

Building Segmentation Using Multiprompts and Fine-tuned Segment Anything Model 2

Hong-Deok Seo and Eui-Myoung Kim*

Department of Spatial Information Engineering, Namseoul University,
91, Daehak-ro, Seonghwan-eup, Seobuk-gu, Cheonan, Chungcheongnam-do 31020, Korea

(Received May 8, 2025; accepted September 4, 2025)

Keywords: building segmentation, multiprompts, fine-tuning, SAM2, YOLOv8

The construction of digital maps typically relies on manual stereoscopic plotting based on aerial imagery, which demands considerable time and cost. As a result, it is challenging to promptly reflect frequent building changes associated with urban development in digital maps. In this study, to automate the modification and updating of digital maps, we proposed an automated segment anything model (SAM) 2-based building segmentation approach that utilizes You Only Look Once (YOLO) v8 for building detection and image processing techniques to extract building boundaries from ortho-images. In the proposed methodology, we were able to automatically generate prompts for SAM2 by applying image processing techniques to the bounding boxes of buildings detected by YOLOv8, removing noise and creating clear masks. Furthermore, through performance comparison experiments between the pretrained and fine-tuned SAM2, we found that the fine-tuned SAM2 significantly improved building segmentation performance because of the additional training specialized for building data. In experiments on comparing a single prompt and multiprompt inputs, we observed that multiprompt inputs enabled a more precise and accurate building segmentation, confirming that prompts play a crucial role in enhancing model performance.

1. Introduction

In the domain of geospatial information science, there has been active research on applying artificial intelligence techniques to detect and segment objects in imagery, as well as to generate outcomes that are applicable to practical industrial contexts.^(1–3) The construction of digital maps involves processes such as terrain and feature description, geo-editing, and structured editing based on aerial imagery, all of which are labor-intensive and require considerable costs and time. Moreover, since these processes are predominantly carried out manually, the resulting quality is highly dependent on the operator's expertise, and rapid changes in terrain and features are difficult to incorporate in a timely manner.⁽⁴⁾ Therefore, it is necessary to develop a methodology of building digital maps in an automated way to solve these problems.

*Corresponding author: e-mail: kemyoung@nsu.ac.kr
<https://doi.org/10.18494/SAM5730>

For applying artificial intelligence technology to geospatial information, researchers have conducted studies on extracting buildings by applying deep learning to images acquired by unmanned aerial vehicles,⁽⁵⁾ building multitemporal high-resolution building object datasets by applying deep learning to Sentinel-2 images,⁽⁶⁾ evaluating the influence of the model used through accuracy comparison and parameter adjustment by applying U-Net and ResUNet to Sentinel-2 images,⁽⁷⁾ and extracting geospatial information such as buildings and roads by applying deep learning to satellite images.⁽⁸⁾

In previous studies of transfer learning on pretrained models, researchers have analyzed the accuracy in terms of the weight importance and transfer learning of deep learning models to automatically extract buildings,⁽⁹⁾ detected building changes through transfer learning by combining the local feature and global spatial pyramid modules for extracting various building features,⁽¹⁰⁾ fine-tuned the segment anything model (SAM) using multiprompts for mobility infrastructure segmentation such as roads, sidewalks, and crosswalks,⁽¹¹⁾ and improved the performance of SAM through multimodal fusion for detailed building identification.⁽¹²⁾

When applying pretrained deep learning to geospatial information, the conditions and environments of the model training and experimental data differ, requiring the fine-tuning of the model or parameter adjustment. To improve this, the accuracy can be effectively enhanced by adjusting the pretrained model using data similar to the experimental data and segmenting by inputting multiprompts for each object.

Accordingly, the purpose of this study was to fine-tune a pretrained object segmentation deep learning model using ortho-images and to generate multiprompts to segment buildings that are highly useful in digital maps.

2. Research Methodology

2.1 Methodology

As shown in Fig. 1, we fine-tuned SAM2, a deep learning model for object segmentation, using aerial ortho-images provided by the Korea National Geographic Information Institute (NGII) platform,⁽¹³⁾ and segmented building objects using the fine-tuned model and multiple prompts. To this end, deep learning training data were first constructed using the ortho-images and digital maps, followed by model training and fine-tuning.

Second, building bounding boxes were detected by applying You Only Look Once (YOLO) v8, an object detection model, to the ortho-image, and box prompts were generated for use with the fine-tuned SAM2. Third, each detected bounding box was cropped and converted into a hue, saturation, value (HSV) image, from which point prompts were generated through edge and contour detection, morphological operations, and other image processing techniques.

Finally, SAM2 was fine-tuned using training data constructed by editing ortho-images and digital maps, and building segmentation was performed using the previously generated prompts and the fine-tuned model.

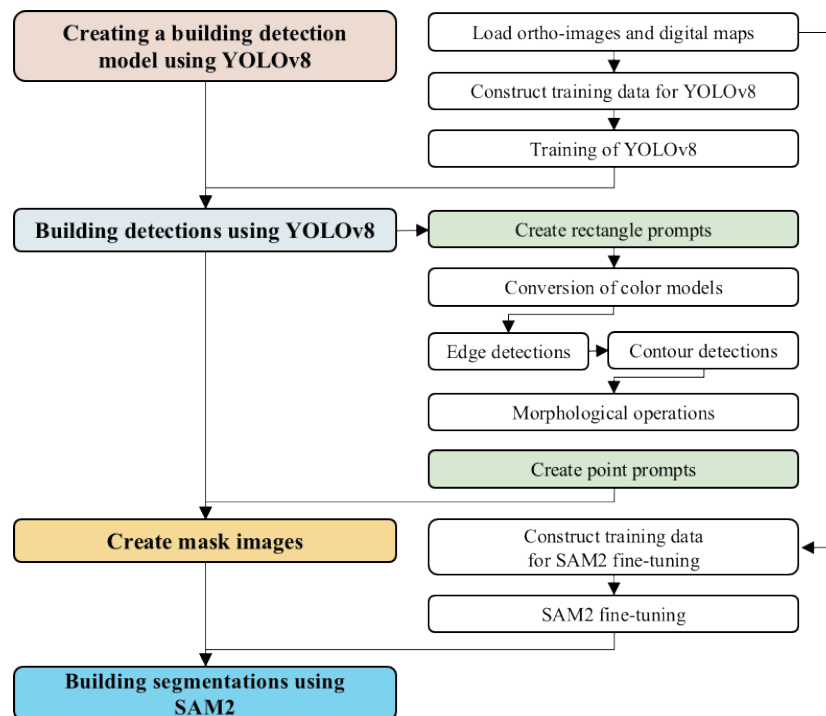


Fig. 1. (Color online) Research flow diagram.

2.2 YOLOv8

YOLO is a model that detects and classifies multiple objects by processing a single image only once. In this study, YOLOv8 was used to generate box prompts. YOLOv8 is an improved version of YOLOv5 and provides object detection, segmentation, and classification capabilities within a unified framework.⁽¹⁴⁾ It demonstrates enhanced performance compared with the previous v5x model by replacing certain modules and adjusting the backbone's kernel size. The model's speed has also been improved, making it effective for real-time object detection.⁽¹⁵⁾

2.3 Image processing

2.3.1 Edge and contour detections

Edge detection is a technique that identifies regions in an image where pixel brightness values change rapidly, recognizing them as edges to detect object boundaries or shapes. It is typically applied to binary images for accurate detection.⁽¹⁶⁾ In this study, the Canny edge detection algorithm, which has been widely used in the field of image processing, was employed to detect object boundaries.⁽¹⁷⁾

2.3.2 Morphological operations

Morphological operations are used to fill the inside of boundary images to convert linear information into polygonal information. In this study, erosion operations were performed to remove polygons with areas within 5% from the filled boundary image, and dilation operations were performed to emphasize polygon areas where building objects exist.^(15,18)

2.4 SAM2

SAM2 is a model that improves the performance of the existing SAM and is optimized for segmenting objects in images and videos by inputting prompts.⁽¹⁹⁾ As shown in the architecture in Fig. 2, instead of directly using frame embeddings like SAM, the decoder of SAM2 utilizes the accumulated memory from past prediction results and the prompt information provided by the user to generate the final mask.⁽²⁰⁾ One of the main features of SAM2 is that it can improve the prediction accuracy by utilizing the information of the prompt frame located in the future rather than the current frame time. In addition, the processing and prediction results of each frame are stored in the memory bank through the memory generation process and then used for the prediction of the subsequent frames. Afterwards, the memory attention mechanism receives the embedding of the current frame extracted from the video encoder and the information and prompt conditions of the past frames stored in the memory bank to output a new embedding, and the output embedding is passed to the mask decoder to obtain the final object segmentation mask.^(19,20)

3. Experiments

3.1 Data preprocessing

The target area of this study was the area around the Sports Complex located in Seobuk-gu, Cheonan-si, Chungcheongnam-do, Korea, and the data necessary for this study, as shown in Table 1, were obtained using ortho-images and digital maps provided by the NGII platform. The ortho-images comprised data created by processing 1:5000 aerial images taken in 2023, and the digital maps had the same spatial resolution as the ortho-images and were updated in 2024.

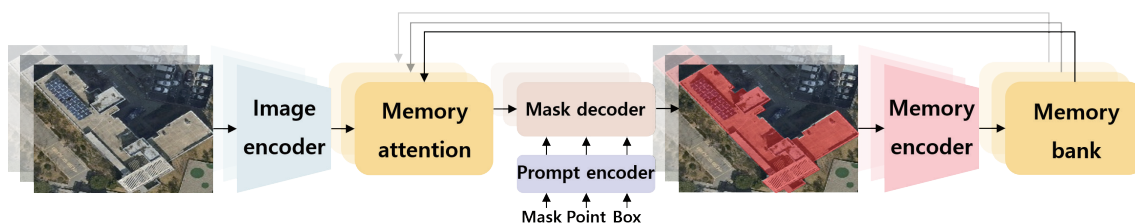


Fig. 2. (Color online) SAM2 architecture.

Table 1
Information of ortho-image and digital map.

Number of maps	Scale	Address
36701064	1:5000	Eumbong-myeon, Asan-si, Chungcheongnam-do
36701065		Chaam-dong, Seobuk-gu, Cheonan-si, Chungcheongnam-do
36701066		Dujeong-dong, Seobuk-gu, Cheonan-si, Chungcheongnam-do
36701067		Budae-dong, Seobuk-gu, Cheonan-si, Chungcheongnam-do
36701074		Tangjeong-myeon, Asan-si, Chungcheongnam-do
36701075		Buldang-dong, Seobuk-gu, Cheonan-si, Chungcheongnam-do
36701076		Seongjeong-dong, Seobuk-gu, Cheonan-si, Chungcheongnam-do
36701077		Wonseong-dong, Dongnam-gu, Cheonan-si, Chungcheongnam-do
36701084		Tangjeong-myeon, Asan-si, Chungcheongnam-do
36701085		Baebang-eup, Asan-si, Chungcheongnam-do
36701086		Yonggok-dong, Dongnam-gu, Cheonan-si, Chungcheongnam-do
36701087		Samnyong-dong, Dongnam-gu, Cheonan-si, Chungcheongnam-do

3.1.1 Editing of ortho-images

The ortho-images provided by the NGII platform comprised data without coordinate system settings. In this study, the coordinate system was set using the range of digital maps with similar time and spatial resolutions. Additionally, to maximize the performance and efficiency of the deep learning model, the image size was tiled to 1024×1024 . Parts of the tiled images are shown in Fig. 3. Figure 3(a) represents an area where mainly nonresidential buildings exist, and Fig. 3(b) represents an area where nonresidential buildings and houses are densely distributed. Additionally, Fig. 3(c) represents an area where nonresidential buildings, houses, and apartments are diversely distributed, and Fig. 3(d) represents an area with only apartments.

3.1.2 Editing of digital maps

In this study, to classify and segment nonresidential buildings, general houses, townhouses, and apartments among building layers, the layers other than the target codes were deleted by referring to the standard feature codes of digital maps shown in Table 2.

Even if digital maps have the same time and spatial resolutions as the ortho-images, they may not match the actual objects owing to digital map update cycles or worker errors, as shown in Fig. 4(a). In this study, by overlaying ortho-images with applied coordinate systems and edited digital maps, areas or locations where objects differ were manually corrected for position as shown in Fig. 4(b). Therefore, the results of position correction for 12 map sheets corresponding to the target area of this study are shown in Fig. 4(c). In Fig. 4, red represents nonresidential buildings, orange represents general houses, yellow represents townhouses, and green represents apartments.



Fig. 3. (Color online) (a) Nonresidential buildings, (b) nonresidential buildings and houses, (c) nonresidential buildings, houses, and apartments, and (d) apartments.

Table 2
Standard layer codes for digital maps.

Code	B0014111	B0014112	B0014113	B0014115
Subcategory	Nonresidential building	House	Townhouse	Apartment

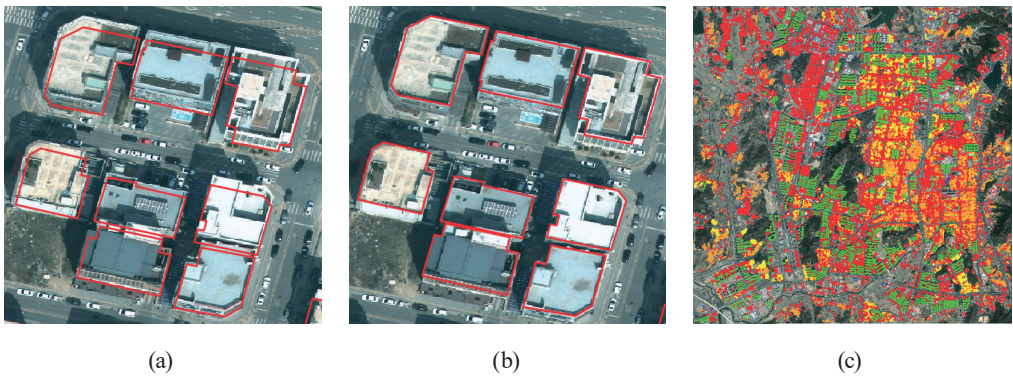


Fig. 4. (Color online) (a) Before correction, (b) after correction and (c) results after correction.

3.2 Training YOLOv8 for building detection

3.2.1 Constructing training data for building detection

In YOLOv8, the training data for building detection consist of bounding boxes that include the top-left and bottom-right coordinates of the target buildings. In this study, the bounding boxes were constructed using images generated by overlaying the previously edited ortho-images and digital maps. Some of the constructed building bounding boxes are shown in Fig. 5. The total number of building objects is 16468, and the number of objects by standard feature code is presented in Table 3.



Fig. 5. (Color online) Training data for YOLOv8: (a) nonresidential buildings, (b) nonresidential buildings and houses, (c) nonresidential buildings, houses, and apartments, and (d) apartments.

Table 3
Number of training data for YOLOv8.

Code	B0014111	B0014112	B0014113	B0014115	Total
Number of bounding boxes	10480	1104	2440	2444	16468

3.2.2 Training model for building detection

To build a model for detecting buildings, parameters were configured as shown in Table 4, and the tiled ortho-images along with the constructed bounding boxes were applied to YOLOv8. In Table 4, Epochs indicates the number of training repetitions for YOLOv8. Learning rate denotes the magnitude of updates applied to the model’s weights during each iteration of the training process. Additionally, Batch size denotes the amount of data used during each training iteration, and Weight decay represents the degree of regularization applied to prevent overfitting by penalizing large weights.

The trained model showed a precision of 99.8% and a recall of 99.7%, indicating relatively high accuracy, as shown in Table 5. Additionally, mAP50, which is the average precision calculated by setting the threshold for average precision to 0.50, was 99.5%, and mAP50-95, which is the average precision calculated by setting the threshold from 0.50 to 0.95, was 98.0%.

3.3 Fine-tuning SAM2 for building segmentation

3.3.1 Constructing training data for fine-tuning

The fine-tuning training data for SAM2 to segment buildings consisted of building mask images generated by overlaying edited ortho-images with digital maps. In the mask images, pixel values corresponding to building areas were set to 255, while those corresponding to background areas were set to 0. Some examples of the generated images are shown in Fig. 6, where Fig. 6(a) shows the ortho-images and Fig. 6(b) shows the corresponding mask images.

Table 4
Parameter set for YOLOv8 training.

Epochs	Learning rate	Resolution	Batch size	Weight decay
20000	0.01	1024	16	5×10^{-4}

Table 5
Results of trained YOLOv8.

Precision (%)	Recall (%)	mAP50 (%)	mAP50-95 (%)
99.8	99.7	99.5	98.0

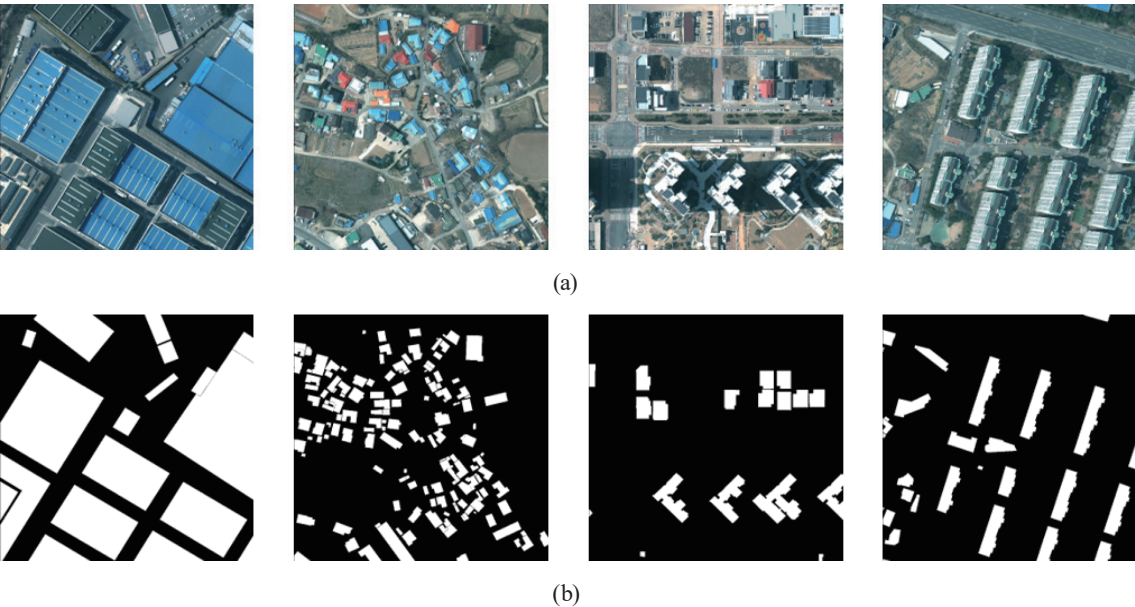


Fig. 6. (Color online) (a) Ortho-images and (b) mask images.

3.3.2 Fine-tuning SAM2

Fine-tuning is one of the representative techniques in transfer learning. It involves optimizing a model for a specific purpose by leveraging a pretrained model and further training it on data related to a particular task or domain.⁽²¹⁾ Pretrained models may not be well suited to specific tasks or domains, as they are typically trained to learn general features. However, fine-tuning helps overcome these limitations by adapting the model to the user’s objectives and enabling it to learn the relevant features and patterns in greater depth, thereby achieving the high level of accuracy and precision required for a specific task.⁽²²⁾ Therefore, in this study, SAM2 was fine-tuned using tiled ortho-images and the corresponding mask images. The parameters used for fine-tuning are presented in Table 6.

As shown in Fig. 7 and Table 7, the accuracy of the trained model gradually increased, and the intersection over union (*IoU*), which represents the object detection accuracy, reached 85.5%. In addition, the mean squared error (*MSE*) was 0.4%, and the mean absolute error (*MAE*) was 0.9%, indicating that the model was fine-tuned with relatively high accuracy.

Table 6
Parameter set for SAM2 fine-tuning.

Epochs	Learning rate	Resolution	Batch size	Weight decay
200000	6×10^{-5}	1024	16	0.1

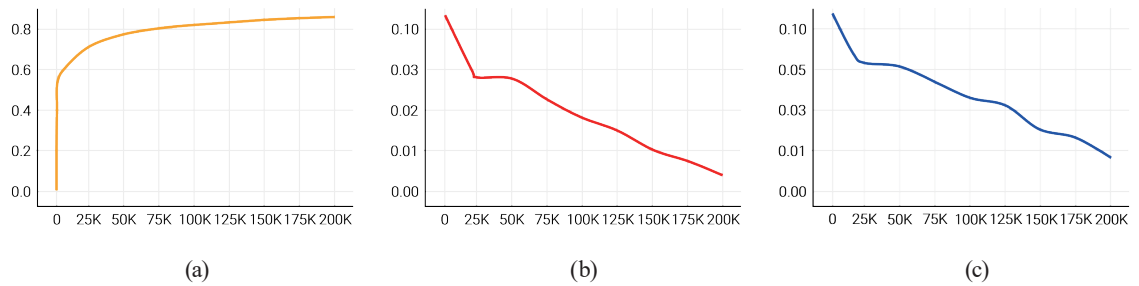


Fig. 7. (Color online) (a) IoU, (b) MSE loss, and (c) MAE loss.

Table 7
Building segmentation using multiprompts.

<i>IoU</i> (%)	<i>MSE</i> Loss (%)	<i>MAE</i> Loss (%)
85.5	0.4	0.9

3.4 Detection and segmentation of buildings

On the basis of the existing code provided by YOLOv8 and the open-source implementation of SAM2, we trained and fine-tuned a deep learning model for building detection and segmentation. Additionally, we developed and applied a method to generate multiple prompts for building segmentation during model evaluation, as shown in Table 8.

3.4.1 Detection of buildings

Figure 8 shows some examples of the results of applying YOLOv8 to tiled images, where Fig. 8(a) shows an area where mainly nonresidential buildings exist and Fig. 8(b) shows an area where nonresidential buildings and houses are densely distributed. Additionally, Fig. 8(c) shows an area where nonresidential buildings, houses, and apartments are diversely distributed, and Fig. 8(d) shows an area with only apartments.

As a result of building object detection, buildings could be detected at a rate of 89.8%, and since we did not train data other than building layers among standard feature codes, there were no cases of false detection of other objects.

Each object detected by YOLOv8 was used as a box prompt, as shown in Fig. 9, where each box represents a quadrilateral defined by the upper-left and lower-right coordinates. Subsequently, the image region corresponding to the box prompt was cropped and processed to generate a point prompt through image processing. The generated point prompt and the original box prompt were then combined and used as a multiprompt.

Table 8

Multiprompt building detection and segmentation algorithm using YOLOv8 and SAM2.

YOLOv8_model = YOLOv8(YOLOv8.pt)	SAM2_model = SAM2(SAM2.pt)
<pre># Create a multiprompt detection = YOLOv8_model.predict(image) for detect in detection : # Create a box prompt Box = detect.bboxes.xyxy # Create a point prompt Crop_image = image[bndry] HSV = cv2.cvtColor(Crop_image) Canny = cv2.Canny(HSV) Conn = connect_edge(Canny) Cnt = cv2.findContours(Conn) FilterImage=apply_contour_filter(Cnt) Erode = cv2.erode(FilterImage) FilterImage=apply_contour_filter(Erode) Mask = cv2.dilate(FilterImage) Point = get_points(Mask)</pre>	<pre># Segmentation using a multiprompt SAM2 = build_sam2(SAM2_model) predictor = SAM2ImagePredictor(SAM2) for image in images predictor.set_image(image) masks = predictor.predict(point_coords=Point, \ box=Box)</pre>

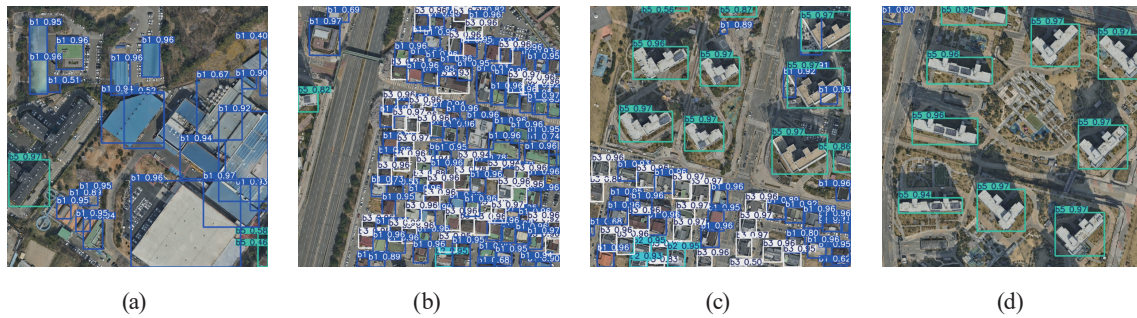


Fig. 8. (Color online) (a) Nonresidential buildings, (b) nonresidential buildings and houses, (c) nonresidential buildings, houses, and apartments, and (d) apartments.



Fig. 9. (Color online) Box prompts.

3.4.2 Point prompt generation

To generate point prompts, the previously cropped images were converted to the HSV color space, followed by edge and contour detection. Figure 10(a) shows the cropped building object

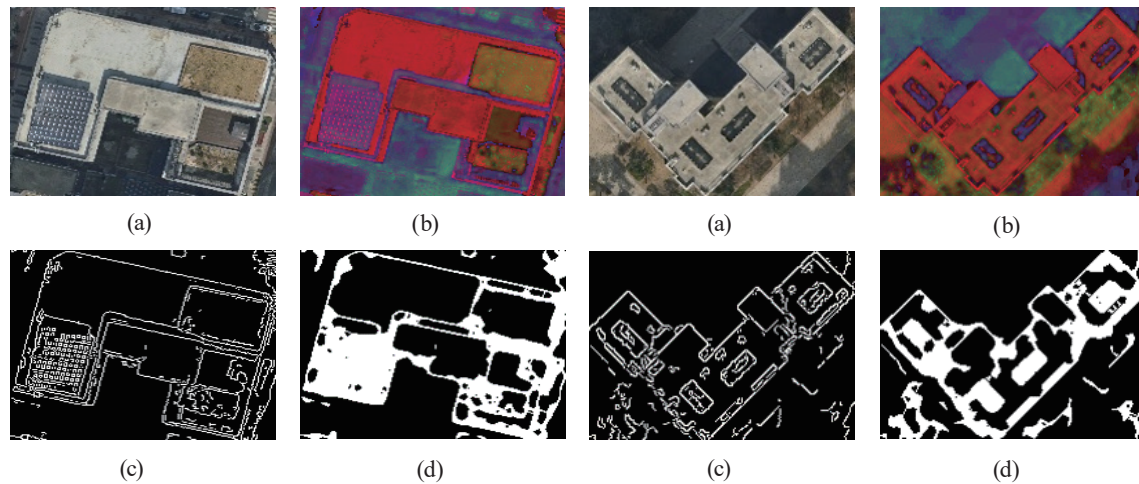


Fig. 10. (Color online) (a) Clipped image, (b) HSV image, (c) edge detection, and (d) contour detection.

and Fig. 10(b) presents the image after HSV conversion. Figure 10(c) illustrates the result of edge detection, while Fig. 10(d) shows the detected building contours based on the edge detection result.

Because of noise in the edge detection process, some linear structures forming the building objects may be partially missing. To address this, the distances between feature points were calculated during contour detection, and points within a certain threshold were connected. Additionally, while filling the interior of the contour lines, redundant lines were removed and connected linear structures were enhanced.

To generate mask images for building segmentation, Gaussian blur was applied to the images with detected building contours, followed by morphological operations. Some of the results of this process are shown in Fig. 11. Figure 11(a) shows the result after applying Gaussian blur, and Fig. 11(b) shows the result after performing morphological operations. In this study, Gaussian blur was applied prior to the morphological operations to reduce noise outside the building boundaries and to enhance the clarity of the object contours.

Point prompts for building segmentation were created as the center points within the masks of each object generated through image processing, as shown in Fig. 12, minimizing the work of manually inputting prompts.

As shown in Fig. 13, we combined box and point prompts, derived from object detection and image processing, respectively, to create a multiprompt that complementarily leverages both types of information. In the figure, the red quadrilaterals represent box prompts, and the blue dots represent point prompts.

3.4.3 Segmentation of buildings

To evaluate the building segmentation accuracy of the fine-tuned deep learning model and the generated prompts, experiments were conducted under two different conditions, as shown in Fig. 14: using the pretrained SAM2 with single prompts and using the fine-tuned model with multiprompts. Figure 14(a) shows the original images for segmentation, while Fig. 14(b) presents

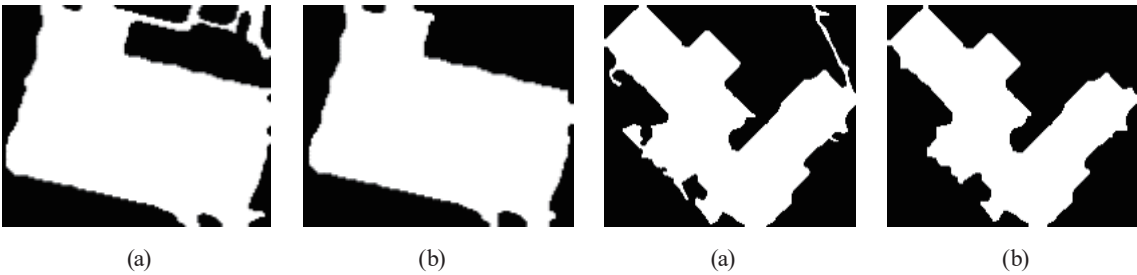


Fig. 11. (Color online) (a) Blur and (b) morphological images.

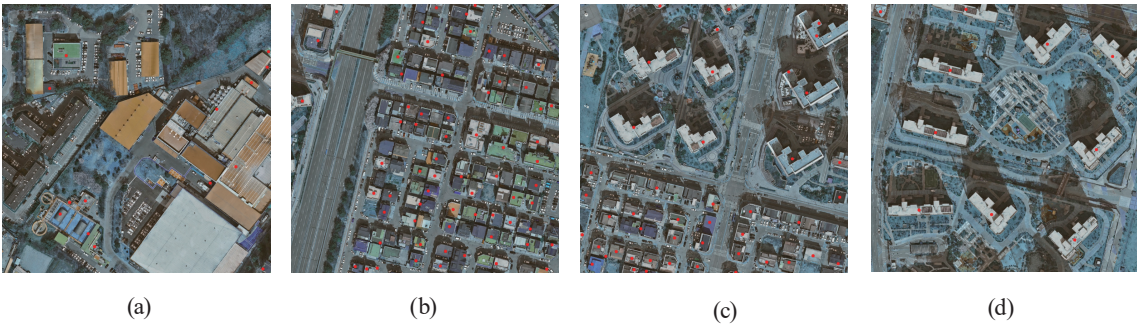


Fig. 12. (Color online) (a) Nonresidential buildings, (b) nonresidential buildings and houses, (c) nonresidential buildings, houses, and apartments, and (d) apartments.

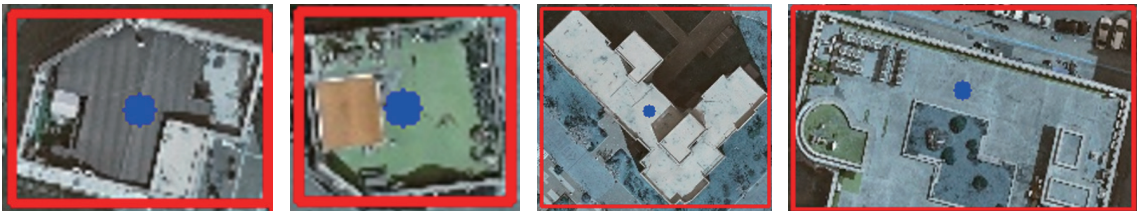


Fig. 13. (Color online) Multiprompts.

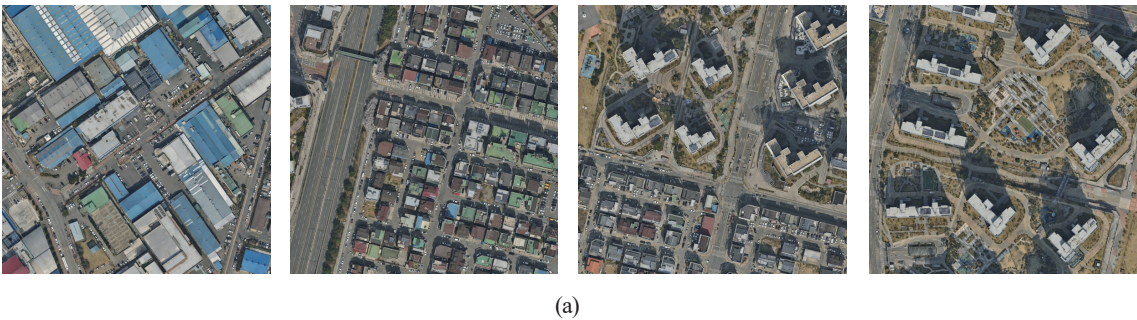


Fig. 14. (Color online) (a) Original images, (b) single prompt applied to a pretrained model, and (c) multiprompts applied to a fine-tuned model.

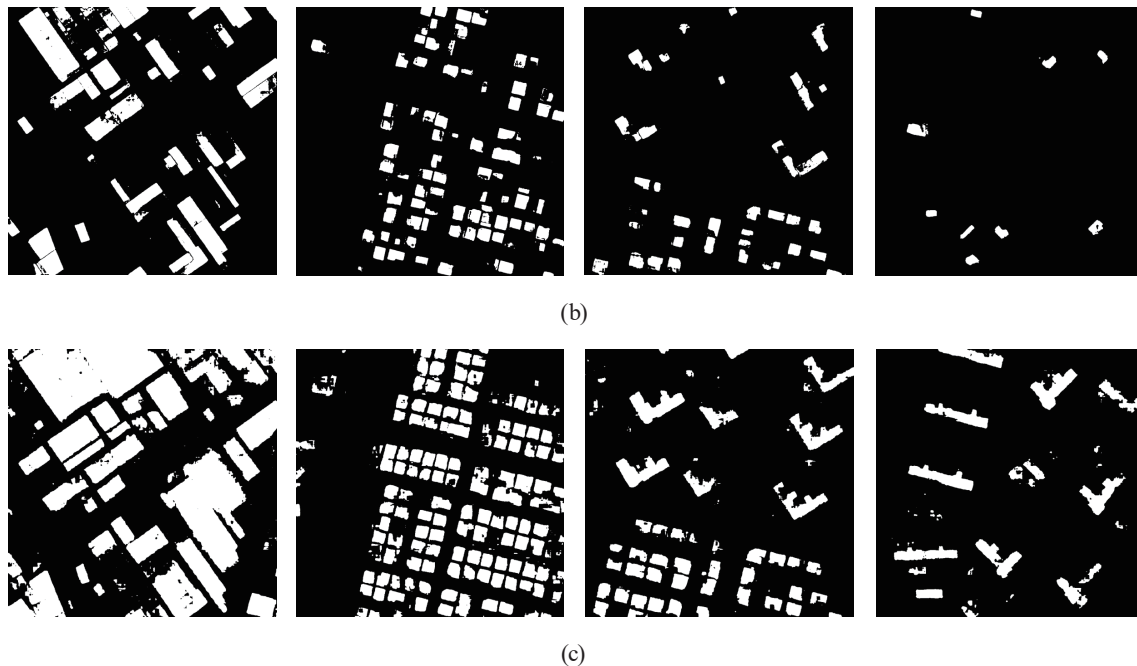


Fig. 14. (Color online) (Continued) (a) Original images, (b) single prompt applied to a pretrained model, and (c) multiprompts applied to a fine-tuned model.

the segmentation results obtained using the pretrained model with single prompts. Figure 14(c) presents the segmentation results obtained using the fine-tuned model with multiprompts.

A comparison of the pretrained model with the model fine-tuned using the building data targeted in this study showed that the pretrained model failed to segment most buildings owing to insufficient training on the dataset used in this study. In contrast, the fine-tuned model showed significantly improved performance in building segmentation, accurately identifying the majority of building objects. This improvement resulted from the fine-tuning process reflecting the characteristics of the building data, thereby achieving a higher accuracy than the pretrained model.

In addition, when performing object segmentation using a single prompt based on the center point of the mask image, we observed that the object was incorrectly segmented into multiple objects if the color values within the object varied. In contrast, when using the multiprompts proposed in this study, the object was correctly segmented as a single object even when the color values varied, confirming that the use of multiple prompts played an important role in improving the model performance.

4. Conclusions

In this study, we proposed a methodology of building segmentation by first detecting building bounding boxes in ortho-images using YOLOv8, then generating multiple prompts based on the detection results, and finally applying them to a fine-tuned SAM2. The following conclusions were drawn:

First, by generating prompts using various image processing techniques within the building regions detected by YOLOv8, building boundaries were effectively delineated and background noise was reduced, enabling the creation of clear and accurate mask images. This approach significantly reduced the need for manual prompt input and enhanced the accuracy and efficiency of the building segmentation process.

Second, a comparison between the pretrained SAM2 and the fine-tuned model revealed that the former exhibited limited segmentation performance owing to its lack of adaptation to building data. In contrast, the fine-tuned model, trained specifically on building imagery, demonstrated substantially improved segmentation performance.

Third, when performing object segmentation using a single prompt, one object was often incorrectly segmented into multiple objects if it contained varying color values. However, when multiprompts were used as proposed in this study, the object was correctly segmented as a single object, even in the presence of different color values. From these results, we confirmed that prompt input plays a critical role in improving the model's performance.

Future work will be focused on vectorizing and regularizing the segmented building regions to generate building objects, and the resulting objects, produced through the vectorization and regularization processes, will be integrated into geospatial information systems.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00554725).

References

- 1 S. Wang, X. Huang, P. Liu, M. Zhang, F. Biljecki, T. Hu, X. Fu, L. Liu, X. Liu, R. Wang, Y. Huang, J. Yan, J. Jiang, M. Chukwu, S. Naghedi, M. Hemmati, Y. Shao, N. Jia, Z. Xiao, T. Tian, Y. Hu, L. Yu, W. Yap, E. Macatulad, Z. Chen, Y. Cui, K. Ito, M. Ye, Z. Fan, B. Lei, and S. Bao: *Int. J. Earth Obs. and Geo.* **128** (2024) 103734. <https://doi.org/10.1016/j.jag.2024.103734>
- 2 A. S. Albahri, Y. L. Khaleel, M. A. Habeeb, R. Ismael, Q. Hameed, M. Deveci, R. Homod, O. S. Albahri, A. H. Alamoodi, and L. Alzubaidi: *Com. Elec. Eng.* **118** (2024) 109409. <https://doi.org/10.1016/j.compeleceng.2024.109409>
- 3 M. H. Je and S. H. Jung: *J. Korea Acad-Indu. Coop. Soc.* **25** (2024) 469. <https://doi.org/10.5762/KAIS.2024.25.1.469>
- 4 H. D. Seo and E. M. Kim: *J. Cad. Land Inf.* **50** (2020) 37. <https://doi.org/10.22640/ixsiri.2020.50.2.37>
- 5 W. Boonpook, Y. Tan, and B. Xu: *Int. J. Remote Sens.* **42** (2021) 1. <https://doi.org/10.1080/01431161.2020.1788742>
- 6 Z. Liu, H. Tang, L. Feng, and S. Lyu: *Earth Sys. Sci. Data* **15** (2023) 3547. <https://doi.org/10.5194/essd-15-3547-2023>
- 7 M. Wahbi, I. El Bakali, B. Ez-zahouani, R. Azmi, A. Moujahid, M. Zouiten, O. Alaoui, H. Boulaassal, M. Maatouk, and O. El Kharki: *Remote Sens. App. Soc. Env.* **29** (2023) 100898. <https://doi.org/10.1016/j.rsase.2022.100898>
- 8 K. Pushparani, G. Roja, R. Anusha, C. Dastagiriah, B. Srilatha, and B. Manjusha: 2024 15th International Conference on Computing Communication and Networking Technologies (IEEE, 2024) 1. <https://doi.org/10.1109/ICCCNT61001.2024.10724802>
- 9 S. H. Yoo and H. G. Sohn: *Korean Soc. Surv. Geodesy Photogramm. Cartogr.* **41** (2023) 605. <https://doi.org/10.7848/ksgpc.2023.41.6.605>
- 10 T. Liu, M. Gong, D. Lu, Q. Zhang, H. Zheng, F. Jiang, and M. Zhang: *IEEE Trans. Geo. and Remote Sens.* **60** (2021) 1. <https://doi.org/10.1109/TGRS.2021.3130940>

- 11 R. I. Sultan, C. Li, H. Zhu, P. Khanduri, M. Brocanelli, and D. Zhu: arXiv preprint (2024) 2311.11319. <https://doi.org/10.48550/arXiv.2311.11319>
- 12 Z. Gong, B. Li, C. Wang, J. Chen, and P. Zhao: Int. J. Geo. Inf. Sci. (2024) 1. <https://doi.org/10.1080/13658816.2024.2399142>
- 13 National Geographic Information Institute: <https://map.ngii.go.kr/mn/mainPage.do> (accessed March 2025).
- 14 Ultralytics: <https://github.com/ultralytics/ultralytics> (accessed March 2025).
- 15 H. D. Seo, C. H. Lee, J. H. Hong, S. G. Park, S. W. Jung, and E. M. Kim: Korean Soc. Surv. Geodesy Photogramm. Cartogr. **42** (2024) 61. <https://doi.org/10.7848/ksgpc.2024.42.1.61>
- 16 K. Muntarina, R. Mostafiz, F. Khanom, S. B. Shorif, and M. S. Uddin: Intel. Sys. App. **20** (2023) 200274. <https://doi.org/10.1016/j.iswa.2023.200274>
- 17 J. Canny: IEEE Trans. Pat. Anal. Mac. Intel. **6** (1986) 679. <https://doi.org/10.1109/TPAMI.1986.4767851>
- 18 S. K. Tummala: Indian J. Eng. Mater. Sci. **29** (2023) 796. <https://doi.org/10.56042/ijems.v29i6.70310>
- 19 N. Ravi, V. Gabeur, Y. T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. Alwala, N. Carion, C. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer: arXiv preprint (2024) 2408.00714. <https://doi.org/10.48550/arXiv.2408.00714>
- 20 A. S. Geetha, and M. Hussain: arXiv preprint (2024) 2408.06305. <https://doi.org/10.48550/arXiv.2408.06305>
- 21 T. Strangmann, L. Purucker, J. K. Franke, I. Rapant, F. Ferreira, and F. Hutter: arXiv preprint (2024) 2411.01195. <https://doi.org/10.48550/arXiv.2411.01195>
- 22 M. Chen and L. Li: Remote Sens. **17** (2025) 290. <https://doi.org/10.3390/rs17020290>

About the Authors



Hong-Deok Seo received his B.S. and M.S. degrees from Namseoul University, Korea, in 2019 and 2021, respectively. Since 2021, he has been a researcher at Spatial Information Industry Promotion Agency (SPACEN). Since 2022, he has been pursuing his Ph.D. degree at Namseoul University. His research interests are in photogrammetry, artificial intelligence (AI), and GIS. (seoing23@gmail.com)



Eui-Myoung Kim received his B.S. and M.S. degrees from Gyeongsang National University, Korea, in 1994 and 1996, respectively, and his Ph.D. degree from Yonsei University, Korea, in 2000. From 2000 to 2002, he was a senior researcher at Korea Institute of Civil Engineering and Building Technology (KICT). He worked as a postdoctoral fellow at the University of Calgary, Canada, from 2003 to 2005. He was with Korea Geospatial Information and Communication (KSIC) from 2005 to 2006. Since 2007, he has been a professor at Namseoul University, Korea. His research interests are in photogrammetry and GIS. (kemyoung@nsu.ac.kr)

