Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 4438861, 15 pages
https://doi.org/10.1155/2021/4438861

Research Article

Development of AI-Based Vehicle Detection and Tracking System for C-ITS Application

Sehyun Tak,[1] **Jong-Deok Lee**,[1] **Jeongheon Song**,[2] **and Sunghoon Kim**[1]

[1]*The Korea Transport Institute, 370 Sicheong-daero, Sejong-si 30147, Republic of Korea*
[2]*CAL Lab., HyperSensing Inc., 169-84 Gwahak-ro, Yuseong-gu, Daejeon 34133, Republic of Korea*

Correspondence should be addressed to Sunghoon Kim; sunghoon@koti.re.kr

Received 28 April 2021; Revised 7 July 2021; Accepted 11 August 2021; Published 19 August 2021

Academic Editor: Wen LIU

There are various means of monitoring traffic situations on roads. Due to the rise of artificial intelligence (AI) based image processing technology, there is a growing interest in developing traffic monitoring systems using camera vision data. This study provides a method for deriving traffic information using a camera installed at an intersection to improve the monitoring system for roads. The method uses a deep-learning-based approach (YOLOv4) for image processing for vehicle detection and vehicle type classification. Lane-by-lane vehicle trajectories are estimated by matching the detected vehicle locations with the high-definition map (HD map). Based on the estimated vehicle trajectories, the traffic volumes of each lane-by-lane traveling direction and queue lengths of each lane are estimated. The performance of the proposed method was tested with thousands of samples according to five different evaluation criteria: vehicle detection rate, vehicle type classification, trajectory prediction, traffic volume estimation, and queue length estimation. The results show a 99% vehicle detection performance with less than 20% errors in classifying vehicle types and estimating the lane-by-lane travel volume, which is reasonable. Hence, the method proposed in this study shows the feasibility of collecting detailed traffic information using a camera installed at an intersection. The approach of combining AI and HD map techniques is the main contribution of this study, which shows a high chance of improving current traffic monitoring systems.

1. Introduction

Urban road traffic is a complex phenomenon caused by interactions among various moving entities, such as vehicles and pedestrians. The growth in urban population during the past decades has raised the severity of urban traffic congestion, leading to socioeconomic and environmental problems in modern cities. To mitigate this issue, brisk trials have been conducted to apply intelligent transportation systems (ITS) in urban roads. In this regard, traffic monitoring is one of the most valuable functions of traffic management systems (TMSs). Particularly in advanced TMSs (ATMSs), real-time collection of precise information through traffic monitoring plays a crucial role for traffic managers when they develop various control strategies [1–3]. Furthermore, the detailed numerical status of real-time traffic such as lane-by-lane travel volume and queue length can be used as supplementary information for cooperative intelligent transportation system (C-ITS) operations based on autonomous vehicles [4, 5].

Traffic monitoring systems have been developed in various ways, and traffic information is collected indirectly or directly depending on the characteristics of a specific monitoring system. Indirect methods estimate traffic status such as travel volume and travel time within a road section based on the data samples collected via roadside units (RSU) or global positioning systems (GPS), which are instances of automatic vehicle identification (AVI) technologies [6–8]. However, the estimation performance of these methods is highly dependent on the market penetration rate (MPR) of equipped vehicles for vehicle-to-infrastructure (V2I) communication. On the contrary, direct methods measure the traffic conditions using point sensors such as loop detectors [9–11], radars [12–14], and video cameras [15, 16]. Loop

detectors have been widely used for traffic monitoring due to relatively higher reliability in collecting travel volume, occupancy, and spot speed, but their installation and maintenance complexity is higher because they are normally installed on road surfaces [17]. Radar-based monitoring systems are relatively easier to install, but the cost of the hardware itself is more expensive [18]. Moreover, the common limitation of both loop detectors and radar is difficulty in classifying vehicle types. However, cameras are relatively cheaper than radars, and camera-based monitoring systems are able to classify vehicle types [19]. They can also distinctively obtain traffic information in each lane of a road spot [17]. They have a high potential for extracting more detailed traffic information at a specific location, but it requires advanced image processing techniques to obtain reliable information, which is problematic. Automatic traffic data collection via camera-based monitoring systems can be operated at lower costs only when proper image processing techniques support the system.

Various methods have been proposed in several studies related to automatic image processing techniques. Some studies from the early 2000s had focused on improving the poor performance with respect to vehicle detection owing to several technical issues, such as segmentation of objects in the background and shadows [20], difficulties in detecting dark-colored vehicles [21], differences in day- and night-vision data [22], and influences of weather conditions [23]. An attempt to develop a technique to detect accidents automatically was also reported [24].

Recently, studies began to focus on using machine-learning or deep-learning techniques, and one of the most popular examples is the application of You Only Look Once (YOLO) to process traffic vision data [25]. YOLO has high applicability to real-time traffic monitoring based on its capability to process multiple images faster than conventional region-based convolutional neural networks (R-CNNs). With the aid of deep-learning techniques such as YOLO or faster R-CNN, the performance of detecting vehicles using real-time traffic vision data has been tried to improve in several studies. Their common purpose was to accurately count vehicles for estimating traffic conditions in specific road spots [26]. Some of them specifically focused on detecting vehicles in captured scenes with several objects (vehicles) with high density [27], while others focused on detecting small objects (vehicles) in complex scenes [28, 29]. Some studies have also attempted to distinctively detect road vehicles and pedestrians [30, 31].

Such object detection techniques have evolved into real-time visual object tracking approaches. Several studies have proposed methods for tracking multiple objects in time series based on convolutional neural networks (CNNs) [32–34]. There are also some examples of using kernelized correlation filter (KCF) for high-speed tracking of objects on roads and even in waterway traffic [35, 36]. Within the context of object tracking on roads, there were a few studies related to tracking moving vehicles particularly for the purpose of collecting more detailed traffic behaviors [37]. They have proposed methods for extracting and analyzing trajectories of multiple vehicles within a specified road spot for capturing lane-change events [38] or measuring the speeds of individual vehicles [39]. However, till now, only rough estimations have been conducted on trajectories without accurately measuring vehicle positions over time. For example, with the current machine-learning-based image processing techniques, a possibility of detecting multiple vehicles as a single object arises when they travel through similar paths and speeds, even though on different road lanes. Hence, it is still difficult to obtain an accurate trajectory of a vehicle by tracking the exact position of the road lane where the vehicle is located. Obtaining accurate trajectories of multiple vehicles would be advantageous to traffic managers intending to improve the accuracy of collecting travel volume or queue length values in each traveling direction at an intersection. Furthermore, it would enable us to obtain information on different road lanes, which can be useful for deeper analysis of traffic flow behavior and supporting autonomous vehicle operations.

Therefore, we present a method for deriving traffic information using a camera installed at an intersection for improving monitoring performance. The method uses a deep-learning-based approach for image processing for vehicle detection and vehicle type classification. Then, the method estimates lane-by-lane vehicle trajectories by matching the detected vehicle locations with the high-definition map (HD map). While estimating the vehicle trajectories, we attempt to reduce the error of estimating the center points of the bounding boxes in the images of vehicles to ensure proper performance of the HD map-matching process. Based on the estimated vehicle trajectories, the traffic volumes of each lane-by-lane traveling direction and queue lengths of each lane were estimated as well. In fact, this is not the first attempt to increase the accuracy of trajectory estimation to the lane level. The work in [40] had a similar purpose and approach but differs from the present study in that recent deep-learning techniques and HD map technology are combined for estimating vehicle positions accurately.

The remainder of this paper is organized as follows. Section 2 provides a description of the method of vehicle detection and classification, along with the method of matching the detected vehicle positions with the HD map for lane-by-lane trajectory estimation. Section 3 describes the settings for testing the performance of the proposed method, and Section 4 presents the test results. Section 5 concludes this paper and offers suggestions for further work.

2. AI-Based Vehicle Detection System at Intersection

2.1. Data Flow Framework. In fact, the image processing technology these days can easily identify a vehicle in a captured image, as long as the image resolution is sufficient. However, the focus of this study is on how to precisely extract traffic information upon multiple vehicles on roads rather than a single vehicle and how to deal with the extracted data from the traffic monitoring perspective. Hence, it is necessary to consider the data flow framework of the camera-based vehicle detection system.

Figure 1 shows the data flow framework of the artificial intelligence (AI) based vehicle detection system for C-ITS. As shown, the system consists of four components: roadside sensor, traffic monitoring center, RSU for communication, and an on-board unit (OBU) in vehicles. In this study, traffic cameras installed at intersections were considered as the main roadside sensors. First, the vision data of the traffic status at an intersection were collected in real-time via a roadside sensor and sent to a data collecting server in the traffic monitoring center. Then, using the vision data, the center conducted the vehicle detection task using the AI-based image processing technique. The information gathered from the vehicle detection task was then used to extract and predict the trajectories of vehicles. Then, the trajectory data information underwent the HD map-matching task to improve the prediction accuracy. The information message of the detected vehicles and their predicted trajectories were sent to the OBU in a subject vehicle via RSU using infrastructure-to-vehicle (I2V) communication. When a message was received, the collision risk of the subject vehicle could be calculated based on the predicted trajectories and also be displayed to the vehicle monitoring system. The status of the subject vehicle could be sent back to the traffic monitoring center via the RSU using V2I communication.

The framework described above provides two major advantages in terms of C-ITS operations. The first is that vehicle-to-vehicle collisions can be prevented by providing vehicles with their detection information traveling through intersections. Implementing a service that provides detailed information, such as vehicle location, speed, and abnormal status, is possible. In addition, it provides predictive information in seconds using the previously detected information. Second, a more detailed road status can be provided by extracting lane-by-lane traffic conditions near intersections. It is possible to provide a service that provides information on the traffic volume and vehicle queue of each lane. Furthermore, a service that detects illegally parked vehicles on streets can also be implemented. In this study, we aim to improve the advantages of the framework. The focus of this study is to develop methodologies for AI-based vehicle detection and HD map matching, which are the tasks of the traffic monitoring center described above.

2.2. AI-Based Vehicle Detection and Trajectory Prediction. In this study, a deep-learning algorithm is adopted using roadside sensors to extract object information such as vehicle location, movement trajectory, and vehicle speed at intersections and surrounding areas, and useful traffic information, such as traffic volume and queue length, is estimated. The proposed algorithm is based on vision data transmitted from the roadside sensors to a vision data collecting server located in the traffic monitoring center, and the predicted data are stored in a real-time database for real-time data communication. As shown in Figure 2, the proposed algorithm consists of (1) vehicle detection and classification with deep learning, (2) trajectory extraction, (3) trajectory correction, and (4) trajectory prediction, and the details are outlined as follows.

2.2.1. Vehicle Detection and Classification with Deep Learning. We used a deep-learning-based algorithm for vehicle detection as it has higher applicability to real-time traffic monitoring compared to other image processing techniques such as traditional labeling due to its capability of processing multiple images faster than others. The proposed system performs real-time detection of vehicle location and speed from the vision data sent from the vision data collecting server based on the YOLOv4 deep-learning algorithm and performs vehicle type classification. The YOLOv4 algorithm uses the state-of-the-art deep-learning method and is optimized, showing 10% improved performance for the detection accuracy index (MAP: mean average prediction) and a 12% improved detection speed index (FPS: frame per second) compared to YOLOv3, the previous version of the algorithm. In particular, YOLOv4 can process vision data with efficiency, enhancing its applicability in the traffic safety sector where detection, preprocessing, and warning message generation must be performed within 0.1 seconds.

In the process of vehicle detection and classification with deep learning, the algorithm processes vision data in frames and primarily generates vehicle type information such as cars, trucks, and buses and vehicle location information based on pixels. As for vehicle type information, data derived from YOLOv4 can be directly used, and additional separate training was performed based on the target site data to improve the accuracy of vehicle type information. Vehicle location information was generated based on the information of each vertex and the center point of the bounding box. This information was then converted into longitude and latitude coordinates based on the center point of the vehicle's bottom through correction.

2.2.2. Trajectory Extraction. The vision data collected from the roadside are distorted when converting 3D real-world images into 2D images. Because of this distortion, a significant error occurs between the actual physical coordinates and the image coordinates depending on the degree of vision data distortion when the location information detected in pixel units is converted directly into longitude and latitude coordinates. In this study, to remove this error, the corrected vision data were generated from the distorted vision data by inverse application of the camera intrinsic parameters extracted through its calibration. Note that the focal length, principal point, and distortion are the intrinsic parameters of the camera. The values of the intrinsic parameters were determined by projecting a 2D image into 3D world space. Also, note that an existing method is used for the distortion correcting process in this study. For a better understanding of the details of the distortion correcting method, refer to the work by Seong et al. [40]. The equation for correcting the vision data distortion is as follows:

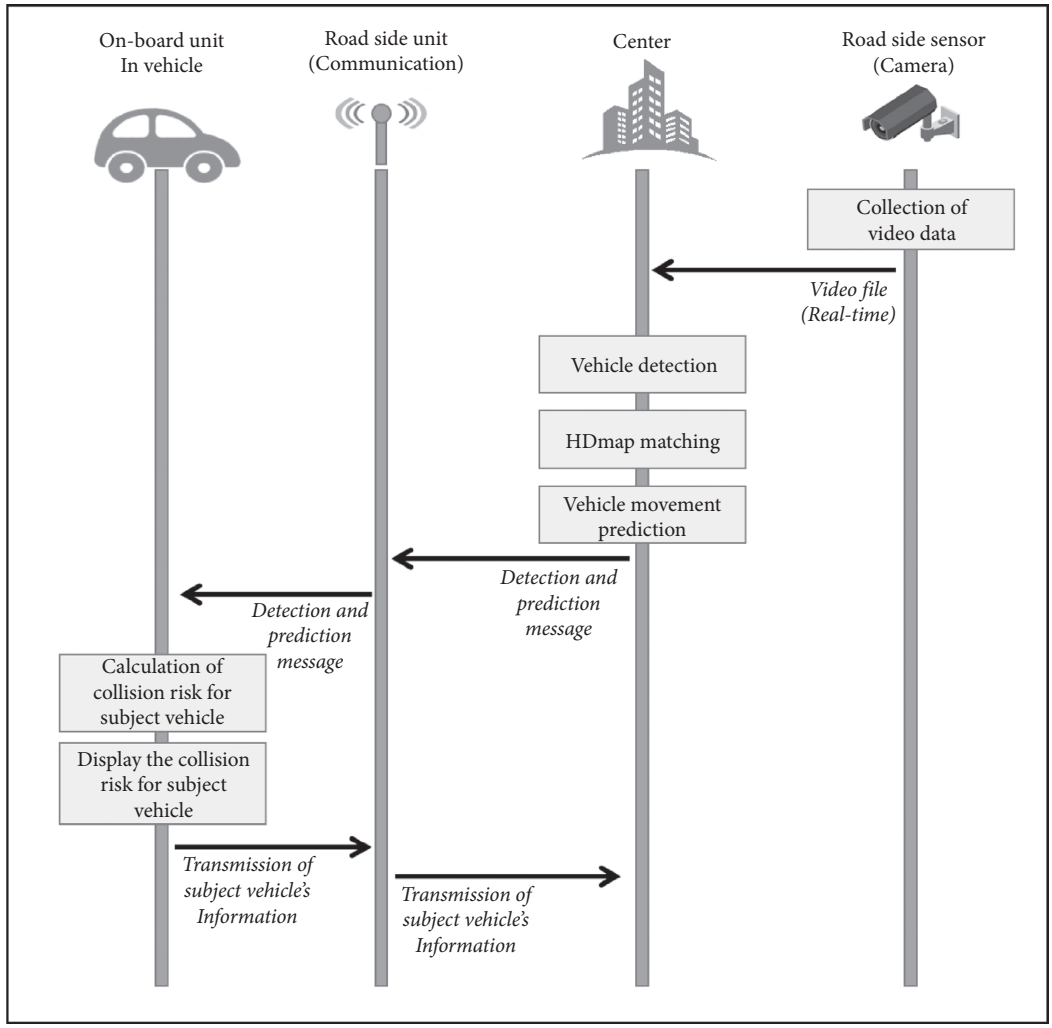


Figure 1: Data flow of AI-based Vehicle Detection system for C-ITS.

$$
\begin{bmatrix} x_v \\ y_v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \text{skew} & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_{\text{pu}} \\ y_{\text{pu}} \\ 1 \end{bmatrix},
$$

$$
r_u^2 = x_v^2 + y_v^2,
$$

$$
\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \left(1 + k_1 r_u^2 + k_2 r_u^4 + k_3 r_u^6\right) \begin{bmatrix} x_v \\ y_v \end{bmatrix} + \begin{bmatrix} 2p_1 x_v y_v + p_2\left(r_u^2 + 2x_v^2\right) \\ p_1\left(r_u^2 + 2y_v^2\right) + 2p_2 x_v y_v \end{bmatrix}, \tag{1}
$$

$$
\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \text{skew} & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix},
$$

where x_p, y_p are the pixel coordinates of an image, x_{pu}, y_{pu} are the pixel coordinates of the image corrected for distortion, x_n, y_n are the normalized planar coordinates with distortion, and x_v, y_v are the normalized planar coordinates with corrected distortion. Focal length: $f_x = 664.821$; $f_y = 668.333$. Principal point: $c_x = 350.377$; $c_y = 350.377$. Distortion: $k_1 = 0.278027$, $k_2 = 0.058863$, $p_1 = 0.000278$, and $p_2 = -0.001996$.

The vehicle location information detected from each image frame was expanded to a continuous frame for extracting the vehicle trajectory information and data for use in vehicle location correction. In the video images captured

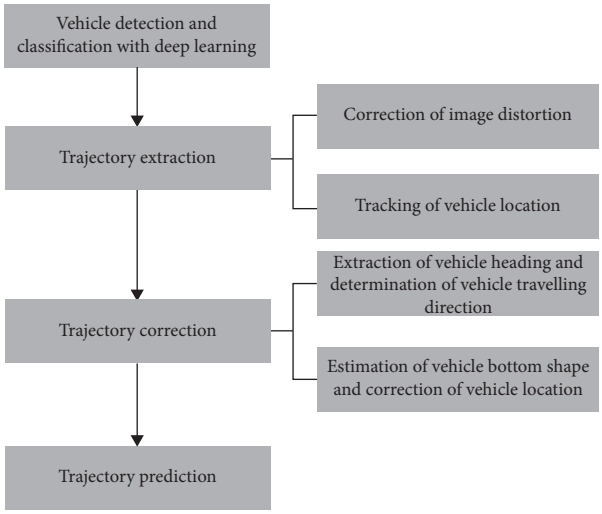


Figure 2: Algorithm for vehicle detection and trajectory prediction with deep learning.

by a camera, the similarity between the feature information of the object in the image frame is used to track the location change of the objects. To track the vehicle's location, its location and size in the previous frame were compared with those of the vehicle object detected in the next frame. As a result, the vehicle with the largest intersection of union (IOU) was classified as an identical vehicle to the vehicle existing in the previous frame; based on this classification, the continuous movement of the vehicle was tracked. In addition, if there was no intersection of union where the location and size of the detected object for a set frame (0.2 s) overlapped with that in the previous frame, the object was recognized as a new object, and a new vehicle tracking ID was assigned.

The pixel coordinates extracted from an image are calculated based on a matrix transformed using Transverse Mercator coordinates of four designated points in the HD map. The transformed matrix is derived by using homography that generalizes transformation relationships after obtaining coordinates corresponding to sample image coordinates. If there are four points (x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4) in a plane and these points are projected to another plane as (x_1', y_1'), (x_2', y_2'), (x_3', y_3'), and (x_4', y_4'), there exists a 3 by 3 homography matrix H satisfying relationship among these corresponding points. The camera image coordinates are converted to real-world coordinates using such a mechanism.

2.2.3. Trajectory Correction. In general, deep-learning-based vehicle detection extracts information in the form of a bounding box, and the central point of the bounding box represents the overall vehicle location information. However, as shown in the example in Figure 3, when vehicle location information is extracted with reference to the center point of the bounding box, the result differs from the location with reference to the center point of the vehicle bottom, which is the actual required information for traffic monitoring. In addition, when the center point is estimated

Figure 3: Comparison before (red-colored dot) and after (orange-colored dot) the trajectory correction.

based on the bounding box, an error occurs in the estimated position according to the heading shown (by captured angle) in the vehicle image. This type of error can lead to another error in trajectory prediction. This subsequent error can lower the performance of the HD map-matching process, which deals with extracting lane-by-lane traffic information later. Furthermore, if we assume that the trajectory prediction with such an error is utilized in a vehicle's collision warning or avoidance system, it can also lead to insufficient performance of the safety system. Hence, it is necessary to give an effort in reducing the errors while estimating the center point of the bounding box.

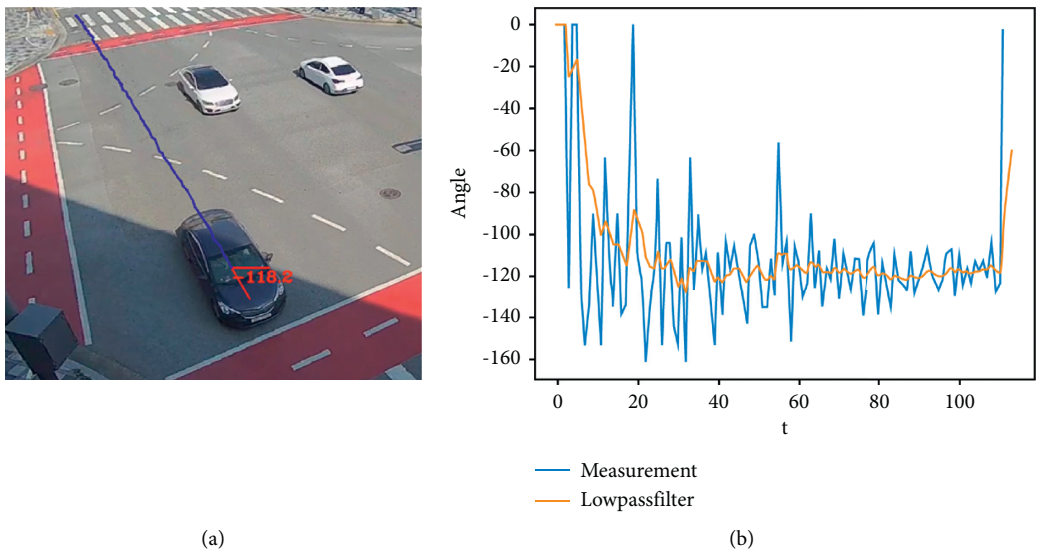
In this study, to reduce the error in center point estimation, real-time correction of vehicle location was performed through the following two steps: (1) extracting the heading and determining the traveling direction of the vehicle and (2) estimating the shape of the vehicle bottom and correcting the location.

For the first task, the vehicle heading was obtained using the pixel coordinates detected in the vision data collected from the road (the bounding box center point value) and the pixel coordinates of the previous frame, as shown in Figure 4. The heading of a vehicle is extracted through the following steps: (1) The vehicle position of the previous image frame and the position of the current image frame are converted into coordinates using a transformation matrix. (2) The angle formed by the two positions is calculated using the Pythagorean equation, and the distance between the two positions is calculated using the coordinate values. The extracted heading for each frame was corrected based on the low-pass filter as follows:

$$\overline{z_n} = \sigma \cdot \overline{z_{n-1}} + (1 - \sigma) \cdot z_n, \tag{2}$$

where $\overline{z_n}$ is the corrected heading, $\overline{z_{n-1}}$ is the heading at previous time, z_n is the heading at current time, and σ is the weight.

The vehicle traveling direction and the vertical direction are derived using the heading obtained from the real-time estimation and the detected pixel coordinates. Figure 4(a) shows the corrected results of the low-pass filter. In Figure 4(b), the orange and blue colored lines represent the



(a) (b)

Figure 4: Example of vehicle heading estimation. (a) Example image. (b) Correction result (example).

filtered and raw data, respectively. Noisy data points and variation in heading information are smoothed using a low-pass filter.

Based on the previously derived information of vehicle traveling direction and vehicle type, the shape of the vehicle bottom was estimated, as shown in Figure 3. Because the vehicle height varies depending on the type, the shape of the bottom surface within the bounding box is estimated by applying the average vehicle height per vehicle type. The bottom surface information is estimated based on the following steps: (1) The center points of the bounding boxes in the previous image frame and in the current image frame are converted into Transverse Mercator coordinates. (2) Since the vector formed by the two center points is the moving direction of the vehicle, a hypothetical vector perpendicular to the moving direction is drawn to create a rectangular vehicle bottoms shape (assuming that vehicles have a rectangular shape from the top view). (3) Let h_{camera} be the height between camera and ground surface, h_{vehicle} be the height of a vehicle, d_1 be the distance on the surface between camera and vehicle, and d_2 be the distance on the surface between the camera and point where the line connecting between the camera and the top of the vehicle meets the surface. Here, h_{camera}, h_{vehicle}, and d_2 are directly obtained from image data, and d_1 then can be calculated by the triangle proportional theorem. Note that the height of the vehicle is assumed to be half of the actual height because the center point of the bounding box detected in the image is half the actual height in usual. Based on this method, the four corner points (in 3D coordinates) of the vehicle bottom are estimated. (4) The 3D coordinates of the vehicle bottom (a', b', c', d') are then converted into the image coordinates (a'', b'', c'', d'') using an inverse transformation matrix, and this finalizes estimating the vehicle bottom. The center point information of the vehicle's bottom surface is extracted based on the estimated pixel information of the bottom surface, and the final pixel-based location information of the vehicle is derived based on this information.

2.2.4. Trajectory Prediction. Using the previously derived real-time trajectory data of the vehicle, the upcoming vehicle trajectory information from 1 to 3 seconds was estimated. Location information for each time slot was used to estimate the future trajectory of the vehicle. In addition, a polynomial curve fitting algorithm was used, as shown in the following equation, by applying a linear equation if the past data is a vehicle traveling forward or a quadratic equation for a turning vehicle, to extract the future location of the vehicle.

$$y(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_n + 1,$$

$$\begin{pmatrix} x_1^n & x_1^{n-1} & \cdots & 1 \\ x_1^n & x_1^{n-1} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ x_m^n & x_m^{n-1} & \cdots & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_1 \\ \vdots \\ p_{n+1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}. \quad (3)$$

When estimating the future location of the vehicle based on the detected vehicle location information alone, the result showed that the prediction performance is decreased at the intersection approach where a fewer number of points exist in the trajectory data. To address this limitation, the HD map previously built at the intersection was used, as shown by the solid black lines in Figure 5. Using the location information per link in the HD map, the future vehicle location was estimated assuming that the vehicle trajectory will follow the shape of the HD map link, and the estimated result is shown in Figure 5. The blue solid line represents the ground truth, the green- and blue-dotted lines represent the link of the HD map where the detected vehicle is assigned, and the red-dotted line represents the estimated future location of the vehicle.

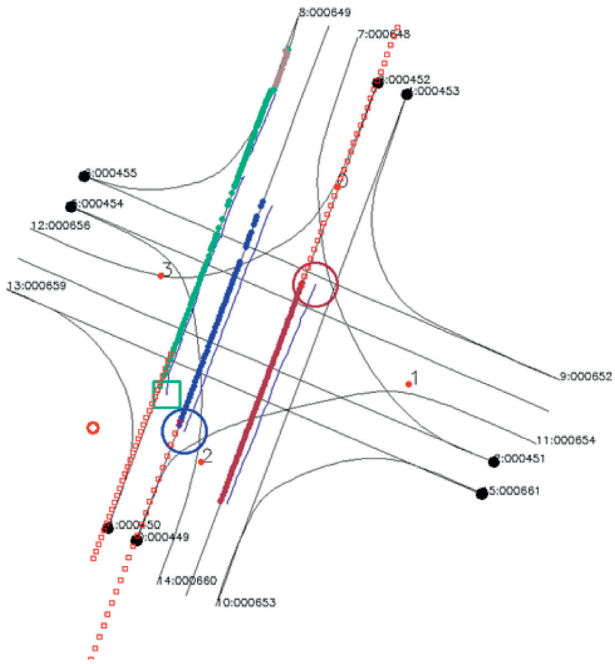


Figure 5: Example of vehicle location prediction.

2.3. Provision of V2X Communication-Based Detection Information

2.3.1. Generation of HD Map-Based Information. The current C-ITS provides information such as unforeseen incidents or accidents via messages that include longitude and latitude data. For such type of information, C-ITS has an advantage in terms of general use of information but is disadvantageous when the number of messages increases sharply with the increase in the number of related pieces of information. Furthermore, in the case of existing C-ITS based on location information, the computational load increases rapidly as the number of messages increases when matching the predicted vehicle trajectory information and point of event occurrence for each event. What is worse in the case of predicting trajectory based on the past trajectory is that the accuracy decreases at curved sections and intersections, leading to reduced accuracy when matching events. Therefore, in this study, to overcome the limitation in sending location information based on longitude and latitude, the AI-based detection and prediction information provided in the previous subsection was combined with the HD map link information, as shown in Figure 6(a).

Figure 6(b) shows the HD map link allocation algorithm. First, in the process of extracting HD map link information, information such as the length, linearity, and type of link and the longitude and latitude of the start and end points are extracted from the link attribute information of the HD map. This information of the HD map is compared with the detected location coordinates of the vehicle, and matching is performed with the nearest link, extracting the lane on which the vehicle is currently traveling. Figure 7 shows an example of the HD map link allocation based on the trajectories of the forward-traveling vehicle and turning vehicle. As shown in the figure, information on whether the vehicle travels forward or turns is extracted based on the vehicle trajectory for the past 1 s. Based on this information, if the vehicle is determined to be traveling forward, links with forward-type traveling are extracted from the HD map links, and the extracted candidate links and vehicle trajectory for 1 s are matched based on the start and end points, thereby extracting the HD map link with the closest matching. Finally, the HD map link extracted based on the distance is compared with the heading of the vehicle traveling direction, and when the latter shows consistency within a set threshold, the HD map link is allocated.

To enhance the applicability of the extracted information based on AI, the information extracted from the vision sensor is allocated in HD map link units. Then, the number of vehicles present in the link representing density, the most necessary information in traffic management, and queue length information are generated by the link. The density is calculated as the difference $(n_{in} - n_{out})$ between the number of vehicles entering the starting point (n_{in}) and that leaving the end point of the link (n_{out}). As for the queue length of a vehicle, when the average speed over the last 1 s is smaller than the set speed for each HD map link, the corresponding vehicle is classified as the vehicle in the queue. To improve the applicability of the information, the queue length is expressed based on the offset of the HD map. For example, if the length of the HD map link is 50 m, the start point of the link is set to 0, and the end point of the link is set to 50 based on the vehicle traveling direction. Based on these values, when the vehicle queue length is 20 m from the end point of the link, the start point of the queue is offset by 30, and the end point by 50.

2.3.2. Data Design for V2X Communication-Based Information Provision. Data converted based on the link format of the HD map are stored in the server in the format shown in Tables 1 and 2 to be utilized in messages in C-ITS in the future. Table 1 shows the storage format of vehicle information, which is used for storing and sharing object information (vehicle type, longitude, and latitude coordinates) extracted from AI. However, to improve the applicability of the information and accuracy of matching with the vehicle trajectory, the information allocated to the HD map link is combined. In addition, providing predicted information of vehicle objects based on the HD map link ID facilitates the calculation of the probability of collision in the future traveling direction of an autonomous vehicle.

Table 2 shows the storage format of data, primarily processed to facilitate the application of the information extracted from AI-based detection information to the traffic management field. As described above, information of the number of vehicles present in the link (density), queue length information, and average speed information is generated with reference to the HD map link. Similar to the storage format of vehicle information (Table 1), the predicted information is provided to facilitate the calculation of the collision probability in the future traveling direction of

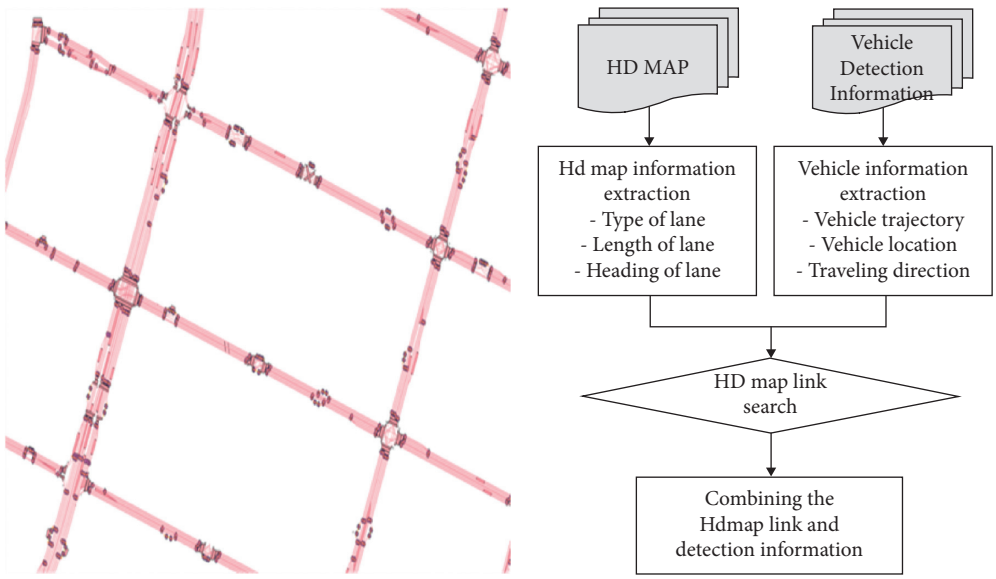


Figure 6: HD map example for generation of HD map-based information of the target site (a) and HD map link allocation algorithm (b).

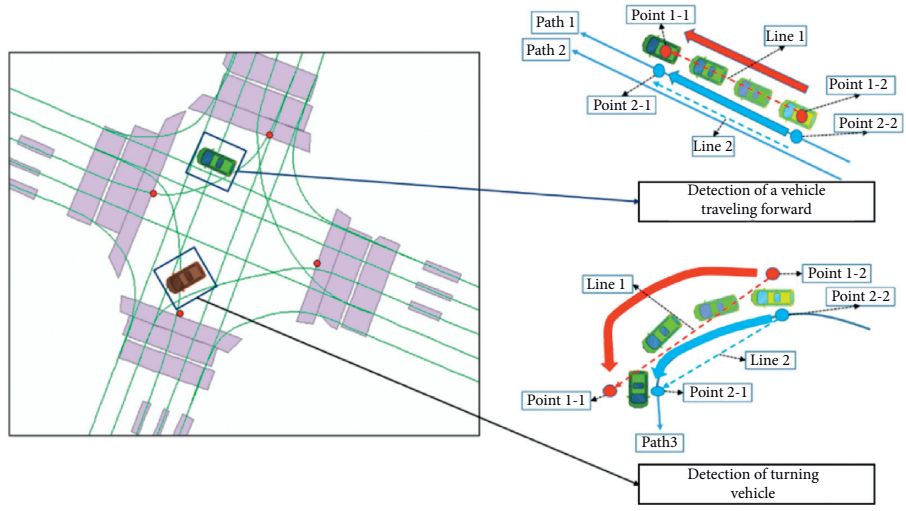


Figure 7: Example of road lane allocation for a vehicle traveling forward and a turning vehicle.

an autonomous vehicle. As shown in Tables 1 and 2, not only does the proposed system enhance the applicability of information by actively utilizing HD map links, but also the object extraction and traffic-related information are provided in combination with a similar format considering the need for other information attributes depending on the situation.

3. Target Site for Application of the Proposed Method and Evaluation

3.1. Target Site. Figure 8 shows the target site for applying the AI-based vehicle detection and prediction technique proposed in this study. The proposed system was evaluated using data collected for three days, and data for accuracy verification were generated in two steps as follows. First, the ground truth data for calculating the accuracy of vehicle location information were generated using a drone, by capturing the same area as the image data collected from the roadside vision sensor and collecting vertical images. Second, information such as the vehicle type, number of vehicles in a link, and queue length was generated based on a field survey, and the vehicle type and the number of vehicles were manually counted from visual observation of image data. To prevent human errors in counting, a cross-check and final check were performed using labeled image data.

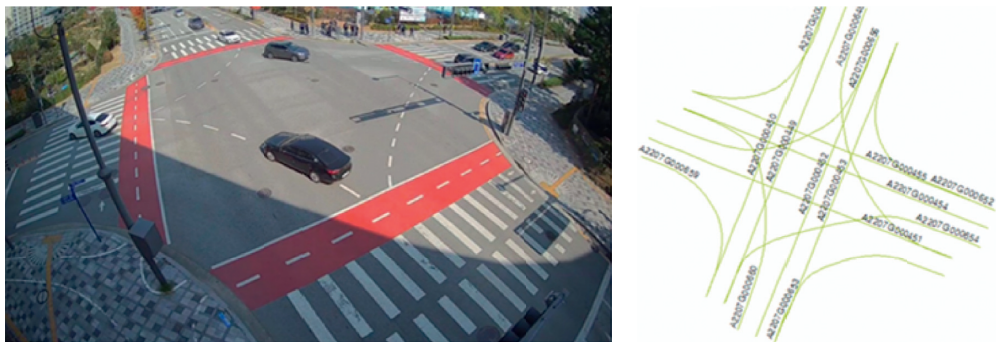
3.2. Accuracy Evaluation. Descriptions of how we evaluate the performance of the proposed methodologies are provided in this subsection, which is based on five different

Table 1: Vehicle information storage type.

Field name		Description
LinkID		Serial number of link ID in road central line of HD map in the detection area
Timestamp		Vehicle detection time: Unix timestamp (UTC) of accuracy in milliseconds
ObjectID		Object ID
Vehicletype		Vehicle type
Vehicletypeprob		Vehicle type probability
Objectstatus		Normal/abnormal traveling status (abnormal when not in motion for a certain period of time)
Offset		Offset of event point for link ID
Posdistance		Distance between the present HD map link and the extracted coordinates in longitude/latitude
Poslong		Longitude
Poslat		Latitude
Speed		Speed of detected vehicle (km/h)
Heading		Heading (°)
Object prediction	Index	Index (0–29 for 3 s prediction in units of 0.1 s)
	Timestamp	Vehicle detection time (prediction)
	Poslong	Longitude
	Poslat	Latitude
	Speed	Speed (km/h)
	Heading	Heading (°)
	LinkID	Link ID of predicted location at the timestamp of the detected object
	Offset	Offset of link ID of predicted location at the timestamp of the detected object

Table 2: Traffic information storage format including the number of vehicles on the link and queue length.

Field name		Description
LinkID		Serial number of link ID in road central line of HD map in the detection area
Timestamp		Vehicle detection time (present)
Avgspeed		Average speed of vehicles with link ID
Linktraveltime		Difference between the entry and exit time of vehicle
Numvehicle		Number of vehicles for the applicable link (present)
Object status	ObjectID	ID of an object with abnormal driving
	Offset	Offset of event point for link ID
Queue	QueueID	Queue event ID
	Offsetstart	Start point of offset of event point for link ID
	Offsetend	End point of offset of event point for link ID
Road prediction	Index	Index (0–29 for 3 s prediction in units of 0.1 s)
	Timestamp	Vehicle detection time (prediction)
	Numvehicle	Number of vehicles for the applicable link (prediction)
	Avgspeed	Average vehicle speed (prediction)



(a) (b)

Figure 8: (a) Example of roadside sensor screenshot. (b) Example of HD map.

evaluation criteria: vehicle detection rate, vehicle type classification, trajectory prediction, traffic volume estimation, and queue length estimation.

3.2.1. Accuracy of Vehicle Detection and Classification. To evaluate the vehicle detection performance, the detection rate was calculated to determine whether all vehicles were

successfully detected regardless of vehicle type. As in equation (4), it is defined as the ratio of the total number of detected objects to that of ground truths:

$$\text{detection rate} = \frac{\text{total number of detected objects}}{\text{total number of ground truths}}. \quad (4)$$

Vehicle classification performance is evaluated through MAP, which is a performance evaluation index widely used in the field of computer vision. MAP is the mean of the average precision (AP) values of each vehicle type. The AP represents the performance of the classification algorithm as a single value, and it is calculated as the area below the graph line in the precision-recall graph. As in equation (5), the precision is calculated as the ratio of the number of correct answers for vehicle type classification (true positives) to that of all detected vehicles (sum of true and false positives). The recall is calculated as the ratio of the number of correct answers (true positives) to that of all ground truths (sum of true positives and false negatives), as shown in equation (5). Precision and recall are inversely related to each other. Hence, the changes in such a relationship are analyzed to properly evaluate the overall performance of the proposed method.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{all detection}}, \quad (5)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{all ground truths}}. \quad (6)$$

3.2.2. Accuracy of Vehicle Trajectory Estimation. The performance of the vehicle trajectory prediction was evaluated by comparing the predicted and actual trajectories. The predicted trajectory is the set of coordinates within an intersection derived by the AI-based detection technique, while the actual trajectory is that directly generated from the image data. The average Euclidean distance is used for calculating the prediction accuracy, as shown in the following equation:

$$\text{average Euclidean distance} = \frac{\sqrt{\sum_{t=1}^{N}\left(x_{a,t} - x_{p,t}\right)^2 + \left(y_{a,t} - y_{p,t}\right)^2}}{N}, \quad (7)$$

where N is the number of sets of t for the comparison, $(x_{a,t}, y_{a,t})$ are the actual coordinates of the vehicle location at t, and $(x_{p,t}, y_{p,t})$ are the predicted coordinates at t.

3.2.3. Accuracy of Traffic Volume Estimation. Traffic volume was estimated by comparing the number of vehicles counted by the image processing (estimated value) technique and that counted manually (actual value). The evaluation was performed by calculating the root mean square error (RMSE) and mean absolute percentage error (MAPE). The former is used to check the degree of difference between the estimated and actual values, which can be calculated using the following equation:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}, \quad (8)$$

where n is the number of data points for the comparison, y_i is the i-th predicted value, and $\hat{y}_i$ is the i-th actual value of the traffic volume.

However, RMSE is highly influenced by the size of the estimation subject (scale-dependent errors), and it may emphasize only greater errors than the small ones. Hence, we calculate MAPE as well, which is independent of the scale of the estimation subject and can be calculated using equation (9):

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{A_i}\right|, \quad (9)$$

where n is the number of data points for the comparison, A_i is the i-th predicted value, and F_i is the i-th actual value of the traffic volume.

3.2.4. Accuracy of Queue Length Estimation. The evaluation of the performance of the queue length estimation is similar to that of the traffic volume estimation. This is done by comparing the queue length in meters derived by the image processing (estimated value) technique and that collected from a drone image (actual value). Here, we calculate the RMSE of the queue length estimation using equation (8), where $\hat{y}_i$ is the i-th actual value of queue length. We also calculate the MAPE of the queue length estimation using equation (9), where F_i is the i-th actual value of the queue length.

4. Result of Applying AI-Based Vehicle Detection and Trajectory Prediction

4.1. Accuracy of Vehicle Detection and Classification. Figure 9 shows an example of vehicle detection using the proposed training model based on YOLOv4. The system detects vehicles within the detection range and saves the results of the vehicle classification and coordinates of the bounding box as an image file (∗ .jpg) and data files (∗ .txt), using the same filename. By using the data collected by the drone (considered as actual data) and that extracted by the classification model, the performance evaluation is performed with the detection rate and MAP described in the previous section. The number of tested samples was 6,804. As a result, the detection rate was 99%, indicating that it

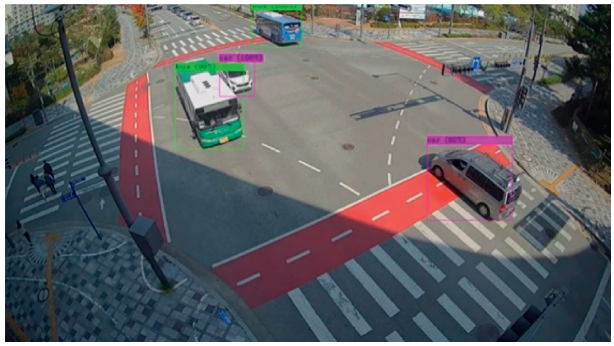


Figure 9: Vehicle detection by the proposed method based on YOLOv4.

could judge the detected objects as vehicles very well. In terms of vehicle type classification, the MAP value was 95% for cars, 87% for trucks, and 81% for buses.

4.2. Accuracy of Vehicle Trajectory Extraction. In vehicle trajectory extraction, during preprocessing, the locations of a vehicle are extracted at each frame by the proposed method. Then, the locations of the vehicles are projected onto the drone image. By matching the vehicle locations of the drone image and those from the proposed method to the same coordinates, it is determined that the vehicle area overlaps the most with the same vehicle. As shown in the figure, the vehicle locations extracted by the proposed method (red box area) are projected onto the drone image, where the actual vehicle locations are displayed in the drone image (blue box area) to determine them as the same vehicle. The performance of vehicle trajectory prediction is evaluated using these two different vehicle trajectories, as described in the previous section. The number of tested samples was 60,531. As a result, the average Euclidean distance was 1.138 m.

4.3. Accuracy of Traffic Volume Estimation. Figure 10 shows an example of comparing drone and camera images for the performance evaluation of traffic volume estimation. The test area is the blue box area within an intersection. As shown in the figure, the identification names are assigned for each in/out lane, and the pairs of the lane-by-lane travel directions of the vehicles can be seen in Tables 3 and 4. Then, the number of vehicles in each lane-by-lane traveling direction is counted from the drone images manually to obtain the actual data. On the contrary, the proposed method extracts the number of vehicles in each lane traveling direction based on the camera images to obtain the estimated data. Table 3 shows the traffic volume in each traveling direction counted from the drone images, and Table 4 shows that extracted from the camera data. In these tables, the notations in the second column represent the identification numbers of departure lanes (from I_1 to I_12) in approaching roads (from Road_1 to Road_4). However, those in the second row are the identification numbers of arrival lanes (from O_1 to O_12) in the roads in each direction. For example, if some vehicles pass through the intersection from the right-most lane of Road_1 (I_1) to the left-most lane of Raod_3 (O_5) and they are counted as 5, we record the counted number as shown in the tables. Hence, the entire table represents the lane-by-lane vehicle count values (travel volumes) of all departure and arrival pairs. The unknown in the latter table is the case when the camera-based system fails to detect a vehicle. When comparing the results of the two, the RMSE is 4.20 vehicles, and the MAPE is 16.41%.

4.4. Accuracy of Queue Length Derivation. Figure 11 shows an example of a drone image for queue length derivation, which was also performed manually. A person selects the starting and ending points of the vehicles within the delayed section on the road. Then, data containing the bounding box information of the vehicles at the starting and end points, map coordinates, and queue length within the image are saved. Using the information from these data, the true value of the queue length is calculated by converting the values into the real-world scale, which is considered the actual queue length. However, the proposed method directly derives the queue length through HD map matching to obtain the estimated data, which is compared with the actual queue length from the drone image. The number of tested samples was 62,205. Comparing the two, the RMSE is 2.37 m, and the MAPE value is 13.25%.

4.5. Comprehensive Evaluation. The overall performance of the proposed method is presented in Table 5. As described in the previous subsections, the detection rate is the total number of detected objects over the total number of ground truths, and a successful detection performance of 99% for 6,804 attempts is achieved, which can be judged to be highly consistent. The performance of the vehicle classification is performed in terms of MAP. With 6,804 test samples, the MAP values were 95%, 87%, and 81% for cars, trucks, and buses, respectively. Hence, the proposed method also shows reasonable performance in classifying the vehicle types. In terms of trajectory prediction, the average Euclidean distance was 1.138 m when 60,531 samples were tested. Such a low degree of error indicates the high performance of the proposed method. In terms of both traffic volume and queue length estimations, the absolute differences are only 4.20 vehicles for vehicle counting and 3.08 m in queue length estimation upon the RMSE values for more than 60,000 test samples. The MAPE values are less than 20%, which means that the performance of the proposed method is reasonable, particularly when estimating the lane-by-lane traffic information. Overall, based on the analyses of the five different evaluation criteria, the method proposed in this study shows the feasibility of collecting detailed traffic information with a camera installed at an intersection. In addition, the average time taken from image collection, data processing, and data storage in the server is 0.034 seconds, showing that the performance of the entire process can be completed within 0.1 seconds in general. Considering the results of this study, the proposed method is a highly optimistic technology to be applied to the fields of ITS and C-ITS.

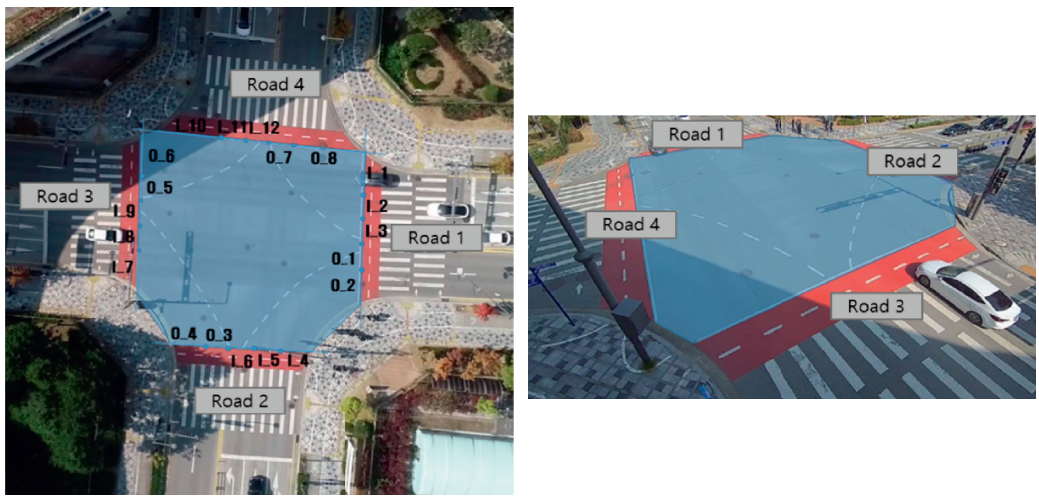


Figure 10: An example of comparing drone image and camera image for traffic volume estimation.

Table 3: Traffic volume in each direction counted from drone images.

Out		Road_1		Road_2		Road_3		Road_4		Total
In		O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	
Road_1	I_1	N/A	N/A	0	0	5	93	0	8	106
	I_2	N/A	N/A	0	0	117	1	0	0	118
	I_3	N/A	N/A	74	0	0	0	0	0	74
Road_2	I_4	0	26	N/A	N/A	0	0	1	9	36
	I_5	0	0	N/A	N/A	0	0	39	0	39
	I_6	0	0	N/A	N/A	47	0	0	0	47
Road_3	I_7	1	59	0	18	N/A	N/A	0	0	78
	I_8	82	1	0	0	N/A	N/A	0	0	83
	I_9	0	0	0	0	N/A	N/A	29	0	29
Road_4	I_10	0	0	0	3	0	35	N/A	N/A	38
	I_11	0	0	32	1	0	0	N/A	N/A	33
	I_12	23	0	0	0	0	0	N/A	N/A	23
Total		106	86	106	22	169	129	69	17	704

Table 4: Traffic volume in each direction extracted through camera data.

Out		Road_1		Road_2		Road_3		Road_4		Unknown	Total
In		O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8		
Road_1	I_1	N/A	N/A	0	0	5	87	0	8	7	107
	I_2	N/A	N/A	0	0	109	1	0	0	3	113
	I_3	N/A	N/A	74	0	0	0	0	0	0	74
Road_2	I_4	0	23	N/A	N/A	0	0	1	8	3	35
	I_5	0	0	N/A	N/A	0	0	37	0	2	39
	I_6	0	0	N/A	N/A	43	0	0	0	3	46
Road_3	I_7	1	49	0	7	N/A	N/A	0	0	12	69
	I_8	79	0	0	0	N/A	N/A	0	0	2	81
	I_9	0	0	0	0	N/A	N/A	25	0	0	25
Road_4	I_10	0	0	0	3	0	32	N/A	N/A	1	36
	I_11	0	0	31	0	0	0	N/A	N/A	2	33
	I_12	22	0	0	0	0	0	N/A	N/A	1	23
Unknown		6	11	0	5	10	6	6	1	0	45
Total		108	83	105	15	167	126	69	17	36	726

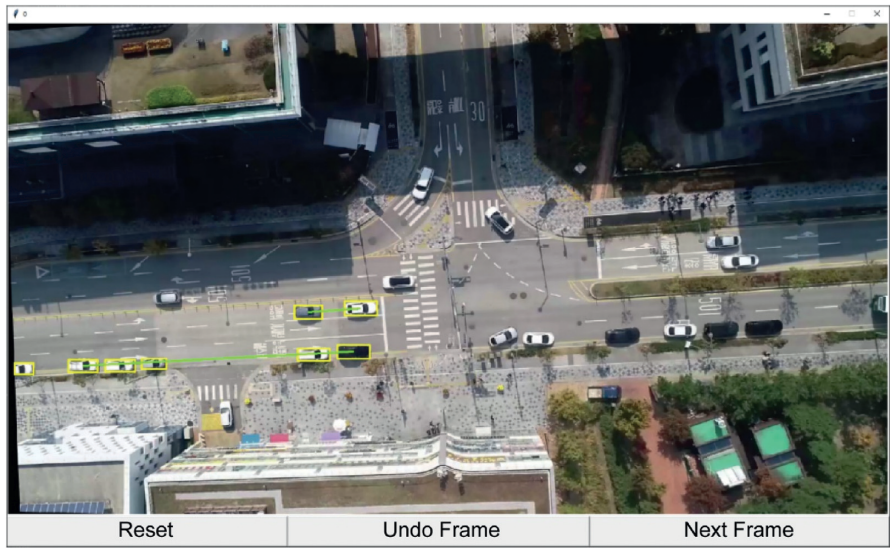


Figure 11: An example of comparing drone image for queue length derivation.

Table 5: Overall performance.

No.	Evaluation item	Number of samples (frame)	Subitems	Value	Unit	Evaluation method
1	Object detection	6804	—	99	%	Detection rate
2	Object classification	6804	Car	95	%	MAP
			Truck	87	%	MAP
			Bus	81	%	MAP
3	Trajectory location	60531	—	1.138	Meter	Mean Euclidean distance
4	Traffic volume	60531	—	4.20	Number of vehicles	RMSE
				16.41	Error rate	MAPE
5	Queue length	62205	—	3.08	Meter	RMSE
				18.28	Error rate	MAPE

5. Conclusion

In this study, we considered a method for deriving traffic information using a camera installed at an intersection to improve the monitoring system for roads. The method uses a deep-learning-based approach for image processing for vehicle detection and vehicle type classification. The method then estimates the lane-by-lane vehicle trajectories using the detected locations of vehicles. Based on the estimated vehicle trajectories, the traffic volumes of each lane-by-lane traveling direction and queue lengths of each lane were estimated. The performance of the proposed method was tested with thousands of samples according to five different evaluation criteria: vehicle detection rate, vehicle type classification, trajectory prediction, traffic volume estimation, and queue length estimation. As a result, the method shows the feasibility of collecting detailed traffic information with a camera installed at an intersection.

The proposed method has two research values. It has shown high accuracy in (1) real-time vehicle detection and classification based on deep-learning-based image processing and (2) estimating lane-by-lane vehicle trajectories by matching the detected vehicle locations with the HD map. While estimating the vehicle trajectories, this study has attempted to reduce the error of estimating the center points of the bounding boxes in the images of vehicles to ensure proper performance of the HD map-matching process. Hence, the approach of combining AI and HD map techniques is the main contribution of this study. This study shows a high chance of improving current traffic monitoring systems.

Although the proposed method has shown reasonable performance, this study is not without limitations. The error rates for both lane-by-lane traffic volume and queue length estimations are greater than 15% even though the vehicle detection showed a 99% performance, which is reasonable but not sufficient in terms of the reliability of traffic information. This is due to intermittent mismatches between the vehicle locations of the camera images and the HD map coordinates. Hence, further studies should consider enhancing the matching performance between camera image-based data and map data. Furthermore, the results of this study confirmed that the error increased with the distance between the camera and vehicle. Thus, investigating the minimum required distance between the camera and the intersection area can be a topic for future studies. In addition, for road lanes, additional research is required to develop a vehicle location correction algorithm. It is also necessary to perform training with trucks and buses to further improve the detection rate. Subsequent studies

should consider these limitations for the further development of image processing-based traffic monitoring systems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge Editage (http://www.editage.co.kr) for English language editing. This research was supported by the Ministry of Land, Infrastructure, and Transport (MOLIT, Korea) under the Connected and Automated Public Transport Innovation National R&D Project (Grant no. 21TLRP-B146733-04).

References

[1] A. H. F. Chow, R. Sha, and S. Li, "Centralised and decentralised signal timing optimisation approaches for network traffic control," *Transportation Research Part C: Emerging Technologies*, vol. 113, pp. 108–123, 2020.

[2] L. Adacher and M. Tiriolo, "Performance analysis of decentralized vs. centralized control for the traffic signal synchronization problem," *Journal of Advanced Transportation*, vol. 2020, Article ID 8873962, 19 pages, 2020.

[3] S. Kim, S. Tak, D. Lee, and H. Yeo, "Distributed model predictive approach for large-scale road network perimeter control," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 5, pp. 515–527, 2019.

[4] S. C. Calvert, W. J. Schakel, and J. W. C. van Lint, "Will automated vehicles negatively impact traffic flow?" *Journal of Advanced Transportation*, vol. 2017, Article ID 3082781, 17 pages, 2017.

[5] S. Tak, J. Yoon, S. Woo, and H. Yeo, "Sectional information-based collision warning system using roadside unit aggregated connected-vehicle information for a cooperative intelligent transport system," *Journal of Advanced Transportation*, vol. 2020, Article ID 1528028, 12 pages, 2020.

[6] M. L. Tam and W. H. K. Lam, "Using automatic vehicle identification data for travel time estimation in Hong Kong using automatic vehicle identification data for travel time estimation in Hong Kong," *Transportmetrica*, vol. 4, no. 3, pp. 179–194, 2008.

[7] P. W. Wang, H. B. Yu, L. Xiao, and L. Wang, "Online traffic condition evaluation method for connected vehicles based on multisource data fusion," *Journal of Sensors*, vol. 2017, Article ID 7248189, 11 pages, 2017.

[8] J. M. Salanova Grau, E. Mitsakis, P. Tzenos, I. Stamos, L. Selmi, and G. Aifadopoulou, "Multisource data framework for road traffic state estimation," *Journal of Advanced Transportation*, vol. 2018, Article ID 9078547, 9 pages, 2018.

[9] J. W. C. Van Lint and N. J. Van der Zijpp, "Improving a travel-time estimation algorithm by using dual loop detectors," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1855, no. 1, pp. 41–48, 2003.

[10] Q. Gan, G. Gomes, and A. Bayen, "Estimation of performance metrics at signalized intersections using loop detector data and probe travel times," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2939–2949, 2017.

[11] J. Tang, Y. Zou, J. Ash, S. Zhang, F. Liu, and Y. Wang, "Travel time estimation using freeway point detector data based on evolving fuzzy neural inference system," *PLoS One*, vol. 11, no. 2, Article ID e0147263, 2016.

[12] A. Roy, N. Gale, and L. Hong, "Automated traffic surveillance using fusion of doppler radar and video information," *Mathematical and Computer Modelling*, vol. 54, no. 1-2, pp. 531–543, 2011.

[13] S. L. Jeng, W. H. Chieng, and H. P. Lu, "Estimating speed using a side-looking single-radar vehicle detector," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 607–614, 2014.

[14] C. Lu and J. Dong, "Estimating freeway travel time and its reliability using radar sensor data," *Transportmetrica B: Transport Dynamics*, vol. 6, no. 2, pp. 97–114, 2018.

[15] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, "Traffic congestion detection from camera images using deep convolution neural networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 45, pp. 222–231, 2018.

[16] A. Fedorov, K. Nikolskaia, S. Ivanov, V. Shepelev, and A. Minbaleev, "Traffic flow estimation with data from a video surveillance camera," *Journal of Big Data*, vol. 6, no. 1, p. 73, 2019.

[17] D. C. Luvizon, B. T. Nassu, and R. Minetto, "A video-based system for vehicle speed measurement in urban roadways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1393–1404, 2017.

[18] M. Bernas, B. Płaczek, W. Korski, P. Loska, J. Smyła, and P. Szymała, "A survey and comparison of low-cost sensing technologies for road traffic monitoring," *Sensors*, vol. 18, no. 10, p. 3243, 2018.

[19] Y. Nam and Y. C. Nam, "Vehicle classification based on images from visible light and thermal cameras," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, 2018.

[20] J. Kato, T. Watanabe, S. Joga, J. Rittscher, and A. Blake, "An HMM-based segmentation method for traffic monitoring movies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1291–1296, 2002.

[21] Y. Wang, Y. Zou, H. Shi, and H. Zhao, "Video image vehicle detection system for signaled traffic intersection," in *Proceedings of the 2009 9th International Conference on Hybrid Intelligent Systems*, vol. 1, pp. 222–227, Shenyang, China, August 2009.

[22] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 758–763, 1999.

[23] J. Zhou, D. Gao, and D. Zhang, "Moving vehicle detection for automatic traffic monitoring," *The IEEE Transactions on Vehicular Technology*, vol. 56, no. 1, pp. 51–59, 2007.

[24] Y. K. Ki and D. Y. Lee, "A traffic accident recording and reporting model at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 188–194, 2007.

[25] J. P. Lin and M. T. Sun, "A YOLO-based traffic counting system," in *Proceedings of the 2018 Conference on Technologies and Applications of Artificial Intelligence TAAI 2018*, pp. 82–85, Taichung, Taiwan, December 2018.

[26] C. S. Asha and A. V. Narasimhadhan, "Vehicle counting for traffic management system using YOLO and correlation filter," in *Proceedings of the 2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, March 2018.

[27] K. J. Kim, P. K. Kim, Y. S. Chung, and D. H. Choi, "Multi-scale detector for accurate vehicle detection in traffic surveillance data," *IEEE Access*, vol. 7, pp. 78311–78319, 2019.

[28] F. Zhang, C. Li, and F. Yang, "Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation," *Sensors*, vol. 19, no. 3, p. 594, 2019.

[29] Z. Xu, H. Shi, N. Li, C. Xiang, and H. Zhou, "Vehicle detection under UAV based on optimal dense YOLO method," in *Proceedings of the 2018 5th International Conference on Systems and Informatics ICSAI 2018*, pp. 407–411, Nanjing, China, November 2019.

[30] A. Forero and F. Calderon, "Vehicle and pedestrian video-tracking with classification based on deep convolutional neural networks," in *Proceedings of the 2019 22nd Symposium on Image, Signal Processing and Artificial Vision*, Bucaramanga, Colombia, April 2019.

[31] D. Ka, D. Lee, S. Kim, and H. Yeo, "Study on the framework of intersection pedestrian collision warning system considering pedestrian characteristics," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 5, pp. 747–758, 2019.

[32] J. Zhang, X. Jin, J. Sun, J. Wang, and K. Li, "Dual model learning combined with multiple feature selection for accurate visual tracking," *IEEE Access*, vol. 7, pp. 43956–43969, 2019.

[33] J. Zhang, Y. Wu, W. Feng, and J. Wang, "Spatially attentive visual tracking using multi-model adaptive response fusion," *IEEE Access*, vol. 7, pp. 83873–83887, 2019.

[34] J. Zhang, X. Jin, J. Sun, J. Wang, and A. K. Sangaiah, "Spatial and semantic convolutional features for robust visual object tracking," *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 15095–15115, 2020.

[35] X. Chen, X. Xu, Y. Yang, H. Wu, J. Tang, and J. Zhao, "Augmented ship tracking under occlusion conditions from maritime surveillance videos," *IEEE Access*, vol. 8, pp. 42884–42897, 2020.

[36] X. Chen, Z. Li, Y. Yang, L. Qi, and R. Ke, "High-resolution vehicle trajectory extraction and denoising from aerial videos," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3190–3202, 2021.

[37] L. Wang, L. Zhang, and Z. Yi, "Trajectory predictor by using recurrent neural networks in visual tracking," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3172–3183, 2017.

[38] D. Koller, J. Weber, T. Huang, and J. Malik, "Towards robust automatic traffic scene analysis in real-time," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 4, Jerusalem, Israel, 1994.

[39] J. Li, S. Chen, F. Zhang, E. Li, T. Yang, and Z. Lu, "An adaptive framework for multi-vehicle ground speed estimation in airborne videos," *Remote Sensing*, vol. 11, no. 10, p. 1241, 2019.

[40] S. Seong, J. Song, D. Yoon, J. Kim, and J. Choi, "Determination of vehicle trajectory through optimization of vehicle bounding boxes using a convolutional neural network," *Sensors*, vol. 19, p. 4263, 2019.