S & M 3782

Synthetic Training Dataset Generation Using a Digital Twin-based Autonomous Driving Simulator

In-Sung Jang,[1] Ki-Joune Li,[2] Eun-Oh Joo,[3] and Min-Soo Kim[3*]

[1]Mobility Infra Research Section, Electronics and Telecommunications Research Institute,
218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea
[2]School of Computer Science and Engineering, Pusan National University,
2 Busandaehak-ro, Geumjeong-gu, Busan 46241, Korea
[3]Department of Computer Engineering, Daejeon University, 62 Daehak-ro, Dong-gu, Daejeon 34520, Korea

(Received August 26, 2024; accepted September 25, 2024)

Keywords: synthetic training data, virtual training data, deep learning, digital twin, autonomous driving

Recently, extensive research has been conducted on generating virtual training data in a digital twin-based simulator to reduce the time and cost associated with acquiring high-quality training data necessary for autonomous driving. In this study, we propose an efficient method of generating synthetic training datasets for autonomous driving by combining real-world and virtual training data. Specifically, we propose a method of implementing a digital twin-based autonomous driving simulator, collecting large amounts of virtual training data using its camera sensor, and generating synthetic training datasets by combining virtual and real-world training data in various ratios. The effectiveness of these datasets is then validated in deep learning applications, particularly for detecting traffic lights and signal information. Validation results indicate that synthetic training datasets significantly improve deep learning performance, provided they include a sufficient amount of real-world training data to avoid class imbalance issues. We conclude that the synthetic training datasets generated using a digital twin-based simulator are cost-effective and practical for deep learning applications.

1. Introduction

In recent years, deep learning has achieved remarkable progress in tasks such as object detection, semantic segmentation, and time series forecasting within geographic information systems (GISs). These studies rely heavily on various types of training data, including road, aerial, and remote sensing images, and location data.(1–4) For example, deep learning studies have utilized remote sensing and aerial images for detecting vehicles, ships, and people,(5–7) and for segmenting forests, soil, and buildings.(8) Numerous studies in autonomous driving have also relied on road images for detecting vehicles, pedestrians, and road infrastructure. The performance of deep learning models in GISs and autonomous driving heavily relies on the availability of large, high-quality training datasets, particularly aerial and road images. However, collecting such high-quality training data in real-world environments is time-consuming and

[*] Corresponding author: e-mail: minsoo@dju.ac.kr

https://doi.org/10.18494/SAM5343

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

costly. To solve this challenge, researchers have developed methods such as data augmentation and synthetic training data generation.(9–11) Data augmentation techniques such as mirroring, rotation, random cropping, shearing, and local warping increase the diversity and variability of training data, thereby improving performance and robustness in deep learning models. On the other hand, synthetic training data generation typically involves using virtual simulation environments or generative adversarial networks(12,13) to generate large quantities of virtual training data, reducing the dependence on real-world data.

In autonomous driving, various studies have been conducted to improve object detection and segmentation performance using synthetic training datasets. In this study, we present a method of generating such synthetic training datasets using a digital twin-based autonomous driving simulator. Specifically, we generate synthetic training datasets for detecting traffic lights and signal information, and we validate their effectiveness in real-world autonomous driving scenarios. Our approach involves collecting various types of road images from the virtual sensor of an autonomous driving simulator, automatically labeling these images with bounding box locations and signal information annotations, and combining the virtual training data with real-world data from the laboratory for intelligent and safe automobile (LISA) dataset in various ratios to generate synthetic training datasets. We validate the effectiveness of these datasets by YOLOv5-based traffic light and signal information detection. The validation results demonstrate performance improvements in terms of precision, recall, and mean average precision (mAP) for both YOLOv5l and YOLOv5s models when using synthetic training datasets.

The remaining sections of this paper are organized as follows: In Sect. 2, we review recent research related to the generation and validation of synthetic training datasets in digital twin-based virtual environments. In Sect. 3, we present the implementation details of the proposed digital twin-based autonomous driving simulation system and the method of generating multiple synthetic training datasets using this system. In Sect. 4, we validate the performance of our synthetic training datasets by YOLOv5-based traffic light and signal information detection. Finally, in Sect. 5, we present our conclusions and discuss future work.

2. Related Works

Autonomous driving technology typically requires large and diverse training datasets for tasks such as object detection, segmentation, and tracking.(14) In recent years, several training datasets, such as KITTI,(15) nuScenes,(16) LISA,(17) and ETRIDriving,(18) have been introduced to advance autonomous driving technology. These datasets consist of real-world data collected from various sensors, including camera, LiDAR, and radar sensors, and cover a wide range of driving scenarios. In addition to these real-world datasets, there is growing interest in generating virtual training datasets using digital twin-based autonomous driving simulators. Virtual training datasets offer advantages such as the rapid generation of large amounts of data, the easy variation of data distribution, and the ability to simulate challenging scenarios that may be difficult to capture in real-world settings. For instance, various autonomous driving simulators, such as CARLA,(19) LGSVL,(20) AirSim,(21) and VISTA 2.0,(22) have been used to generate virtual training datasets. CARLA, in particular, is widely recognized for its realistic driving environment and ability to generate various sensor data such as RGB images, depth maps, and LiDAR point clouds.

Jeon *et al.*(23) generated virtual training data for object detection using CARLA and demonstrated improved performance when combining virtual training data with real-world data from the Waymo dataset.(24) They argued that adding the virtual training data in deep learning is more cost-effective than adding the real-world data. Deschaud(25) used CARLA to generate a virtual dataset of point clouds and images similar to the KITTI dataset by simulating a vehicle equipped with the same sensors used to collect the KITTI dataset. They collected a total of 5000 virtual data samples across seven different CARLA maps, each representing a different environment, for example, cities, suburbs, mountains, rural areas, or highways. Labels for the virtual training dataset were manually annotated to match the format of the KITTI dataset. They showed that when using the virtual dataset, object detection models achieved similar performance to those trained on the original KITTI dataset. Deschaud *et al.*(26) also presented a synthetic Paris-CARLA-3D dataset consisting of both virtual and real-world point clouds, and validated its applicability to 3D segmentation tasks: semantic segmentation, instance segmentation, and scene completion. Pena *et al.*(27) proposed PerDevKit, a tool to generate virtual training datasets for road objects using CARLA, and showed performance improvements in YOLOv5 when combining virtual and real-world KITTI datasets. In particular, they showed that performance improvements become more evident when the virtual training dataset ratio is relatively low compared with the KITTI dataset ratio. Niranjan *et al.*(28) generated a virtual training dataset using CARLA and validated it using a single-shot multibox detector (SSD)(29) model, showing that combining virtual and real-world datasets enhances performance. Specifically, they generated and validated virtual training datasets for five objects: vehicles, bicycles, motorbikes, traffic lights, and traffic signs. Dworak *et al.*(30) collected virtual point clouds from CARLA and combined them with KITTI data to generate synthetic training and test datasets. They compared the performance characteristics of three object detection models of VoxelNet,(31) YOLO3D,(32) and PointPillars(33) using synthetic training and test datasets. They argued that CARLA can serve as a valuable tool for collecting virtual training data and that the combination of virtual and real-world training datasets can lead to improved object detection performance compared with using either type of dataset in isolation. Weng *et al.*(34) proposed the AIODrive dataset, a comprehensive virtual dataset designed to support various perception tasks in autonomous driving, including challenging conditions such as adverse weather and lighting. This dataset includes multiple sensor modalities, including RGB cameras, depth cameras, and LiDAR, and provides annotations for various tasks, such as object detection, object tracking, object trajectory prediction, and segmentation. They validated the AIODrive dataset's effectiveness in 2D and 3D object detection tasks.

In contrast to most studies that focus on large objects such as vehicles and pedestrians, our research aims to generate synthetic training datasets for smaller objects such as traffic signals and to validate their effectiveness in real-world scenarios. Specifically, we first implement a digital twin-based autonomous driving simulator to generate various types of virtual training data. We then propose a method of automatically generating virtual training data from the simulator, combine it with real-world data to generate various synthetic training datasets, and validate their effectiveness in traffic light and signal information detection.

3. Synthetic Training Dataset Generation by Digital Twin-based Simulation

In this section, we describe the implementation details of a digital twin-based autonomous driving simulation system for collecting virtual road images and the generation of synthetic training datasets using the virtual road images. The process consists of four steps: data collection, auto-labeling, refinement, and combining.

3.1 Implementation of digital twin-based autonomous driving simulation system

Figure 1 illustrates the configuration of the digital twin-based autonomous driving simulation system used in this study. The simulation system collects various virtual road images by extending the framework implemented by Kim and Jang[(35)] using CARLA. The system supports digital twin-based simulation by importing real-world 3D spatial data and high-definition road maps. It also allows customized vehicle driving routes and dynamic traffic light changes, providing a versatile environment for generating diverse training data.

In the server system, we implemented an import module and a conversion module to build a simulation environment identical to the real world using 3D spatial data and high-definition road maps. The import module imports various 3D spatial data in format such as FBX or OBJ, which can be used in the simulation system. In contrast, the conversion module, implemented from scratch, converts high-definition road maps into the OpenDRIVE format usable in the simulation system, because most high-definition road maps have proprietary formats that are not standardized. Specifically, we implemented the National Geographic Information Institute high-definition map (NGII HD map)-to-OpenDRIVE conversion function by expanding the conversion module partially implemented by Kim and Jang.[(35)] Finally, we further implemented a digital twin-based simulation module by expanding the CARLA agent library to collect various virtual road images. The simulation module can dynamically set user-defined driving routes instead of the default driving route set based on the A^* algorithm and manually control the real-time route of an ego vehicle. It can also configure all traffic signal information during the simulation to the user's specifications.

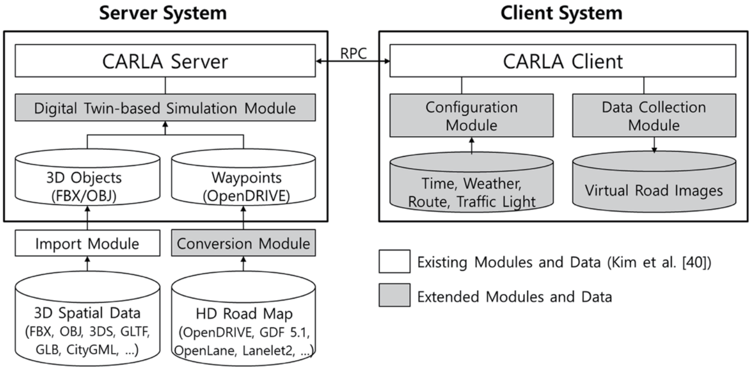

Server System
Client System
CARLA Server
RPC
CARLA Client
Digital Twin-based Simulation Module
Configuration Module
Data Collection Module
3D Objects (FBX/OBJ)
Waypoints (OpenDRIVE)
Time, Weather, Route, Traffic Light
Virtual Road Images
Import Module
Conversion Module
Existing Modules and Data (Kim et al. [40])
Extended Modules and Data
3D Spatial Data (FBX, OBJ, 3DS, GLTF, GLB, CityGML, ...)
HD Road Map (OpenDRIVE, GDF 5.1, OpenLane, Lanelet2, ...)

Fig. 1. Configuration of digital twin-based autonomous driving simulation system.

Figure 2 shows an example of our system's simulation. Specifically, Fig. 2(a) shows a real-world street view image, Fig. 2(b) shows the 3D spatial data built for the same area, and Fig. 2(c) shows the digital twin-based simulation peerformed using both real-world 3D spatial data and the NGII HD map.

In the client system, we additionally implemented a configuration module and a data collection module. These modules, also implemented by expanding the CARLA API, enable the collection of various virtual road images. As shown in Fig. 3, the configuration module sets various driving environmental settings such as weather, time, the number of vehicles, user-specified driving routes, and user-specified signal information. The data collection module collects virtual road images by specifying various options for image format, size, acquisition interval, and camera sensor type. For instance, in this study, we collected 70000 RGB images with a resolution of 1280 × 720 in driving environmental settings of daytime, clear weather, and 30 vehicles.

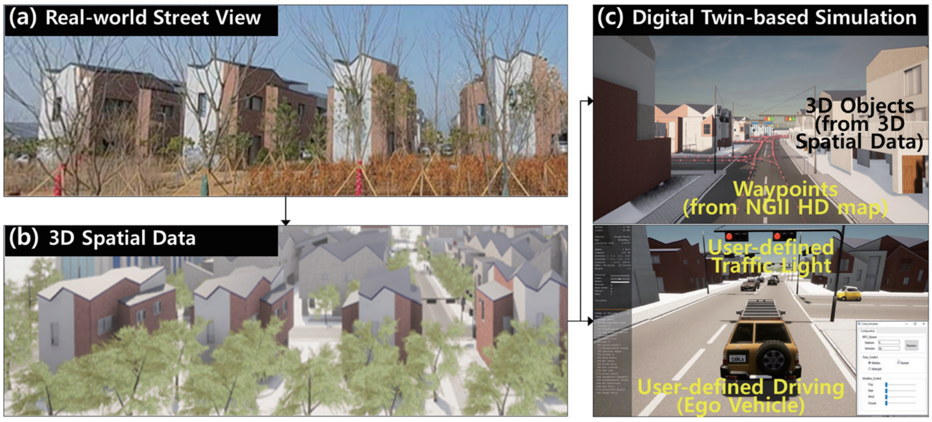
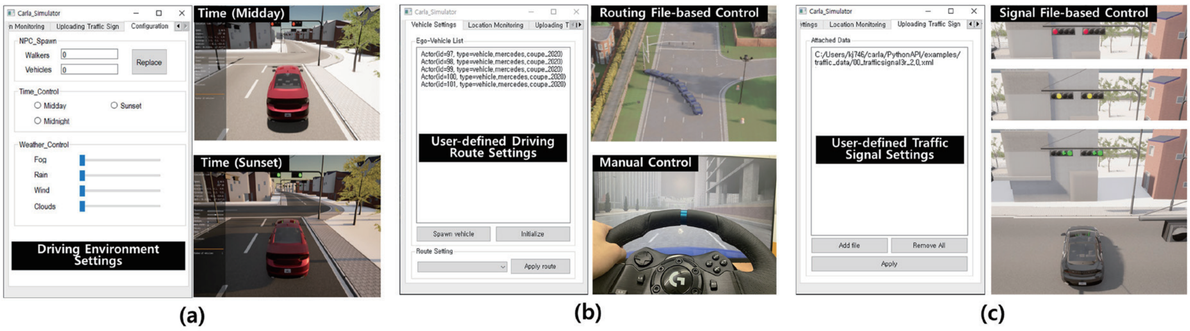

(a) Real-world Street View
(b) 3D Spatial Data
(c) Digital Twin-based Simulation
3D Objects (from 3D Spatial Data)
Waypoints (from NGII HD map)
User-defined Traffic Light
User-defined Driving (Ego Vehicle)

Fig. 2. (Color online) Implementation of digital twin-based autonomous driving simulation server: (a) real-world street view, (b) 3D spatial data built for real-world buildings, and (c) digital twin-based simulation performed using both 3D spatial data and NGII HD map.



Walkers 0
Vehicles 0

```
Actor(id=97, type=vehicle.mercedes.coupe_2020)
Actor(id=98, type=vehicle.mercedes.coupe_2020)
Actor(id=99, type=vehicle.mercedes.coupe_2020)
Actor(id=100, type=vehicle.mercedes.coupe_2020)
Actor(id=101, type=vehicle.mercedes.coupe_2020)
```

```
C:/Users/kj746/carla/PythonAPI/examples/
traffic_data/00_trafficsignal3r_2.0.xml
```

Fig. 3. (Color online) Implementation of configuration module: (a) driving environmental settings such as weather, time, and number of vehicles, (b) user-specified driving route settings, and (c) user-specified signal information

3.2 Synthetic training dataset generation

The process of generating synthetic training datasets includes four major steps: data collection, auto-labeling, refinement, and combining. Figure 4 depicts each step in detail.

Before data collection, we first set up a simulation environment to collect various high-quality virtual road images. Table 1 shows the simulation map, vehicle, weather, time, and sensor settings.

For the map setting, we simulated various maps covering intersections and roads from small villages to downtowns to collect road images including traffic lights in various cases. For the weather and time setting, we simulated clear weather and noontime to collect large numbers of virtual road images in a consistent environment. Finally, we simulated a fleet of 30 vehicles, each equipped with RGB camera sensors and set to the autopilot mode. Figure 5 shows the downtown and village maps used in the simulation.

In the data collection step, we collected 70,000 virtual road images with a resolution of 1280 × 960 using the simulation system. These images captured various urban and suburban environments under consistent weather and lighting conditions. In the auto-labeling step, we trained two YOLOv5-based models, TM_1 and TM_2, to detect traffic lights and signal information in road images, respectively, using training data of 14000 real-world traffic light images acquired from the LISA dataset. Using TM_1, we detected traffic lights in all captured virtual road images and generated the first labeled road images (L_1) with 2D bounding boxes and location annotations for traffic lights. Similarly, using TM_2, we detected signal information for each traffic light in the first labeled road images (L_1) and generated the second labeled road

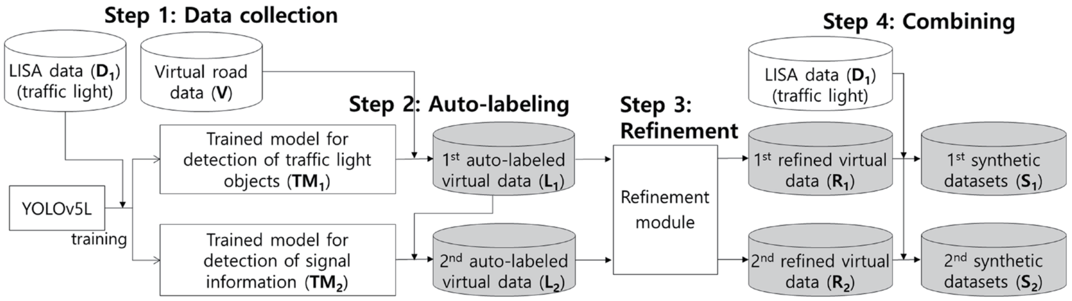

Step 1: Data collection
LISA data (D1) (traffic light)
Virtual road data (V)
Step 2: Auto-labeling
YOLOv5L
training
Trained model for detection of traffic light objects (TM1)
Trained model for detection of signal information (TM2)
1st auto-labeled virtual data (L1)
2nd auto-labeled virtual data (L2)
Step 3: Refinement
Refinement module
Step 4: Combining
LISA data (D1) (traffic light)
1st refined virtual data (R1)
2nd refined virtual data (R2)
1st synthetic datasets (S1)
2nd synthetic datasets (S2)

Fig. 4. Synthetic training dataset generation process including four major steps: data collection, auto-labeling, refinement, and combining.

Table 1
Simulation map, vehicle, weather, time, and sensor settings.

Setup	Description
Map	Eight maps representing various road environments such as urban and suburban areas
Weather and time	Consistent data collection environment with clear weather and noontime
Vehicle	Randomly generated 30 vehicles and random selection of ego vehicles for data collection
Sensor	RGB camera sensor capturing road images with a resolution of 1280 × 960

Fig. 5. (Color online) Downtown and village maps used to collect road images including traffic lights in various cases.

images (L_2) with additional annotations for red, green, and yellow signal information. In the refinement step, we manually corrected invalid bounding boxes and annotations in the auto-labeled data, generating the refined results R_1 and R_2. This refinement was performed using an annotation tool developed to efficiently edit bounding boxes and signal information for each traffic light. Finally, in the combining step, we generated two types of synthetic training dataset: S_1 and S_2. S_1 and S_2 were generated by combining R_1 and R_2, respectively, with the real-world LISA data (D_1). To sufficiently validate the performance of our synthetic training datasets, we generated multiple sets of S_1 and S_2 by combining R_1 and R_2 with D_1 in various ratios. Figure 6 shows examples of R_1, R_2, and D_1 used to generate S_1 and S_2.

In this study, we generated as much synthetic training data as possible to accurately validate their performance in traffic light and signal information detection. Table 2 shows the data counts and descriptions of the various datasets collected or generated during the synthetic training dataset generation process. As shown in Table 2, we first acquired 14000 D_1 and 10954 D_2 images from the LISA dataset to have adequate amounts of training and testing data for traffic light and signal information detection. We then collected 70000 virtual road images (V) from the simulator and created 15392 and 12000 auto-labeled data (L_1 and L_2) from V and 14000 and 8400 refined training data (R_1 and R_2) from L_1 and L_2, respectively. Finally, we generated six synthetic training datasets (S_1) by combining R_1 with D_1 in various ratios, where the ratio of R_1 increases in 20% increments from 0 to 14000. Similarly, we generated four synthetic training datasets (S_2) by combining R_2 with D_1 in various ratios, where the R_2 ratio increases in 20% increments from 0 to 8400.

4. Validation of Synthetic Training Dataset

In this section, we present the validation results for the synthetic training datasets. We evaluated the performance of the YOLOv5 model for traffic light and signal information detection. For traffic light and signal information detection, we validated the YOLOv5 model trained using synthetic training datasets by determining whether it can detect traffic lights and the red, green, and yellow signals in real-world images. We used two variants of the YOLOv5

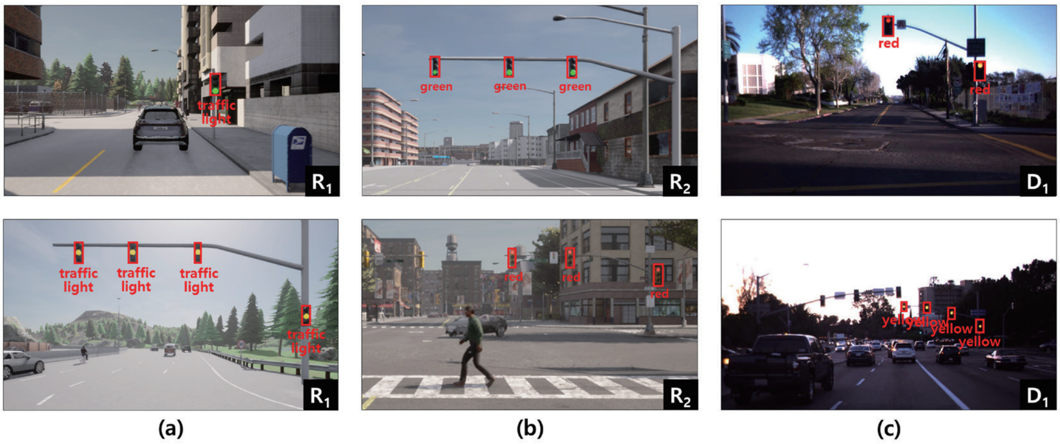


Fig. 6. (Color online) Composition of synthetic training datasets (S_1 and S_2): (a) refined result R_1, (b) refined result R_2, and (c) LISA data D_1.

Table 2
Summary of datasets collected or generated in synthetic training dataset generation process.

Dataset	Count	Description
LISA data for training (D_1)	14000	Real-world training data from LISA for traffic light and signal information detection
LISA data for testing (D_2)	10954	Real-world testing data from LISA for traffic light and signal information detection
Virtual data (V)	70000	Virtual road images collected at resolution of 1280 × 960 from proposed digital twin-based simulator
1st auto-labeled virtual data (L_1)	15392	Road images with 2D bounding box and location for traffic light, automatically labeled from V
2nd auto-labeled virtual data (L_2)	12000	Road images with annotation of red, green, and yellow signal information in addition to bounding box, automatically labeled from L_1
1st refined virtual data (R_1)	14000	Refined road images generated by correcting invalid bounding boxes and annotations in L_1
2nd refined virtual data (R_2)	8400	Refined road images generated by correcting invalid bounding boxes and annotations in L_2
1st synthetic training datasets (S_1)	14000–28000	Six synthetic training datasets generated by combining D_1 with R_1, where R_1 increases from 0 to 100% in 20% increments
2nd synthetic training datasets (S_2)	14000–22400	Four synthetic training datasets generated by combining D_1 with R_2, where the number of R_1 increases from 0 to 100% in 20% increments

model, YOLOv5l and YOLOv5s, to determine the effectiveness of synthetic training datasets on deep learning models that perform high-precision detection and fast inference, respectively. In addition, we used S_1 and S_2 as synthetic training datasets for traffic light and signal information detection, respectively, and used D_2, consisting of real-world data from LISA, as the test dataset. D_2 contains annotations for both traffic light location and signal information, making it suitable for traffic light and signal information detection.

To validate the effectiveness of the synthetic training datasets in various scenarios, we conducted two types of validation. In the first validation, we incrementally increased the total number of synthetic training datasets by adding virtual training data in 20% increments to the real-world training data, performing validation on six different synthetic training datasets (S_1)

for traffic light detection, ranging from 14000 to 28000. Similarly, we performed validation on four different synthetic training datasets (S_2) for signal information detection, ranging from 14000 to 22400. In the second validation, we fixed the total number of synthetic training datasets and compared the performance with varying ratios of virtual to real-world training data (100:0, 50:50, and 0:100). We performed validations on three different synthetic training datasets for traffic light detection and signal information detection, with the total number fixed at 14000 and 8400, respectively. Finally, we compared precision, recall, and mAP as performance metrics to evaluate the accuracy and completeness of object detection using YOLOv5s and YOLOv5l models.

4.1 Validation by traffic light detection

In this section, we present the validation results of YOLOv5l and YOLOv5s for traffic light detection when using the synthetic training datasets. Tables 3 and 4 show the results for the first and second validations, respectively.

In Table 3, we can see that the overall performance of YOLOv5l improves as more virtual training data are added to the synthetic training dataset. Specifically, when 14000 virtual training data are added, precision and mAP significantly improve from 0.706 to 0.878 and from 0.666 to 0.836, respectively, with recall also improving from 0.652 to 0.772. Although the

Table 3
First validation results of YOLOv5l and YOLOv5s for traffic light detection.

Model	Synthetic training dataset	Test dataset	Precision	Recall	F1-score	mAP
YOLOv5l	D_1(14000)	D_2(10954)	0.706	0.652	0.678	0.666
	D_1(14000) + R_1(2800)		0.690	0.677	0.684	0.679
	D_1(14000) + R_1(5600)		0.756	0.688	0.721	0.743
	D_1(14000) + R_1(8400)		0.724	0.694	0.709	0.690
	D_1(14000) + R_1(11200)		0.699	0.747	0.723	0.745
	D_1(14000) + R_1(14000)		0.878	0.772	0.822	0.836
YOLOv5s	D_1(14000)	D_2(10954)	0.703	0.609	0.652	0.653
	D_1(14000) + R_1(2800)		0.719	0.678	0.698	0.699
	D_1(14000) + R_1(5600)		0.750	0.665	0.705	0.701
	D_1(14000) + R_1(8400)		0.734	0.670	0.701	0.714
	D_1(14000) + R_1(11200)		0.686	0.779	0.730	0.737
	D_1(14000) + R_1(14000)		0.724	0.708	0.715	0.717

Table 4
Second validation results of YOLOv5l and YOLOv5s for traffic light detection.

Model	Synthetic training dataset	Test dataset	Precision	Recall	F1-score	mAP
YOLOv5l	D_1(14000)	D_2(10954)	0.706	0.652	0.678	0.666
	D_1(7000) + R_1(7000)		0.815	0.725	0.768	0.771
	R_1(14000)		0.786	0.457	0.578	0.533
YOLOv5s	D_1(14000)	D_2(10954)	0.703	0.609	0.652	0.653
	D_1(7000) + R_1(7000)		0.719	0.658	0.688	0.673
	R_1(14000)		0.434	0.464	0.449	0.358

performance of YOLOv5s is not as good as that of YOLOv5l, it still shows overall improvement. When 14000 virtual training data are added, precision, recall, and mAP improve from 0.703 to 0.724, from 0.609 to 0.708, and from 0.653 to 0.717, respectively. Thus, meaningful performance improvement can be achieved by incorporating a large number of virtual training data into the synthetic training dataset for traffic light detection. In particular, this training data augmentation using virtual training data appears more suitable for YOLOv5l, which prioritizes high-precision inference, than for YOLOv5s, which prioritizes faster inference.

In Table 4, both YOLOv5l and YOLOv5s achieve the best precision, recall, and mAP when the training dataset consists of a 50:50 ratio of real-world and virtual training data, with a total number of 14000 data. Additionally, the worst recall and mAP can be seen when using a training dataset consisting of 100% virtual training data. Specifically, the recall and mAP of YOLOv5l are 0.457 and 0.533, whereas those of YOLOv5s are 0.464 and 0.358, respectively. Thus, for example, when there are more than 50% of real-world training data, meaningful performance improvement can be achieved by appropriately combining real-world training data and virtual learning data. This suggests that the training process benefits from the augmentation of diverse virtual training data that do not exist in the real-world training data.

4.2. Validation by signal information detection

In this section, we present validation results for signal information detection to validate the effectiveness of synthetic training datasets in detecting small objects in road images. Tables 5 and 6 show the results for the first and second validations, respectively.

Table 5
First validation results of YOLOv5l and YOLOv5s for signal information detection.

Model	Synthetic training dataset	Test dataset	Precision	Recall	F1-score	mAP
YOLOv5l	D_1(14000)	D_2(10954)	0.748	0.679	0.712	0.752
	D_1(14000) + R_2(2800)		0.885	0.709	0.788	0.808
	D_1(14000) + R_2(5600)		0.883	0.721	0.794	0.809
	D_1(14000) + R_2(8400)		0.929	0.808	0.865	0.848
YOLOv5s	D_1(14000)	D_2(10954)	0.703	0.609	0.653	0.653
	D_1(14000) + R_2(2800)		0.776	0.668	0.718	0.729
	D_1(14000) + R_2(5600)		0.728	0.704	0.716	0.712
	D_1(14000) + R_2(8400)		0.762	0.709	0.735	0.699

Table 6
Second validation results of YOLOv5l and YOLOv5s for signal information detection.

Model	Synthetic training dataset	Test dataset	Precision	Recall	F1-score	mAP
YOLOv5l	D_1(8400)	D_2(10954)	0.766	0.695	0.729	0.755
	D_1(4200) + R_2(4200)		0.718	0.598	0.653	0.647
	R_2(8400)		0.691	0.504	0.583	0.468
YOLOv5s	D_1(8400)	D_2(10954)	0.729	0.643	0.684	0.670
	D_1(4200) + R_2(4200)		0.725	0.659	0.691	0.643
	R_2(8400)		0.643	0.448	0.529	0.457

In Table 5, YOLOv5l shows significant performance improvements as more virtual training data are added to the synthetic training dataset. Specifically, when 8,400 virtual training data are added, precision, recall, and mAP improve from 0.748 to 0.929, from 0.679 to 0.808, and from 0.752 to 0.848, respectively. YOLOv5s also shows slight performance improvements, with precision, recall, and mAP increasing from 0.703 to 0.762, from 0.609 to 0.709, and from 0.653 to 0.699, respectively. These results indicate that training data augmentation using virtual training data can achieve significant performance improvement even in small object detection tasks such as signal information detection.

In Table 6, both YOLOv5l and YOLOv5s achieve the best precision, recall, and mAP when the training dataset consists entirely of real-world data. When the ratio of real-world to virtual training data is 50:50, the precision, recall, and mAP of YOLOv5l decrease slightly from 0.776 to 0.718, from 0.695 to 0.598, and from 0.755 to 0.647, respectively, with similar trends observed for YOLOv5s. The worst performance is observed when using a training dataset consisting of 100% virtual training data such as Table 4. Thus, when the number of real-world training data in the synthetic training dataset is insufficient, combining virtual training data does not significantly improve deep learning performance, likely owing to the lack of various types of real-world signal information data.

4.3 Discussion

From the results of the validation of the synthetic training datasets by traffic light and signal information detection, we conclude the following:

- If the synthetic training datasets contain a sufficient number of real-world training data to avoid class imbalance problems, the addition of virtual training data can significantly improve deep learning performance. Conversely, if the number of real-world training data is insufficient, the addition of virtual training data does not have a significant effect on improving deep learning performance.
- The synthetic training datasets are effective in recognizing small objects, such as signal information, as well as traffic lights in road images.
- The synthetic training datasets have a greater impact on the performance improvement of models such as YOLOv5l, which prioritizes accuracy, than models such as YOLOv5s, which prioritize inference speed.
- Synthetic training datasets consisting only of virtual data significantly decrease deep learning performance.

These conclusions suggest that synthetic training datasets are sufficiently usable for detecting various objects, such as road signs and pedestrians, in future work. Additionally, we consider that these synthetic training datasets will be effective for not only the YOLO model, but also various object detection models such as SSD and Faster RCNN.

5. Conclusions

In this study, we generated multiple synthetic training datasets by combining real-world training data with virtual training data collected using a digital twin-based autonomous driving simulator. We validated the effectiveness of these synthetic training datasets by traffic light and signal information detection. To perform validation in various environments, we varied the ratio of virtual to real-world training data in the synthetic training datasets and used two deep learning models, YOLOv5s and YOLOv5l. The validation results showed that synthetic training datasets can significantly improve deep learning performance if they include a sufficient amount of real-world training data to avoid class imbalance problems. Additionally, they showed that these synthetic training datasets are effective even in detecting small objects, such as signal information.

However, this study has limitations in that we only used the YOLO model for performance validation and did not consider various road objects other than traffic lights. In addition, it has a limitation in that we cannot conduct experiments in a domestic digital twin environment owing to the lack of high-resolution 3D spatial data and high-definition maps. Therefore, future work will extend this research in two directions. First, we aim to generate and validate domestic synthetic training datasets for various road objects such as road signs, lanes, pedestrians, and bicycles. Second, we plan to more accurately validate the effectiveness of synthetic training datasets using various object detection models, such as SSD and Faster RCNN, in addition to YOLO.

Acknowledgments

This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (24ZR1210, DNA-Based National Intelligent Core Technology Development).

References

1 A. Goel, A. K. Goel, and A. Kumar: Spatial Inf. Res. **31** (2023) 275. https://doi.org/10.1007/s41324-022-00494-x
2 B. Mishra, A. Dahal, N. Luintel, T. B. Shahi, S. Panthi, S. Pariyar, and B. R. Ghimire: Spatial Inf. Res. **30** (2022) 215. https://doi.org/10.1007/s41324-021-00425-2
3 X. Jin, H. Yang, X. He, G. Liu, Z. Yan, and Q. Wang: Remote Sens. **15** (2023) 3160. https://doi.org/10.3390/rs15123160
4 N. E. N. Sey, M. Amo-Boateng, M. K. Domfeh, A. T. Kabo-Bah, and P. Antwi-Agyei: Spatial Inf. Res. **31** (2023) 501. https://doi.org/10.1007/s41324-023-00518-0
5 A. Mehran, S. Tehsin, and M. Hamza: Spatial Inf. Res. **31** (2023) 61. https://doi.org/10.1007/s41324-022-00482-1
6 B. He, X. Li, B. Huang, E. Gu, W. Guo, and L. Wu: Remote Sens. **13** (2021) 4999. https://doi.org/10.3390/rs13244999
7 C. Yu, Z. Feng, Z. Wu, R. Wei, B. Song, and C. Cao: Remote Sens. **15** (2023) 3551. https://doi.org/10.3390/rs15143551
8 F. Manzouri, M. Zare, and S. Shojaei: Spatial Inf. Res. **30** (2022) 551. https://doi.org/10.1007/s41324-022-00452-7
9 A. Figueira and B. Vaz: Mathematics **10** (2022) 2733. https://doi.org/10.3390/math10152733
10 B. Kiefer, D. Ott, and A. Zell: Proc. 2022 26th Int. Conf. Pattern Recognition (ICPR, 2022) 3564–3571. https://doi.org/10.1109/ICPR56361.2022.9956710
11 I. A. Ribeiro, T. Ribeiro, G. Lopes, and A. F. Ribeiro: Algorithms **16** (2023) 411. https://doi.org/10.3390/a16090411
12 A. Bousmina, M. Selmi, M. A. B. Rhaiem, and I. R. Farah: Remote Sens. **15** (2023) 3626. https://doi.org/10.3390/rs15143626

13 Z. Zhang, Z. Yan, J. Jing, H. Gu, and H. Li: Remote Sens. **15** (2023) 265. https://doi.org/10.3390/rs15010265

14 J. Kang, S. Han, N. Kim, and K. Min: ETRI J. **43** (2021) 630. https://doi.org/10.4218/etrij.2021-0055

15 A. Geiger, P. Lenz, and R. Urtasun: Proc. 2012 IEEE Conf. Computer Vision and Pattern Recognition (CVPR, 2012) 3354-3361. https://doi.org/10.1109/CVPR.2012.6248074

16 H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom: Proc. 2020 IEEE Conf. Computer Vision and Pattern Recognition (CVPR, 2020) 11618–11628. https://doi.org/10.1109/CVPR42600.2020.01164

17 LISA Traffic Light Dataset: https://www.kaggle.com/datasets/mbornoe/lisa-traffic-light-dataset (accessed July 2024).

18 D. Choi, S. Han, K. Min, and J. Choi: ETRI J. **44** (2022) 1004. https://doi.org/10.4218/etrij.2021-0192

19 A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun: Proc. 2017 1st Int. Conf. Robot Learning (2017) 1–16. https://doi.org/10.48550/arXiv.1711.03938

20 G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim: Proc. 2020 IEEE Conf. Intelligent Transportation Systems (ITSC, 2020) 1–6. https://doi.org/10.1109/ITSC45102.2020.9294422

21 S. Shah, D. Dey, C. Lovett, and A. Kapoor: Proc. 2017 17th Int. Conf. Field and Service Robotics (2017) 621–635. https://doi.org/10.1007/978-3-319-67361-5_40

22 A. Amini, T. H. Wang, I. Gilitschenski, W. Schwarting, Z. Liu, S. Han, S. Karaman, and D. Rus: Proc. 2022 IEEE Conf. Robotics and Automation (ICRA, 2022) 2419–2426. https://doi.org/10.1109/ICRA46639.2022.9812276

23 H. M. Jeon, L. H. Pham, D. N. N. Tran, H. H. Nguyen, and J. W. Jeon: Proc. 2022 IEEE Conf. Consumer Electronics-Asia (ICCE-Asia, 2022) 1–4. https://doi.org/10.1109/ICCE-Asia57006.2022.9954858

24 P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, I. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov: Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2020) 2443–2451. https://doi.org/10.1109/CVPR42600.2020.00252

25 J. E. Deschaud: Computer Vision and Pattern Recognition (2021). https://doi.org/10.48550/arXiv.2109.00892 (accessed July 2024).

26 J. E. Deschaud, D. Duque, J. P. Richa, S. Velasco-Forero, B. Marcotegui, and F. Goulette: Remote Sens. **13** (2021) 4713. https://doi.org/10.3390/rs13224713

27 J. Pena, L. M. Bergasa, M. Antunes, F. Arango, C. Gomez-Huelamo, and E. Lopez-Guillen: Proc. 2022 IEEE Conf. Intelligent Transportation Systems (ITSC, 2022) 4095–4100. https://doi.org/10.1109/ITSC55140.2022.9922369

28 D. R. Niranjan, B. C. VinayKarthik, and Mohana: Proc. 2021 IEEE Conf. Smart Electronics and Communication (ICOSEC, 2021) 1251–1258. https://doi.org/10.1109/ICOSEC51865.2021.9591747

29 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg: LNCS **9905** (2016) 21. https://doi.org/10.1007/978-3-319-46448-0_2

30 D. Dworak, F. Ciepiela, J. Derbisz, I. Izzat, M. Komorkiewicz, and M. Wojcik: Proc. 2019 IEEE Conf. Methods and Models in Automation and Robotics (MMAR, 2019) 600–605. https://doi.org/10.1109/MMAR.2019.8864642

31 Y. Zhou, and O. Tuzel: Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2018) 4490–4499. https://doi.org/10.1109/CVPR.2018.00472

32 W. Ali, S. Abdelkarim, M. Zidan, M. Zahran, and A. E. Sallab: LNCS **11131** (2019) 716. https://doi.org/10.1007/978-3-030-11015-4_54

33 A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom: Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2019) 12689–12697. https://doi.org/10.1109/CVPR.2019.01298

34 X. Weng, Y. Man, D. Cheng, J. Park, Y. Yuan, M. O'Toole, and K. M. Kitani: All-In-One Drive: A large-scale comprehensive perception dataset with high-density long-range point clouds https://doi.org/10.13140/RG.2.2.21621.81122 (accessed July 2024).

35 M. Kim and I. Jang: Sens. Mater. **34** (2022) 4813. https://doi.org/10.18494/SAM3966