electronics

MDPI

Article

Machine Learning-Based Vehicle Trajectory Prediction Using V2V Communications and On-Board Sensors

Dongho Choi, Janghyuk Yim, Minjin Baek and Sangsun Lee *

Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, Korea; cdh5375@hanyang.ac.kr (D.C.); jhyim@hanyang.ac.kr (J.Y.); mjbaek@hanyang.ac.kr (M.B.)
* Correspondence: ssnlee@hanyang.ac.kr; Tel.: +82-2-2299-0372

Abstract: Predicting the trajectories of surrounding vehicles is important to avoid or mitigate collision with traffic participants. However, due to limited past information and the uncertainty in future driving maneuvers, trajectory prediction is a challenging task. Recently, trajectory prediction models using machine learning algorithms have been addressed solve to this problem. In this paper, we present a trajectory prediction method based on the random forest (RF) algorithm and the long short term memory (LSTM) encoder-decoder architecture. An occupancy grid map is first defined for the region surrounding the target vehicle, and then the row and the column that will be occupied by the target vehicle at future time steps are determined using the RF algorithm and the LSTM encoder-decoder architecture, respectively. For the collection of training data, the test vehicle was equipped with a camera and LIDAR sensors along with vehicular wireless communication devices, and the experiments were conducted under various driving scenarios. The vehicle test results demonstrate that the proposed method provides more robust trajectory prediction compared with existing trajectory prediction methods.

Keywords: machine learning; random forest; LSTM encoder-decoder; collision warning system; lane changing prediction; trajectory prediction

check for updates

Citation: Choi, D.; Yim, J.; Baek, M.; Lee, S. Machine Learning-Based Vehicle Trajectory Prediction Using V2V Communications and On-Board Sensors. *Electronics* **2021**, *10*, 420. https://doi.org/10.3390/electronics10040420

Academic Editor: Arturo de la Escalera Hueso

Received: 29 December 2020
Accepted: 2 February 2021
Published: 9 February 2021

1. Introduction

Autonomous vehicles have undergone phenomenal development over the past decade both for safety and efficient mobility [1]. The development of advanced driver assistance systems (ADAS) is of interest to automotive original equipment manufacturing (OEMs) to reduce the number of traffic accidents. Vehicles equipped with ADAS such as adaptive cruise control (ACC), lane keeping assist system (LKAS) and emergency braking system (EBS) are already extant on the road. One ADAS, the collision warning system (CWS), is able to predict a collision situation and warn the driver in advance. This means that perceiving the traffic scene and predicting trajectory of surrounding vehicles (SV) are critical tasks. However, predicting the trajectory of the SV is quite difficult work since it depends on characteristics of each driver and various traffic situations. To overcome this problem, many approaches to trajectory prediction have been proposed in [2]. The traditional approaches for trajectory prediction are made by assuming a physics-based model such as one based on kinematics or dynamics. Dynamic models describe motion based on many internal parameters of the vehicle such as the longitudinal and lateral tire forces [3,4]. A kinematics model, on the other hand, describes a vehicle's trajectory based on the parameters of movement such as velocity, acceleration and position. Kinematics models are used more than dynamics models for trajectory prediction of SV because the internal parameters of the SV are not observable by sensors mounted on the ego vehicle (EV). A comparison and survey of different kinematics models for tracking vehicle trajectory was given in [5]. In this paper, constant turn rate and acceleration models (CTRA) show better results in tracking the vehicle trajectory than the constant turn rate and velocity models (CTRV). However, this model shows high accuracy only in a monotonous driving environment. If the road is curved or the SV is overtaking the EV, the accuracy of the trajectory prediction is very poor. To overcome such limitations, research using maneuver-based models has been proposed [6–8]. This model predicts the trajectory of SV based on knowing the driver's intention in advance. If EV can identify the maneuver intention of SV, EV can predict trajectory more reliably in the long term than the kinematic models. Trajectory prediction using a kinematic model has high prediction accuracy when the SV is driving at a constant speed/acceleration within one lane, but the prediction accuracy is low in situations where SV are accelerating/decelerating and attempting to change lanes [9,10]. For this reason, the maneuver intention prediction of SV has been investigated by many researchers [11–16]. Among these works, machine learning-based algorithms are very popular because they are affected by the driver's habit [14–16]. After predicting the intentions of the SV in this way, trajectory prediction is executed based on maneuver-based model. Prediction methods can be classified into methods using predefined prototypes and machine learning algorithms. The predefined prototype method first clusters all trajectories of vehicles on the road and then each cluster is used as a predefined prototype trajectory for prediction [17–19]. Subsequent trajectory prediction can be performed by searching for any similarity between the predefined prototype trajectory and partial trajectory of SV. However, when trajectory prediction is executed using a finite prototype, the limitation is that the trajectory is strictly determined. The machine learning algorithm prediction method, a recent area of study, predicts the trajectory of SVs using a machine learning algorithm. Research that predicts the trajectory of SV using machine learning mainly uses recurrent neural network (RNN), which has good performance in predicting time series data [20–24]. Many previous studies use the publicly available NGSIM as a dataset for training prediction models [20,21]. Since the NGSIM dataset was created from vehicle trajectories collected through image processing from a camera mounted on the road, this data suffers from considerable tracking noise [25]. Additionally, in other previous studies, the relative coordinates of the SV were used as the result of the prediction model [22–24]. As can be seen in Figure 1, lateral position accuracy is very important in the trajectory prediction model of SV for CWS. Even if the position accuracy of the prediction model has an error of less than 1 m, CWS may not operate in a collision situation. Therefore, in our paper, we propose a method of predicting the trajectory by dividing the longitudinal prediction model and the lateral prediction model, and expressing the final position in a grid.

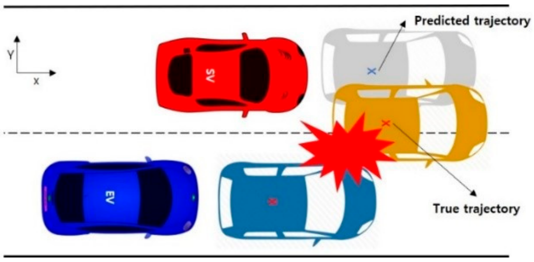


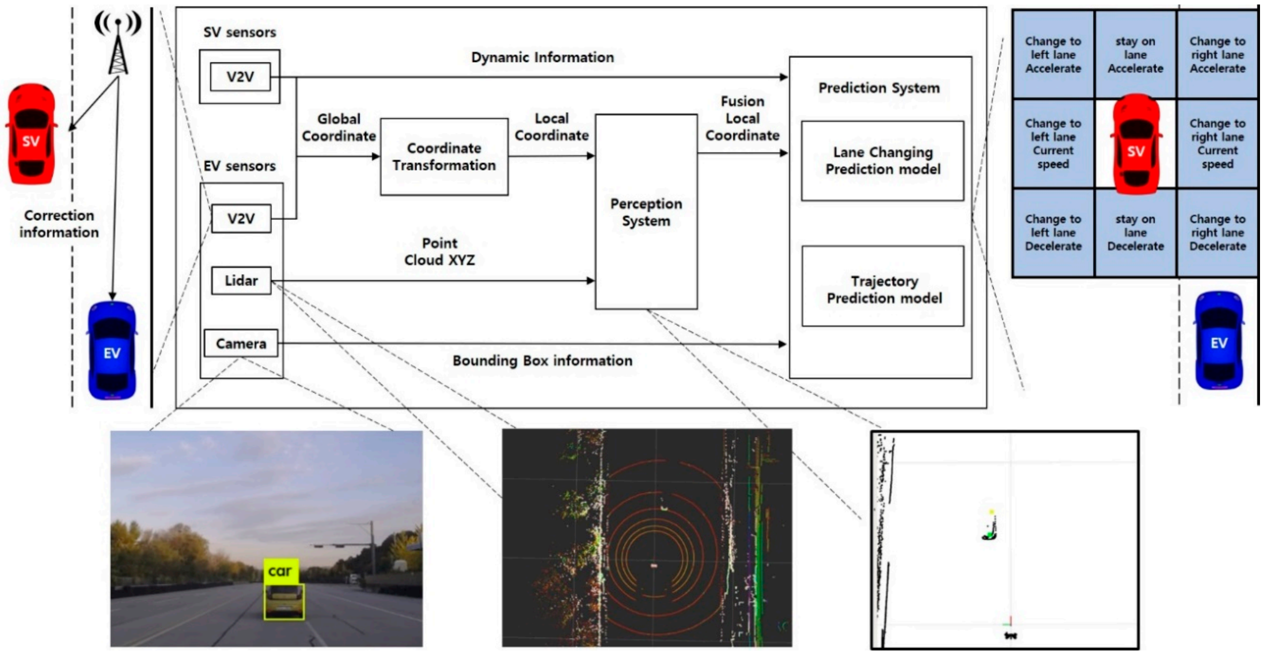
Figure 1. Collision situation caused by lateral position error.

The contribution of this paper is to present a trajectory prediction model using a grid. To ensure accuracy of the lateral position, a lane change prediction model was used. Instead of the public datasets used in previous studies, we used raw data without filters as input data. The bounding box coordinate value and size obtained from the camera sensor and dynamic information of the SV obtained through vehicle-to-vehicle (V2V) communication were used as input data of the prediction model.

The remainder of the paper is as follows: Section 2 describes the entire prediction system. The perception system of surrounding environment is described in Section 3. In Section 4, the system for predicting the trajectory is addressed in detail. The experimental results are presented in Section 5. Finally, in Section 6, we present conclusions and deal with future work.

2. System Overview

We propose a system that predicts the trajectory of SV using on-board sensors and V2V communication as shown in Figure 2. The EV is equipped with V2V communication equipment, LIDAR, and camera sensor to recognize the surrounding environment. The SV is equipped with only V2V communication equipment. The perception system is implemented to estimate the exact position of the SV using V2V communication and LIDAR. Among the information exchanged through V2V communication is location data acquired through the Global Positioning System (GPS). However, GPS has the disadvantage of being vulnerable to the surrounding environment. For example, when the road environment is an open area, location information has accuracy reliable to ~1 m, but in the urban areas reliability of location drops to 5–10 m due to signal blockage and multipaths [26]. Therefore, we applied differential GPS (DGPS) to our system to overcome this problem. DGPS can increase the accuracy of location information by receiving correction information from a base station, where the correction information is calculated from a fixed base station. Since the fixed base station knows the exact location information, it can periodically calculate the GPS error and broadcast it [27]. GPS uses a coordinate system known as earth centered-earth fixed geodetic (ECEF-g), which consists of latitude, longitude and altitude. We converted this to a local tangent plane and applied the local coordinate system by rotating the EV heading angle. Our perception system additionally includes location information obtained from LIDAR. This is because an error occurs depending on the heading accuracy of EV in the process of changing the global coordinate system to the local coordinate system. The LIDAR can get a 3D point cloud for the surrounding objects and the position is accurate to less than 10 cm. Finally, the perception system calculated the location of the SVs by matching the location information obtained through V2V communication and LIDAR. To predict the SV position, we propose an ensemble of two models: lane changing prediction and trajectory prediction. We used RF algorithm to predict lane changing of SV, and the trajectory prediction model used the Encoder-Decoder LSTM model. The lateral position prediction accuracy is important in predicting trajectory of SVs for the precollision warning system because even if EV predicts that the relative longitudinal distance of the SV will be close, it is in a safe state if the SV is in the adjacent lane. Therefore, we propose a prediction system that divides the location of SVs into nine grid cells and finally predicts the grid where the SVs will exist after a certain period of time.



Change to left lane Accelerate	stay on lane Accelerate	Change to right lane Accelerate
Change to left lane Current speed	SV	Change to right lane Current speed
Change to left lane Decelerate	stay on lane Decelerate	Change to right lane Decelerate

Figure 2. System architecture.

3. Perception System

3.1. Camera Sensor

In the field of autonomous vehicles, the camera is the most used and essential sensor because it guarantees a high object recognition rate and is cheaper than other sensors that recognize the surrounding environment. The main role of the camera sensor is to detect and classify surrounding objects. The most traditional method is to sequentially apply classifiers that classify simple feature values [28]. However, in order to improve the performance, a lot of learning data is required, and there is a difference in performance depending on the quality of the learning data. Another way is to use the Histogram of Oriented Gradient (HOG) to classify surrounding objects [29,30]. This is a method of classifying objects by using the gradient distribution characteristics of the objects in the camera image. However, the slow image processing speed constitutes a fatal drawback in the autonomous vehicle field where real-time guarantee is important. Recent research is actively underway to classify surrounding objects using a Convolution Neural Network (CNN). The first object detection using CNN started with classifying handwritten numbers in the 1990s [31]. However, it has not been in the spotlight because of algorithms with better performance and less complex machine learning such as support vector machine (SVM). In 2012, as AlexNet reduced the object recognition error rate from 26% to 15.3%, research on object recognition using CNN was underway [32]. Then, in 2015, ResNet achieved an error rate of 3.6% lower than that of classifying objects with the human eye. CNN's ability to distinguish a single object in an image is powerful, but distinguishing multiple objects remains a problem. To remedy this problem, Regions with CNN (R-CNN) was proposed, which first selects candidates that likely have objects in the image [33]. However, this has disadvantages in that a large storage space is required and the operation processing speed is slow because a network exists for each region selected as a candidate. To address these problems, Faster R-CNN was proposed [34]. It gets regional proposal from the final image obtained through the network process. The above two models (R-CNN/Faster R-CNN) are called 2-stage detectors because localization and classification are separately configured. This is the main cause of the slowdown in image processing. An algorithm called You Only Look Once (YOLO) can detect real-time objects because localization and classification are processed simultaneously [35]. Since YOLO's recognition speed is over 45 fps, and real-time guarantee is important for our system, we used YOLO to recognize the surrounding environment and get the bounding box value. As can be seen in Figure 3, the coordinate value and size of the bounding box are correlated with the relative coordinates of the SV. If the SV is far from the EV, the size of the bounding box is small. Conversely, if it is close, the size of the bounding box is large. Further, if the SV is driving in the adjacent left lane, the bounding box is to the left from the center point of the image. For this reason, we used the bounding box obtained through the camera as the input value of the prediction model.

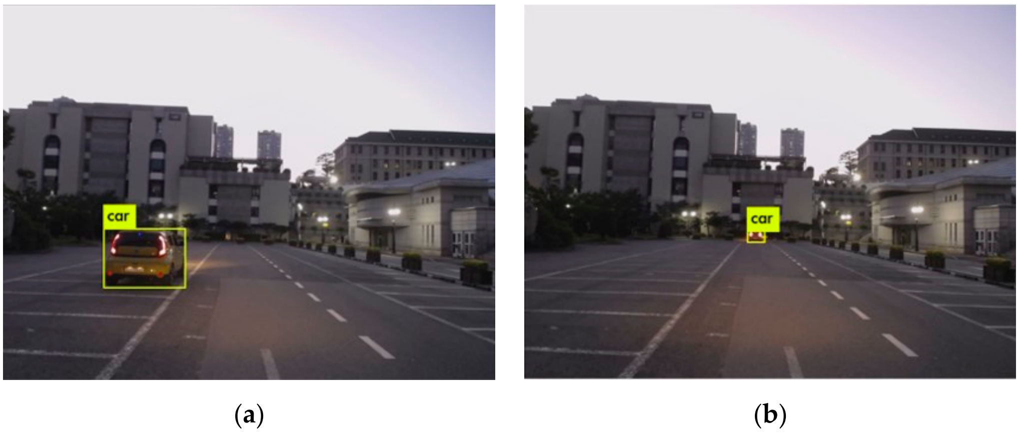


Figure 3. (**a**) The bounding box of the SV in the adjacent lane and close up; (**b**) the bounding box of the SV in the same lane and far away.

3.2. V2V Communication and LIDAR

Vehicle-to-Vehicle (V2V) communication is considered an essential factor for more advanced ADAS. V2V communication follows the wireless access in vehicular environments (WAVE) standard defined by IEEE, which is based on IEEE 802.11P and IEEE 1609 [36]. V2V communication using the WAVE standard exchanges information every 100 ms using the 5.9 GHz DSRC frequency band. The information exchanged between these vehicles follows the basic safety message (BSM) among the set of j2735 messages defined by SAE [37]. As shown in Table 1, the BSM contains dynamic information of the vehicle such as steering angle and acceleration that is difficult to obtain with existing on-board sensors apart from V2V communication.

Table 1. SAE J2735 BSM.

Message	Content
Part 1	Message count Temporary ID Time Latitude Longitude Elevation Position accuracy Transmission state Speed Heading Steering wheel angle Acceleration Yaw rate Brake system status Vehicle size (width, length)
Part 2	Event flags Path history Path prediction RTCM package

The relative coordinates of EV and SV should be calculated using the GPS information (latitude/longitude) in BSM. The GPS coordinate system is known as the earth centered-earth fixed geodetic (ECEF-g) coordinate system. To change this to a local coordinate system with EV as the origin, it is necessary to change it to the earth centered-earth fixed rectangular (ECEF-r) coordinate system and to a local tangent plane (LTP). The ECEF-r coordinates can be calculated from the ECEF-g coordinate system as shown in Equation (1).

$$\begin{aligned} x &= (h+N)\cos\lambda\cos\phi \\ y &= (h+N)\cos\lambda\sin\phi \\ z &= \left(h+\left(1+e^2\right)N\right)\sin\lambda \end{aligned} \tag{1}$$

where λ is the latitude, ϕ is the longitude, e is the eccentricity. N is the distance from the surface to the z axis along the ellipse normal and is defined by Equation (2).

$$N = \frac{a}{\sqrt{1-e^2\ \sin^2\ (\lambda)}} \tag{2}$$

where a is WGS-84 earth semimajor axis. As shown in Equation (3), the ECEF-r coordinate system is converted to an LTP coordinate system.

$$\begin{bmatrix} -\sin\phi & \cos\phi & 0 \\ -\cos\phi\sin\lambda & -\sin\lambda\sin\phi & \cos\lambda \\ \cos\lambda\cos\phi & \cos\lambda\sin\phi & \sin\lambda \end{bmatrix} \cdot \begin{bmatrix} x-x_0 \\ y-y_0 \\ z-z_0 \end{bmatrix} \tag{3}$$

If $<x, y, z>$ is the ECEF-r coordinate of SV, and $<x_0, y_0, z_0>$ is the ECEF-r coordinate of EV, the local coordinate system is completed by rotating the heading angle of EV. However, if there is an error in the heading angle of EV, there is a disadvantage in that an error occurs in the relative position. To overcome this problem, we estimated the location of the SV using the point cloud obtained through the LIDAR. The point cloud is just a set of points in the XYZ coordinate system around the LIDAR position. These are the points scanned by multiple rotating laser beams. The advantage is that the accuracy of the distance to the object measurement is high, but there is a disadvantage in that additional work is required to classify the object. Therefore, research on accurately detecting the surrounding environment using LIDAR is largely divided into three categories: object clustering, object classification and tracking the movement [38,39]. If two objects exist nearby, there is a problem of recognizing them as one during the object clustering process because the traditional clustering method classifies objects based on Euclidean distance. Recent research approaches object classification using machine learning algorithms by learning the shape and characteristics of objects [40,41]. This approach shows better performance than the traditional clustering method, but incurs high computational cost and requires high vertical-resolution. This means that autonomous vehicles should be equipped with expensive LIDAR.

To overcome the limitations described above, we proposed a method of estimating the exact location by a fusion of the location information calculated from V2V communication and the point cloud obtained from LIDAR. Although there is an error in the relative position obtained through V2V communication, the position of the SV can be estimated. If clustering is performed around this estimated position, we can distinguish the points reflected from the SV. As shown in Figure 4a, these reflected points are mostly reflected from the rear of the SV. Of course, as shown in Figure 4b, when the SV changes lanes or drives on a curved road, there are also points that have returned from the side of the SV. The closest points among the reflected points are recognized as the rear of the SV, and the final position of the SV is calculated by considering the specifications (width/length) of the vehicle obtained through V2V communication.

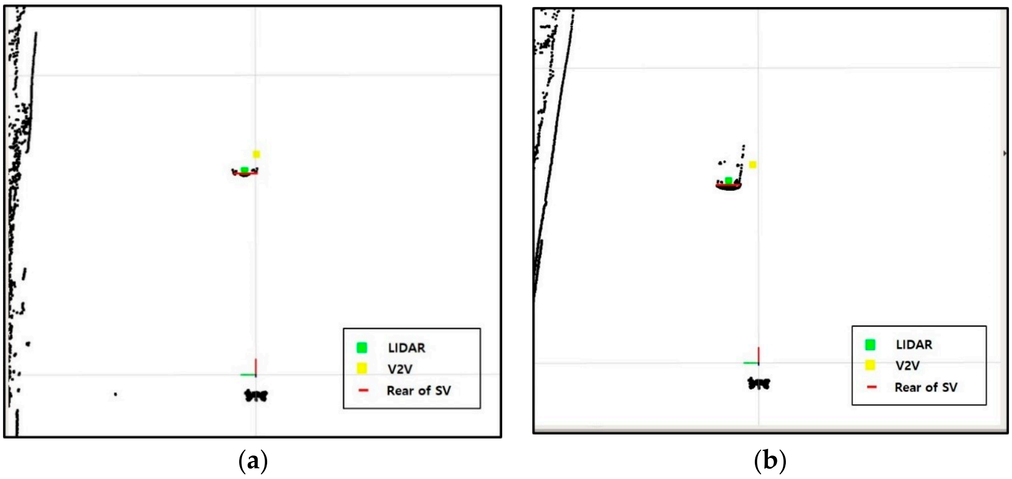
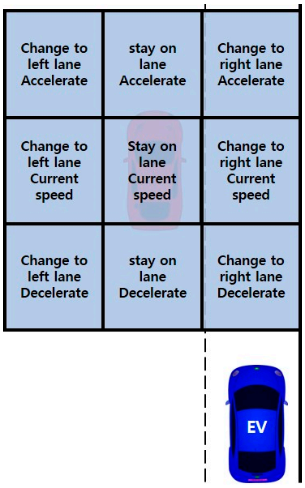


Figure 4. (**a**) In the case of keeping the lane in the same lane, Estimated location using V2V communications and LIDAR; (**b**) In the case of changing the lane in the adjacent lane, Estimated location using V2V communications and LIDAR.

4. Prediction System

The vehicle trajectory is characterized by having a continuous value. Therefore, recurrent neural network (RNN) structures are widely used in many papers for predicting trajectory [23,24]. It is necessary to predict the location of the SV after a certain period of time to give the driver an advance warning to prevent a collision. In the prediction system, the lateral position accuracy is very important when considering the width of the lane within the road because given the width of each vehicle, a false negative/positive pre-warning may occur even if the location accuracy error is only 0.5–1 m. Therefore, as shown in Figure 5, we divided the locations where the SV is likely to exist after a certain period of time into nine grid cells. In addition, we propose a system that predicts the intention to change lanes to determine the vertical lines of the grid and one that predicts the trajectory to determine the horizontal lines.



Change to left lane Accelerate	stay on lane Accelerate	Change to right lane Accelerate
Change to left lane Current speed	Stay on lane Current speed	Change to right lane Current speed
Change to left lane Decelerate	stay on lane Decelerate	Change to right lane Decelerate

Figure 5. Grid for the possible locations of SV.

4.1. Input Features

Our proposed system for predicting the trajectory of SV consists of a lane changing prediction model and a trajectory prediction model. To avoid overfitting of the prediction model, an input data that has a high correlation to the output data should be selected. We selected the feature value using the correlation coefficient as shown in Equation (4).

$$\begin{aligned} r &= \frac{\sigma_{xy}}{\sigma_x \sigma_y} \\ \sigma_x &= \sqrt{\sum_{i=1}^{n} \left(x^i - \mu_x\right)^2} \\ \sigma_y &= \sqrt{\sum_{i=1}^{n} \left(y^i - \mu_y\right)^2} \\ \sigma_{xy} &= \sum_{i=1}^{n} \left[\left(x^i - \mu_x\right)\left(y^i - \mu_y\right)\right] \end{aligned} \tag{4}$$

Here, x and y are each feature, and μ is the average of the feature. σ_{xy} is the covariance between features x and y, and σ_x and σ_y are the standard deviations of each feature. This coefficient represents the linear dependence between features. As shown in Figure 6a, the coordinate value and size of the bounding box obtained from the camera have a high correlation with the relative position between EV and SV. It can be seen in Figure 6b that velocity, longitudinal acceleration, heading, and steering angle are correlated with the relative position among BSM. Therefore, we defined feature values to be used as input in the prediction model as shown in Equation (5).

$$\mathrm{X} = \left[x_{bbox,bsm}^{t-(h-1)}, \cdots, x_{bbox,bsm}^{t-1}, x_{bbox,\, bsm}^{t}\right]$$

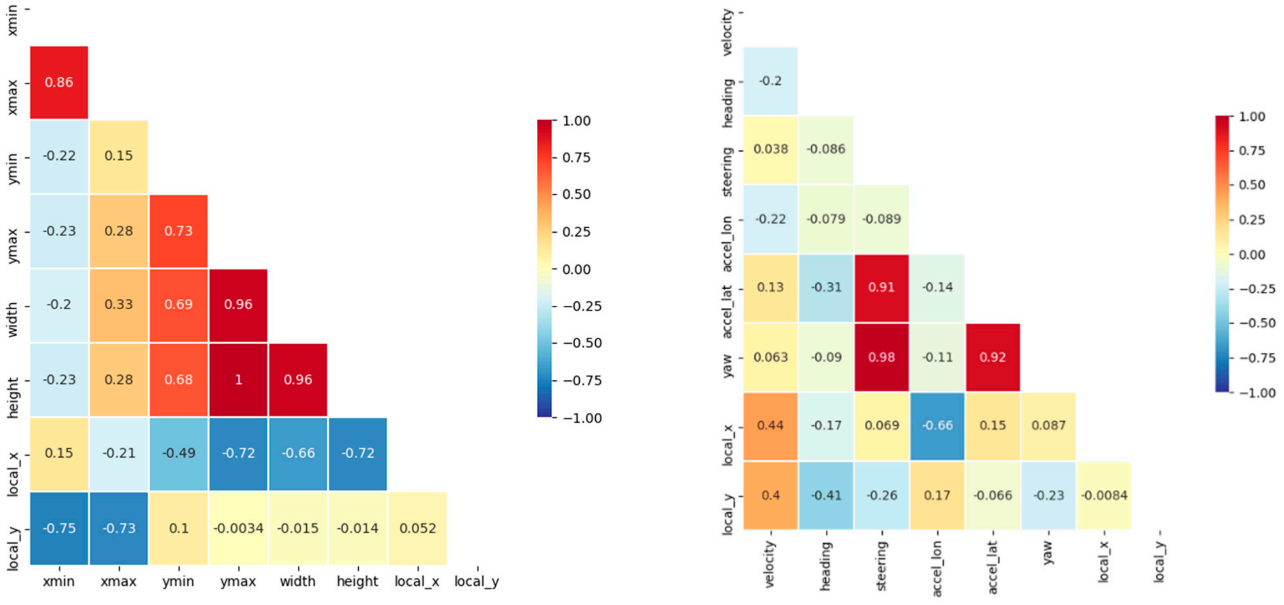
Here,

$$\begin{aligned} x_{bbox}^{t} &= [xmin,\ xmax,\ ymin,\ ymax,\ width,\ height,\,] \\ x_{bsm}^{t} &= \left[x^t, y^t, v^t, a^t,\ \theta^t,\ \delta^t\right] \end{aligned} \tag{5}$$

where *xmin* is the left coordinate value of the bounding box in the image, *xmax* is the right coordinate value of the bounding box, *ymin* is the top coordinate value of the bounding box, $ymax$ is the bottom coordinate value of the bounding box, x and y are the relative coordinates of SV with EV as the origin, v is the SV's velocity, a is the SV's acceleration, θ is the SV's heading, δ is the steering angle of SV, and h is observation time of previous trajectory. The output value of the model is defined as follows.

$$\begin{gathered} Y = \left[y^{t+1}_{position}, y^{t+2}_{position}, \cdots, y^{t+p}_{position}\right] \\ \text{Here,} \quad y^{t}_{position} = [x,\ y] \end{gathered} \tag{6}$$

where x and y are relative coordinates of SV with EV as the origin, and p is prediction time of the future trajectory.



(**a**) (**b**)

Figure 6. (**a**) Correlation matrix between bounding box and relative distance; (**b**) correlation matrix between BSM and relative distance.

4.2. Lane Change Prediction Model

Random Forest (RF) is an ensemble method with multiple decision trees [42]. Although each decision tree has a high variance problem, the ensemble reduces the variance of bagging and the risk of overfitting. The four steps of the random forest algorithm are as follows:

Step 1: Extract n bootstrap samples from the original dataset (permission for repetition).

Step 2: Train the base classifier from the bootstrap sample.

- Select d base classifiers from all base classifiers.
- Split into the node with the best classifier performance using information gain.

Step 3: Iterate the previous step K times.

Step 4: Assign a class label using majority voting.

In general, the more decision trees, the higher the computational cost, but the better the RF classifier performance. In addition, if there are factors that affect the results a lot, a simple and intuitive RF algorithm has good classification performance. The driver's intention to change lanes is closely related to the lateral position. Therefore, the lateral motion of the bounding box obtained from the camera sensor is an important factor in the prediction model. In addition, we can obtain the steering angle and accurate heading angle for SV through V2V communication. These two features are also decisive factors in lane change prediction models. In many previous studies, there has been a lot of research on predicting lane change using machine learning algorithms [43,44], but since we can obtain a decisive factor through camera and V2V communication, we use RF to predict lane changing intent. In Section 4.1, we defined the feature value to be used in the prediction model. Then the label value for feature value was defined as the maneuver state of the SV (lane keeping: 0, left lane changing: 1 and right lane changing: 2). As shown in Figure 7, the end point of the lane changing maneuver T_e is defined as the point at which the center of the SV reaches the center line of the target lane. The lane changing maneuver duration T_d is determined by the driver characteristics. For example, if the driver attempts an aggressive lane change, T_d will be shortened. As shown in Figure 8, the initiation point of a lane change T_i is defined as an instance that goes back by T_d from T_e.

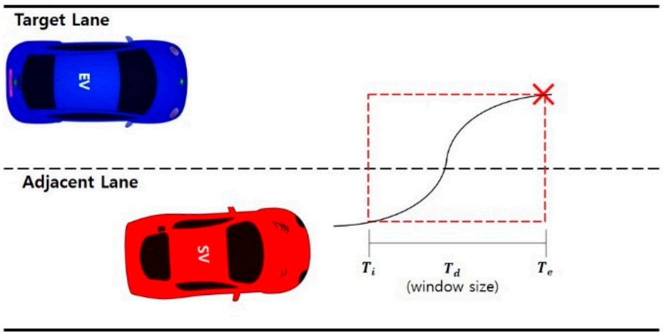
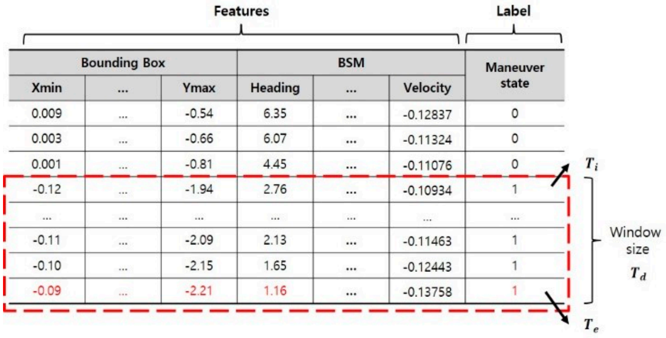


Figure 7. Define lane changing maneuvers.


Features
Label
T_i
Window size T_d
T_e

Bounding Box			BSM			Maneuver state
Xmin	...	Ymax	Heading	...	Velocity	
0.009	...	-0.54	6.35	...	-0.12837	0
0.003	...	-0.66	6.07	...	-0.11324	0
0.001	...	-0.81	4.45	...	-0.11076	0
-0.12	...	-1.94	2.76	...	-0.10934	1
...	...	...	...	...	...	...
-0.11	...	-2.09	2.13	...	-0.11463	1
-0.10	...	-2.15	1.65	...	-0.12443	1
-0.09	...	-2.21	1.16	...	-0.13758	1

Figure 8. Example dataset for lane change prediction.

4.3. Trajectory Prediction Model

4.3.1. LSTM

Machine learning algorithms for supervised learning assume that the input data are independent and identically distributed. However, to predict the trajectory of the SV, time series data is entered as an input value. This means that the input data is not independent. Therefore, we make a trajectory prediction model using an RNN that uses sequentially structured data as input data. However, it has the disadvantage of gradient vanishing or exploding for a long input sequence. To overcome this problem, LSTM was proposed [45]. As shown in Figure 9, the updated LSTM has a cell state idea that is regulated by three gates: the forget gate, input gate, and output gate [46].

The forget gate determines the information to be forgotten using the input data at the current time step x_t and the hidden state at the previous time step h_{t-1}, and is given as:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{7}$$

The input gate determines which cell state must be updated with new information. It is composed of i_t and $\widetilde{C}_t$ as described by:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{8}$$

$$\widetilde{C}_t = \tan \mathrm{h}(W_c x_t + U_c h_{t-1} + b_c) \tag{9}$$

where i_t determines how much information at the current time should contribute to the new cell state, and $\widetilde{C}_t$ proposes the new candidate cell state. As shown in Equation (10), the new cell state C_t is updated as described by:

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t \tag{10}$$

As shown in Equations (11) and (12), the hidden state h_t is calculated from the cell state C_t and the output gate. Output gate o_t determines the state of the cell that contains a large amount of information.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{11}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{12}$$

where W_f, U_f, W_i, U_i, W_C, U_C, W_o, and U_o are weight matrices connecting x_t, h_{t-1} to the three gates and the cell input, and where b_f, b_i, b_c, b_o are bias terms of the three gates and the cell input. The sigmoid function is represented by $\sigma(x)$, and tanh represents the hyperbolic tangent function.

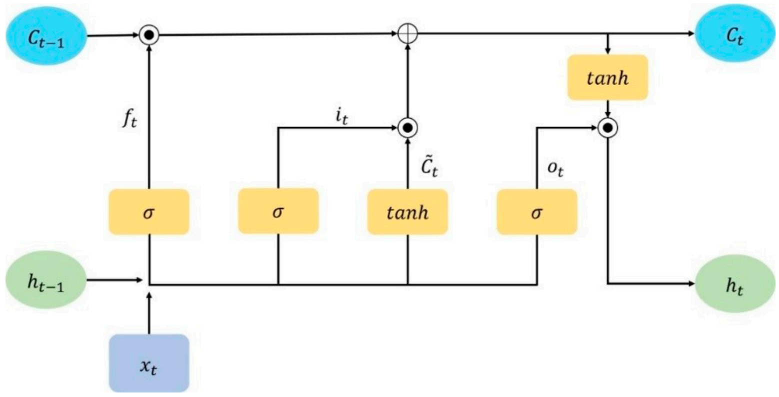


Figure 9. LSTM architecture.

4.3.2. LSTM Encoder-Decoder

The LSTM encoder-decoder architecture is also called a sequence-to-sequence model, and it converts an input data in a sequence form into an output data in a sequence form. This model is most used in the fields of machine translation, text summarization, and image captioning [47,48]. The trajectory prediction model of the SV also predicts a future trajectory sequence by training input sequence. Therefore, it has been widely used in the field of autonomous vehicles in recent years [24,49]. LSTM encoder-decoder is based on the RNN model and is divided into an encoder part and a decoder part. First, it receives an input value from the encoder part and creates a vector containing the information of the input value. After that, the decoder uses this vector to recursively generate an output value.

The LSTM encoder-decoder architecture of the trajectory prediction model proposed in our paper is shown in Figure 10. Our model defined the observation time of previous trajectory and the prediction time of future trajectory as 2 s. The input data is divided into information obtained through the camera and information obtained through V2V communication. The feature values obtained through the camera are the coordinates and width/height of the bounding box. Feature values obtained through V2V communication include relative coordinates, velocity, acceleration, heading, and steering angle values from among BSM information for SV.

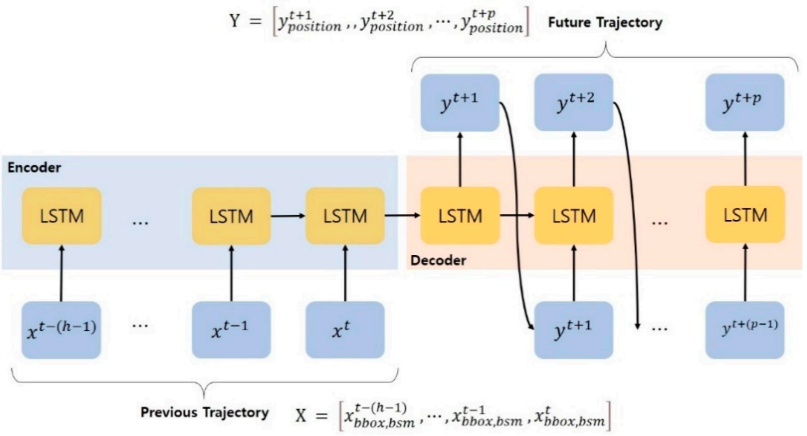


Figure 10. LSTM encoder-decoder architecture for trajectory prediction model.

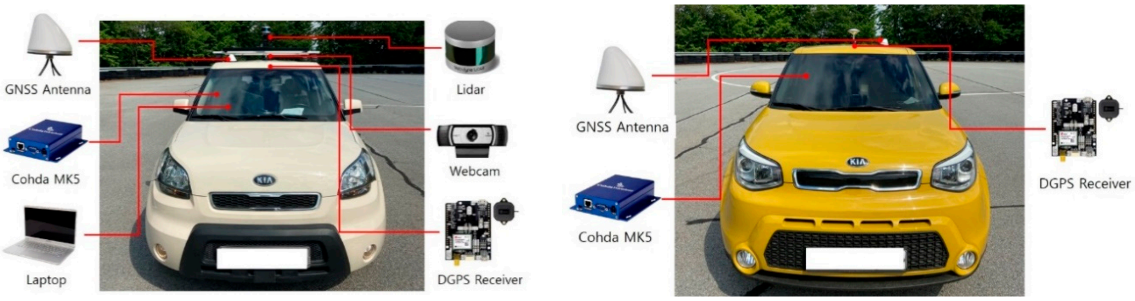
5. Experiment and Results

5.1. Vehicle Configuration

In our experiment, EV is equipped with sensors to obtain feature values to be used in the trajectory prediction model. We obtained the training dataset to be used for the model using two vehicles as shown in Figure 11. We obtained BSM of SV using Cohda MK5(Cohda Wireless, Wayville, Australia) as V2V communication device. However, since the GPS receiver of the Cohda MK5 provides low location accuracy, we improved the location accuracy by using a DGPS receiver that can use NTRIP correction information as shown in Figure 12. Although the improved location information can be transmitted/received through V2V communication, a location error occurs in the process of changing the global coordinate system to the local coordinate system as mentioned in Section 3. In order to correct the location error, we calculated the location of the SV using the point cloud obtained from the Velodyne LIDAR-VLP 16 sensor as shown in Figure 13.

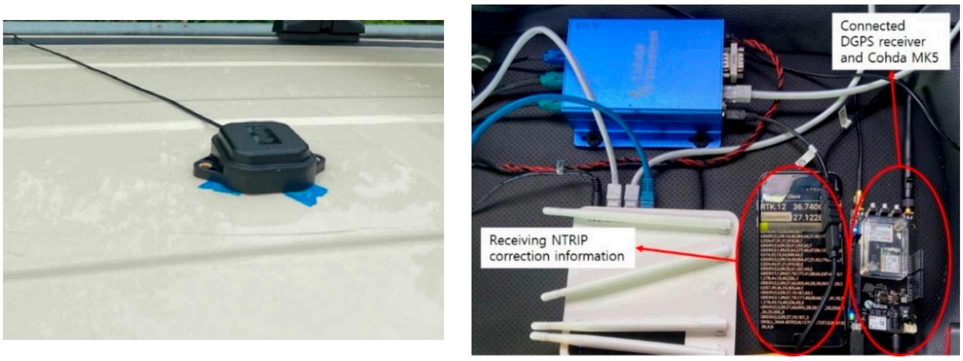
5.2. Dataset Collection

In the training dataset, the actual vehicle driving trajectory was collected in the testbed similar to a highway environment as shown in Figure 14. We did not set a specific trajectory to create various driving trajectories; rather, we defined only the scenario as shown in Figure 15, and the actual driving was done freely by the driver. This scenario is divided into four action types: acceleration and lane-keeping, acceleration and lane-changing, deceleration and lane-keeping, and deceleration and lane-changing. To avoid overfitting and to reflect various driver characteristics as much as possible, the number of trajectories was increased by reflecting the lateral inversion and longitudinal shift to the trajectory [50]. These four scenarios are sampled at the dataset sampling rate of 10 Hz. We used the robot operating system (ROS) to synchronize the time input data from camera sensor, lidar, and V2V communication [51]. As a result, we collected 932 trajectories from four scenarios, which consist of a total of 9428 instances as shown in Table 2.



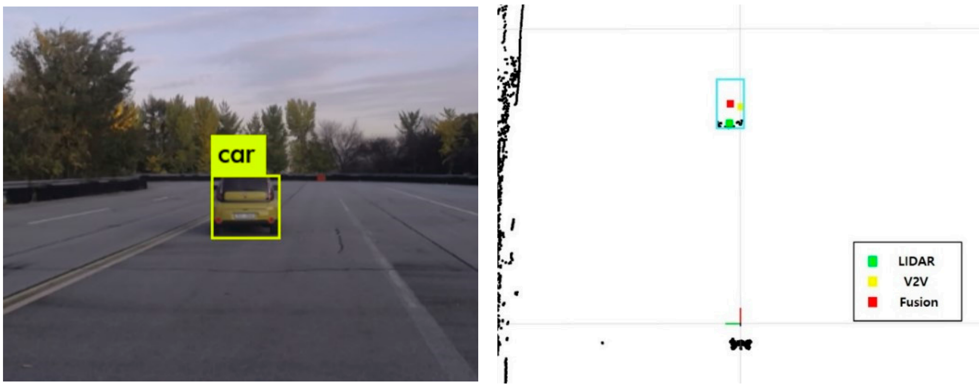
(**a**) (**b**)

Figure 11. (**a**) EV’s sensor configuration; (**b**) SV’s sensor configuration.



(**a**) (**b**)

Figure 12. (**a**) DGPS antenna mounted in the center of the vehicle; (**b**) environment inside the vehicle.



(**a**) (**b**)

Figure 13. Perception system result. (**a**) Bounding box obtained from camera sensor; (**b**) estimated location of SV obtained by fusion of LIDAR and V2V communications.

Figure 14. Testing ground at the Korea Automotive Technology Institute.

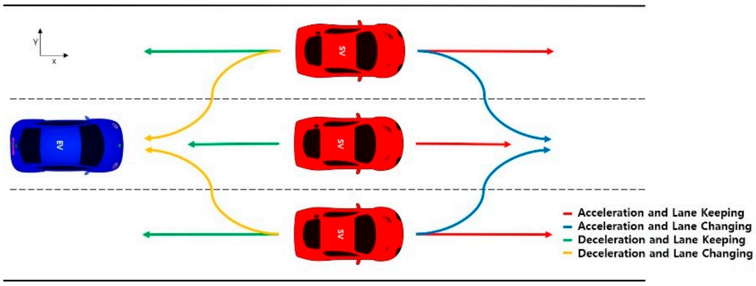

Y
x
SV
SV
SV
AV
Acceleration and Lane Keeping
Acceleration and Lane Changing
Deceleration and Lane Keeping
Deceleration and Lane Changing

Figure 15. Trajectory definition for data set: acceleration and lane keeping (same/adjacent), acceleration and lane changing (left/right).

Table 2. Dataset statistics.

Scenario	Number of Trajectory
Acceleration and Lane-Keeping	242
Acceleration and Lane-Changing	230
Deceleration and Lane-Keeping	231
Deceleration and Lane-Changing	229

5.3. Trajectory Prediction Model

5.3.1. Lane Change Prediction Model

We used an RF algorithm that ensembles multiple decision trees to predict the lane change intention of SV. RF has the advantage of not being sensitive to hyperparameters, but it has a disadvantage of low accuracy in the case of complex classification because it is an ensemble of decision trees. For example, the coordinate value of the bounding box for SV obtained through the camera sensor clearly shows the characteristic of moving to the center. In addition, the steering angle, which is part of the BSM information obtained through V2V communication, is an important feature value that SV uses to distinguish between lane-keeping and lane-changing. As can be seen in Figure 16, using the learning curve, we analyzed whether overfitting or underfitting was occurring.

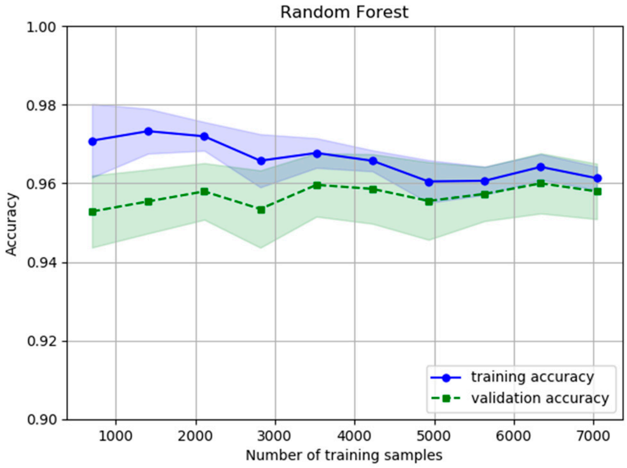


Figure 16. Learning curve of lane change prediction model using the RF.

5.3.2. Trajectory Prediction Model

We used the LSTM encoder-decoder architecture to predict the trajectory of the SV. The input vectors defined in Section 4.1 were embedded with a 256 unit fully connected (FC) layer prior to being input to the RNN cell of encoder architecture. Each fully connected layer used rectified linear unit (ReLU) as an activation function. The RNN cells of the encoder architecture used a two stacked LSTM of width 256, batch size of 100. Symmetrically, we also used two stacked LSTMs in the decoder architecture. Then, the output value from the LSTM was fed to 256 FC layers to calculate the final output vector. We defined both the observation time and prediction time as 2 s. Since the sampling period of the dataset is 100 ms, the sequence lengths of both the input vector and the output vector are 20 steps. To compare the results of the path prediction model using the LSTM encoder-decoder architecture, we used the path prediction results using only the stacked LSTM model and the CTRV model.

As shown in Figure 17a,b, the performance of the trajectory prediction model is excellent when driving straight ahead with constant acceleration and deceleration. However, when the SV attempts to change lanes while increasing velocity, the path prediction accuracy falls off as shown in Figure 17c. In Figure 17, the marker plots the trajectory of 2 s at intervals of 0.5 s, and the coordinate system is the x-axis in the vertical direction and the y-axis in the horizontal direction. Table 3 shows the Mean Absolute Error (MAE) of the

LSTM encoder-decoder architecture used in this paper, and the MAE of the CTRV model and the stacked LSTM model for comparison.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|(x,y)_t - (\hat{x},\ \hat{y})_t\right| \tag{13}$$

where (x,y) is the actual driving trajectory of the SV, $(\hat{x},\ \hat{y})$ is the driving trajectory predicted through three models and t is each timestep within the prediction horizon.

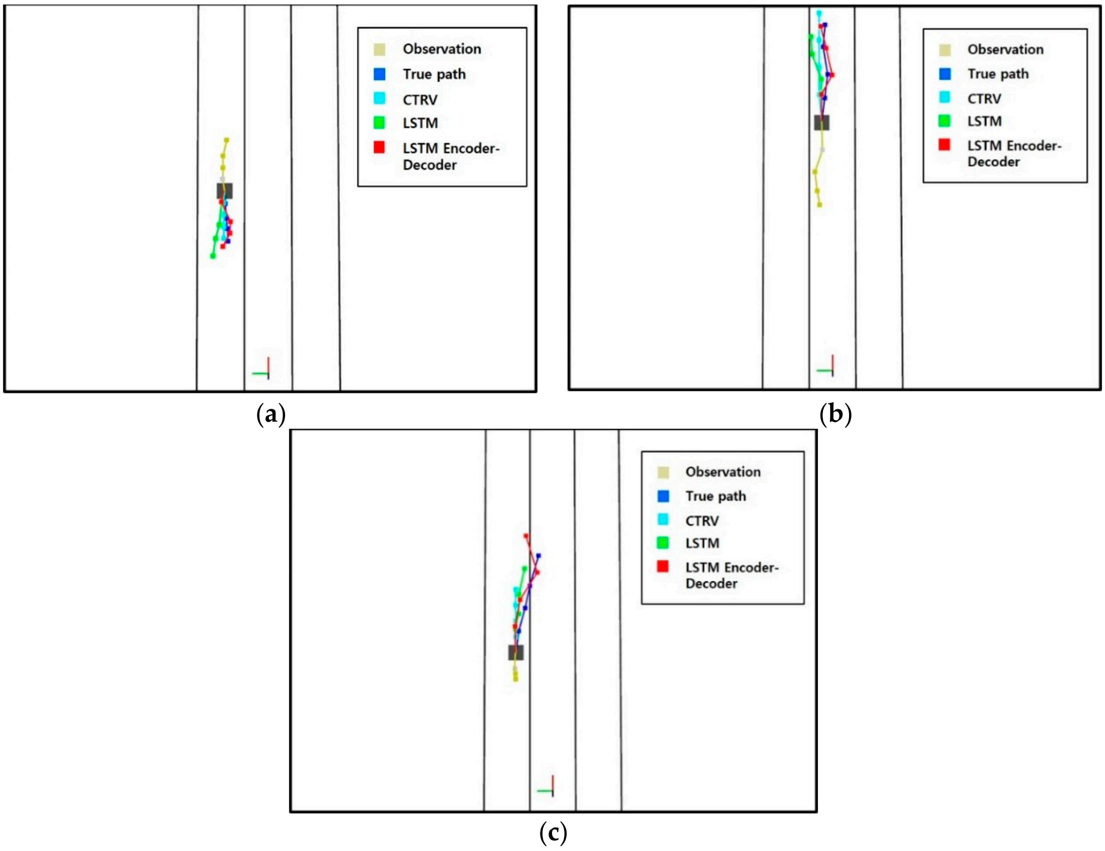


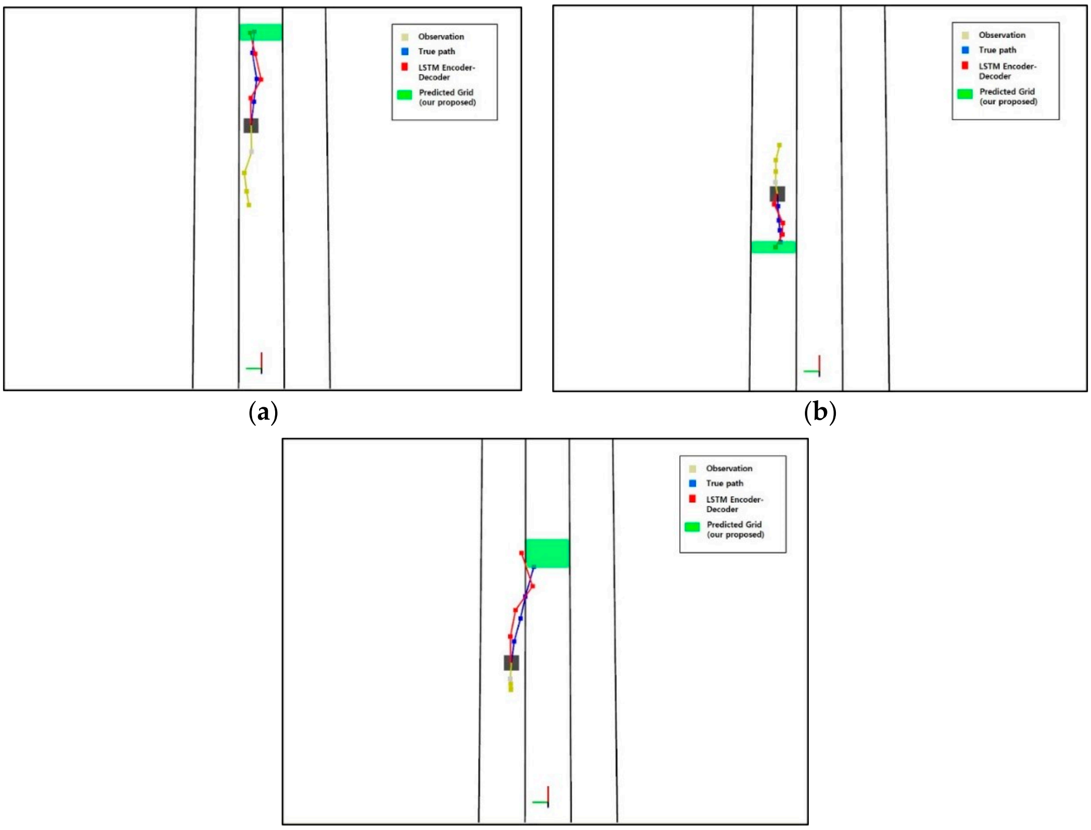
Figure 17. Comparison of trajectory prediction models: (**a**) deceleration and lane-keeping; (**b**) acceleration and lane-keeping; (**c**) acceleration and lane-changing.

Table 3. Trajectory prediction accuracy (MAE).

Prediction Horizon (s)	**Trajectory Prediction Model**		
	CTRV (m)	**LSTM (m)**	**LSTM Encoder-Decoder (m)**
0.5	0.21	0.62	0.58
1	0.52	1.19	0.82
1.5	1.84	1.42	1.23
2	2.11	1.81	1.47

5.3.3. Grid Prediction Model

The horizontal direction of the grid is predicted through the lane changing prediction model, which is the method we propose in this paper, and the trajectory prediction mode is used for vertical prediction. The length of the grid varies depending on the relative speed of SV and EV. For example, if the relative speed differs by 1 m/s, the width of the grid is defined as 1 m. The width of the grid was defined as 3 m, the width of the lane of the experimental test bed. Using RF, the lateral position of the grid was determined by predicting whether or not the lane change of the SV in the adjacent lane occurred. In addition, the longitudinal position of the grid was determined by the predicted longitudinal distance from the LSTM encoder-decoder architecture. Figure 18 shows the predicted grid where the SV will be located after 2 s.



(**a**) (**b**)

(**c**)

Figure 18. Trajectory prediction results using the proposed grid prediction model: (**a**) acceleration and lane-keeping; (**b**) deceleration and lane-keeping; (**c**) acceleration and lane-changing.

Table 4 shows the probability that a true trajectory is included in the predicted grid compared with other models. As shown in Equation (14), we used the accuracy to express this probability.

$$\text{ACC} = \frac{TP + TN}{FP + FN + TP + TN} \tag{14}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. We defined nine grid cells defined in advance as labels, and defined as true if the actual trajectory and the predicted trajectory of SV exist in the same cell, otherwise false. The horizontal length of the cell is the same as the lane width, and the vertical length is determined according to the relative speed of EV and SV. If the relative speed is 1 m/s, the vertical length of the cell is defined as 1 m. The result of trajectory prediction using the grid is rather worse for short prediction times such as 0.5 or 1 s because the lateral grid is determined by predicting the intention to change lanes before the lane change is completed. However, the accuracy of prediction after 1 s is far higher than that of other models.

Table 4. Trajectory prediction accuracy.

Prediction Horizon (s)	Trajectory Prediction Model		
	CTRV (%)	LSTM (%)	LSTM Encoder-Decoder (%)
0.5	87.21	86.74	87.18
1	88.32	88.81	89.46
1.5	82.24	87.01	91.83
2	80.93	86.84	90.87

6. Conclusions

In this paper, we presented grid prediction model using RF and LSTM encoder-decoder architecture. This model uses RF to predict the lane change intention of SV and determines the horizontal position of the grid. Then, the LSTM encoder-decoder architecture is used to predict the trajectory of the SV and determine the vertical position of the grid. In order to record the dataset to be used for training the proposed prediction model, we used a vehicle equipped with a V2V communication device, camera sensor and LIDAR. In this experiment, 932 trajectories were collected in the testbed, an environment similar to a highway, and the training data and test data were divided into 70 to 30. As a result of the experiment, the positional accuracy of the proposed model was high even after 1 s. Conversely, it showed relatively low position accuracy before 1 s because the horizontal position of the grid was determined by predicting lane changes before the SV crosses the lane.

The proposed method assumes that the bounding box of the SV is accurately acquired through image processing. It also calculates the exact location of SV using V2V communication and LIDAR. However, there is a limitation in that it is not possible to recognize SV in areas where V2V communication is not possible. Therefore, it is necessary to improve the recognition system by using only the LIDAR or through fusion of the LIDAR and camera sensors so that the SV can be recognized even in an area where V2V communication is impossible or a delay occurs. Since our test vehicle is equipped with only Velodyne LIDAR-VLP 16, we are planning a study to fuse camera sensors and LIDAR to improve the perception system [52,53]. As a future study, we plan to compare the accuracy of trajectory prediction for the surrounding vehicles by gradually increasing the number of SVs. In the case of object detection using a camera sensor in an actual road environment, there is also a disadvantage that objects not of interest are also detected. For example, the vehicle is detected not only for vehicles traveling in the same direction as the EV, but also for vehicles approaching from opposite lanes. Therefore, we are planning a study to increase the accuracy of the SV's bounding box and the algorithm for selecting the target vehicle.

It is very important to ensure real-time performance of CWS, which predicts the path of surrounding vehicles and warns in advance. We used a laptop equipped with a GTX 2070 graphic card and conducted an experiment in a driving environment with only one SV. In order to increase the maximum computing performance, video processing was performed using NVIDIA's CUDA, and research was conducted in the ROS environment. As a result of the experiment, there was no delay caused by a large amount of computation. As a future study, we plan to gradually increase the number of surrounding vehicles and compare the delay problems caused by the accuracy of trajectory prediction and the processing speed of the SVs.

The proposed model that predicts the trajectory of SV using the grid determined in this way is utilized as an advantage in CWS. The performance of CWS depends on the prediction accuracy of the lateral position. The proposed prediction model performs robustly in prediction accuracy for the lateral position. For future work, we are planning to create a system that predicts the trajectory of EV and sounds a prewarning to prevent collision when the SV and EV grids overlap.

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

Funding: This work was supported by the Technology Innovation Program (Development of AI-Based Autonomous Computing Modules and Demonstration of Services) funded by the Ministry of Trade, Industry and Energy (MOTIE), South Korea, under Grant 20005705.

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

References

1. Bengler, K.; Dietmayer, K.; Farber, B.; Maurer, M.; Stiller, C. Three decades of driver assistance systems: Review and future perspectives. *IEEE Intell. Transp. Syst. Mag.* **2014**, *6*, 6–22. [CrossRef]
2. Lefevre, S.; Vasquez, F.; Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH J.* **2014**, *1*, 1–14. [CrossRef]
3. Lin, C.F.; Ulsoy, A.G.; LeBlanc, D.J. Vehicle dynamics and external disturbance estimation for vehicle path prediction. *IEEE Trans. Control Syst. Technol.* **2000**, *8*, 508–518.
4. Brannstrom, M.; Coelingh, E.; Sjoberg, J. Model-based threat assessment for avoiding arbitrary vehicle collisions. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 658–669. [CrossRef]
5. Schubert, R.; Richter, E.; Wanielik, G. Comparison and evaluation of advanced motion models for vehicle tracking. In Proceedings of the 11th International Conference on Information Fusion, Cologne, Germany, 30 June–3 July 2008; pp. 1–6.
6. Tamke, A.; Dang, T.; Breuel, G. A flexible method for criticality assessment in driver assistance systems. In Proceedings of the 4th IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 697–702.
7. Laugier, C.; Paromtchik, I.E.; Perrollaz, M.; Yong, M.; Yoder, J.D.; Tay, C.; Mekhnacha, K.; Negre, A. Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety. *IEEE Intell. Transp. Syst. Mag.* **2011**, *3*, 4–19. [CrossRef]
8. Aoude, G.S.; Luders, B.D.; Lee, K.K.; Levine, D.S.; How, J.P. Threat assessment design for driver assistance system at intersections. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 19–22 September 2010; pp. 19–22.
9. Houenou, A.; Bonnifait, P.; Cherfaoui, V.; Yao, W. Vehicle trajectory prediction based on motion model and maneuver recognition. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 4363–4369.
10. Xie, G.; Gao, H.; Qian, L.; Huang, B.; Li, K.; Wang, J. Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5999–6008. [CrossRef]
11. McCall, J.C.; Wipf, D.P.; Trivedi, M.M.; Rao, B.D. Lane change intent analysis using robust operators and sparse Bayesian learning. *IEEE Trans. Intell. Transp. Syst.* **2007**, *3*, 431–440. [CrossRef]
12. Hou, Y.; Edara, P.; Sun, C. Modeling mandatory lane changing using Bayes classifier and decision trees. *IEEE Trans. Intell. Transp. Syst.* **2013**, *15*, 647–655. [CrossRef]
13. Greene, D.; Liu, J.; Reich, J.; Hirokawa, Y.; Shinagawa, A.; Ito, H.; Mikami, T. An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 942–953. [CrossRef]
14. Morris, B.; Doshi, A.; Trivedi, M. Lane change intent prediction for driver assistance: On-road design and evaluation. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 895–901.
15. Kumar, P.; Perrollaz, M.; Lefevre, S.; Laugier, C. Learning-based approach for online lane change intention prediction. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia, 23–36 June 2013; pp. 797–802.
16. Mandalia, H.M.; Salvucci, M.D.D. Using support vector machines for lane-change detection. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* **2005**, *49*, 1965–1969. [CrossRef]
17. Hermes, C.; Wohler, C.; Schenk, K.; Kummert, F. Long-term vehicle motion prediction. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 652–657.
18. Vasquez, D.; Fraichard, T.; Laugier, C. Growing hidden markov models: An incremental tool for learning and predicting human and vehicle motion. *Int. J. Robot. Res.* **2009**, *28*, 1486–1506. [CrossRef]
19. Tran, Q.; Firl, J. Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 8–11 June 2014; pp. 918–923.
20. Altché, F.; de La Fortelle, A. An LSTM network for highway trajectory prediction. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 353–359.
21. Deo, N.; Trivedi, M.M. Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1179–1184.

22. Zyner, A.; Worrall, S.; Nebot, E. Naturalistic driver intention and path prediction using recurrent neural networks. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1584–1594. [CrossRef]
23. Kim, B.; Kang, C.M.; Kim, J.; Lee, S.H.; Chung, C.C.; Choi, J.W. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Changshu, China, 26–30 June 2018; pp. 399–404.
24. Park, S.H.; Kim, B.; Kang, C.M.; Chung, C.C.; Choi, J.W. Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1672–1678.
25. Thiemann, C.; Treiber, M.; Kesting, A. Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data. *Transp. Res. Rec.* **2008**, *2088*, 90–101. [CrossRef]
26. Tan, H.-S.; Huang, J. DGPS-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 415–428. [CrossRef]
27. Weber, G.; Dettmering, D.; Gebhard, H. Networked transport of RTCM via internet protocol (NTRIP). In *A Window on the Future of Geodesy*; Springer: Heidelberg/Berlin, Germany, 2005; pp. 60–64.
28. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 8–14 December 2001; pp. 511–518.
29. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
30. Zhu, Q.; Yeh, M.C.; Cheng, K.T.; Avidan, S. Fast human detection using a cascade of histograms of oriented gradients. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; pp. 1491–1498.
31. LeCun, Y.; Boser, B.E.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.E.; Jackel, L.D. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990; pp. 396–404.
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
33. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
35. Redmon, J.; Farhadi, A. Yolov3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
36. *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)—Multi-Channel Operation*; IEEE Std/1609.4; IEEE: New York, NY, USA, 2016; No. 16264705.
37. *Dedicated Short Range Communications (DSRC) Message Set Dictionary*; SAE J2735; SAE International: Warrendale, PA, USA, 2016; No. J2735_200911.
38. Morsdorf, F.; Meier, E.; Kötz, B.; Itten, K.I.; Dobbertin, M.; Allgöwer, B. LIDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. *Remote Sens. Environ.* **2004**, *92*, 353–362. [CrossRef]
39. Tonini, M.; Abellan, A. Rockfall detection from terrestrial LiDAR point clouds: A clustering approach using R. *J. Spat. Inf. Sci.* **2014**, *2014*, 95–110. [CrossRef]
40. Wojke, N.; Häselich, M. Moving vehicle detection and tracking in unstructured environments. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 3082–3087.
41. Azim, A.; Aycard, O. Detection, classification and tracking of moving objects in a 3D environment. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Alcala de Henares, Spain, 3–7 June 2012; pp. 802–807.
42. Breiman, L. *Random Forests. Machine Learning*; Springer: Heidelberg/Berlin, Germany, 2001; pp. 5–32.
43. Dogan, Ü.; Edelbrunner, J.; Iossifidis, I. Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior. In Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics, Karon Beach, Thailand, 7–11 December 2011; pp. 1837–1843.
44. Motamedidehkordi, N.; Amini, S.; Hoffmann, S.; Busch, F.; Fitriyanti, M.R. Modeling tactical lane-change behavior for automated vehicles: A supervised machine learning approach. In Proceedings of the 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS 2017), Naples, Italy, 26–28 June 2017; pp. 268–273.
45. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **2006**, *9*, 1735–1780. [CrossRef]
46. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef] [PubMed]
47. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
48. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.

49. Zhao, Z.; Fang, H.; Jin, Z.; Qiu, Q. Gisnet: Graph-based information sharing network for vehicle trajectory prediction. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–7.
50. Deo, N.; Rangesh, A.; Trivedi, M.M. How would surround vehicles move? a unified framework for maneuver classification and motion prediction. *IEEE Trans. Intell. Veh.* **2018**, *3*, 129–140. [CrossRef]
51. Koubâa, A. *Robot Operating System (ROS)*; Springer: Heidelberg/Berlin, Germany, 2019.
52. Melotti, G.; Premebida, C.; Gonçalves, N. Multimodal deep-learning for object recognition combining camera and LIDAR data. In Proceedings of the 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 15–17 April 2020; pp. 177–182.
53. Wang, Y.; Chao, W.L.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8445–8453.