infrastructures

MDPI

Article

Deep Learning-Based Real-Time Traffic Sign Recognition System for Urban Environments

Chang-il Kim, Jinuk Park, Yongju Park, Woojin Jung and Yong-seok Lim *

Korea Electronics Technology Institute, Seongnam 13509, Republic of Korea
* Correspondence: busytom@keti.re.kr

Abstract: A traffic sign recognition system is crucial for safely operating an autonomous driving car and efficiently managing road facilities. Recent studies on traffic sign recognition tasks show significant advances in terms of accuracy on several benchmarks. However, they lack performance evaluation in driving cars in diverse road environments. In this study, we develop a traffic sign recognition framework for a vehicle to evaluate and compare deep learning-based object detection and tracking models for practical validation. We collect a large-scale highway image set using a camera-installed vehicle for training models, and evaluate the model inference during a test drive in terms of accuracy and processing time. In addition, we propose a novel categorization method for urban road scenes with possible scenarios. The experimental results show that the YOLOv5 detector and strongSORT tracking model result in better performance than other models in terms of accuracy and processing time. Furthermore, we provide an extensive discussion on possible obstacles in traffic sign recognition tasks to facilitate future research through numerous experiments for each road condition.

Keywords: traffic sign recognition; deep learning; object detection; real-time application; urban road scene

check for updates

Citation: Kim, C.-i.; Park, J.; Park, Y.; Jung, W.; Lim, Y.-s. Deep Learning-Based Real-Time Traffic Sign Recognition System for Urban Environments. *Infrastructures* **2023**, *8*, 20. https://doi.org/10.3390/infrastructures8020020

Academic Editor: António Couto

Received: 18 November 2022
Revised: 14 January 2023
Accepted: 26 January 2023
Published: 31 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Owing to the increasing market share of the autonomous vehicle industry, fundamental technologies for driving assistants and artificial intelligence have been increasingly studied [1]. One of the most crucial technologies for Advanced Driver Assistance Systems (ADAS), including self-driving, forward collision warning, or pedestrian recognition, is contextual awareness of road environments [2]. In particular, traffic sign recognition systems are core methods for providing vital instructions in safety-critical road regulations and should perform at highly stringent confidence levels.

Many autonomous vehicles utilize high-definition (HD) maps to provide richer information for road environments [3,4]. However, because of the manual and time-consuming efforts in production, the usage of HD maps is costly [5]. More importantly, HD maps can suffer from the discrepancies between the stored traffic signs and real-time changes [6]. In addition to assisting drivers, intelligent object recognition systems can facilitate the maintenance of road surroundings, such as traffic signs, lane lines, and guard rails [7]. For instance, traffic sign recognition systems can effectively analyze damage or defects through autonomous vehicles for monitoring purposes because it is nontrivial to inspect an entire road scene using human resources [8]. Therefore, the traffic sign recognition technique is an important component both for decision-making systems in vehicles and for monitoring road management systems.

A traffic sign recognition system should provide stable support in real-time driving vehicles, but faces two major challenges. The first problem is the low quality of images owing to diverse environmental conditions, such as weather, illuminance, and occlusion [9,10]. Given that blurred and contaminated images can severely deteriorate recognition performance, a recognition model should be robust to the noise from various causes. Secondly,

another critical issue for the traffic sign recognition technique is to detect and recognize signs in real-time. Therefore, to respond to the frequent changes in driving car scenarios, the model should guarantee a short processing time for the recognition systems.

Conventional traffic sign recognition methods have been developed in the context of studying salient features to capture traffic signs. The feature-based models such as color- or shape-based methods have been proposed to extract candidate regions and classify the signs in road scene images [11]. However, these feature-based methods are illumination-sensitive. A recent trend in traffic sign recognition systems is to employ deep learning-based object detection models, especially Convolutional Neural Networks (CNNs) [2,12]. Deep learning-based object detection approaches including YOLO models facilitate accurate traffic sign recognition in several benchmark datasets because they are capable of flexible and expressible nonlinear representation. Indeed, studies [2,13,14] based on YOLO models have achieved state-of-the-art performances for publicly available benchmark datasets (e.g., GTSDB [15], GTSRB [16], and RTSD [17]) for traffic sign recognition.

Although the recent studies for traffic sign recognition systems outperform the conventional methods, they have three prominent limitations. Firstly, there are limited studies to demonstrate the performance of deep learning-based traffic sign recognition methods in real-world urban road environments with various noise. Because there can be significant differences in the quality between benchmark images and urban road scenes, it is crucial to inspect the types of noise and to verify the recognition results of the models. Secondly, deep learning-based approaches for traffic sign recognition systems barely consider object tracking models coupled with detection models for practical usage. Lastly, there is a lack of studies with respect to the processing time or frames per second (FPS) of the recent deep learning-based models in driving cars coupled with intelligent cameras. To evaluate a traffic sign recognition system, it is necessary to implement the entire edge system, from an edge camera to an edge computing system (e.g., NVIDIA Jetson) in a driving car, rather than only assessing the performance of models in laboratories.

To address these issues, we propose a traffic sign recognition framework based on deep learning models to implement the entire system from cameras to processing units and to validate the accuracy and latency of the proposed system architecture. As our main goal is to evaluate the traffic sign recognition performance in a driving car, we install the entire system inside a car, including a camera device, an edge computing unit, and a standalone server. We categorize the urban road environments based on diverse noise types and group road images with the corresponding settings [18]. We train several versions of object detection models and examine the framework with respect to the defined categories of environmental conditions in terms of accuracy and latency. In addition, we evaluate the real-time performance of two object tracking models with a detection model. From our experimental results, we identify the possible obstacles in traffic sign recognition tasks with extensive discussion to facilitate future research.

The main contributions of this study are as follows:

1. We propose a traffic sign recognition framework coupled with deep learning-based object detection models. Furthermore, we collect real-world road scene images from a driving car and define the main categories of noise for the environmental conditions;
2. We evaluate and prove the efficiency of deep learning-based object detection and tracking models through in-depth experiments considering model types and environmental conditions;
3. We derive the candidate issues from the experimental results and provide insightful analyses to facilitate future research on traffic sign recognition systems.

The remainder of the paper is organized as follows. Section 2 briefly reviews related works on traffic sign recognition tasks. In Section 3, we introduce the proposed framework and then describe data collection and categorization methods for urban road scene images. Section 4 includes the experimental setup and evaluation metrics. In Sections 5 and 6, we present comprehensive results and provide a detailed discussion through in-depth ablation studies. Finally, Section 7 provides a conclusion and considers future research directions.

2. Related Work

In the last decade, extensive research has been conducted to detect and recognize traffic signs as well as moving objects in road scenes. Among the intelligent object detection tasks, the traffic sign recognition method is core because it is the most critical in safety-critical road applications. The various types of urban environments (e.g., downtown, residential areas) are relevant to traffic sign recognition because different types of traffic signs tend to occur in different environments and traffic signs are also different in size and location [7]. The core challenge in detecting and recognizing traffic signs on roads is the low quality of images owing to variable urban environments including the influence of weather, the time of day, and driving of vehicles at high speed. Alongside environmental effects, it is still problematic to develop intelligent recognition systems because the installation methods of devices can prompt vibrations in vehicles and result in a restricted view of the surrounding objects [10]. To reduce the impact of these problems, various image preprocessing methods are being studied: one approach is based on traditional feature-based methods, and another approach uses deep learning-based methods.

The traditional feature-based methods typically gather specifically engineered visual features including Histogram of Oriented Gradient (HOG) features and use those features to classify traffic signs [1,19,20]. A method based on HOG features [21] was proposed for object detection purposes in computer vision and image processing; HOG descriptors are calculated on dense grids of cells placed at uniform intervals and are designed to use nested local contrast normalization to improve accuracy. These methods have been used as improvements to balance scale-invariant characteristic transformations with descriptor and shape contexts, feature-invariance (e.g., illumination, translation, and scale), and nonlinearity. HOG methods, however, mainly utilize manual features to capture significant characteristics in images; hence, they often fail to model complex surroundings in road scene images.

On the other hand, methods utilizing features from the color and shape of a given image have been proposed for traffic sign recognition tasks [11,22–25]. The main steps of these methods are to extract visual information contained in candidate areas, capture and segment traffic signs within an image, and correctly classify the signs through pattern classification [26]. The color-based methods typically segment specific colors to generate a color map. These methods find solid color regions and then determine the candidate regions that are computationally efficient [27]; however, they usually require delicate color and shape information to increase recognition accuracy. The intrinsic issues of the environment, such as color fading, subtle illumination change, and occlusion of traffic signs, are critical in these methods [2]. For instance, color-based methods are dependent on specific characteristics (e.g., solid colors) of traffic signs; hence they show significant performance degradation with contaminated images. Similarly, shape-based methods fail to recognize objects with unclear shapes in blurred images.

Prominent traffic sign recognition methods have used machine learning models including Support Vector Machine [28–30] and AdaBoost based traffic signs detection [31]. These approaches result in relatively fair recognition performances when coupled with specific image features (e.g., HOG, color- or shape-features). However, they are heavily reliant on hand manual feature engineering. For this reason, they fail to model complex urban road surroundings, which is a similar shortcoming inherited from features-based methods. Moreover, different feature engineering is required for different environmental settings.

Deep learning models, specifically CNN-based models, have achieved rapid advances in computer vision tasks [32–34]. Similar to object detection tasks, computer vision processing tasks in intelligent transportation systems are also following this trend, and these tasks are being used practically in ADAS as well as in autonomous vehicles. Accordingly, several researchers have tried to solve the traffic sign recognition problem using a CNN-based object detection framework [2,11]. These studies mainly employ object detection models that can be represented by Faster R-CNN [34] and YOLO models [32,33]. In particular, Faster R-CNN combines the regression of bounding boxes and object classification. It uses

end-to-end methods to detect visual objects, which not only improve the accuracy of object detection but also improve the speed of object recognition. Visual object detection consists of classification and positioning. Before the advent of YOLO, these two tasks differed in visual object detection. In YOLO models, object detection is simply transformed into a regression problem. In addition, YOLO follows the end-to-end structure of the neural network for visual object detection, and thus simultaneously obtains the coordinates of the predicted boundary box, the reliability of the target, and the probability of the class to which the target belongs through one image input [35].

Recently, three versions of YOLO have been proposed: YOLOv3 [32], YOLOv4 [33], and YOLOv5; YOLOv4 succeeded the Darknet and obtained a remarkable performance improvement based on the Microsoft COCO dataset [36]. Compared with YOLOv3, accuracy and speed have been effectively improved in YOLOv4. Furthermore, it can be considered a real-time object detection model for traffic sign recognition tasks on Tesla V100 [37], which has strong computational resources. The most recent model, YOLOv5, outperformed the previously models both in accuracy and efficiency. It uses several techniques, including efficient CNN blocks, adversarial training, and augmentation. Compared with YOLOv3, the detection accuracy of YOLOv5 increased by 4.30%, indicating that the performance was superior to that of the previous model [38].

Object tracking is another technology that has been actively researched recently. Object tracking is an algorithm that assigns a unique ID to a detected object and maintains the ID value unchanged even as frames flow. One of the most successful object tracking methods is Simple Online and Realtime Tracking (SORT) [39], which employs Kalman filters [40] and the Hungarian algorithm [40,41]. SORT is proposed for a multiple object tracking task that efficiently associates detected objects for real-time tracking. Based on the position information of objects using noise and speed information, SORT obtains the position information of objects to come in the next frame. DeepSORT [42] has been proposed to utilize expressive CNN features to the existing SORT method. DeepSORT has advanced the tracking accuracy because the model uses more informative features than SORT. Recently, [43] proposed StrongSORT which extends DeepSORT. StrongSORT outperformed the previous models by introducing an appearance-free link model to generate efficient trajectories and Gaussian-smoothed interpolation to compensate missing detections.

Recent deep learning-based traffic sign recognition studies [2,13,14] have shown significant improvement by outperforming state-of-the-art scores on several benchmark datasets [15–17]. However, there are limited studies on real-world urban road environments with the diverse noise types. Since urban road scenes can have different aspects from benchmark images, it is crucial to investigate deep learning models for real-world application. Additionally, deep learning-based approaches for traffic sign recognition systems barely consider object tracking models, which are essential for practical implementation. In this study, we develop a traffic sign recognition framework using different YOLO models followed by two tracking methods (DeepSORT and StrongSORT), and explore those models on urban roads in a driving car to ensure real-time applicability.

3. Methods

3.1. System Overview

Our goal is to develop and compare the performance of deep learning-based object detection and tracking models for traffic sign recognition in urban road environments. For ease of description, we first describe the overall framework for the recognition and tracking models. In the following section, we then introduce the categorization and labeling method for urban environment.

Figure 1 shows the overall flow of the proposed real-time traffic sign recognition framework. As depicted in the figure, the framework has a sign recognition and tracking model. A camera installed on the vehicle takes videos (a sequence of frames) while driving, and trained YOLO detectors are applied to recognize traffic signs in those images. Afterward, images of detected traffic signs are fed into the object tracking models (DeepSORT and

StrongSORT). Note that the overview represents the inference flow in a driving vehicle scenario. Those models are trained and evaluated in a local server, not in edge resources.

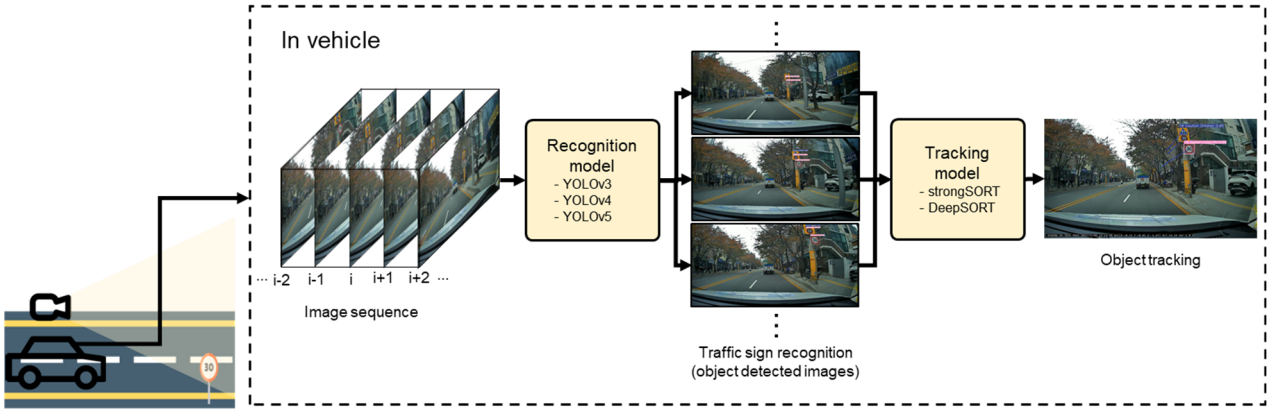


Figure 1. Traffic sign recognition flows in a vehicle using YOLO and tracking models.

The entire physical framework in Figure 2 of the proposed system has two main components: a camera device and computing systems. To collect and capture road scene images in a driving car, we chose a high performance PoE camera, which has 2056 × 1542 (3.17 megapixels) resolution and 36 FPS. The computing resources include a standalone storage server and a processing unit. We employ a standalone storage server with a capacity of 930 GB from NEOUSYS to store images captured from the camera. In addition, we utilize NVIDIA's Jetson AGX Xavier with built-in GPU (512-core NVIDIA Volta™ GPU with 64 Tensor Cores) as an edge device to inference using the trained YOLO models in real-time.

Figure 2. Overview of the installed system in a driving car.

3.2. Environment Categorization and Data Collection

Inspired by the previous description of weather characteristics in [44], we propose novel categories of distinct environmental conditions for urban road scenes. There can be several possible reasons for changes in environments, including seasonal effects, terrain conditions, and road facilities. Traffic sign recognition performance even with the same traffic sign can be severely affected by these conditions. Therefore, we categorized possible conditions of road environments and collected the corresponding images for training. The environmental conditions included in the data set are clean, cloud, tunnel, night, rain, and sunlight. The details for each category are described in Table 1.

Table 1. Proposed categories for various urban environments.

Categories	Characteristics of Images
Clean	There is no direct interference of strong sunlight and the peripheral vision is clear.
Cloud	Because there is no direct sunlight and clouds cover the sun, the surrounding environment is slightly dark.
Rain	The surrounding environment is dark and the traffic signs are usually blurred by rainwater.
Sunlight	Intense sunlight directly interferes with the vision camera, causing light scattering, and making it difficult to recognize traffic signs.
Night	As sunlight is absent, the surrounding environment is very dark. It is hard to recognize traffic signs with human eyes without a light source.
Tunnel	Tunnels have artificial light sources which are different from natural light. Additionally, the environments are usually surrounded by static conditions, such as tunnel walls.

Extreme weather conditions can temporarily degrade the quality of images from a camera. In addition, deviation in the amount of light or abnormal light types, such as dim light, overexposure, and glare, can negatively affect the recognition of traffic signs [45,46]. Therefore, we have to consider the impact of urban environmental conditions on the detection of traffic signs.

A vehicle equipped with a physical system collected video data while traveling on 21 highways in Korea (e.g., Seocheon-Gongju: 62.2 km, Muan-Gwangj: 40.5 km, and Pyeongtaek-Paju: 36.2 km Expressways). The total distance driven by the vehicle was 1246.65 km, and the data collection period was from 1 April 2019 to 22 August 2019.

The driving for data collection was performed with one driver and an assistant. After collecting videos, for each appearance of the traffic sign, the acquired video frame was transformed into an image and stored in the database as post-processing. The data information stored in this way included the date, time, and road information. Consequently, we gathered a total of 1,005,162 (one million) images for training; Figure 3 shows examples of each category in Table 1.

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

Figure 3. *Cont.*

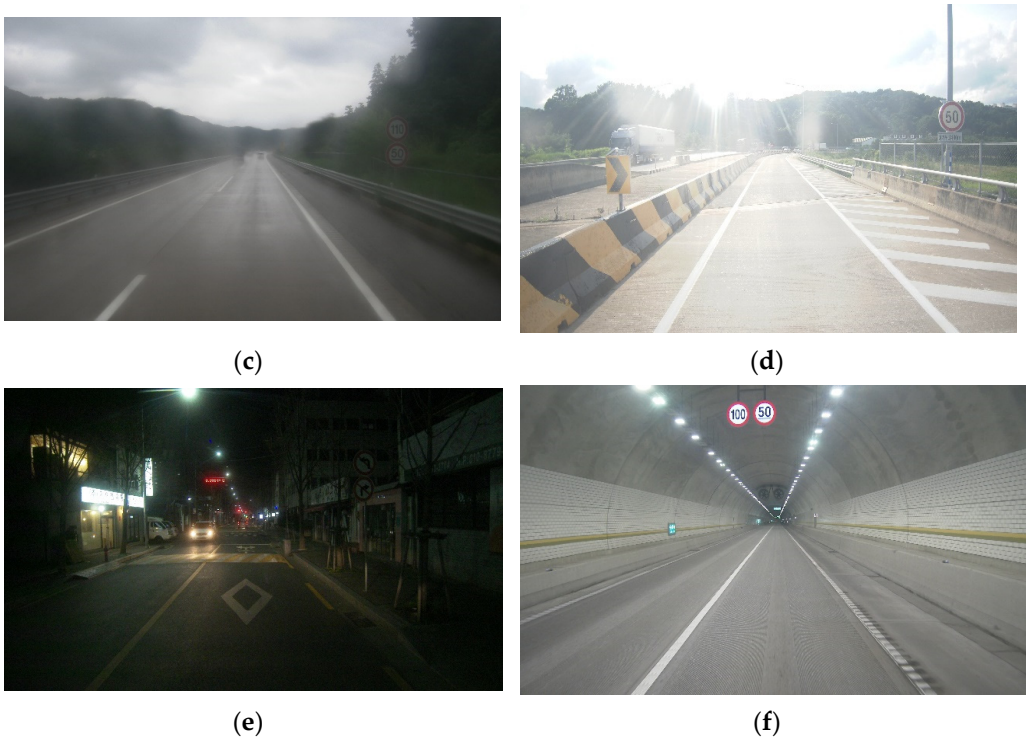


Figure 3. Examples for each environment category: (**a**) Clean, (**b**) Cloud, (**c**) Rain, (**d**) Sunlight, (**e**) Night, and (**f**) Tunnel.

3.3. Labeling Methods

According to the traffic sign information table from the Korean Institution of Road Traffic (KoROAD), each sign in the one million images was labeled and assigned into one of 98 possible classes. The labels consisted of 36 caution signs, 30 regulation signs, 28 indication signs, and 4 other signs. The signs followed specific standards of shape and color, to effectively deliver information to road users. In particular, all caution signs, which inform road users that road conditions are dangerous or problematic, have a triangular shape, and a yellow background with a red edge. To inform about prohibited or restricted road situations, regulation signs are primarily circular in shape and have a white background with a red border. Indication signs mainly have a circular shape and a blue background without boundary to inform road users that they give directions. Examples of the main three sign categories, caution, regulation, and indication are listed in Table 2.

Table 2. Examples of traffic sign classifications: Caution, Regulation, Indication signs.

Class No.	Class Name	Image	Sign Classification
0	Cross Intersection		Caution
24	Signal		Caution
31	Watchout Children		Caution

Table 2. *Cont.*

Class No.	Class Name	Image	Sign Classification
43	No Trucks Allowed		Regulation
56	No Parking		Regulation
61	Speed Limit 60		Regulation
74	Cross Walk		Indication
86	Right Turn		Indication
92	U-Turn		Indication

Furthermore, we found that the number of traffic signs had biased distributions. In particular, the smallest class had 61 samples, while the largest one had 238,857 samples. The numbers in each class are depicted in Figure 4. To be specific, the average number of samples of caution, regulation, and indication category were 1495, 4713, and 6780, respectively. This can be considered to be an intrinsic distribution for road condition because direction indicating signs are typically much more numerous than other signs. We further discuss the effect of the number of signs in Section 5.5.

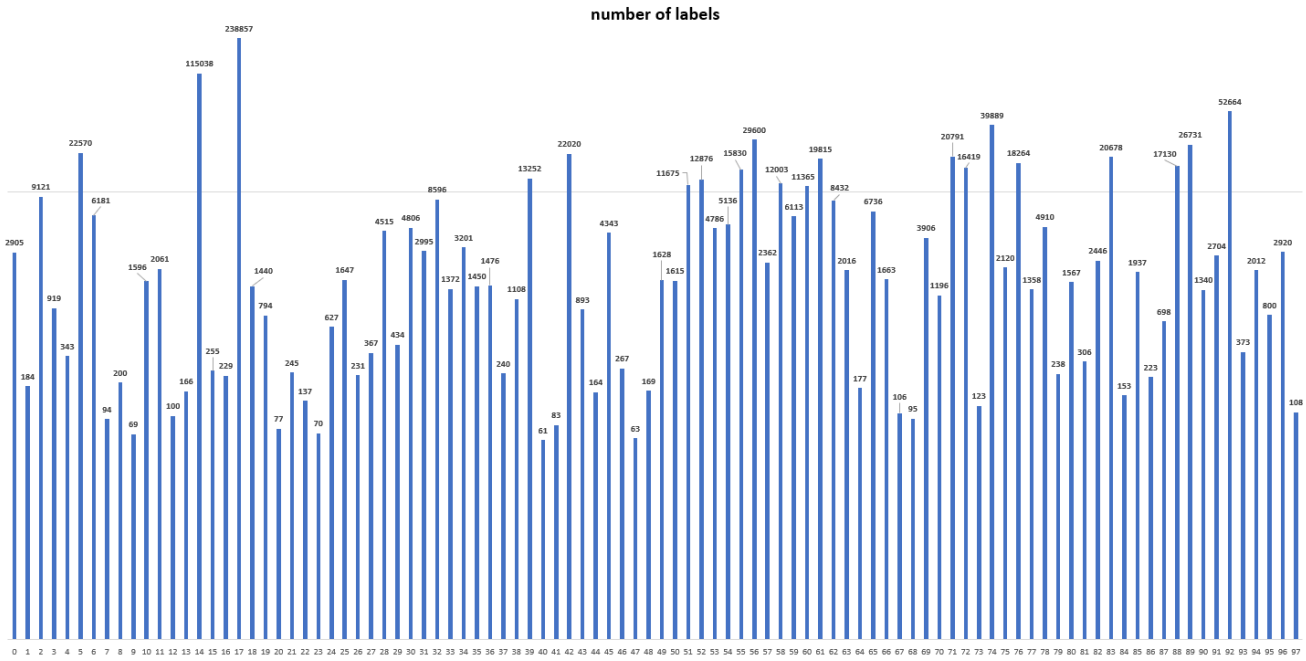


Figure 4. The number of samples for each class (in logarithmic scale).

4. Experiment

4.1. Implementation Details

We extracted one million images containing traffic signs from the captured videos in the driving car. After extraction, we resized each image into 640 × 640. Given that the entire set of images cannot be stored in the edge computing unit, we employed a standalone storage in the car. Additionally, we utilized four GPUs (Nvidia RTX A6000) in parallel during the training process for all the deep learning models. We used a Python 3.8 environment with Pytorch 1.9, OpenCV 4.5, and CUDA 11.2.

We randomly divided the one million images into training and validation sets in a 7:3 ratio for each class. Then, we trained each object detection model with a batch size of 64 and 300 epochs. The training hours could be slightly different for each model, but a single model took approximately 14 days with four GPUs for one million training images. Obviously, the training was conducted on the on-premise server with RTX GPUs and the inference was performed on the edge computing unit (Nvidia Jetson AGX Xavier).

4.2. Evaluation Metrics

To evaluate the performance in terms of accuracy, we compute the widely used metrics for object detection tasks—mean average precision (mAP)—which takes the average of average precisions (AP) of the different classes. In particular, we compute mAP scores from detected bounding box locations with the Intersection over Union (IoU) 0.5 and report them as mAP@.5. We also average the mAP scores over uniformly spaced IoUs between 0.5 and 0.95 as mAP@.5:.95. In addition, we report Precision, Recall, and F1 scores.

To compare the inference speed of the models, we report FPS to inspect the number of forwarded image frames per second. For real-time application, including traffic sign recognition systems, the application must be higher than 30 FPS.

5. Experimental Results

In this section, we validate the recognition performances of the proposed framework for various environments during road driving. We first compare performance for three different detection models trained with the same dataset, and then derive an applicable model for real-time traffic sign recognition system. We also investigate the proposed categorization method for environmental conditions through in-depth ablation studies, including different weather and light conditions.

5.1. Comparison of Detection Models

We trained the three different YOLO models for the comparison: YOLOv3 (https://github.com/ultralytics/yolov3, 10 November 2022), YOLOv4 (https://github.com/AlexeyAB/darknet, 10 November 2022), and YOLOv5 (https://github.com/ultralytics/yolov5, 10 November 2022). The models were trained using the same training set. We employed the original source codes without modification from the official GitHub pages. The YOLOv4 model is built in C language, whereas the other models are implemented in Python. After training all the models, we employed the trained YOLOv5-S model for inference during a test drive, while collecting a test set. The test set comprises the proposed categories in Section 3.2. To be specific, we additionally collected 1740 Clean, 244 Cloud, 377 Rain, 179 Sunlight, 1309 Night, and 168 Tunnel images for the test set. Except for YOLOv5-S, which was already evaluated in a driving car, the other models were evaluated using this test set.

Table 3 reports the training results obtained with different YOLO models. Note that the training epochs are set to 300, except for YOLOv5-L, which is trained for 200 epochs because it needs much longer GPU days owing to the heavy computation for large layers inside the model. It could be an unfair comparison with other models because the YOLOv5-L model does not have enough training iteration. Aside from the accuracy performance of YOLOv5-L, it shows the second slowest inference speed of 76 FPS. From this result, we conclude that YOLOv5-L model is not suitable for real-time application.

Table 3. The results of applying each training result model with the same test set.

Models	Image Size	Precision	Recall	mAP@.5	mAP@.5:.95	F1	FPS
YOLOv3	640 × 640	0.935	0.949	0.95	0.814	0.94	102
YOLOv4	416 × 416	0.879	0.880	0.775	0.718	0.88	123
	608 × 608	0.880	0.890	0.835	0.776	0.88	70
YOLOv5-S	640 × 640	0.958	0.977	0.977	0.850	0.97	133
YOLOv5-L *	640 × 640	0.898	0.916	0.916	0.784	0.89	76

* YOLOv5-L is trained for 200 epochs where others are trained for 300 epochs.

In terms of accuracy performance, YOLOv5-S achieved the best performance with an mAP@.5:.95 score of 0.850, followed by YOLOv3. The YOLOv4 model shows the worst performance when the model is trained with a smaller input size (416 × 416), but it has slightly improved performance with a similar input size (608 × 608). We could obtain a higher accuracy value because of the higher resolution, but it has a lower FPS because of the complexity. One can confirm that the size of a road scene image should be at least 608 × 608.

Based on the FPS performance, YOLOv5-S also shows the best performance followed by YOLOv4, and YOLOv3. Therefore, we choose YOLOv5-S as the preferred detection model, because it achieves both the highest mAP and FPS. In addition, if we consider applying the object tracking algorithm, one can conclude that YOLOv5-S can be used in real-time applications with 133 FPS.

5.2. Effect of Weather Conditions

It is possible to have diverse weather conditions, such as clouds and rain, as well as sunlight in summer because of the four distinct seasons. To evaluate traffic sign recognition in such environments, we conducted an ablation test for Clean, Cloud, and Rain.

Table 4 compares the results for the different weather conditions. Clean shows the best results in Precision (0.856) and F1 (0.86). Surprisingly, the performance in the Rain condition has a better mAP@.5:.95 than that in Clean, where the exact Precision is higher than that of Rain. By definition, the mAP@.5:.95 measure averages several precisions while changing the thresholds. Therefore, the objects are detected with high confidence scores when it is Clean weather. However, the objects could still be detected in other conditions, though with slightly lower confidence scores.

Table 4. Comparison of the recognition performance in different weather conditions.

Categories	# of Imgs.	Precision	Recall	mAP@.5	mAP@.5:.95	F1
Clean	1740	0.856	0.923	0.943	0.774	0.86
Cloud	244	0.826	0.899	0.917	0.637	0.81
Rain	377	0.813	0.927	0.947	0.79	0.82

In the case of Cloud, in Table 4 we observe the lowest performance in all indicators except Precision. To further investigate this result, we report sample images for each condition in Figure 5. In Figure 5c,d, we can see that Cloud has lower image quality than the other two conditions. Furthermore, the overall environment appears dark, and the boundaries between objects are not clear. For these reasons, we evaluate that the Cloud environment adversely affects recognition of traffic signs, resulting in poor performance indicators. To improve the performance in cloud weather conditions, we suggest using bright lighting or post-processing of images to aid the discrimination of objects. In addition, removal of rainwater formed on the lens during rainfall will elevate the confidence levels for precision and lead to better results.

Figure 5. Examples of traffic sign recognition results in different weather conditions: (**a**) Clean (0.89), (**b**) Clean (0.84), (**c**) Cloud (0.63), (**d**) Cloud (0.72), (**e**) Rain (0.82), (**f**) Rain (0.77), (**g**) Cloud (0.45), and (**h**) Rain (0.75). Note that the numbers in the brackets indicate the minimum confidence score for the accurately recognized traffic signs in the examples.

Figure 5g,h show the failure results of traffic sign recognition. In particular, in Figure 5h, one can expect a class named Left-Merge, but we obtain Right-Merge. The possible explanation is that those classes have a similar shape but an imbalanced number of samples. Note that the Right-Merge class has as twice many samples as Left-Merge (i.e., Left-Merge: 794, Right-Merge: 1440). To solve this problem, one could consider setting the number of samples similarly for each class by data augmentation or oversampling.

5.3. Effect of the Amount of Light

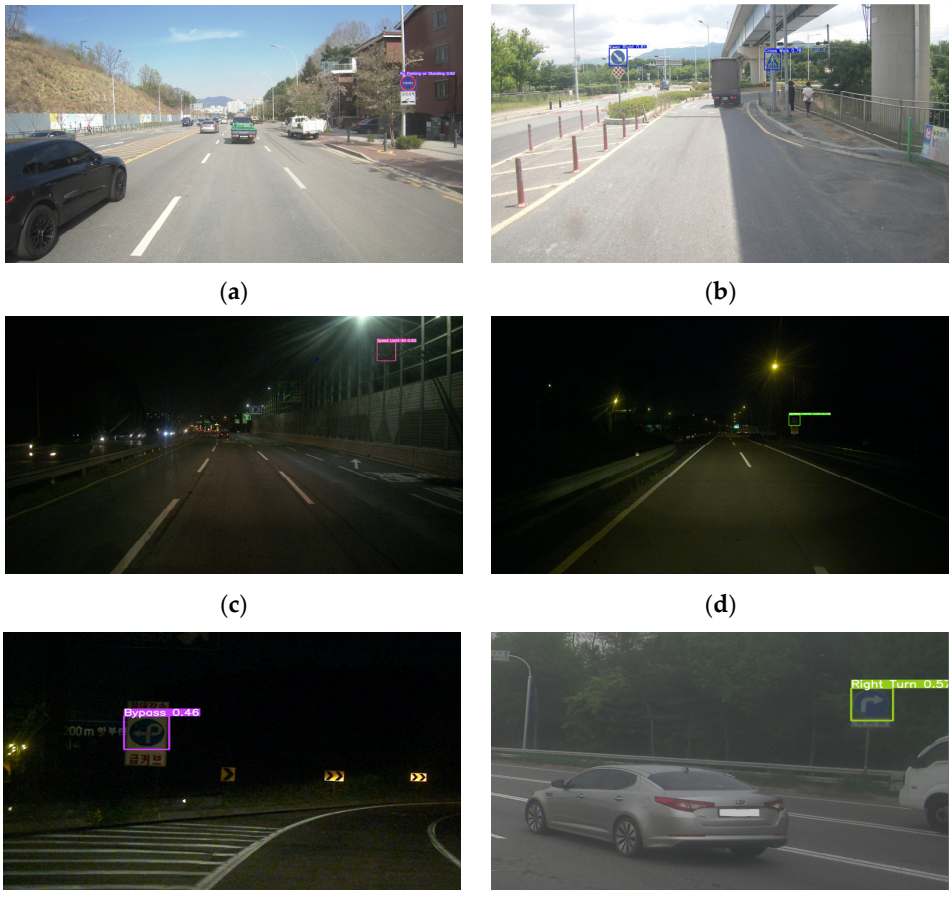
In this subsection, we conduct a test to determine the degree of traffic sign recognition with a vision camera. In places with dim light, it is difficult to recognize traffic signs with human eyes. Similarly, the vision camera system and the accompanying detection model

could be heavily affected by the amount of light. To evaluate Daytime, we use the same test samples as Clean in Table 4 because of similar environmental conditions.

Table 5 shows that the Night condition results in significantly lower performance than Daytime in all indicators: Precision (0.815), Recall (0.838), mAP@.5 (0.886), mAP@.5:.95 (0.635), and F1 (0.78), as we expected. Although we conducted the test with a similar number of images, the difference between the two results is significant. We also observe significant qualitative differences between the two conditions in terms of the amount of light, as visualized in Figure 6. One can confirm that it is difficult to recognize traffic signs at night because of the dim light. To improve the recognition of traffic signs in an environment where there is little light, we propose to install a device that provides illumination to aid the recognition of objects. In addition, we suggest equipping the traffic signs with LEDs to enable them to be well recognized at night. On the other hand, we can see the low confidence results of traffic sign recognition in Figure 6e,f because the number of samples for learning is not sufficient (e.g., Bypass: 108, Right Turn: 223). To alleviate this issue, it is important to obtain a lot of high-quality data for training.

Table 5. Comparison of the recognition performance in different amounts of light.

Categories	# of Imgs.	Precision	Recall	mAP@.5	mAP@.5:.95	F1
Daytime	1740	0.856	0.923	0.943	0.774	0.86
Night	1309	0.815	0.838	0.886	0.635	0.78



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 6. Examples of traffic sign recognition results in daytime and at night time: (**a**) Daytime (0.92), (**b**) Daytime (0.78), (**c**) Night (0.60), (**d**) Night (0.78), (**e**) Night (0.46), and (**f**) Daytime (0.57). Note that the numbers in the brackets indicate the minimum confidence score for the accurately recognized traffic signs in the examples.

5.4. Effect of the Type of Light

In this subsection, we investigate how the type of light affects the vision cameras used to recognize traffic signs. We define natural light as the amount of light during the daytime that does not cause discomfort in daily life. We use test samples of natural light on the same test set as Clean in Table 4 because of similar environmental conditions. We also evaluate Sunlight, which is the amount of light that is intense, as defined in Table 1. Finally, we conduct a test on Tunnel, which has artificial light sources.

Table 6 compares the results for the different types of light. Tunnel shows the best results in all metrics, except for mAP@.5:.95. It can be observed that Tunnel condition has almost no blurred effect caused by direct sunlight entering the camera; hence, this results in the best performance scores.

Table 6. Comparison of the recognition performance in the different types of light.

Categories	# of Imgs.	Precision	Recall	mAP@.5	mAP@.5:.95	F1
Natural light	1740	0.856	0.923	0.943	0.774	0.86
Sunlight	179	0.939	0.909	0.976	0.875	0.89
Tunnel	168	0.973	0.999	0.995	0.831	0.98

On the other hand, we observe that the performance in the Sunlight condition shows better results than in the Natural light condition, contrary to our expectations. As shown in Figure 7, objects in the Sunlight condition have boundaries to their background and show distinct shapes or colors. One can conclude that the YOLOv5-S model is powerful enough to classify these small deviations as an object. Nonetheless, to improve traffic sign recognition, we propose to install filtering to reduce the influence of light interference on the vision camera.

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

(**c**) (**d**)

Figure 7. *Cont.*

(**e**) (**f**)

Figure 7. Examples of traffic sign recognition results in different types of light: (**a**) Sunlight (0.84), (**b**) Sunlight (0.89), (**c**) Tunnel (0.89), (**d**) Tunnel (0.89), (**e**) Sunlight (0.26), and (**f**) Natural light (0.80). Note that the numbers in the brackets indicate the minimum confidence score for the accurately recognized traffic signs in the examples.

Figure 7e,f report the failure results of traffic sign recognition. In Figure 7e, we can confirm that the class name No Automobiles Allowed has a low recognition result of 0.26. Note that there is a small number of samples for the class (i.e., 267 samples), but the sign contains relatively complicated images (small cars and automobiles). Figure 7f shows that Left Lane Ends is recognized as another class, Right Lane Ends, despite the recognition result value of 0.8. It should be noted that both classes are similar shapes but include small differences in directions. To alleviate these problems, one could use an extra classifier for detailed classification to improve the accuracy.

5.5. Evaluation of Real-Time and Tracking Performance

In this subsection, we analyze the detailed real-time performance of the YOLO models and conduct an additional experiment to evaluate real-time performance for the object tracking algorithms DeepSORT and StrongSORT. To validate real-time performance, it is necessary to implement an edge computing system, and we employ Jetson AGX Xavier. For the test video, we recorded a 36 s video during an additional test drive on urban roads in Seoul, Korea. While driving, the StrongSORT was evaluated and other models were tested using the same video.

Table 7 reports the real-time performance results obtained by the different YOLO models. We report results for two GPU environments: one for the local server with RTX A6000, where the models have been trained; and the other for the edge device installed in vehicle, Jetson AGX Xavier. Obviously, one can confirm that all models have lower FPS results on Jetson AGX Xavier compared with the results on the local server. Nevertheless, in Table 8, one can see that the FPS performance of YOLOv5-S is remarkably higher in the edge device. YOLOv5-L shows a lower FPS than YOLOv5-S because of the complexity. It indicates that YOLOv5-S is optimized for real-time compared with YOLOv3 and YOLOv4. Based on the results in Table 8, we conducted the experiments for object tracking using the YOLOv5-S model.

Table 7. The results of real-time performance (FPS) testing of each model. In this section, we use two GPU environments: one is Nvidia's RTX A6000 equipped on the local server in laboratory; and the other GPU mounted on the Xavier in vehicle is Nvidia's 512-Core Volta GPU with Tensor Cores.

Models	Image Size	A6000	Xavier
YOLOv3	640 × 640	102	12
YOLOv4	416 × 416	123	14
	608 × 608	70	8
YOLOv5-S	640 × 640	133	42
YOLOv5-L	640 × 640	76	8

Table 8. The results of real-time performance (FPS) applied tracking algorithms DeepSORT and StrongSORT coupled with YOLOv5-S.

YOLOv5-S	A6000	Xavier
Without tracking	133	42
DeepSORT	40	13
StrongSORT	75	23

We employed the original source codes and pretrained checkpoints for DeepSORT (https://github.com/nwojke/deep_sort, 10 November 2022) and StrongSORT (https://github.com/dyhBUPT/StrongSORT, 10 November 2022) from the official GitHub site. Note that our framework first detects traffic signs and then the detected features are fed into the tracking models. Table 8 reports the FPS results from DeepSORT and StrongSORT coupled with YOLOv5-S. In both GPU environments, one can see that object tracking methods are highly costly because the FPS drops sharply with the tracking methods. Nonetheless, we observed that the FPS of StrongSORT is twice as good as that of DeepSORT. We achieved 23 FPS with StrongSORT for the best speed in the edge device, but this is reduced by half compared with the result before StrongSORT was applied. We can conclude that StrongSORT should be recommended over DeepSORT for real-time traffic sign recognition system.

Figure 8 represents the results of StrongSORT coupled with YOLOv5-S on Jetson AGX Xavier. We confirm the consecutive image frames maintain recognized object information when drawing tracking paths. We also can see that high confidence is maintained while tracking the detected traffic signs (e.g., Caution Children: 0.89, Speed Limit 30: 0.90).

Figure 8. Examples of object tracking results applying StrongSORT coupled with YOLOv5-S. These pictures are the result of tracking traffic signs detected in consecutive frames flowing from (**a**–**d**). Note that the dotted lines on the image represent the trajectories of the detected traffic signs.

6. Discussion

In this section, we perform an additional experiment for the number of training samples. For deep learning models, the number of training data is one of the most important factors. Although we trained the models using a dataset with one million images, in this study, the number of training data varies for each class. We compare results according to the number of samples in each class.

In Table 9, we divide the whole class into six clusters, based on the number of class samples. We define each cluster based on the number of samples using the following ranges: less than 100, 100 to 500, 500 to 1000, 1000 to 5000, 5000 to 10,000, and over 10,000. In addition, the mAP@.5:.95 values of each class are classified according to each range and averaged. We can confirm that the larger the number of samples, the greater the value of mAP@.5:.95. Evidently, the model performance increases sharply when we train the model with more samples.

Table 9. Recognition performances of each cluster based on the number of samples.

# of Samples	# of Classes	Average of mAP@.5:.95 Values
<=100	20	0.5204
101~500	20	0.7647
501~1000	17	0.8013
1001~5000	19	0.8879
5001~10,000	7	0.9279
10,001<=	15	0.9417

To further investigate, Table 10 reports the recognition performances for the sampled classes. For instance, Up-Hill shows results significantly lower than for the other classes in all indicators except Recall: Precision (0.54), mAP@.5 (0.556), and mAP@.5:.95 (0.408). We deduce that Up-Hill has the worst results because it used less than 100 samples of data for training. In contrast, Speed Limit 40, Speed Limit 50, and Bicycle Cross Walk all show good results compared with the class with more than 1000 samples. In addition, the Bicycle Cross Walk class presents the best results in all indicators: Precision (0.991), Recall (0.998), mAP@.5 (0.995), and mAP@.5:.95 (0.974).

Table 10. Recognition performances for each traffic sign class (sampled classes).

Class Labels	Images	# of Samples	Precision	Recall	mAP@.5	mAP@.5:.95
Up-Hill		96	0.540	0.979	0.556	0.408
No Pedestrian Passing		101	0.949	0.861	0.889	0.742
Right Lane Ends		667	0.921	0.975	0.973	0.774
Speed Limit 40		3425	0.975	0.966	0.991	0.925
Speed Limit 50		6386	0.969	0.978	0.993	0.951
Bicycle Cross Walk		12,598	0.991	0.998	0.995	0.974

Furthermore, Figure 9 shows the relationship between the number of samples and mAP@.5:.95 for each class. We observe that the results of mAP@.5:.95 do not exceed 0.8 for the class with less than 200 samples. In general, the higher the number of samples, the better the value of mAP@.5:.95. In particular, when the trained model is applied to images, a class with a small number of samples is not recognized at all or is recognized as a similar but different class. To prevent this problem, a potential guideline for reasonable recognition performance is to exploit as many training samples as possible.

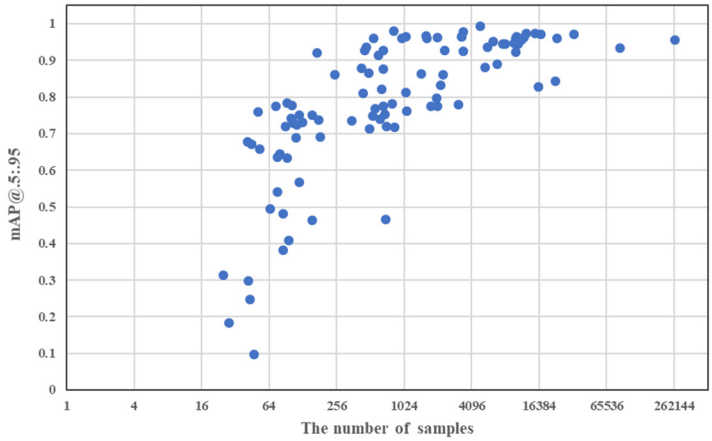


Figure 9. Recognition performance plot for all clusters based the number of samples.

7. Conclusions

In this study, we propose a traffic sign recognition framework based on deep learning from a camera to an edge processing unit to validate an entire system of complex urban road environments in a driving car. Recent traffic sign recognition studies using deep learning-based detection models have shown significant advances in addressing the performance constraints of traffic sign systems resulting from the low quality of images. However, they lack both practical validation on real-world urban road environments including object tracking models, and assessment of the processing time for heavy computation in deep learning models. To alleviate these limitations and provide insightful discussion, we developed a traffic sign recognition system using YOLO models as detectors and SORT-variants models as trackers. We also propose a novel categorization method for the frequently changing urban road environments based on diverse noise types.

We evaluate the proposed framework in a driving car on several expressways to validate the system in terms of accuracy and latency. In particular, we investigate the results from different types of YOLO architectures and tracking models to understand the effect of different road environment conditions on models in terms of mAP performance and inference time. Furthermore, we conduct in-depth ablation studies for a proposed categorization method for complex urban road environments. Additional analyses performed include the effect of weather, light conditions, and the number of traffic signs. Finally, we provide detailed analyses and identify potential issues for recognizing traffic signs in urban road scenes. We believe that this study can become a solid stepping stone and facilitate future research on traffic sign recognition systems. Potential future work includes the identification of broken traffic signs from road scenes. Furthermore, although it is currently difficult to collect data across all environments, we will be able to obtain more informative results if we add various environments (e.g., ice, snow cover, snow fall). Moreover, pretraining the detection models and incorporating them in downstream tasks could also be an interesting future study that maximizes the advantages of knowledge transfer.

Author Contributions: Conceptualization, J.P. and Y.-s.L.; Data curation, C.-i.K. and W.J.; Formal analysis, C.-i.K., and Y.P.; Funding acquisition, Y.-s.L.; Investigation, C.-i.K. and J.P.; Methodology, J.P.; Project administration, Y.P.; Resources, Y.-s.L.; Software, C.-i.K. and W.J.; Supervision, J.P. and Y.-s.L.; Validation, Y.P. and W.J.; Visualization, C.-i.K. and W.J.; Writing—original draft, C.-i.K. and J.P.; Writing—review and editing, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by a Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 21AMDP-C160853-01).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Xie, Y.; Liu, L.F.; Li, C.H.; Qu, Y.Y. Unifying visual saliency with HOG feature learning for traffic sign detection. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009.
2. Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J.Z.; Langer, D.; Pink, O.; Pratt, V.; et al. Towards fully autonomous driving: Systems and algorithms. In Proceedings of the 2011 IEEE intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011.
3. Ziegler, J.; Bender, P.; Schreiber, M.; Lategahn, H.; Strauss, T.; Stiller, C.; Dang, T.; Franke, U.; Appenrodt, N.; Keller, C.G.; et al. Making bertha drive—An autonomous journey on a historic route. *IEEE Intell. Transp. Syst. Mag.* **2014**, *6*, 8–20. [CrossRef]
4. Zhang, P.; Zhang, M.; Liu, J. Real-time HD map change detection for crowdsourcing update based on mid-to-high-end sensors. *Sensors* **2021**, *21*, 2477. [CrossRef] [PubMed]
5. Kim, K.; Cho, S.; Chung, W. HD map update for autonomous driving with crowdsourced data. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1895–1901. [CrossRef]
6. Rajendran, S.P.; Shine, L.; Pradeep, R.; Vijayaraghavan, S. Real-time traffic sign recognition using YOLOv3 based detector. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019.
7. Fazekas, Z.; Balázs, G.; Gyulai, C.; Potyondi, P.; Gáspár, P. Road-Type Detection Based on Traffic Sign and Lane Data. *J. Adv. Transp.* **2022**, *2022*, 6766455. [CrossRef]
8. Kortmann, F.; Fassmeyer, P.; Funk, B.; Drews, P. Watch out, pothole! featuring road damage detection in an end-to-end system for autonomous driving. *Data Knowl. Eng.* **2022**, *142*, 102091. [CrossRef]
9. Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J.; Zhang, L. Image-adaptive YOLO for object detection in adverse weather conditions. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 22 February–1 March 2022.
10. Ellahyani, A.; El Ansari, M.; El Jaafari, I. Traffic sign detection and recognition based on random forests. *Appl. Soft Comput.* **2016**, *46*, 805–815. [CrossRef]
11. Bahlmann, C.; Zhu, Y.; Ramesh, V.; Pellkofer, M.; Koehler, T. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In Proceedings of the IEEE Proceedings Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005.
12. Tao, J.; Wang, H.; Zhang, X.; Li, X.; Yang, H. An object detection system based on YOLO in traffic scene. In Proceedings of the 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 21–22 October 2017.
13. Huang, R.; Pedoeem, J.; Chen, C. YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers. In Proceedings of the 2018 IEEE International Congerence on Big Data, Seattle, WA, USA, 10–13 December 2019.
14. Liu, C.; Tao, Y.; Liang, J.; Li, K.; Chen, Y. Object detection based on YOLO network. In Proceedings of the 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018.
15. Houben, S.; Stallkamp, J.; Salmen, J.; Schlipsing, M.; Igel, C. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013.
16. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. The German traffic sign recognition benchmark: A multi-class classification competition. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011.
17. Shakhuro, V.I.; Konouchine, A.S. Russian traffic sign images dataset. *Comput. Opt.* **2016**, *40*, 294–300. [CrossRef]
18. Fazekas, Z.; Gerencsér, L.; Gáspár, P. Detecting Change between Urban Road Environments along a Route Based on Static Road Object Occurrences. *Appl. Sci.* **2021**, *11*, 3666. [CrossRef]
19. Yang, Y.; Luo, H.; Xu, H.; Wu, F. Towards real-time traffic sign detection and classification. *IEEE trans. Intell. Transp. Syst.* **2016**, *17*, 2022–2031. [CrossRef]
20. Ellahyani, A.; El Ansari, M.; Lahmyed, R.; Trémeau, A. Traffic sign recognition method for intelligent vehicles. *J. Opt. Soc. Am. A* **2018**, *35*, 1907–1914. [CrossRef] [PubMed]
21. 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.
22. Zeng, Y.; Lan, J.; Ran, B.; Wang, Q.; Gao, J. Restoration of motion-blurred image based on border deformation detection: A traffic sign restoration model. *PLoS ONE* **2015**, *10*, e0120885. [CrossRef] [PubMed]
23. Fleyeh, H. Color detection and segmentation for road and traffic signs. In Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 1–3 December 2004; Volume 2, pp. 809–814.
24. Won, W.J.; Lee, M.; Son, J.W. Implementation of road traffic signs detection based on saliency map model. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 542–547.

25. Belaroussi, R.; Foucher, P.; Tarel, J.P.; Soheilian, B.; Charbonnier, P.; Paparoditis, N. Road sign detection in images: A case study. In Proceedings of the IEEE 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 484–488.
26. Wang, C. Research and application of traffic sign detection and recognition based on deep learning. In Proceedings of the IEEE International Conference on Robots & Intelligent System (ICRIS), Changsha, China, 26–27 May 2018; pp. 150–152.
27. Chourasia, J.N.; Bajaj, P. Centroid based detection algorithm for hybrid traffic sign recognition system. In Proceedings of the IEEE 3rd International Conference on Emerging Trends in Engineering and Technology, Goa, India, 19–21 November 2010; pp. 96–100.
28. Wang, G.; Ren, G.; Wu, Z.; Zhao, Y.; Jiang, L. A robust, coarse-to-fine traffic sign detection method. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013.
29. Liang, M.; Yuan, M.; Hu, X.; Li, J.; Liu, H. Traffic sign detection by ROI extraction and histogram features-based recognition. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013.
30. Wang, G.; Ren, G.; Wu, Z.; Zhao, Y.; Jiang, L. A hierarchical method for traffic sign classification with support vector machines. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013.
31. 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.
32. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
33. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015.
35. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
36. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 740–755.
37. Xing, J.; Yan, W.Q. Traffic sign recognition using guided image filtering. In *International Symposium on Geometry and Vision*; Nguyen, M., Yan, W.Q., Ho, H., Eds.; Springer: Cham, Switzerland, 2021; pp. 85–99.
38. Kuznetsova, A.; Maleva, T.; Soloviev, V. Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images. In *International Symposium on Neural Networks*; Han, M., Qin, S., Zhang, N., Eds.; Springer: Cham, Switzerland, 2020; pp. 233–243.
39. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
40. Gunjal, P.R.; Gunjal, B.R.; Shinde, H.A.; Vanam, S.M.; Aher, S.S. Moving object tracking using kalman filter. In Proceedings of the 2018 International Conference on Advances in Communication and Computing Technology (ICACCT), Sangamner, India, 8–9 February 2018.
41. Sahbani, B.; Adiprawita, W. Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system. In Proceedings of the 2016 6th International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 3–4 October 2016.
42. Hou, X.; Wang, Y.; Chau, L.P. Vehicle tracking using deep sort with low confidence track filtering. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019.
43. Yeh, K.H.; Hsu, I.C.; Chou, Y.Z.; Chen, G.Y.; Tsai, Y.S. An Aerial Crowd-Flow Analyzing System for Drone Under YOLOv5 and StrongSort. In Proceedings of the 2022 International Automatic Control Conference (CACS), Kaohsiung, Taiwan, 3–6 November 2022.
44. Zhang, J.; Zou, X.; Kuang, L.D.; Wang, J.; Sherratt, R.S.; Yu, X. CCTSDB 2021: A more comprehensive traffic sign detection benchmark. *Hum. Cent. Comput. Inf. Sci.* **2022**, *12*, 23.
45. Seraj, M.; Rosales-Castellanos, A.; Shalkamy, A.; El-Basyouny, K.; Qiu, T.Z. The implications of weather and reflectivity variations on automatic traffic sign recognition performance. *J. Adv. Transp.* **2021**, *2021*, 5513552. [CrossRef]
46. Fazekas, Z.; Simonyi, E.; Gáspár, P. Glare in Street View images may signify unsafe road locations. In Proceedings of the International Scientific Conference Modern Safety Technologies in Transportation, Kosice, Slovakia, 24–26 September 2013.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.