Sensors and Materials, Vol. 38, No. 7 (2026) 3815–3836
MYU Tokyo

S & M 4536

Impact of Object Coverage on Aerial Semantic Segmentation: A Data-centric Approach for Efficient Training and Real-time Inference in Drone Applications

Kwon-Cheol Lee, Yeong-Wuk Kim, Chan-Ui Song, and Min-Soo Kim*

Department of Computer Engineering, Daejeon University,
62 Daehak-ro, Dong-gu, Daejeon 34520, Korea

(Received March 30, 2026; accepted June 22, 2026)

Keywords: aerial image segmentation, object coverage, lightweight models, real-time inference, UAV localization

The rapid advancement of advanced air mobility (AAM) and urban air mobility (UAM) has increased the demand for the precise localization of drones in complex urban environments, where global navigation satellite systems (GNSS) sensors often suffer from degradation due to buildings, interference, and multipath effects. Vision-based localization using aerial imagery has therefore emerged as a promising alternative, relying on the reliable segmentation of quasi-static objects such as buildings and roads. In this study, we address the challenge of achieving both high segmentation accuracy and real-time performance under resource-constrained drone environments. We propose a coverage-aware, data-centric training strategy that systematically organizes training datasets on the basis of the proportion of building and road pixels. In addition, we develop lightweight segmentation models by adapting backbone networks and model scales of YOLO11-seg and DeepLabV3+. Extensive experiments demonstrate that midrange-coverage datasets (20–40%) yield the most stable and balanced performance, while removing low-coverage samples improves data efficiency without notable performance degradation. Furthermore, all proposed models achieve real-time inference exceeding 60 frames per second (FPS), confirming their suitability for onboard deployment. Overall, the results of this study provide practical guidelines for jointly optimizing training data composition and model efficiency in AAM/UAM applications.

1. Introduction

Recent advances in advanced air mobility (AAM) and urban air mobility (UAM) have significantly increased the demand for precise localization technologies for an unmanned aerial vehicle (UAV) operating in complex urban environments. In such environments, the reliability of global navigation satellite systems (GNSS) sensors is often degraded by high-rise buildings, signal blockage, and multipath interference, leading to GNSS-denied conditions. To enable

*Corresponding author: e-mail: minsoo@dju.ac.kr
https://doi.org/10.18494/SAM6357

ISSN 0914-4935

robust navigation in these scenarios, vision-based localization has emerged as a promising complementary solution to GNSS sensors.[1–6] Aerial imagery, in particular, contains abundant quasi-static features such as buildings and roads, whose visual characteristics remain largely stable over time in urban environments. Accordingly, it is widely utilized in vision-based localization technologies that estimate absolute position by extracting feature points from these structures and matching them with preexisting geospatial data. In this context, semantic segmentation is widely employed to reliably extract and delineate building and road regions, which serve as stable references for feature matching.[7]

However, aerial image semantic segmentation for AAM/UAM localization must simultaneously achieve high accuracy and real-time performance under the strict computational and power constraints of UAV platforms. Although recent deep learning models have demonstrated remarkable accuracy, their complex architectures and high computational demands limit their practical deployment in real-time UAV systems. To address this issue, numerous lightweight models and efficient backbone networks have been proposed. Nevertheless, a systematic analysis of the trade-off between segmentation accuracy and inference speed under varying model complexities remains limited. Furthermore, existing studies are primarily focused on model architecture and training strategies, while the impact of training data composition on segmentation performance has not been sufficiently investigated. In particular, the spatial coverage of target objects significantly affects both visual context and feature distribution in aerial imagery, yet it has been largely overlooked as a systematic factor in training data design.

To address these challenges, in this study, we propose a unified framework that jointly considers model lightweighting and training data composition. Specifically, we design lightweight variants of YOLO11-seg and DeepLabV3+ by adjusting backbone networks and model scales, and systematically analyze their accuracy–speed trade-offs. In addition, we introduce a coverage-based training data construction strategy that accounts for the spatial proportion of building and road objects in aerial imagery, and evaluate its impact on segmentation performance.

The main contributions of this study are summarized as follows.

1) We propose a coverage-aware training data construction strategy that explicitly models the proportion of building and road objects, enabling a systematic analysis of its impact on segmentation performance.
2) We develop multiple lightweight configurations of YOLO11-seg and DeepLabV3+, and evaluate the trade-off between segmentation accuracy and inference speed.
3) We derive practical guidelines for optimal data composition and model selection in real-world AAM/UAM applications.

The remainder of this paper is organized as follows. Section 2 contains a review of related work on aerial image semantic segmentation and training data composition. In Sect. 3, we present the proposed coverage-based training data construction method. The lightweight model configurations and backbone design strategies are described in Sect. 4. In Sect. 5, we introduce the experimental setup and present the performance evaluation results in terms of accuracy and inference speed. Finally, in Sect. 6, we conclude the paper and discuss future research directions.

2. Related Work

2.1 Semantic segmentation of buildings and roads in aerial imagery

Buildings and roads in aerial imagery are representative quasi-static structures in urban environments and serve as essential elements for applications such as localization, map updating, and disaster response. With the rapid advancement of deep-learning-based semantic segmentation, extensive research has been conducted to accurately extract these structures from aerial images.[8–17] Early approaches primarily relied on encoder–decoder architectures such as U-Net[18] and SegNet,[19] which improved segmentation performance by integrating multiscale features and recovering spatial details through upsampling. Subsequently, the introduction of atrous separable convolutions and atrous spatial pyramid pooling (ASPP) led to the development of DeepLab-based models, which demonstrated strong performance in handling objects of varying scales and complex geometries while maintaining relatively efficient computational structures.[20,21] In parallel, the You Only Look Once (YOLO) family, originally designed for real-time object detection using a single-stage architecture, has been extended to segmentation tasks through the incorporation of segmentation heads. These YOLO-based segmentation models have gained attention for their ability to provide fast inference while maintaining competitive accuracy, making them suitable for real-time applications.[22,23]

More recently, increasing attention has been directed toward lightweight semantic segmentation models to enable real-time deployment on resource-constrained platforms such as UAVs. Representative approaches include the adoption of efficient backbone networks such as MobileNet[24] and EfficientNet,[25] as well as the incorporation of attention mechanisms into compact architectures. For instance, LOANet[26] integrates an encoder–decoder structure with object-level attention to improve both segmentation accuracy and computational efficiency for building and road extraction. Furthermore, Transformer-based models have been actively explored, with SegFormer[27] demonstrating strong performance and generalization capability through a lightweight and scalable design.

Despite these advances, existing studies have predominantly focused on improving segmentation accuracy through architectural innovations, while fewer works have explicitly addressed real-time deployment constraints. In particular, a systematic and controlled comparison of the trade-off between segmentation accuracy and inference speed under consistent aerial imaging conditions and varying levels of model lightweighting remains limited. This limitation is critical for AAM/UAM applications, where both real-time performance and robustness are required for safe operation. To address this gap, we adopt representative and practically deployable models, YOLO11-seg and DeepLabV3+, and systematically evaluate multiple lightweight configurations, providing a comprehensive analysis of the trade-off between segmentation accuracy and real-time performance.

2.2 Training data composition for aerial image semantic segmentation

The performance of deep-learning-based semantic segmentation is strongly affected not only by model architecture but also by the composition and characteristics of training data.[28] In aerial imagery, object appearance and spatial distribution vary significantly depending on factors such as flight altitude, image resolution, and regional characteristics, which directly affect the segmentation performance of buildings and roads.[29–32] To support standardized evaluation, several large-scale aerial and UAV image datasets have been developed, including iSAID,[33] UAVid,[34] ISPRS Potsdam/Vaihingen,[35] and UrbanBIS.[36] These datasets provide high-resolution imagery with pixel-level annotations for multiple object classes and have become widely adopted benchmarks in aerial image semantic segmentation research. Studies based on these datasets have identified class imbalance, object scale variation, and complex background patterns as major factors that degrade segmentation performance. To mitigate these challenges, previous research has primarily focused on data augmentation techniques, multi-dataset training, and loss function design. For example, combining datasets captured under different conditions or altitudes has been shown to improve model generalization,[37,38] while sampling strategies and class-balanced loss functions have been proposed to address class imbalance issues.

However, these approaches largely focus on class-level imbalance and global dataset diversity, while the impact of intra-image object distribution has been relatively overlooked. In particular, the variation in object coverage—defined as the proportion of image area occupied by specific object classes—has not been systematically analyzed in relation to segmentation performance. In aerial imagery, variations in coverage can significantly alter visual context, spatial relationships, and feature distributions, thereby affecting model learning behavior. Despite its importance, a principled understanding of how object coverage affects segmentation performance remains largely unexplored, particularly in the context of real-time UAV deployment.

To this end, in this study, we propose a coverage-aware training data construction strategy by explicitly defining object coverage as a key factor, constructing coverage-stratified datasets, and conducting a comprehensive quantitative evaluation of its impact on segmentation performance. This approach enables a deeper understanding of the relationship between data composition and model performance, and provides practical guidelines for efficient dataset design in AAM/UAM applications.

3. Construction of Training Datasets

Here we present a coverage-aware, data-centric training dataset construction strategy designed to improve both the accuracy and robustness of aerial image semantic segmentation under the computational constraints of AAM/UAM drones. To this end, publicly available aerial imagery datasets containing building and road objects are first analyzed to assess their suitability for urban UAV environments. Subsequently, to analyze the impact of the coverage of buildings and roads in aerial imagery on semantic segmentation performance, we propose various training dataset configuration methods based on object coverage criteria.

3.1 Analysis of real-world aerial training data

To develop a reliable training dataset for aerial image-based semantic segmentation, publicly available aerial and satellite datasets containing building and road classes were analyzed. The selection criteria included spatial resolution, labeling accuracy, the diversity of object distribution, and similarity to real-world urban UAV operating environments. Datasets from AI-hub[39] and Kaggle were primarily examined. As shown in Fig. 1, the Massachusetts building dataset[40] and the Inria aerial image dataset[41] provide high-resolution imagery with accurate building annotations. However, these datasets do not include road classes and are based on satellite imagery, resulting in notable differences in viewpoint and spatial characteristics compared with UAV-based aerial imagery. In contrast, the AI-hub land cover dataset includes both building and road classes, provides pixel-level high-precision annotations, and consists of high-resolution aerial imagery closely aligned with real-world urban UAV flight conditions. Because of these characteristics, the AI-hub dataset provides a more realistic representation of urban UAV operating conditions than existing satellite-based datasets, making it particularly suitable for evaluating real-time aerial segmentation in AAM/UAM scenarios.

The selected dataset consists of aerial images with a spatial resolution of 25 cm and a fixed size of 512 × 512 pixels, along with corresponding pixel-wise annotation masks. To construct the training and evaluation datasets suitable for this study, we applied a preprocessing pipeline consisting of the filtering, cleansing, and conversion stages. First, in the filtering stage, only images containing building or road classes were selected from the original dataset, which includes 11 classes (buildings, parking lots, roads, street trees, rice paddies, greenhouses, fields, broadleaf forests, coniferous forests, bare land, and water). Second, in the cleansing stage, all classes except buildings and roads were merged into a single background class to simplify the segmentation problem and focus on target objects. Third, in the conversion stage, pixel-wise annotation masks were converted into text formats compatible with various segmentation models. This process utilized the findContours function from the OpenCV library[42] to extract object boundaries. To verify annotation integrity, the generated polygon annotations were reconstructed into masks using the fillPoly function and compared with the original pixel-wise masks. The resulting intersection over union (IoU) values were 0.99999 for buildings and 1.00000 for roads, with no lost pixels and less than 0.0006% additional pixels, confirming that

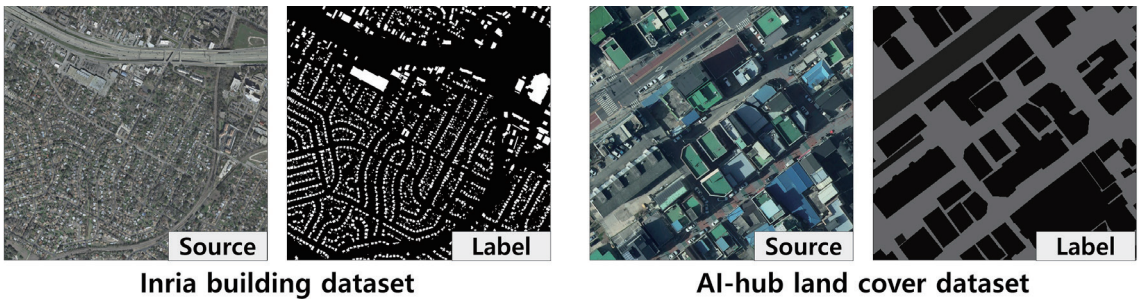

Fig. 1. (Color online) Examples of AI-hub land cover and Inria building datasets.

the conversion process introduced negligible information loss. As a result, a total of 70038 candidate images were constructed for training, validation, and evaluation from the original 90260 samples after preprocessing.

3.2 Coverage-based training dataset configuration

The spatial coverage of building and road objects in aerial imagery varies significantly depending on the imaging environment and urban density. Such variations are expected to directly affect the learning behavior and performance of semantic segmentation models. Therefore, we introduce a coverage-based dataset configuration strategy to systematically analyze these effects. The object coverage within an image is defined as the proportion of pixels occupied by building and road classes:

$$Coverage\ (\%) = \left(\frac{N_{building} + N_{road}}{N_{total}} \right), \tag{1}$$

where $N_{building}$ and N_{road} denote the numbers of pixels belonging to building and road classes, respectively, and N_{total} represents the total number of pixels in the image. This definition captures the relative dominance of target objects within an image and serves as a proxy for scene complexity and contextual richness, both of which are expected to affect model training and generalization. To investigate the impact of object coverage on segmentation performance, two types of datasets, T_1 and T_2, were constructed.

3.2.1 Dataset T_1: Coverage interval-based sampling

Dataset T_1 was designed to evaluate how different coverage intervals affect semantic segmentation performance. The entire dataset was partitioned into five coverage intervals: 0–10%, 10–20%, 20–30%, 30–40%, and ≥40%. From each interval, 5000 images were randomly sampled to construct five training datasets (T_{11}–T_{15}), ensuring balanced comparisons across coverage ranges. The upper threshold was set at 40% to ensure sufficient sample availability. Specifically, the dataset contained 11869 images (13.1%) with coverage ≥40%, while only 5369 images (5.9%) exceeded 50%, making higher thresholds less suitable for stable dataset construction. As shown in Fig. 2, images in the 30–40% and ≥40% intervals effectively represent dense urban environments typically encountered in UAV operations.

For validation, a dataset (V_1) was constructed by randomly sampling 250 images from each of the first four intervals (0–40%), ensuring balanced representation across coverage ranges. For evaluation, five datasets (E_{11}–E_{15}) were constructed by randomly sampling 1000 images from each coverage interval. Additionally, a mixed evaluation dataset (E_{16}) was created by combining 250 samples from each of the first four intervals to assess generalization performance.

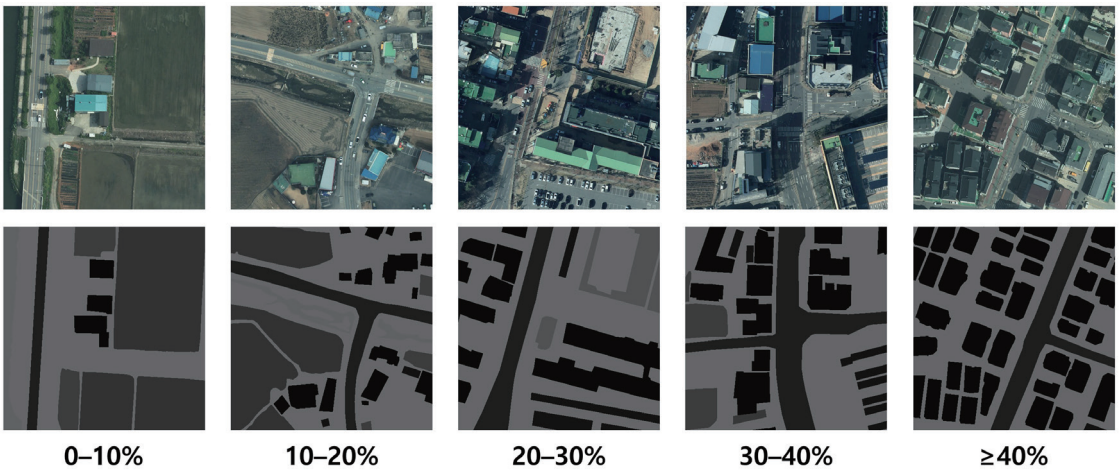

Fig. 2. (Color online) Training data examples across intervals.

3.2.2 Dataset T_2: Progressive coverage filtering

Dataset T_2 was designed to evaluate the effect of progressively removing low-coverage samples from the training data. To this end, coverage thresholds were increased from 0 to 40% in 10% increments, and images below each threshold were removed to construct five datasets (T_{21}–T_{25}). This design enables the systematic analysis of how excluding low-coverage samples affects segmentation performance and training efficiency. For dataset T_2, validation dataset V_2 was constructed by randomly sampling 3000 images without considering coverage intervals. Similarly, evaluation dataset E_{21} was constructed using 3000 randomly selected images. To further evaluate performance in dense urban environments, an additional evaluation dataset (E_{22}) was constructed by randomly sampling 3000 images with coverage ≥40%. The overall configuration of the training, validation, and evaluation datasets is illustrated in Fig. 3, forming the basis for the comprehensive experimental analysis presented in Sect. 5.

4. Deep Learning Model Configuration

AAM/UAM drones operate under strict constraints in computational resources and power consumption. Therefore, semantic segmentation models must achieve a balance between high segmentation accuracy and inference speed. To complement the proposed coverage-aware data construction strategy, in this section, we present model configuration approaches designed to systematically analyze the trade-off between segmentation accuracy and inference speed under varying coverage conditions by employing two structurally distinct model families.

4.1 Characteristics of semantic segmentation models for AAM/UAM environments

YOLO11-seg and DeepLabV3+ are widely adopted models for building and road segmentation in aerial imagery; however, they differ fundamentally in architectural design and information processing mechanisms.

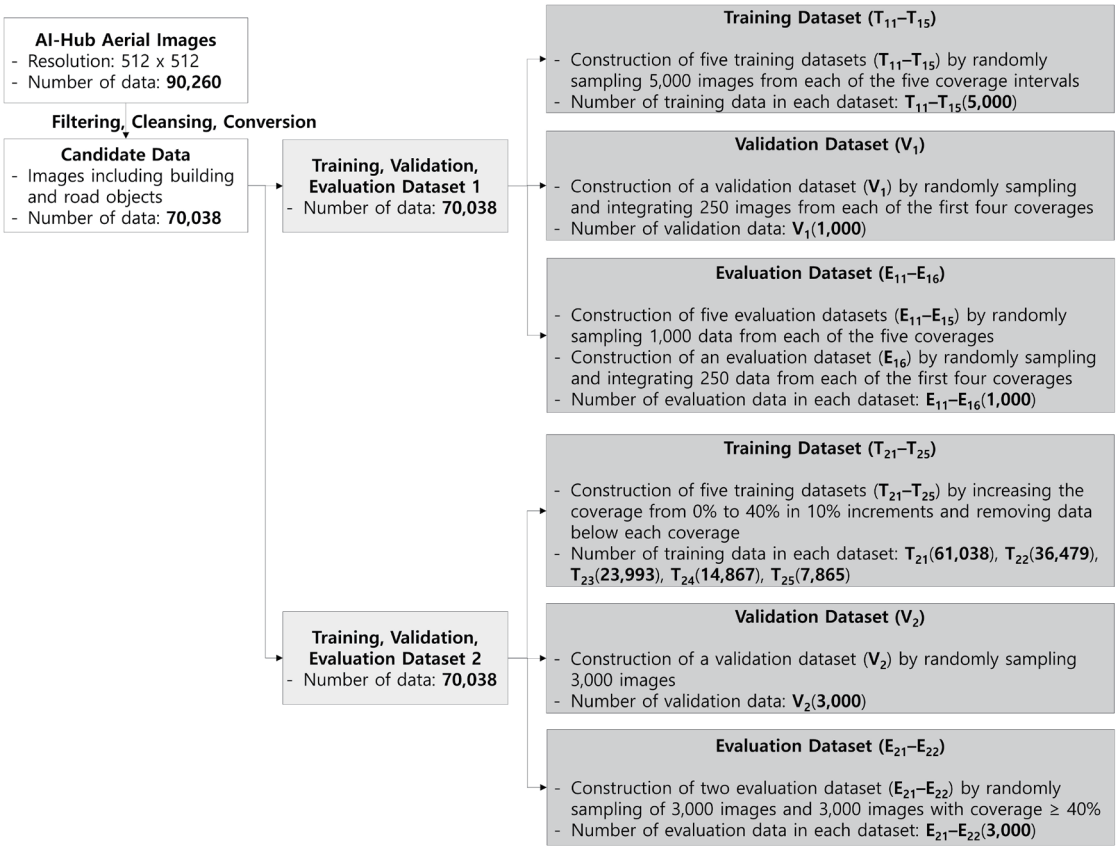

Fig. 3. Overall dataset configuration used in this study.

YOLO11-seg is based on a single-stage architecture that performs feature extraction, object detection, and mask prediction within a single network. This end-to-end design eliminates the need for region proposal generation, resulting in low latency and high computational efficiency. Such characteristics make YOLO11-seg well suited for real-time applications in aerial environments where continuous frame processing is required.

In contrast, DeepLabV3+ is a semantic segmentation–specific model based on an encoder–decoder architecture. The encoder extracts multiscale contextual information with various receptive fields using the backbone network and the ASPP module. The decoder then restores spatial resolution by combining these features with low-level features to precisely reconstruct object boundaries. Although this architecture requires higher computational cost, it provides robust performance in complex urban environments where objects exhibit intricate spatial structures. These architectural differences are particularly relevant when considering object coverage, as variations in coverage alter scene complexity and spatial context. Consequently, the two model families are expected to exhibit complementary performance characteristics across different coverage regimes. Accordingly, YOLO11-seg is adopted as an efficiency-oriented baseline model, whereas DeepLabV3+ is used as an accuracy-oriented reference model, enabling systematic comparison across varying coverage conditions.

4.2 Model lightweighting and backbone configuration strategies

To achieve an optimal balance between segmentation accuracy and inference speed, lightweight configurations of YOLO11-seg and DeepLabV3+ were constructed by adjusting model scale and backbone networks.

For YOLO11-seg, two model variants—small (YOLO11s-seg) and medium (YOLO11m-seg)—were selected to represent different levels of computational complexity, enabling the controlled evaluation of accuracy–speed trade-offs under varying resource constraints. The small model has relatively low parameter count and computational cost, making it suitable for deployment in resource-constrained environments such as onboard drone systems. It is primarily evaluated under low- to medium-coverage conditions where real-time processing is critical. The medium model, on the other hand, increases network depth and representational capacity, enabling the improved modeling of complex object structures and evaluation under higher-coverage conditions.

For DeepLabV3+, ResNet-101 was adopted as a baseline backbone owing to its strong feature representation capability and widespread use in semantic segmentation. However, its high computational cost limits its applicability in real-time scenarios. To systematically investigate the impact of backbone efficiency, lightweight configurations were constructed by replacing the backbone with EfficientNet-B0 and EfficientNet-B1, enabling the controlled analysis of performance degradation relative to computational savings. These architectures employ compound scaling and mobile inverted bottleneck convolution (MBConv) to effectively reduce parameter count and computational load while maintaining competitive performance.

Table 1 shows the number of parameters for all model configurations. As shown, the YOLO11s-seg model contains less than 50% of the parameters of YOLO11m-seg, while the EfficientNet-based DeepLabV3+ models require only approximately 18–26% of the parameters of the ResNet-101 backbone. These differences in model complexity are further analyzed in Sect. 5 to quantitatively evaluate the trade-off between segmentation accuracy and inference speed.

5. Experimental Setup and Performance Analysis

The experimental setup designed to validate the effectiveness of the proposed coverage-based training data configuration strategy is presented in this section, along with a

Table 1
Parameter sizes of the semantic segmentation models used in this study.

Model	Model/backbone variants	Parameter size
YOLO11	Small	10.1M
	Medium	22.4M
DeepLabV3+	ResNet-101	60-65M
	EfficientNet-B0	11–13M
	EfficientNet-B1	15–17M

comprehensive analysis of semantic segmentation accuracy and inference speed. The experimental framework integrates the training and evaluation datasets introduced in Sect. 3 with the lightweight model configurations described in Sect. 4. On the basis of this framework, the relationship between object coverage, segmentation accuracy, and inference efficiency is quantitatively analyzed.

5.1 Experimental setup

Two experimental scenarios based on the training dataset configurations T_1 and T_2 were designed, as illustrated in Fig. 4.

In Experiment 1, we evaluate how different coverage intervals affect segmentation performance. It combines training datasets T_{11}–T_{15}, validation dataset V_1, evaluation datasets E_{11}–E_{16}, and both YOLO11-seg and DeepLabV3+ models. Each model trained on T_{11}–T_{15} is evaluated across all evaluation datasets to analyze generalization across coverage distributions.

In Experiment 2, we investigate the effect of progressively removing low-coverage samples from the training data, employing training datasets T_{21}–T_{25}, validation dataset V_2, and evaluation datasets E_{21}–E_{22} to examine performance degradation under reduced data diversity.

To ensure strict experimental fairness, all models were trained using identical hyperparameter settings (Table 2). The experiments were conducted on a single NVIDIA GeForce RTX 5070 Ti GPU with Python 3.12.9, PyTorch 2.8.0, CUDA 12.9, cuDNN 9.1.0, and OpenCV 4.12.0. All input images were resized to 512 × 512 pixels. The number of training epochs was set to 50 on the basis of preliminary experiment results confirming stable convergence without overfitting. AdamW was adopted as the optimizer with a weight decay of 10^{-4} to mitigate overfitting caused

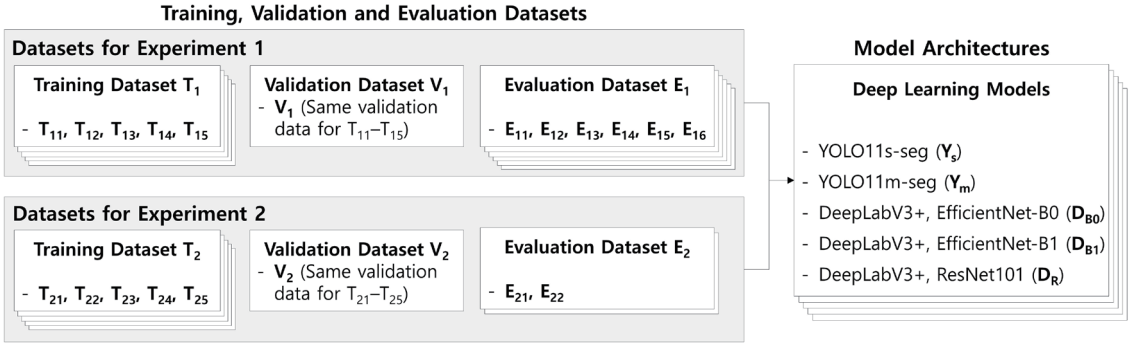

Fig. 4. Experimental configurations for Experiments 1 and 2, illustrating the combinations of coverage-based training datasets, validation datasets, evaluation datasets, and semantic segmentation models.

Table 2
Hyperparameter settings of the semantic segmentation models used in this study.

Item	Setting
Input resolution	512 × 512
Batch size	16
Epochs	50
Optimizer	AdamW
Learning rate	10^{-4}
Weight decay	10^{-4}

by coverage imbalance. The learning rate was fixed at 10^{-4} to ensure stable boundary learning. Segmentation performance was evaluated using precision, recall, and IoU for each class (building and road), reducing the bias caused by background dominance. Inference speed was measured in frames per second (FPS), averaged over 500 single-image inference runs.

5.2 Performance analysis

In this section, segmentation accuracy and inference speed under varying coverage conditions are analyzed. Quantitative and qualitative accuracy results are presented in Sect. 5.2.1, inference speed is analyzed in Sect. 5.2.2, and key findings are discussed in Sect. 5.2.3.

5.2.1 Semantic segmentation accuracy analysis

Figures 5 and 6 present the segmentation performance for buildings and roads in Experiment 1. When trained with T_{11} (low coverage), the YOLO11-seg models exhibit competitive performance for buildings but notably lower performance for roads. For example, the lightweight model Y_s achieves balanced performance for buildings, with mean precision, recall, and IoU of 0.7721, 0.7111, and 0.7890, respectively, and an F1-score of 0.7403, as shown in Fig. 5(a). However, for roads, the same model shows degraded performance, with a mean recall of 0.6224 and an IoU of 0.6370, as illustrated in Fig. 6(a). The Y_m model exhibits a similar trend. This indicates that YOLO-based models have difficulty capturing elongated and structurally complex road features under low-coverage conditions. In contrast, DeepLabV3+ models trained with T_{11} achieve significantly higher precision but relatively low recall and IoU, indicating a conservative prediction behavior in which false positives are reduced at the expense of increased false negatives.

As coverage increases from T_{12} to T_{14}, performance consistently improves across all models. Notably, T_{13} yields the most balanced results, where both YOLO11-seg and DeepLabV3+ achieve high and stable precision, recall, and IoU. For instance, when trained with T_{13}, the Y_s model achieves building segmentation performance of 0.7994 (precision), 0.7839 (recall), and 0.8417 (IoU), while the D_{B0} model achieves 0.8644, 0.8823, and 0.7733, respectively. For road segmentation, the Y_s model achieves 0.8032, 0.7550, and 0.7878, whereas the D_{B0} model achieves 0.8193, 0.8096, and 0.6794, respectively. These results indicate consistently strong and balanced performance across both classes. The data in Table 3 further confirm that T_{13} improves both segmentation accuracy and the balance between precision and recall compared with T_{11}. In particular, as shown in Figs. 5(c) and 6(c), performance on E_{16} demonstrates strong generalization capability, indicating that mid-coverage datasets enhance robustness under unseen conditions. These results demonstrate that, under limited data availability (e.g., 5000 samples), training datasets centered around the 20–40% coverage interval provide the most stable and balanced segmentation performance.

In Experiment 2, the impact of removing low-coverage data is analyzed (Figs. 7 and 8). The results show that segmentation performance remains stable from T_{21} to T_{23} despite a substantial reduction in dataset size (from 61038 to 23993 images). This finding provides strong evidence

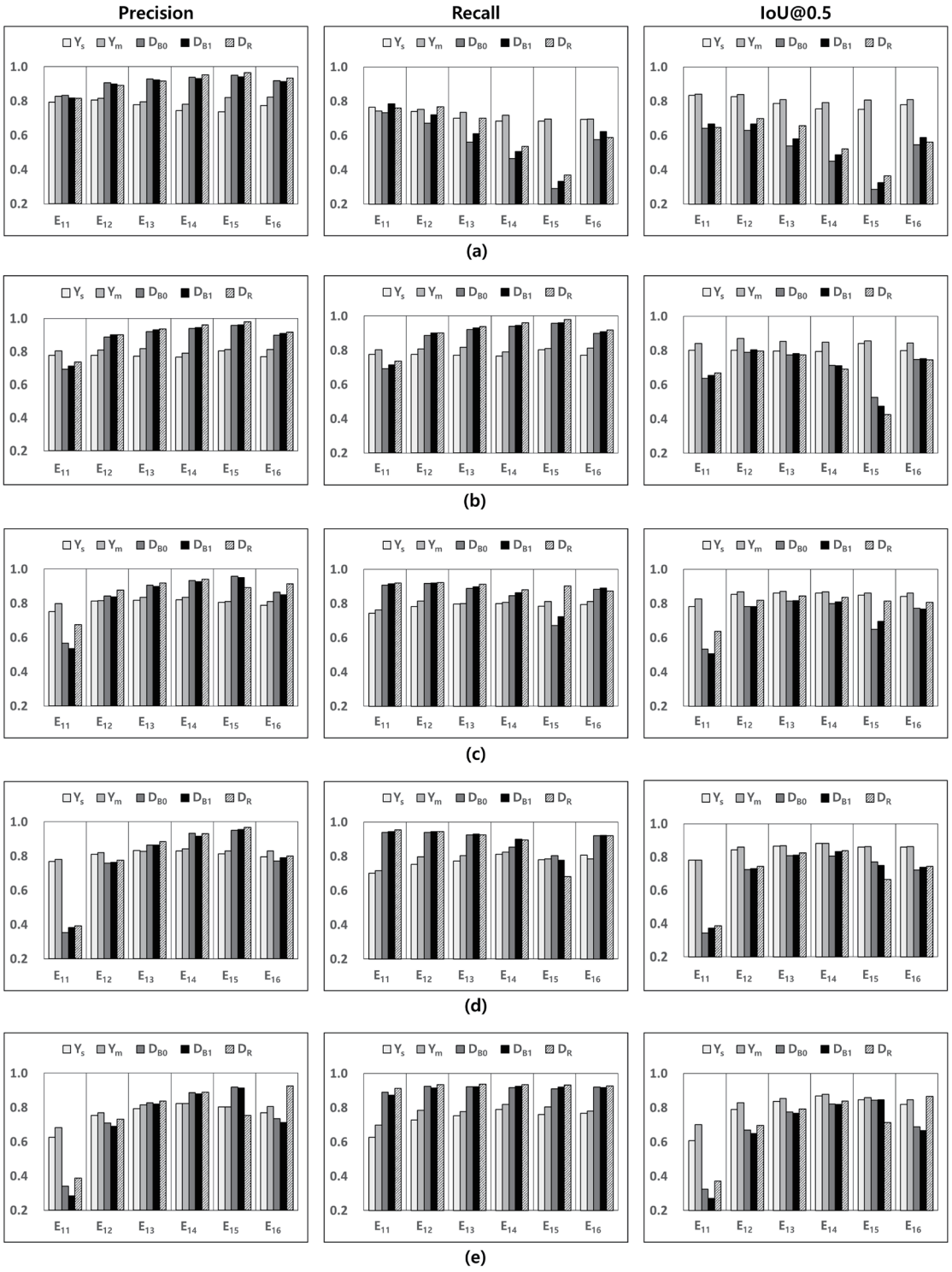

Fig. 5. Quantitative segmentation performance for the building class in Experiment 1 using training datasets T_{11}–T_{15} and evaluation datasets E_{11}–E_{16}: (a) T_{11}, (b) T_{12}, (c) T_{13}, (d) T_{14}, and (e) T_{15}.

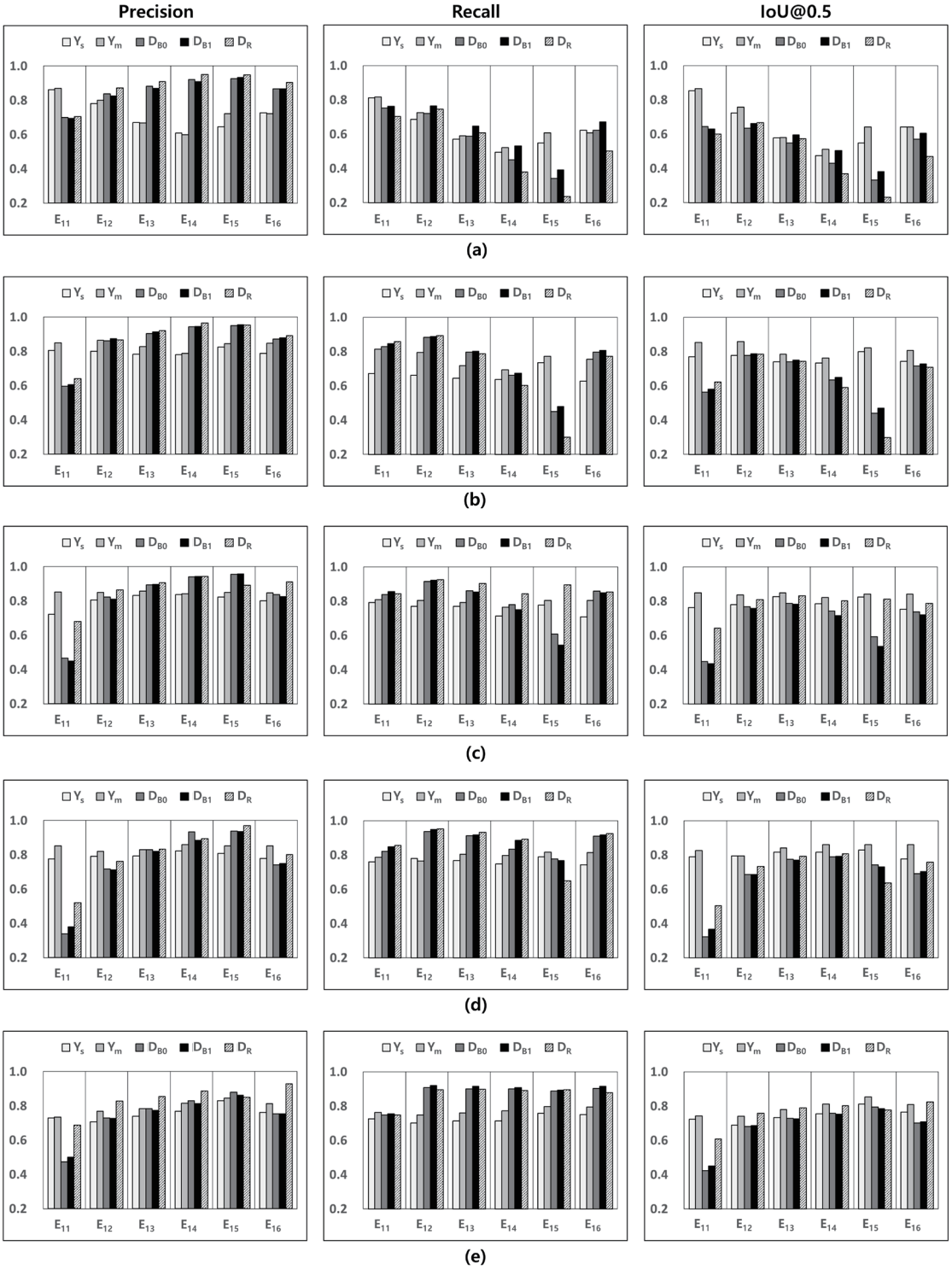

Fig. 6. Quantitative segmentation performance for the road class in Experiment 1 using training datasets T_{11}–T_{15} and evaluation datasets E_{11}–E_{16}: (a) T_{11}, (b) T_{12}, (c) T_{13}, (d) T_{14}, and (e) T_{15}.

Table 3
Mean precision, recall, F1-score, and IoU@0.5 of Y_s and D_{B0} when using (a) training data T_{11} and evaluation datasets E_{11}–E_{16}, and (b) training data T_{13} and evaluation datasets E_{11}–E_{16}.

	(a)				(b)			
	Y_s		D_{B0}		Y_s		D_{B0}	
	Building	Road	Building	Road	Building	Road	Building	Road
mPrecision	0.7721	0.7157	0.9125	0.8584	0.7994	0.8032	0.8451	0.8193
mRecall	0.7111	0.6224	0.5498	0.5792	0.7839	0.7550	0.8520	0.8096
mF1-score	0.7403	0.6658	0.6862	0.6917	0.7916	0.7784	0.8485	0.8184
mIou@0.5	0.7890	0.6370	0.5150	0.5267	0.8417	0.7878	0.7255	0.6794

that low-coverage samples contribute minimally to model learning and can be selectively pruned without performance loss. However, performance degradation begins to appear from T_{24}, where both dataset size and coverage diversity are significantly reduced, and becomes more pronounced in T_{25}. This confirms that excessive filtering limits the model's ability to capture spatial variability.

Qualitative results in Fig. 9 further support these observations. The input images I_1 and I_2 were selected to evaluate segmentation performance in building- and road-dominant scenarios, respectively. In Fig. 9(a), low-coverage datasets (T_{11}–T_{12}) exhibit high false negative rates for both building and road objects, particularly in the DeepLabV3+ models. Mid-coverage datasets (T_{13}–T_{14}) significantly reduce both false negatives and false positives, yielding the most visually consistent results, although minor false positives remain in complex road structures such as intersections. The high-coverage dataset (T_{15}) further reduces false negatives but introduces boundary-level errors, where adjacent objects are merged owing to over-segmentation. These errors are particularly pronounced for road objects with narrow boundaries. Figure 9(b) presents the results of Experiment 2. For datasets T_{21} to T_{23}, DeepLabV3+ models demonstrate consistently high segmentation quality with only minor false positives, indicating robust performance. In contrast, YOLO11-seg models exhibit higher false positive rates for both building and road objects. As the training dataset is further reduced (T_{24} and T_{25}), DeepLabV3+ models show a gradual increase in false negatives, although overall performance remains acceptable. Moreover, YOLO11-seg models continue to exhibit false positives, particularly for road objects in complex scenes such as I_2. In summary, the qualitative analysis indicates that training datasets centered around T_{13} and T_{14} provide the most balanced performance by minimizing both false negatives and false positives. Furthermore, when sufficient training data and wide coverage distributions are available (e.g., T_{21}–T_{23}), DeepLabV3+ models consistently outperform YOLO11-seg models in terms of segmentation robustness and boundary accuracy.

5.2.2 Inference speed analysis

In this section, the impact of training dataset configurations and model architectures on semantic segmentation inference speed is analyzed using FPS. In this study, FPS was computed as the average single-image inference speed measured over 500 test images.

Table 4 presents the FPS results obtained using the training datasets T_{11}–T_{15} and T_{21}–T_{25} across the YOLO11-seg and DeepLabV3+ models. The results indicate that the coverage-based

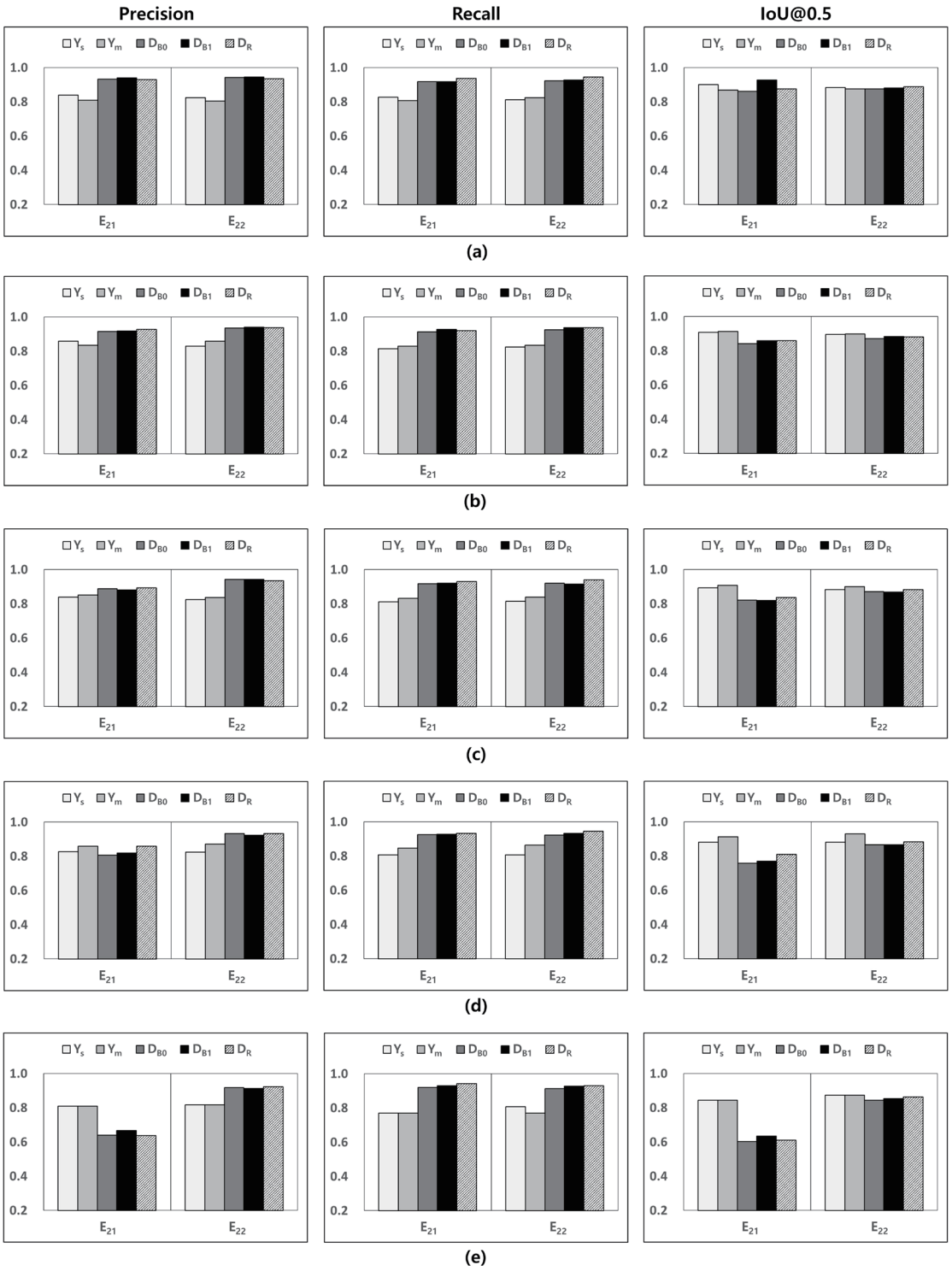

Fig. 7. Quantitative segmentation performance for the building class in Experiment 2 using training datasets T_{21}–T_{25} and evaluation datasets E_{21}–E_{22}: (a) T_{21}, (b) T_{22}, (c) T_{23}, (d) T_{24}, and (e) T_{25}.

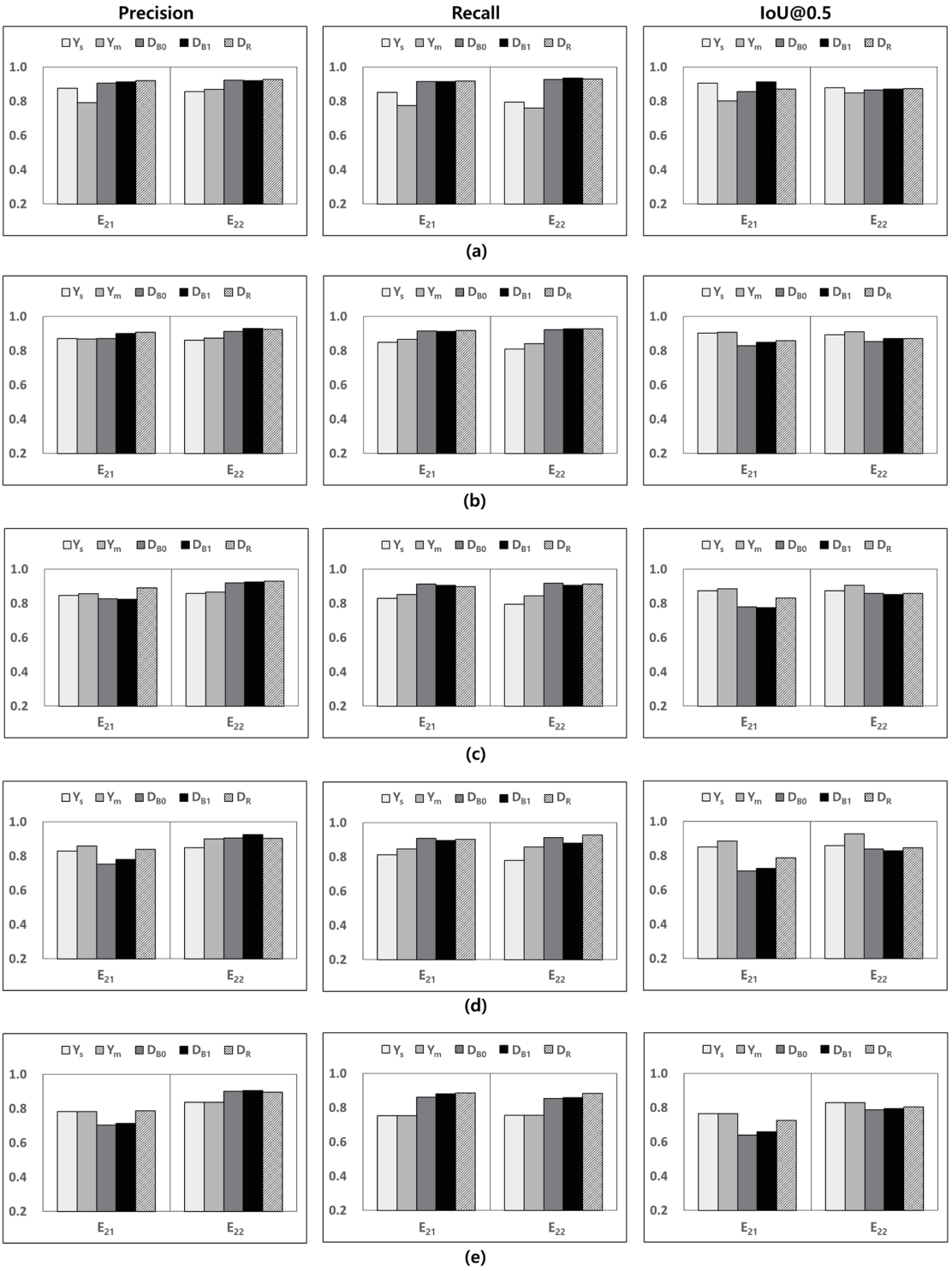

Fig. 8. Quantitative segmentation performance for the road class in Experiment 2 using training datasets T_{21}–T_{25} and evaluation datasets E_{21}–E_{22}: (a) T_{21}, (b) T_{22}, (c) T_{23}, (d) T_{24}, and (e) T_{25}.

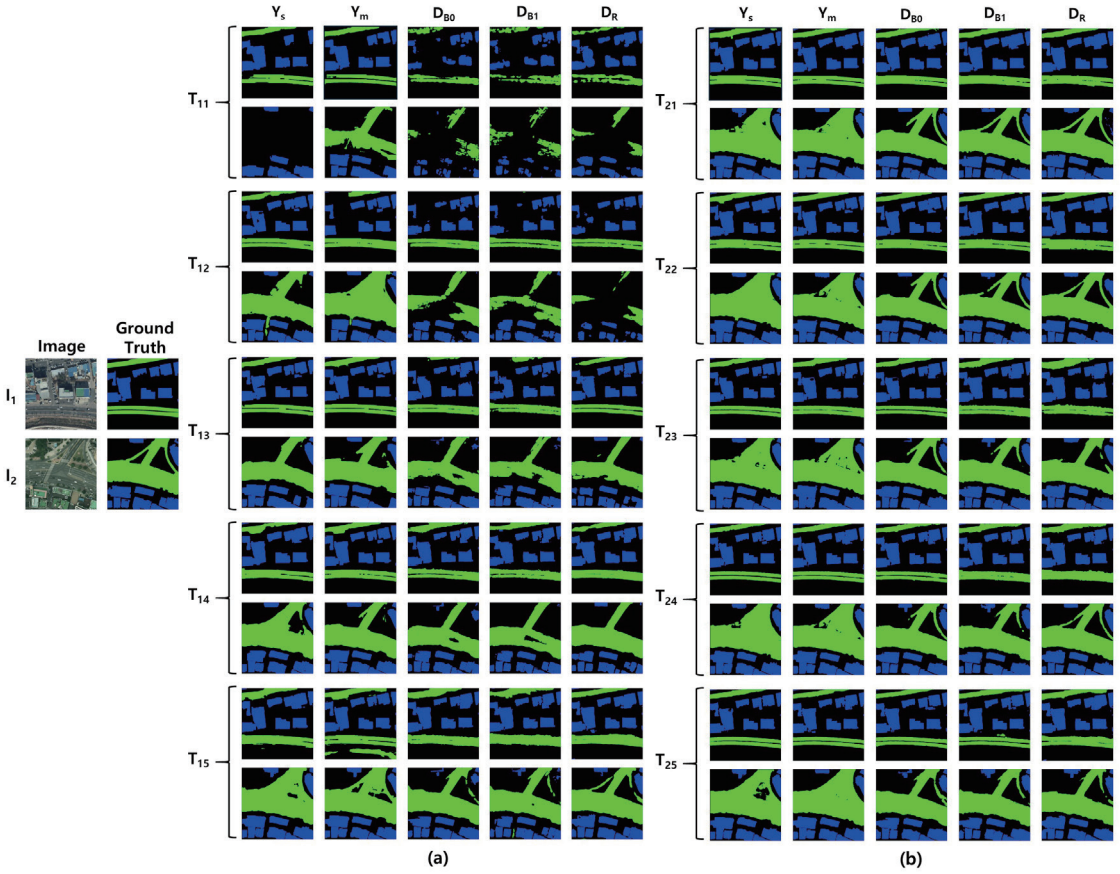

Fig. 9. (Color online) Qualitative segmentation results for representative test images. (a) Experiment 1 (T_{11}–T_{15}): performance variations across coverage intervals and (b) Experiment 2 (T_{21}–T_{25}): effect of progressively removing low-coverage training samples.

Table 4
FPS results for each model trained using T_{11}–T_{15} and T_{21}–T_{25}.

Model	Metric	Training dataset									
		T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{21}	T_{22}	T_{23}	T_{24}	T_{25}
Y_s	FPS	87.22	86.28	87.04	86.93	87.43	86.58	87.63	86.52	87.47	86.83
	mFPS	86.99									
Y_m	FPS	78.50	77.11	74.96	75.40	78.34	71.16	71.95	72.50	71.20	71.05
	mFPS	73.02									
D_{B0}	FPS	63.76	63.99	64.51	64.23	67.37	67.87	70.08	71.03	70.06	70.42
	mFPS	67.33									
D_{B1}	FPS	58.28	57.80	58.68	58.38	58.59	58.12	57.93	58.13	56.91	57.42
	mFPS	58.02									
D_R	FPS	61.56	61.60	62.06	62.17	62.06	60.37	60.87	61.17	60.11	61.04
	mFPS	61.30									

training data configuration has a negligible impact on inference speed. For a fixed model, FPS remains consistent across different dataset configurations. For example, across T_{11}–T_{15} and T_{21}–T_{25}, the Y_s and Y_m models achieve 86.28–87.63 and 71.05–78.50 FPS, while the D_{B0} and D_{B1} models achieve 63.76–71.03 and 56.91–58.68 FPS, respectively. These results confirm that variations in training data composition do not significantly affect inference efficiency. In contrast, a notable difference of 10–20 FPS is observed between the different model architectures. Considering the mean FPS (mFPS), the Y_s model achieves the highest inference speed (86.99 FPS), followed by the Y_m model (73.02 FPS). DeepLabV3+ models exhibit lower inference speeds, with D_{B0}, D_{B1}, and D_R achieving 67.33, 58.02, and 61.30 FPS, respectively. This difference can be attributed to the higher computational complexity of the DeepLabV3+ architecture, which includes the encoder–decoder structure, the ASPP module, and high-resolution feature reconstruction.

These results clearly indicate that inference speed is predominantly governed by model architecture rather than training data composition. This decoupling enables the independent optimization of data and model design, which is critical for real-time UAV deployment. Furthermore, since all models achieve approximately ≥60 FPS, they are suitable for real-time applications even under resource-constrained environments.

5.2.3 Discussion

In this study, we provide a systematic and data-centric perspective on semantic segmentation, highlighting the critical role of training data composition—particularly object coverage—in determining model performance.

- The results demonstrate that increasing object coverage in training data does not necessarily lead to improved segmentation performance. In Experiment 1, the best performance was consistently observed in the intermediate coverage intervals (T_{13}–T_{14}), rather than in the highest coverage interval (T_{15}). In particular, the Y_s and D_{B0} models achieved balanced and high performance across precision, recall, and IoU in these intervals. This finding suggests that an appropriate level of background information is essential for training robust contextual relationships between objects and their surroundings. Excessively high coverage may reduce contextual diversity, while excessively low coverage may limit object representation.
- The results of Experiment 2 indicate that training data efficiency can be significantly improved through selective filtering. Although the total number of training samples remains an important factor, removing low-coverage samples with limited training contribution does not substantially degrade performance up to a certain threshold. The stable performance observed from T_{21} to T_{23} suggests that many low-coverage samples contribute relatively little to the learning of building and road representations. This finding highlights the potential for cost-effective dataset construction, which is particularly valuable for AAM/UAM applications where large-scale data collection and annotation are resource-intensive.
- The inference speed analysis reveals that coverage-based training data composition has negligible effect on runtime performance. Instead, inference speed is primarily determined by model architecture and parameter complexity. In particular, the YOLO11-seg models

achieve higher FPS owing to their single-stage design, whereas DeepLabV3+ models incur additional computational overhead from encoder–decoder processing and multiscale feature aggregation. Nevertheless, all evaluated models maintain real-time performance (approximately ≥60 FPS), demonstrating their practical applicability in UAV-based systems.

- An interesting observation is that the lightweight Y_s achieved a higher road IoU than the lightweight D_{B0} in certain coverage intervals, despite the common understanding that DeepLab-based architectures are particularly effective for elongated structures such as roads. This behavior is likely attributable to the lightweight backbone configuration adopted in D_{B0}. Specifically, replacing the original ResNet-101 backbone with EfficientNet-B0 substantially reduces computational complexity but may also limit the feature representation capacity for preserving fine-grained spatial details of narrow and elongated road structures. Since the effectiveness of the ASPP module depends on the quality of backbone features, aggressive backbone compression may partially reduce the advantages of DeepLabV3+ for road segmentation. In contrast, the prediction characteristics of the single-stage YOLO11-seg architecture appear to preserve road continuity effectively under intermediate coverage conditions. Therefore, the observed result should not be interpreted as evidence that YOLO11-seg generally outperforms DeepLabV3+ for road segmentation; rather, it highlights the trade-off between computational efficiency and feature representation capacity introduced by model lightweighting.

Overall, these findings suggest that optimal system design for aerial image semantic segmentation should consider both data composition and model efficiency. Specifically, training datasets centered around the 20–40% coverage interval, combined with lightweight models such as Y_s and D_{B0}, provide the most effective balance between segmentation accuracy and real-time performance. More broadly, the results demonstrate that data-centric optimization can be as important as architectural optimization when designing semantic segmentation systems for resource-constrained AAM/UAM environments.

6. Conclusions

We addressed the challenge of achieving both high accuracy and real-time performance for aerial image semantic segmentation in AAM/UAM drone environments with limited computing and sensor resources. To this end, we proposed a unified framework that jointly optimizes training data composition and model efficiency through coverage-aware dataset construction and lightweight model design. Specifically, we introduced a coverage-based training data construction strategy in which datasets are systematically organized in accordance with the spatial proportion of building and road pixels. In parallel, we developed multiple lightweight configurations of YOLO11-seg and DeepLabV3+ by varying backbone networks and model scales, enabling a comprehensive evaluation of accuracy–efficiency trade-offs under diverse conditions. Extensive experiments yielded three key findings. First, segmentation performance does not monotonically increase with object coverage; instead, mid-coverage datasets (20–40%) consistently provide the most stable and balanced performance by preserving both object saliency and contextual diversity. Second, low-coverage samples contribute minimally to model

learning and can be selectively removed without significant performance degradation up to a certain threshold, thereby substantially improving data efficiency. Third, inference speed is predominantly governed by model architecture rather than training data composition, and all evaluated lightweight models achieve real-time performance (≥60 FPS), confirming their suitability for onboard deployment. On the basis of these findings, we provide practical guidelines for designing aerial image semantic segmentation systems in drone environments. In particular, combining mid-coverage training data (20–40%) with lightweight models such as Y_s and D_{B0} yields an effective balance between segmentation accuracy and real-time inference.

Despite these contributions, several limitations remain. First, the experiments were conducted using the AI-hub dataset, and inference speed was evaluated in a desktop GPU environment rather than using actual onboard UAV hardware. Second, object coverage was defined using the combined area of buildings and roads, which does not explicitly account for class-specific geometric characteristics or differences between area-based and linear structures.

Future research will therefore focus on validating the proposed framework on real-world UAV platforms using onboard computing systems and live aerial video streams. In addition, comparative evaluations involving other efficient architectures, such as MobileNetV2,[43] MSTDeepLabV3+,[44] DDRNet,[45], and SegFormer, will be conducted to further refine model selection and deployment strategies. Another important direction is the systematic investigation of data-centric learning strategies for aerial image semantic segmentation. Specifically, direct ablation studies for comparing the proposed coverage-aware dataset construction approach with conventional random-sampling baselines under equivalent training data sizes will be performed to further quantify the effectiveness of coverage-based data selection. Furthermore, class-specific coverage distributions, building-to-road coverage ratios, and scene structural complexity metrics will be investigated in future studies to better characterize the geometric properties of aerial imagery. Finally, we will explore the relationship between coverage-aware dataset construction and optimization-based imbalance-handling approaches, such as online hard example mining[46] and focal loss,[47] including integrated frameworks that combine data-centric and loss-function-based learning strategies. These investigations are expected to provide deeper insights into the joint optimization of training data composition, learning strategy, and model architecture for real-time aerial image semantic segmentation in AAM/UAM environments.

Acknowledgments

This research was supported by the Daejeon University Research Grants (2025).

References

1 Schleiss: Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. **XLII-2/W13** (2019) 575. https://doi.org/10.5194/isprs-archives-XLII-2-W13-575-2019
2 A. Nassar, K. Amer, R. ElHakim, and M. ElHelw: Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW, 2018) 1626–1636. https://doi.org/10.1109/CVPRW.2018.00201
3 A. P. Tanchenko, A. M. Fedulin, R. R. Bikmaev, and R. N. Sadekov: Gyroscopy Navig. **11** (2020) 293. https://doi.org/10.1134/S2075108720040100

4 K. Hong, S. Kim, J. Park, and H. Bang: J. Aerosp. Inf. Syst. **18** (2021) 964. https://doi.org/10.2514/1.I010957
5 Z. Lu, F. Liu, and X. Lin: arXiv:2211.11988 (2022). https://doi.org/10.48550/arXiv.2211.11988
6 I. Jarraya, A. Al-Batati, M. Kadri, M. Abdelkader, A. Ammar, W. Boulila, and A. Koubaa: Satell. Navig. **6** (2025) 1. https://doi.org/10.1186/s43020-025-00162-z
7 V. Mnih and G. E. Hinton: Lect. Notes Comput. Sci. **6316** (2010) 210. https://doi.org/10.1007/978-3-642-15567-3_16
8 E. Shelhamer, J. Long, and T. Darrell: IEEE Trans. Pattern Anal. Mach. Intell. **39** (2017) 640. https://doi.org/10.1109/TPAMI.2016.2572683
9 O. Ronneberger, P. Fischer, and T. Brox: Lect. Notes Comput. Sci. **9351** (2015) 234. https://doi.org/10.1007/978-3-319-24574-4_28
10 X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfe: IEEE Geosci. Remote Sens. Mag. **5** (2017) 8. https://doi.org/10.1109/MGRS.2017.2762307
11 E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez: IEEE Trans. Geosci. Remote Sens. **55** (2017) 7092. https://doi.org/10.1109/TGRS.2017.2740362
12 S. Azimi, C. Henry, L. Sommer, A. Schumann, and E. Vig: Proc. 2019 IEEE/CVF Conf. Computer Vision (ICCV, 2019) 7392–7402. https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00749
13 A. Tran, A. Zonoozi, J. Varadarajan, and H. Kruppa: Proc. Workshop on Structuring and Understanding of Multimedia heritAge Contents (SUMAC, 2020) 57–64. https://doi.org/10.1145/3423323.3423407
14 D. G. Lee, J. H. You, S. G. Park, S. H. Baeck, and H. J. Lee: Sens. Mater. **31** (2019) 3335. https://doi.org/10.18494/SAM.2019.2472
15 Y. Li, Y. Wang, H. W. Tseng, H. Huang, and C. C. Chen: Sens. Mater. **33** (2021) 3325. https://doi.org/10.18494/SAM.2021.3402
16 A. Vatresia, F. Utama, N. Sugianto, A. Widyastiti, R.Rais, and R. Ismanto: Spatial Inf. Res. **32** (2024) 327. https://doi.org/10.1007/s41324-023-00560-y
17 A. Khatua, A. Bhattacharya, A. K. Goswami, and B. H. Aithal: Spatial Inf. Res. **32** (2024) 511. https://doi.org/10.1007/s41324-024-00574-0
18 O. Ronneberger, P. Fischer, and T. Brox: Lect. Notes Comput. Sci. **9351** (2015) 234. https://doi.org/10.1007/978-3-319-24574-4_28
19 V. Badrinarayanan, A. Kendall, and R. Cipolla: IEEE Trans. Pattern Anal. Mach. Intell. **39** (2017) 2481. https://doi.org/10.1109/TPAMI.2016.2644615
20 L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille: IEEE Trans. Pattern Anal. Mach. Intell. **40** (2018) 834. https://doi.org/10.1109/TPAMI.2017.2699184
21 L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam: Lect. Notes Comput. Sci. **11211** (2018) 833. https://doi.org/10.1007/978-3-030-01234-2_49
22 R. Khanam, and M. Hussain: arXiv:2410.17725 (2024). https://doi.org/10.48550/arXiv.2410.17725
23 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: Proc. 2016 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2016) 779–788. https://doi.org/10.1109/CVPR.2016.91
24 A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam: arXiv:1704.04861 (2017). https://doi.org/10.48550/arXiv.1704.04861
25 M. Tan, and Q. Le: Proc. 2019 Int. Conf. Machine Learning (PMLR, 2019) 6105–6114. https://arxiv.org/abs/1905.11946
26 X. Han, Y. Liu, G. Liu, Y. Lin, and Q. Liu: PeerJ Comput. Sci. **9** (2023) e1467. https://doi.org/10.7717/peerj-cs.1467
27 E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. Alvarez, and P. Luo: Proc. 2019 Int. Conf. Advances in Neural Information Processing Systems 34 (NeurIPS, 2021) 12077–12090. https://arxiv.org/abs/2105.15203
28 D. Zha, Z. P. Bhat, K. H. Lai, F. Yang, Z. Jiang, S. Zhong, and X. Hu: ACM Comput. Surv. **57** (2025) 1. https://doi.org/10.1145/3711118
29 L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson: ISPRS J. Photogramm. Remote Sens. **152** (2019) 166. https://doi.org/10.1016/j.isprsjprs.2019.04.015
30 W. Cai, K. Jin, J. Hou, C. Guo, L. Wu, and W. Yang: J. Visual Commun. Image Represent. **109** (2025) 104429. https://doi.org/10.1016/j.jvcir.2025.104429
31 G. Dinga: ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. **V-2-2022** (2022) 245. https://doi.org/10.5194/isprs-annals-V-2-2022-245-2022
32 B. Kiefer, D. Ott, and A. Zell: Proc. 2022 Int. Conf. Pattern Recognition. (ICPR, 2022) 3564–3571. https://doi.org/10.1109/ICPR56361.2022.9956710
33 S. W. Zamir, A. Arora, A. Gupta, S. Khan, G. Sun, F. S. Khan, F. Zhu, L. Shao, G. Xia, and X. Bai: Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2019) 28–37. https://doi.org/10.48550/arXiv.1905.12886

34 Y. Lyu, G. Vosselman, G. Xia, A. Yilmaz, and M. Yang: ISPRS J. Photogramm. Remote Sens. **165** (2020) 108. https://doi.org/10.1016/j.isprsjprs.2020.05.009
35 ISPRS 2D Semantic Labeling: https://www.isprs.org/resources/datasets/benchmarks/UrbanSemLab/semantic-labeling.aspx (accessed May 31 2026).
36 G. Yang, F. Xue, Q. Zhang, K. Xie, C. Fu, and H. Huang: Proc. ACM SIGGRAPH 2023 Conf. (SIGGRAPH, 2023) 1–11. https://doi.org/10.1145/3588432.3591508
37 A. Song: Aerospace **10** (2023) 880. https://doi.org/10.3390/aerospace10100880
38 J. Sherrash: arXiv:1606.02585 (2016). https://doi.org/10.48550/arXiv.1606.02585
39 AI-Hub Land Cover Maps from Aerial Satellite Images: https://aihub.or.kr/aihubdata/data/view.do?dataSetSn=71361 (accessed May 31 2026).
40 Kaggle Massachusetts Buildings Dataset: https://www.kaggle.com/datasets/balraj98/massachusetts-buildings-dataset (accessed May 31 2026).
41 Kaggle Inria Aerial Image Labeling Dataset: https://www.kaggle.com/datasets/huanranye/inria-aerial-image-labeling-dataset (accessed May 31 2026).
42 Open Computer Vision Library: https://opencv.org/ (accessed May 31 2026).
43 M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen: Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2018) 4510–4520. https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00474
44 Y. Wang, L. Yang, X. Liu, and P. Yan: Sci. Rep. **14** (2024) 9716. https://doi.org/10.1038/s41598-024-60375-1
45 H. Pan, Y. Hong, W. Sun, and Y. Jia: IEEE Trans. Intell. Transp. Syst. **24** (2023) 3448. https://doi.org/10.1109/TITS.2022.3228042
46 A. Shrivastava, A. Gupta and R. Girshick: Proc. 2016 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2016) 761–769. https://doi.org/10.1109/CVPR.2016.89
47 T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár: IEEE Trans. Pattern Anal. Mach. Intell. **42** (2020) 318. https://doi.org/10.1109/TPAMI.2018.2858826