight	Date: 13/10/2022 12:01:49
has			

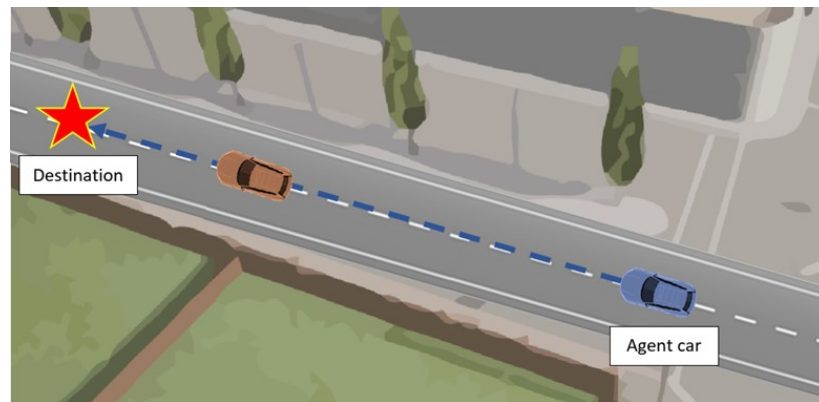


Figure 7. The scenario obstacle avoidance in low-speed control.

One of highest accident case in Thailand is that the agent vehicle performs on the main road and the obstacle vehicle drives across with the main road as shown in Figure 7. The agent vehicle cannot brake or avoid the time, which has challenge to design the function for avoiding the obstacle vehicle using DQN algorithm.

Therefore, the case scenario in this study is defined that the initial velocity of the agent car is 40 km/h and the destination is at 100 meters. The front wheel's angle is bounded at ± 35 degrees. Then, the agent vehicle is driven to the destination and the obstacle is placed on the trajectory. The agent vehicle should learn to brake or avoid without the crash.

7. Simulation result

According to the case scenario, the parameters are configured as Table 1. DQN model has been developed relative to the number of training episodes. The results as shown in Figure 9 and Figure 10 present distance from vehicle to destination, velocity, and rewards. The motion will be presented by 3D visualization in CARLA simulator as shown in Figure 8.

Considering distance and velocity of agent vehicle in Figure 9, the distance from the vehicle to destination is shown in Figure 9a. The training episode is increased, the final position of vehicle becomes nearly to the destination. The decision of control system may be switching between stop and forward, after it stops the motion as shown in blue line in Figure 9. It presents the vehicle motion for DQN model at 1400 training episode. The other cases the vehicle will permanently stop. Therefore, the decision of control system using 1400 training episode may not enough to use as obstacle avoiding because some nodes in DQN model may still blank node so that the control system may be finding the better path, which tries to move forward after the vehicle stop.

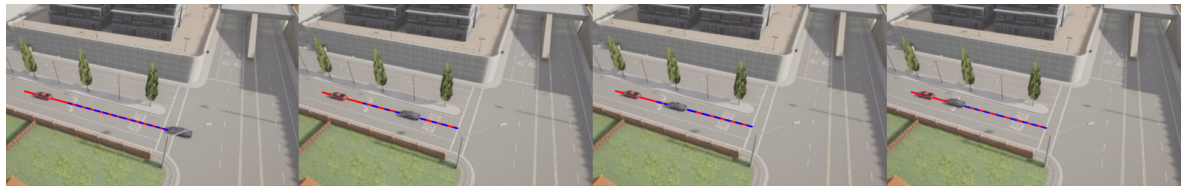
The velocity profile of each case is shown in Figure 9b. The initial velocity in all cases is about 40 km/h and the velocity using DQN model at 1400 and 5000 training episode will decrease with low deceleration after that the deceleration will increase to zero velocity. The velocity using DQN model at 7000 and 8000 training episode almost decrease with constant deceleration to zero velocity.

Figure 10 illustrates the reward during the motion from initial position to goal. The rewards increase when the vehicle becomes to destination but the rewards for DQN model at 1400, 5000, and 8000 training episode increased to the convergent values and the oscillation of rewards may be occurred by penalty reward because the minimum distance from vehicle to obstacle may be swayed by the shape of obstacle.

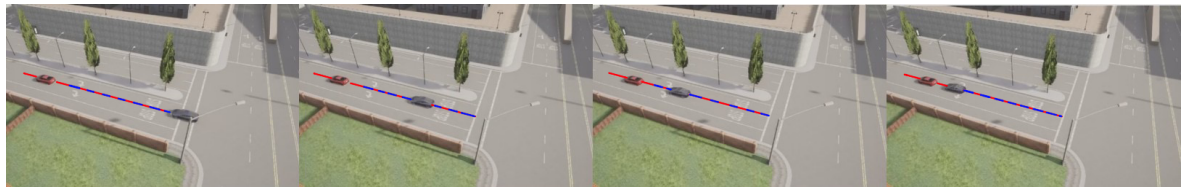
As above mentions, the amount of training episode affects to the vehicle's behavior. The control system uses DQN model, which is trained at 1400, 5000, 7000, and 8000 training episode, respectively. The demonstration is shown in Figure 8. The vehicle control system using DQN model at 1400 and 5000 training episodes can be stop before the vehicle crash

Page: 9

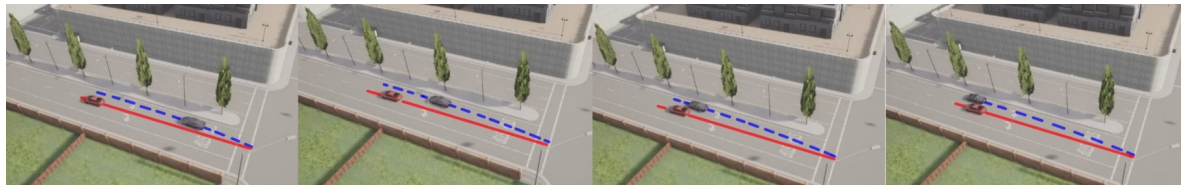
	Number: 1	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:03:09
in time				
	Number: 2	Author: llxs	Subject: Inserted Text	Date: 13/10/2022 12:03:41
suggest to change to "is challenging in designing"				
	Number: 3	Author: llxs	Subject: Cross-Out	Date: 13/10/2022 12:03:54
	Number: 4	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:05:23
You need to explain why this speed is used in the case study? Is this the normal speed for cars driving in Thailand				
	Number: 5	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:10:03
why "or"? Do you have other means to avoid crash except "braking"? You only defined two actions, straight forward and brake. There is no other way to avoid the crash without a brake.				
	Number: 6	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:10:48
which present				
	Number: 7	Author: llxs	Subject: Cross-Out	Date: 13/10/2022 12:11:05
	Number: 8	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:14:52
motion is				
	Number: 9	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:16:37
missing a verb				
	Number: 10	Author: llxs	Subject: Cross-Out	Date: 13/10/2022 12:17:15
is				
	Number: 11	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:19:50
what do you mean by "vehicle becomes to destination"? You need to clearly describe the status of the vehicle and the obstacle/destination".				
	Number: 12	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:20:59
wrong word				
	Number: 13	Author: llxs	Subject: Highlight	Date: 13/10/2022 12:21:24
affects				



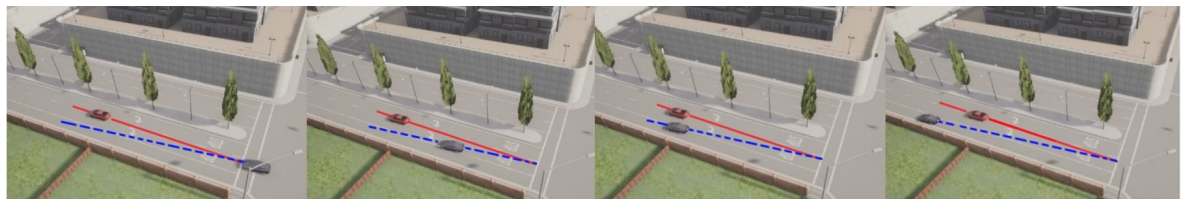
(a) 1400 training episodes



(b) 5000 training episodes

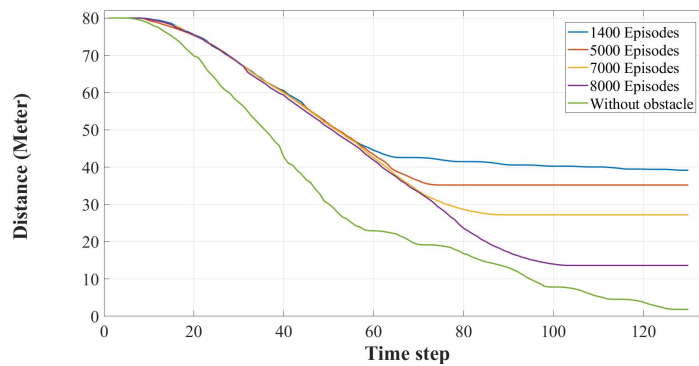


(c) 7000 training episodes

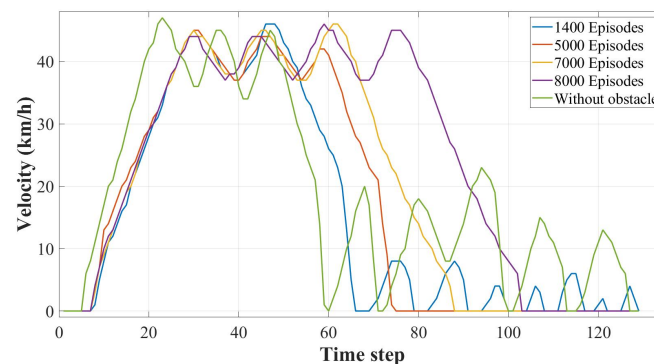


(d) 8000 training episodes

Figure 8. The agent performs using DQN model at 1400, 5000, 7000, and 8000 training episodes.



(a) Distance from agent to destination.



(b) Velocity of agent

Figure 9. The motion of the agent relative to the number of training episode for straight motion test.

with the obstacle but it is not good enough to avoid the obstacle as shown in Figure 8a and Figure 8b, respectively.

After DQN model is trained at 7000 and 8000 training episode, the vehicle's behavior will be improved that the vehicle can avoid the obstacle as shown in Figure 8c and 8d. The vehicles become avoid the obstacle with deceleration and they can stop the destination. The vehicle using model at 7000 training episode immediately avoids the obstacle when LiDAR detected the obstacle. The vehicle using model at 8000 training episode avoids the obstacle when the vehicle nears the obstacle rather than other cases.

8. Conclusion

According to the results, this paper successes to develop a control system using the DQN algorithm for obstacle avoidance. CARLA is used to present the result by visualization and graph. The vehicle motion using the DQN model at the high number of training episodes is greater than the low number of training episodes.

To train the DQN model, the best training result is recognized to replay memory size. If the model is trained by TensorFlow version 1, the recognized memory may be divergence and some memories may not best results. It causes the failure of the simulation results. TensorFlow version 2.8 is used instead of version 1, and the results are better than the first version.

Referring to ϵ -greedy exploration, it influences the quality of replay memory. If the model uses a small initial exploration value, the model may not find a new solution. If the final exploration value is a large value, the model won't make the decision by itself. For the 0.5 ϵ -greedy exploration value, the model is slowly improving but it cannot be suitable for the scenario case and it uses a lot of time in training. Thus, the preference in this project

 Number: 1 Author: llxs Subject: Highlight Date: 13/10/2022 12:25:43

You need to explain what actions/mechanisms you used in the simulation to avoid the obstacle. Recall that you only have straight forward and brake action, there is no control on the steering wheel, how could in your figure 8c and 8d the agent car sway from collision with the obstacle if you are not able to change the direction?

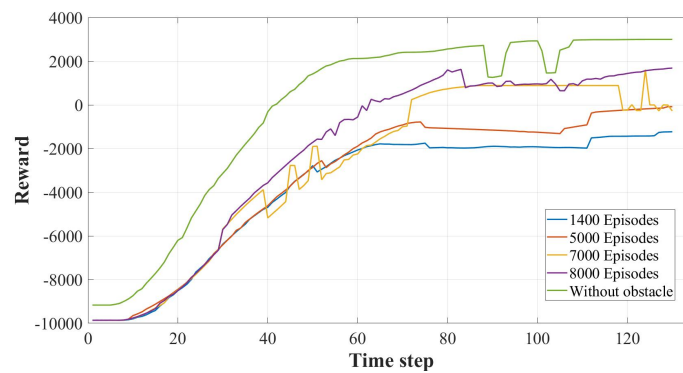


Figure 10. Reward for each number of training episode for straight motion test.

has been designed to slowly decay from 1 to 0.0001 as shown in table 1. The purpose aims to let the model search for a new solution and after that try to gradually increase to make the decision by itself. The replay memory size has been designed to make sure that it can collect all the states that happen enough to generate in DQN-model.

According to the reward graph, as shown in Figure 10, the value of training without obstacles between the time steps at 85 to 95 and 100 to 110 was presented as an attraction for DQN making decisions that balance the coefficient of main reward and penalty reward. Thus, even if the LiDAR sensor detected any obstacle in the lane, it didn't have an effect for change the decision as long as it was not in the critical area that is within 15 meters. Comparing training with obstacles, the reward result also shows the trend for training episodes that were better when episodes were increased. When comparing the reward graph between 7000 episodes of training and 8000 episodes of training, the shape of the 7000 episodes of training was swinging more than 8000 episodes of training which was disturbed by the LiDAR sensor. Thus, DQN algorithm in this study was trying to avoid sudden changes in reward value. The limitation of this work is using only this specific case scenario which is to avoid the static obstacle in front of the vehicle. If the obstacle is a dynamic obstacle or obstacles are on the side of the vehicle, the algorithm may not be successful. The difference boundary condition must be setting the new critical data set which has risks of collision and training the algorithm again.

This work is one of the functions in an autonomous control system. This work purposes to apply DQN model with the other function and combine them for fully autonomous vehicle control in future work.

Author Contributions: Conceptualization, J.R., D.P. and P.K.; methodology, J.R., W.T., P.K. and D.P.; software, W.T. and P.K.; validation, J.R., P.K., D.P. and W.T.; formal analysis, D.P.; investigation, J.R.; resources, J.R.; data curation, W.T.; writing—original draft preparation, W.T.; writing—review and editing, W.T., D.P., and J.R.; visualization, W.T.; supervision, J.R. and D.P.; project administration, W.T.; funding acquisition, J.R. and D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research project is supported by Thailand Science Research and Innovation (TSRI) Basic Research Fund: Fiscal year 2022 under project number FRB650048/0164.

Institutional Review Board Statement: Not applicable

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

References

- World Health Organization (WHO), Thailand's status against 12 Global Road Safety Performance Targets. 2020. Available online: <https://www.who.int/docs/default-source/thailand/roadsafety/thailand-status-against-12-global-road-safety-performance-targets.pdf?sfvrsn=92a24b064> (accessed on 10 June 2021).

 Number: 1 Author: llxs Subject: Highlight Date: 13/10/2022 12:27:54
it is better to present this in a subsection of discussion in section 7.

2. Suphanchaimat, R.; Sornsrivichai, V.; Limwattananon, S.; Thammawijaya, P. Economic Development and Road Traffic Injuries and Fatalities in Thailand: An Application of Spatial Panel Data Analysis, 2012–2016. *BMC Public Health* **2019**, *19*, 1449 331
332
3. Klinjun, N.; Kelly, M.; Praditsathaporn, C.; Petsirasan, R. Identification of Factors Affecting Road Traffic Injuries Incidence and Severity in Southern Thailand Based on Accident Investigation Reports. *Sustainability* **2021**, *13*, 12467 333
334
4. Kiran, G.R.K.; Deo, I.K.; Agrawal, S.; Haldar, S.; Shah, H.; Rudra, S.; Maheshwari, H.; Rathore, A.; Shah, P.; Nehete, A.; et al. Design and Implementation of Autonomous Ground Vehicle for Constrained Environments. In *Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC)* **2019**, 236–239. 335
336
337
5. Benterki, A.; Boukhnifer, M.; Judalet, V.; Choubeila, M. Prediction of Surrounding Vehicles Lane Change Intention Using Machine Learning. In *Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* **2019**, 839–843. 338
339
340
6. Min, K.; Han, S.; Lee, D.; Choi, D.; Sung, K.; Choi, J. SAE Level 3 Autonomous Driving Technology of the ETRI. In *Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC)* **2019**, 464–466. 341
342
7. Sanil, N.; venkat, P.A.N.; Rakesh, V.; Mallapur, R.; Ahmed, M.R. Deep Learning Techniques for Obstacle Detection and Avoidance in Driverless Cars. In *Proceedings of the 2020 International Conference on Artificial Intelligence and Signal Processing (AISP)* **2020**. 343
344
8. Barea, R.; Perez, C.; Bergasa, L.M.; Lopez-Guillen, E.; Romera, E.; Molinos, E.; Ocana, M.; Lopez, J. Vehicle Detection and Localization Using 3D LIDAR Point Cloud and Image Semantic Segmentation. In *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC); IEEE, November 2018* **2018**, 3481–3486. 345
346
347
9. Pérez-Gil, Ó.; Barea, R.; López-Guillén, E.; Bergasa, L.M.; Revenga, P.A.; Gutiérrez, R.; Díaz, A. DQN-Based Deep Reinforcement Learning for Autonomous Driving. In *Proceedings of the Advances in Physical Agents II; Springer International Publishing: Cham* **2021**, 60–67. 348
349
350
10. Pérez-Gil, Ó.; Barea, R.; López-Guillén, E.; Bergasa, L.M.; Gómez-Huélamo, C.; Gutiérrez, R.; Díaz-Díaz, A. Deep Reinforcement Learning Based Control for Autonomous Vehicles in CARLA. *Multimed Tools Appl* **2022**, *81*, 3553–3576. 351
352
11. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. **2013**. 353
354
12. Pan, Y.; Cheng, C.-A.; Saigol, K.; Lee, K.; Yan, X.; Theodorou, E.A.; Boots, B. Imitation Learning for Agile Autonomous Driving. *Int J Rob Res* **2020**, *39*, 286–302. 355
356
13. Dworak, D.; Ciepiela, F.; Derbisz, J.; Izzat, I.; Komorkiewicz, M.; Wójcik, M. Performance of LiDAR Object Detection Deep Learning Architectures Based on Artificially Generated Point Cloud Data from CARLA Simulator. In *Proceedings of the 2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)* **2019** pp.600–605. 357
358
359
14. Sung, K.; Min, K.; Choi, J. Driving Information Logger with In-Vehicle Communication for Autonomous Vehicle Research. In *Proceedings of the 2018 20th International Conference on Advanced Communication Technology (ICACT)* **2018**, 300–302. 360
361
15. Serban, A.C.; Poll, E.; Visser, J. A Standard Driven Software Architecture for Fully Autonomous Vehicles. In *Proceedings of the 2018 IEEE International Conference on Software Architecture Companion (ICSA-C)* **2018**, 120–127. 362
363
16. Wang, C. 2D Object Detection and Semantic Segmentation in the Carla Simulator, 2020. 364
17. Marti, E.; de Miguel, M.A.; Garcia, F.; Perez, J. A Review of Sensor Technologies for Perception in Automated Driving. *IEEE Intelligent Transportation Systems Magazine* **2019**, 94–108. 365
366
18. Warren, M.E. Automotive LIDAR Technology. In *Proceedings of the 2019 Symposium on VLSI Circuits* **2019**, 254–255. 367
19. CARLA Documentation, Sensors reference. 2020. Available online: <https://carla.readthedocs.io/en/latest/refensors/> (accessed on 22 June 2021). 368
369
20. Carla Simulator, Open-source simulator for autonomous driving research. 2020. Available online: <https://carla.org/> (accessed on 1 June 2021). 370
371
21. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Conference on Robot Learning (CoRL 2017)*, Mountain View, United States; November 10 2017. 372
373
22. Velodyne Lidar, High Resolution Real-Time 3D Lidar Sensor. 2020. Available online: <https://velodynelidar.com/products/hdl-32e/> (accessed on 1 June 2021). 374
375

This page contains no comments