Sensors and Materials, Vol. 37, No. 6 (2025) 2663–2677
MYU Tokyo

S & M 4080

Improving Object Detection Performance in Built Areas from Drone Imagery Using Deep-learning-based Super-resolution Techniques

Phillip Kim[1,2] and Junhee Youn[1*]

[1]Department of Future & Smart Construction Research, Korea Institute of Civil Engineering and Building Technology, Goyang-si 10223, Republic of Korea
[2]Department of Educational Facilities & Environment Support, Korea Research Foundation of Local Educational Administration, Seoul 03923, Republic of Korea (Current Address)

(Received March 31, 2025; accepted June 13, 2025)

Keywords: drone imagery, object detection, super-resolution, construction management, dynamic spatial information

Drones equipped with various imaging sensors have become essential tools for urban monitoring, with applications spanning environmental change detection and traffic analysis. However, challenges such as small object sizes, viewpoint variability, and low resolution in high-altitude drone imagery limit the accuracy of object detection. In this study, we investigated the use of deep-learning-based super-resolution techniques to enhance object detection in drone imagery. The Super-Resolution Generative Adversarial Network (SRGAN) model was used to generated super-resolved imats at 2× and 4× scales to improve image quality. Objects were detected using the PaddlePaddle-You Only Look Once Enhanced-Small Object Detection (PP-YOLOE-SOD) algorithm, which enabled a comparative analysis of the object detection performance between original and super-resolved imagery. The findings indicate that 2× super-resolution significantly enhances the detection of small objects, such as pedestrians and two-wheeled vehicles, leading to improved recall and F2-scores. In contrast, 4× super-resolution reduced the detection accuracy. In this study, we demonstrated that super-resolution techniques can effectively address challenges associated with drone imagery at high altitudes, enhancing detection performance for small objects. However, the results underscore the importance of selecting appropriate resolution scales to avoid diminishing returns. These findings offer valuable insights for optimizing drone-based monitoring systems in urban environments, with implications for traffic management and object tracking under challenging conditions.

1. Introduction

With significant advancements in drone technology, including airframe and battery manufacturing, sensor integration, and AI-based operational systems, drones are now widely utilized across various sectors of society. Equipped with diverse sensors such as cameras and

*Corresponding author: e-mail: younj@kict.re.kr
https://doi.org/10.18494/SAM5648

ISSN 0914-4935

thermal imaging devices, drones collect a wide range of data that are analyzed using technologies such as edge computing, thereby expanding the scope and utility of the information gathered. The combined application of drones and imaging sensors has been extensively employed to detect natural and environmental phenomena,[1–10] monitor disaster situations,[11–23] and survey urban environments and built areas.[24–32] Specifically, drone applications for urban surveillance have concentrated heavily on the transportation sector. These applications include studies leveraging video imagery to identify individual moving objects, estimate traffic parameters (speed and traffic density), and track entities,[24–27] as well as those focused on deriving traffic flow by analyzing traffic parameters.[28,29] Together, these technological advances have enabled the development of comprehensive systems for monitoring urban traffic conditions based on information extracted from drone imagery.[30–32] Construction projects can be categorized into the planning, design, construction, and maintenance phases. With advancements in drone technology, collecting and monitoring urban traffic information in built areas can be utilized in the maintenance phase of construction projects. Traffic flows in built areas, such as cars, undergo rapid state changes, which impact urban infrastructure, including roads, and create the need for maintenance. Efficient traffic monitoring in built areas using drones can be actively utilized in the maintenance phase of construction projects.

Despite the rapidly expanding applications of drone technology across various fields, studies utilizing object detection techniques to monitor urban traffic conditions using drone imagery encounter several limitations. Imagery acquisition in urban areas is typically conducted at high altitudes to ensure safety from collisions with terrain features and structures, which results in smaller object sizes in the captured imagery. This reduction in object size can adversely affect detection performance, increasing both the numbers of false positives and false negatives when applying object detection methods.[33–38] Furthermore, unlike fixed cameras such as closed-circuit television, drones continuously change their direction and position during operation, causing variations in viewpoint.[39] These challenges are compounded by factors such as mixed object sizes[40] and imbalances in temporal perspectives during recording,[41] further diminishing detection accuracy. The key limitations of using drone imagery for urban traffic monitoring can thus be summarized as issues related to object size, density, viewpoint variability, and lighting conditions.[42] To address these challenges, various studies have focused on developing new datasets to overcome the shortcomings of existing ones or constructing novel models based on current datasets. Such efforts aim to improve the object detection performance and optimize the use of drone imagery for urban traffic monitoring.

One notable research focus is the creation of new datasets to address the limitations of existing ones. For example, Zhang *et al.*[36] developed a large-scale benchmark dataset to address challenges in building high-performing models with unmanned aerial vehicle (UAV)-acquired imagery, which often varies in orientation and scale. They evaluated its performance using six models.[36] Similarly, Razakarivony and Jurie[37] created a dataset capturing small objects under diverse constraints, such as varying viewpoints, lighting, and shadows, whereas Robicquet *et al.*[38] developed another dataset to study human trajectories, incorporating entities such as pedestrians and cyclists. Du *et al.*[43] tackled detection challenges by providing a new dataset for

UAV imagery, and Kim and Youn[44] proposed an evaluation methodology to automate dataset construction.

Efforts to enhance existing algorithms and develop new ones have been demonstrated in several studies. Li *et al.*[33] optimized convolutional neural network (CNN) models by refining anchor box selection to improve the detection of small, densely packed objects, achieving high performance across four large-scale benchmark datasets. Zhang *et al.*[42] introduced a novel algorithm to address environmental challenges such as small object sizes and viewpoint changes, validating its effectiveness on existing datasets. Similarly, Khoshboresh Masouleh and Shah-Hosseini[45] developed an algorithm for ground vehicle detection using thermal imagery, diverging from traditional dataset approaches. Zhu *et al.*[46] proposed a deep-learning-based solution for estimating traffic density, demonstrating through experiments that algorithm performance declined as image quality deteriorated.

In some studies, dataset construction has been integrated with algorithm development. Hsieh *et al.*[34] observed that existing detection and counting models were primarily designed for fixed, static cameras. To address this limitation, they developed a model suitable for dynamic environments such as those involving drones and simultaneously constructed a large-scale dataset to validate their model. Similarly, Jensen *et al.*[39] created a dataset featuring synchronized videos from a fixed camera and drone, using it to perform detection tasks. However, they noted significant challenges in synchronizing detections between videos taken from different viewpoints. Li *et al.*[35] proposed an algorithm capable of preprocessing imagery to account for object size changes caused by altitude variations while also excluding outliers based on image scale. This algorithm was validated using both proprietary and publicly available datasets. Li *et al.*[41] introduced an algorithm for detecting objects in nighttime imagery using daytime data, addressing imbalances in the constructed dataset while accounting for environmental differences between day and night. They validated their approach using the newly created dataset. Lastly, Wang *et al.*[47] developed a new dataset to improve the utility of drone imagery in traffic management, reducing false and missed detections by modifying the Faster R-CNN model.

The above literature review highlights several approaches to addressing various limitations, including the creation of new datasets, the development of novel algorithms, and the integration of both. Moreover, key factors affecting object detection in drone imagery have been identified, such as small object sizes and changes in viewpoint.

In this study, we proposed a method to enhance the object detection performance by addressing object size variability in drone imagery captured at different altitudes. By applying deep-learning-based super-resolution techniques, we can realize high-altitude imagery that can provide object sizes comparable to those captured at lower altitudes. Although previous studies have demonstrated that super-resolution methods improve detection performance in terms of mean average precision and accuracy,[48–50] they did not account for variations in performance across different object sizes. Against this backdrop, we evaluated the application of super-resolution techniques to drone imagery and the object detection performance of these techniques by comparing the drone imagery with the original imagery to assess their effectiveness. Through

this, dynamic spatial information, such as traffic changes in built areas, collected using drones, can serve as a foundation for enhancing the utilization of spatial information in construction management.

2. Data, Materials, and Methods

2.1 Collection and preprocessing of drone imagery

Drone imagery was collected as the basis for generating super-resolved imagery using super-resolution techniques and evaluated their object detection performance by comparing between the original and super-resolved imagery. The imagery was captured in Ilsanseo-gu, Republic of Korea, using a commercial drone. The manufacturers and models of the drone and camera used are detailed in Table 1.

Imagery was captured twice over the same region using different flight paths to explore different areas, ensuring data diversity and verifying the applicability of super-resolution under varying conditions. The first set of imagery was used for training and validating the super-resolution model, and the second set was manually labeled for objects within the imagery and used to evaluate the object detection performance by comparing between the original and super-resolved imagery. The schedule, altitude, resolution, and number of imagery sets for each collection are outlined in Table 2.

Super-resolution processes using deep learning models typically involve dividing imagery into tiles of a predefined size. Training a super-resolution algorithm on the unsplit original imagery demands substantial memory capacity, rendering it impractical for standard computing resources. Conversely, using excessively small tiles hampers the extraction of individual object features. Therefore, selecting an appropriate tile size is crucial.[51] On the basis of a review of previous studies, a tile size of 256 × 256 pixels was identified as optimal for training the super-resolution algorithm and was used in this study.[51–53]

In training the super-resolution algorithm, data pairs consisting of high-resolution (HR) and

Table 1
Drone and camera manufacturers and models used for collecting imagery.

	Drone	Camera
Manufacturer	ARGOSDYNE (Korea)	YUNEEC (China)
Product name	AQUILA-II	E90x

Table 2
Details of drone imagery collection.

No.	Purpose	Date	Altitude (m)	Resolution	No. of imagery sets
Dataset 1	Model training and validation for super-resolution	2023.05.12.	100	5472 × 3080	121
Dataset 2	Comparison of object detection performance following super-resolution	2023.10.13./16.	100	5472 × 3080	120

low-resolution (LR) imagery must be created using the same original imagery. The HR dataset was generated by dividing the original imagery into tiles of 256 × 256 pixels, whereas the LR datasets were created by downscaling the 256 × 256 tiles into 64 × 64 and 128 × 128 pixels. The 64 × 64 (LR) to 256 × 256 (HR) dataset was used for training and validating a 4× super-resolution algorithm, whereas the 128 × 128 (LR) to 256 × 256 (HR) dataset was utilized for a 2× super-resolution algorithm. To train, validate, and test the super-resolution algorithm, the entire dataset was divided into training and testing subsets at a 7:3 ratio. Additionally, 10% of the training data was allocated as a validation set to assess accuracy during model training

The resulting super-resolution algorithms produced two outputs: a 4× super-resolution model that generates 256 × 256 imagery from 64 × 64 inputs and a 2× super-resolution model that generates 256 × 256 imagery from 128 × 128 inputs. Detailed information on the original imagery and the split and downscaled imagery used in the training process is presented in Table 3. Figure 1 shows data pairs for the 256 × 256 HR and 64 × 64 LR datasets, as well as the 256 × 256 HR and 128 × 128 LR datasets.

After completing the training, validation, and testing of the super-resolution algorithm using Dataset 1, the resulting models were applied to Dataset 2 for super-resolution and object detection performance comparison. To ensure consistency, the original imagery sets in Dataset 2 were also divided similarly. For the 2× super-resolution model, 128 × 128 imagery sets were used as inputs to generate 256 × 256 outputs, whereas for the 4× super-resolution model, 64 × 64 imagery sets served as inputs to produce 256 × 256 outputs. The results of dividing Dataset 2 are presented in Table 4.

2.2 Training and validation of super-resolution model

The Super-resolution Generative Adversarial Network (SRGAN) algorithm was selected as the super-resolution model for drone imagery. In our previous study, in which we used imagery captured from the same region, SRGAN demonstrated detection performance superior to that of the Enhanced Deep Residual Networks algorithm.[51] Likewise, in various previous studies on the development of a super-resolution model for drone imagery, SRGAN was used as the baseline model.[52–54] Furthermore, multiple studies on super-resolution on aerial imagery, comparable to drone imagery, also adopted SRGAN as their baseline model.[54–57] On the basis of these findings, we selected SRGAN as the super-resolution model and trained it using our newly constructed dataset. The parameters used for training the model are listed in Table 5.

Table 3
Detailed information on datasets used for training super-resolution.

Image size	Count	Training dataset	Validation dataset	Test dataset	Remarks
5472 × 3080	121	–	–	–	Original imagery
256 × 256	86273	54351	6040	25882	Split HR imagery
128 × 128	86273				Downscaled HR imagery
64 × 64	86273				

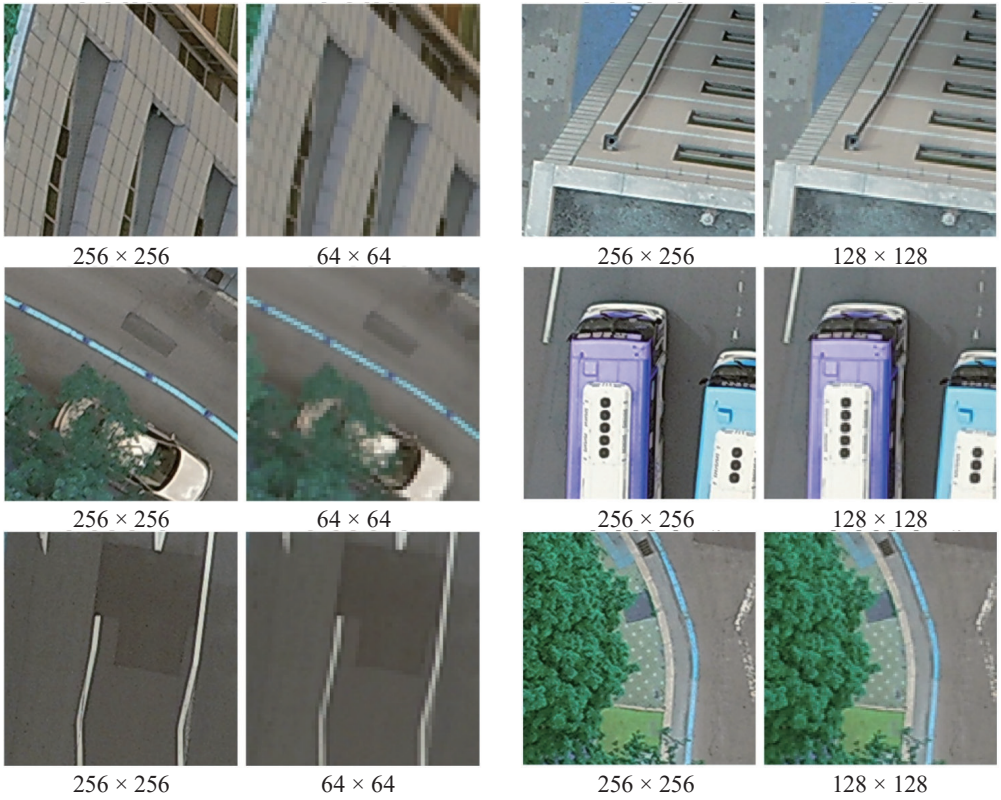

Fig. 1. (Color online) Examples of dataset pairs used for training and validation.

Table 4
Split results of Dataset 2.

Image size	Count	Category
5472 × 3080	120	Original imagery
128 × 128	264600	Split imagery for 2× super-resolution
64 × 64	1162800	Split imagery for 4× super-resolution

Table 5
Parameters used to train the super-resolution algorithm.

Initial learning rate	0.0002
Learning rate decay	0.99 × per epoch
Batch size	8
Epoch	100
Optimizer	Adam
Loss function	BCE with Logits
Contents loss function	L1

BCE, binary cross entropy.

2.3 Selection and application of the object detection algorithm

We previously proposed a method to automatically construct AI training datasets using drone imagery by applying existing object detection algorithms to detect moving objects and evaluated

their performance.[44] The PaddlePaddle-You Only Look Once Enhanced-Small Object Detection (PP-YOLOE-SOD) algorithm was selected to assess the object detection performance.[58] The same model was also used for object detection in this study, as the pretrained model had already demonstrated effective performance in detecting objects in drone imagery. Instead of undergoing a training process, a pretrained model was employed. The five object classes for the 120 imagery sets in Dataset 2 were manually labeled. The labeled results served as the ground truth (GT) for evaluating the performance of the object detection algorithm. For object detection, the intersection over the union threshold was set to 0.5, and the confidence threshold (CT) was set to 0. The classification framework used for object detection, the summary of the manually labeled results, and detection performance are summarized in Table 6. The object detection results for the original imagery were used as a baseline for comparing the performances of object detection algorithms applied to super-resolved imagery.

2.4 Performance evaluation of super-resolution model

Various image quality metrics can be used to evaluate the model performance of the super-resolution algorithm by comparing the results of applying a super-resolution model with the original imagery. In this study, peak signal-to-noise ratio (*PSNR*) and structural similarity index measure (*SSIM*) were used to assess changes in image quality resulting from super-resolution.

PSNR is a widely used metric for comparing the qualities of original and transformed imagery. It is commonly applied in image compression, transmission, and restoration tasks and used to quantify the ratio of signal to noise in decibels (dB). *PSNR* is calculated using the following equation:

$$PSNR = 10 \times \log_{10}\left(\frac{MAX^2}{MSE}\right), \tag{1}$$

where *MAX* is the maximum pixel value of the image, and *MSE* is the mean squared error between the original and transformed imagery, defined as

$$MSE = \frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - K(i,j)\right]^2, \tag{2}$$

Table 6
Object classification framework, manually labeled number of objects (ground truth), and object detection performance on original imagery.

Class	Pedestrian	Two wheeled vehicle	Car	Truck	Bus	Overall
Manually labeled No. of objects	640	335	1380	153	42	2,550
Performance of object detection	0.8094 (518/640)	0.8030 (269/335)	0.9993 (1379/1380)	0.9935 (152/153)	0.9524 (40/42)	0.9247 (2358/2550)

where m and n represent the height and width of the image, respectively. I and K denote the original and transformed imagery, respectively, with (i,j) indicating the pixel value at each location in the image.

SSIM is a quantitative metric used to assess the visual similarity between two imagery sets. It was proposed to overcome some limitations of *PSNR* by considering human visual perception, which focuses on the structural information within imagery. *SSIM* is computed using the following equation:

$$SSIM\left(x,y\right)=\frac{\left(2\mu_x\mu_y+C_1\right)\left(2\sigma_{xy}+C_2\right)}{\left(\mu_x^2+\mu_y^2+C_1\right)\left(\sigma_x^2+\sigma_y^2+C_2\right)}, \tag{3}$$

where μ_x and μ_y represent the mean values of imagery x and y, respectively, σ_x and σ_y denote their variances, σ_{xy} is the covariance between x and y, and C_1 and C_2 are constants used for stabilization.

3. Results and Discussion

The experimental procedure consisted of two main steps: (1) training the super-resolution algorithm using Dataset 1 and evaluating its performance on test data excluded from training and (2) applying the trained super-resolution model to Dataset 2, and performing object detection on the resulting super-resolved imagery. The results were then compared with those obtained from the original imagery.

3.1 Results of applying the super-resolution model

Table 7 shows the quality metrics obtained by applying the two models for 2× and 4× super-resolutions to the test dataset and comparing them with the original dataset. Figures 2 and 3 show examples of image triplets compared in the order of LR-HR-SR after performing the 2× and 4× super-resolution processes, respectively.

We compared the image quality metrics obtained by applying the super-resolution model with those reported in previous studies with similar methodologies to evaluate the effectiveness of the super-resolution model's training process. To this end, studies utilizing SRGAN for aerial imagery captured by drones or similar media were reviewed. The comparison results are summarized in Table 8.

In comparable previous studies utilizing 2× super-resolution, the average *PSNR* and *SSIM* values were 27.82 and 0.74, representing 93 and 71% of the values achieved in our study,

Table 7
Quality metrics (*PSNR* and *SSIM*) for 2× and 4× super-resolutions.

Scale	*PSNR*	*SSIM*
2×	26.8468	0.9164
4×	24.2645	0.7417

PSNR: peak signal-to-noise ratio; *SSIM,* structural similarity index measure.

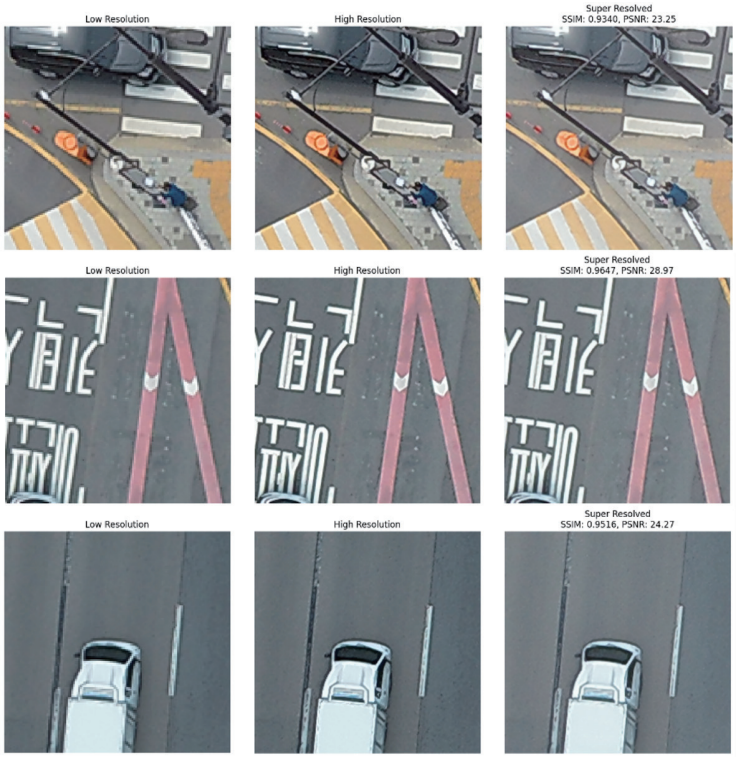

Fig. 2. (Color online) Examples of LR-HR-SR(2×) image pairs.

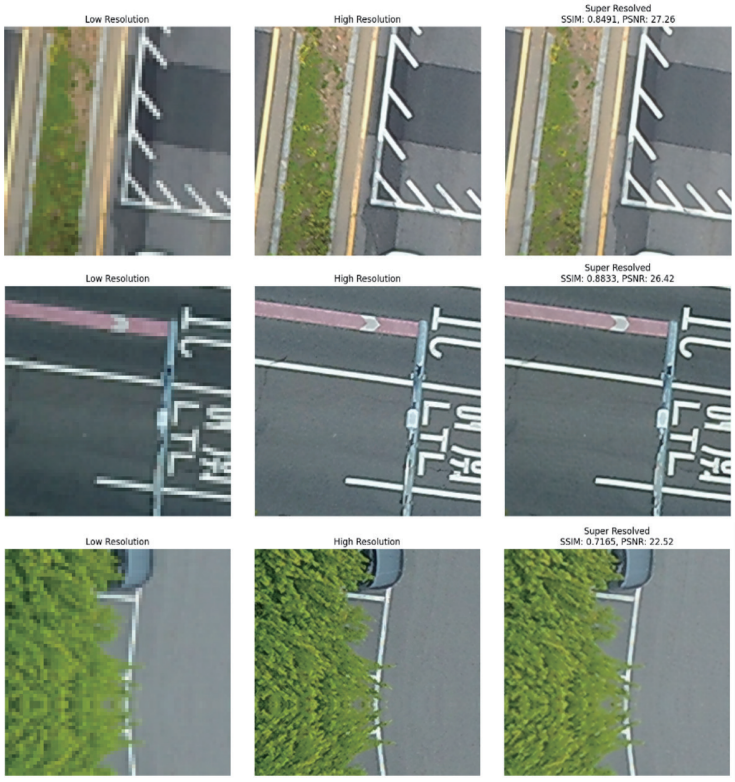

Fig. 3. (Color online) Examples of LR-HR-SR(4×) image pairs.

Table 8
Comparison of image quality metrics across studies.

Study	Imaging System	Scale	*PSNR*	*SSIM*
Fan *et al.*[52]	Drone	2×	22.572	0.528
		4×	21.287	0.467
Ren *et al.*[54]	Satellite	3×	30.7	0.8002
		3×	31.26	0.8114
		3×	31.61	0.8137
		3×	32.18	0.8189
Karwowska and Wierzbicki[55]	Drone, Satellite	2×	19.92	0.62
Pang *et al.*[56]	Satellite	2×	32.5216	0.8776
		3×	31.6962	0.8224
		4×	30.3971	0.7912
Zhao *et al.*[57]	Satellite	2×	24.7729	0.5922
El amraoui *et al.*[53]	Drone	4×	26.98	0.707
		4×	26.67	0.787
Our study	Drone	2×	26.8468	0.9164
		4×	24.2645	0.7417

respectively. For 4× super-resolution, the average *PSNR* and *SSIM* values were 26.3 and 0.70, corresponding to 108 and 94% of the values achieved in our study, respectively. Except the 4× *PSNR* value, the image quality metrics for super-resolution in this study demonstrated superior performance compared with those obtained in previous studies.

3.2 Results of applying the object detection model

The object detection model was applied to original imagery, as well as to 2× and 4× super-resolved imagery. The detected objects were compared against the GT, and the detection results (recall) are presented in Table 9.

In the object detection results, a 2× super-resolution improved detection performance for the pedestrian and two-wheeled vehicle classes by approximately 5 percentage points. However, the 4× super-resolution led to an average decrease in detection performance of 5 percentage points.

In deep-learning-based object detection, setting a CT is essential. Objects detected by the model are assigned a confidence score (CS), and only those with a CS exceeding the CT are considered valid detections. The object detection performance discussed above was evaluated with the CT set to 0.

In our previous study, the F2-score has been suggested as a metric for evaluating the object detection performance.[44] The F2-score is specifically designed to minimize false negatives and positives, making it an effective tool for constructing high-quality AI training datasets by considering the task of corrections to detection results. A high F2-score indicates that the object detection model exhibited good performance. The F2-score was calculated for original, 2× super-resolved, and 4× super-resolved imagery. The results of this performance evaluation are summarized in Table 10.

The comparison of F2-scores under different CT settings revealed that applying the 2× super-resolution model improved the object detection performance by 3 percentage points for

Table 9
Object detection results (recall).

	Pedestrian	Two wheeled vehicle	Car	Truck	Bus	Overall
Original	0.8094 (518/640)	0.8030 (269/335)	0.9993 (1379/1380)	0.9935 (152/153)	0.9524 (40/42)	0.9247 (2358/2550)
2×	0.8609 (551/640)	0.8627 (289/335)	0.9935 (1371/1380)	0.9935 (152/153)	0.9524 (40/42)	0.9524 (2403/2550)
	+0.0515 (+33)	+0.0597 (+20)	−0.0058 (−8)	+0 (+0)	+0 (+0)	+45 (+0.0474)
4×	0.7484 (479/640)	0.6030 (202/335)	0.9913 (1368/1380)	0.9804 (150/153)	0.9048 (38/42)	0.8773 (2237/2550)
	−0.061 (−39)	−0.2 (−67)	−0.08 (−11)	−0.0131 (−2)	−0.0476 (−2)	−0.0474 (−121)

Table 10
Results of object detection performance evaluation using different metrics.

Resolution	Metrics	Pedestrian	Two wheeled vehicles	Car	Truck	Bus
Original	CT	0.35	0.22	0.60	0.36	0.73
	F2-score	**0.6295**	**0.4687**	0.8763	**0.8099**	0.7258
	Recall	0.6797 (435/640)	0.5940 (199/335)	0.9826 (1356/1380)	0.9020 (138/153)	0.8571 (36/42)
	Precision	0.4860 (435/895)	0.2542 (199/783)	0.6116 (1356/2217)	0.5750 (138/240)	0.4500 (36/80)
2×	CT	0.39	0.32	0.60	0.40	0.59
	F2-score	**0.6589**	**0.5208**	0.8665	**0.8178**	0.6944
	Recall	0.7016 (449/640)	0.5761 (193/335)	0.9783 (1350/1380)	0.9150 (140/153)	0.8333 (35/42)
	Precision	0.5301 (449/847)	0.3762 (193/513)	0.5947 (1350/2270)	0.5738 (140/244)	0.4167 (35/84)
4×	CT	0.35	0.26	0.55	0.54	0.70
	F2-score	0.5478	0.3239	0.8095	0.6164	0.3009
	Recall	0.5656 (362/640)	0.4418 (148/335)	0.9101 (1256/1380)	0.6993 (107/153)	0.6190 (26/42)
	Precision	0.4866 (362/744)	0.1566 (148/945)	0.5612 (1256/2238)	0.4180 (107/256)	0.0985 (26/264)

pedestrians, 5 percentage points for two wheeled vehicle, and 1 percentage point for trucks. However, applying the 4× super-resolution model resulted in decreased detection performance across all object classes.

When detecting objects using the 2× super-resolution model and adjusting the CT to maximize the F2-score, the results indicated improved detection performance for small objects, such as pedestrians and two-wheeled vehicles, compared with the original imagery. In contrast, performance changes for cars and trucks were minimal, remaining below 1%, whereas the detection performance for buses declined by 3 percentage points. Considering the challenges highlighted in previous studies regarding small object detection, these findings validate the potential of super-resolution models to enhance detection performance, particularly for smaller objects.

The applicability of super-resolution for enhancing the object detection performance was examined using imagery collected from a different region. The imagery was acquired in Anyang-si, Republic of Korea, on September 27, 2024. A DJI Matrice 300 drone equipped with a Zenmuse H20T payload was used for data acquisition. The drone operated at an altitude of 100 meters, capturing imagery at a resolution of 4,056 × 3,040. The object detection performance in this area was evaluated using recall and is summarized in Table 11.

In the case of 2× super-resolution, performance improvements were observed for car, truck, and bus, with truck detection improving by 3 percentage points and bus detection by 6 percentage points. For 4× super-resolution, truck detection improved by 8 percentage points, whereas bus detection improved by 11 percentage points. Unlike the previous results, where applying super-resolution improved the detection performance of small objects, the detection

Table 11
Object detection results in another region (recall).

	Pedestrian	Two wheeled vehicle	Car	Truck	Bus	Overall
Original	0.7541 (1420/1883)	0.5851 (110/188)	**0.9516** (7020/7377)	**0.7181** (721/1004)	**0.6613** (41/62)	0.8857 (9312/10514)
2×	0.7366 (1387/1883) −0.0175 (−33)	0.5532 (104/188) −0.0319 (−6)	**0.9555** (7049/7377) +0.0039 (+29)	**0.744** (747/1004) +0.0259 (+26)	**0.7258** (45/62) +0.0645 (+4)	0.8876 (9332/10514) +0.0019 (+20)
4×	0.5438 (1024/1883) −0.2103 (−396)	0.5532 (104/188) −0.0319 (−6)	0.9498 (7007/7377) −0.0018 (−13)	0.8018 (805/1004) +0.0837 (+84)	0.7742 (48/62) +0.1129 (+7)	0.8549 (8988/10514) −0.0308 (−24)

Fig. 4. (Color online) Examples of object detection for original-HR(2×)-HR(4×) image pairs. The red circle means the objects were not detected in the original imagery.

performance of larger objects improved in this case. This difference is likely due to variations in the camera specifications and image resolution between the two datasets. Nevertheless, on the basis of recall, super-resolution can enhance the object detection performance even for imagery captured in different regions. Figure 4 shows the results of object detection performed on the original, 2× super-resolved, and 4× super-resolved imagery. The red circles in the figure indicate objects that were not detected in the original image but were successfully detected in the super-resolved imagery.

4. Conclusions

We investigated the application of super-resolution algorithms to enhance object detection accuracy and standardize resolution in drone imagery captured at varying altitudes. The SRGAN model was trained on drone imagery and evaluated using *PSNR* and *SSIM*. For 2× super-resolution, *PSNR* and *SSIM* were 26.85 and 0.92, and for 4×, they were 24.26 and 0.74, respectively. These results are generally comparable to previously reported averages.

Object detection using the pretrained PP-YOLOE-SOD algorithm showed that 2× super-resolution improved the detection performance for small objects by approximately 5 percentage points, whereas 4× super-resolution led to performance degradation across all classes. A similar trend was observed in F2-score comparisons. Additional analysis using imagery from another region showed that 2× super-resolution improved detection for cars, trucks, and buses, whereas 4× super-resolution was effective only for trucks and buses.

The decrease in *SSIM* at 4× super-resolution suggests that structural distortions may negatively impact detection performance in complex urban scenes. These findings highlight the importance of selecting appropriate super-resolution scales, particularly for high-density environments where small objects dominate. Future research will focus on systematically analyzing detection performance across various object types and scenes to optimize super-resolution parameters for UAV-based monitoring in urban environments.

Acknowledgments

This research was funded by a grant from the Korea Agency for Infrastructure Technology Advancement (KAIA) funded by the Ministry of Land, Infrastructure, and Transport [Grant number RS-2022-00143782 (Development of Fixed/Moving Platform Based Dynamic Thematic Map Generation Technology for Next-generation Digital Land Information Construction)].

References

1 Y. Chen, W. S. Lee, H. Gan, N. Peres, C. Fraisse, Y. Zhang, and Y. He: Remote Sens. **11** (2019) 1584. https://doi.org/10.3390/rs11131584
2 S. J. Hong, Y. Han, S. Y. Kim, A. Y. Lee, and G. Kim: Sensors **19** (2019) 1651. https://doi.org/10.3390/s19071651
3 J. Gao, Z. Hu, K. Bian, X. Mao, and L. Song: IEEE Internet Things J. **8** (2021) 428. https://doi.org/10.1109/JIOT.2020.3004582
4 J. H. Han, M. S. Suh, H. Y. Yu, and N. Y. Roh: Remote Sens. **12** (2020) 3181. https://doi.org/10.3390/rs12193181
5 S. Mahdavi, M. Amani, T. Bullock, and S. Beale: IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **14** (2020) 1363. https://doi.org/10.1109/jstars.2020.3036815

6 W. Shao, R. Kawakami, R. Yoshihashi, S. You, H. Kawase, and T. Naemura: Int. J. Remote Sens. **41** (2020) 31. https://doi.org/10.1080/01431161.2019.1624858
7 L. Wittstruck, I. Kühling, D. Trautz, M. Kohlbrecher, and T. Jarmer: Sensors **21** (2020) 118. https://doi.org/10.3390/s21010118
8 W. Yuan and D. Choi: Remote Sens. **13** (2021) 273. https://doi.org/10.3390/rs13020273
9 A. Mukundan, C. C. Huang, T. C. Men, F. C. Lin, and H. C. Wang: Sensors **22** (2022) 6231. https://doi.org/10.3390/s22166231
10 R. Lyu, J. Zhang, J. Pang, and J. Zhang: J. Cleaner Prod. **437** (2025) 140613. https://doi.org/10.1016/j.jclepro.2024.140613
11 C. Kyrkou, G. Plastiras, T. Theocharides, S. I. Venieris, and C. S. Bouganis: Automation & Test in Europe Conf. & Exhivition. (IEEE, 2018) 967.
12 E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, and A. Gasteratos: Sensors **19** (2019) 3542. https://doi.org/10.3390/s19163542
13 G. D. Georgiev, G. Hristov, P. Zahariev, and D. Kinaneva: 28th National Conf. with Int. Participation (TELECOM). (IEEE, 2020) 57.
14 Z. Jiao, Y. Zhang, L. Mu, J. Xin, S. Jiao, H. Liu, and D. Liu: Chinese Control and Decision Conf. (IEEE, 2020) 4963
15 J. Vasconcelos Reinolds de Sousa, and P. Vieira Gamboa: Int. Cong. Engineering - Engineering for Evolution. (KnE Engineering) 242.
16 Y. Pi, N. D. Nath, and A. H. Behzadan: Adv. Eng. Inf. **43** (2020) 101009. https://doi.org/10.1016/j.aei.2019.101009
17 S. Sudhakar, V. Vijayakumar, C. S. Sathiya Kumar, V. Priya, L. Ravi, and V. Subramaniyaswamy: Comput. Commun. **49** (2020) 1. https://doi.org/10.1016/j.comcom.2019.10.007
18 R. Zhang, H. Li, K. Duan, S. You, K. Liu, F. Wang, and Y. Hu: Remote Sens. **12** (2020) 2621. https://doi.org/10.3390/rs12162621
19 X. Wang, C. E. Wittich, T. C. Hutchinson, Y. Bock, D. Goldberg, E. Lo, and F. Kuester: J. Comput. Civ. Eng. **34** (2020) 4020045. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000928
20 A. Sharma, and P. K. Singh: Int. J. Commun. Syst. **38** (2021) e4826. https://doi.org/10.1002/dac.4826
21 J. Ding, J. Zhang, Z. Zhan, X. Tang, and X. Wang: Remote Sens. **14** (2022) 663. https://doi.org/10.3390/rs14030663
22 V. L. Kasyap, D. Sumathi, K. Alluri, P. Reddy Ch, N. Thilakarathne, and R. M. Shafi: Comput. Intell. Neurosci. **2022** (2022) 3170244. https://doi.org/10.1155/2022/3170244
23 X. Wang, E. Lo, L. De Vivo, T. C. Hutchinson, and F. Kuester: Struct. Control Health Monit. **2022** (2022) e2862. https://doi.org/10.1002/stc.2862
24 W. Sun, L. Dai, X. Zhang, P. Chang, and X. He: Appl. Intell. **52** (2022) 8448. https://doi.org/10.1007/s10489-021-02893-3
25 N. Balamuralidhar, S. Tilon, and F. Nex: Remote Sens. **13** (2021) 573. https://doi.org/10.3390/rs13040573
26 J. Li, S. Chen, F. Zhang, E. Li, T. Yang, and Z. Lu: Remote Sens. **11** (2019) 1241. https://doi.org/10.3390/rs11101241
27 H. Zhang, M. Liptrott, N. Bessis, and J. Cheng: 16th IEEE Int. Conf. Advanced Video and Signal Based Surveillance. (IEEE, 2019) 1.
28 R. Ke, Z. Li, S. Kim, J. Ash, Z. Cui, and Y. Wang: IEEE Trans. Intell. Transport. Syst. **18** (2017) 890. https://doi.org/10.1109/TITS.2016.2595526
29 R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang: IEEE Trans. Intell. Transport. Syst. **20** (2019) 54. https://doi.org/10.1109/TITS.2018.2797697
30 Y. Y. Wang, C. W. Huang, Y. H. Huang, C. F. Wang, and Y. H. Hung: Int. Conf. Consumer Electronics-Taiwan (IEEE, 2023) 821.
31 J. Garau Guzman, and V. M. Baeza: Drones **8** (2023) 7. https://doi.org/10.3390/drones8010007
32 L. Jian, Z. Li, X. Yang, W. Wu, A. Ahmad, and G. Jeon: IEEE Consum. Electron. Mag. **8** (2019) 81. https://doi.org/10.1109/MCE.2019.2892286
33 W. Li, H. Li, Q. Wu, X. Chen, and K. N. Ngan: IEEE Trans. Ind. Electron. **66** (2019) 9651. https://doi.org/10.1109/TIE.2019.2899548
34 M. R. Hsieh, Y. L. Lin, and W. H. Hsu: Proc. of the IEEE Int. Conf. Computer Vision (IEEE, 2017) 4165.
35 X. Li, X. Li, and H. Pan: IEEE Access **8** (2020) 208643. https://doi.org/10.1109/ACCESS.2020.3036075
36 H. Zhang, M. Sun, Q. Li, L. Liu, M. Liu, and Y. Ji: Neurocomputing **421** (2021) 173. https://doi.org/10.1016/j.neucom.2020.08.074
37 S. Razakarivony, and F. Jurie: J. Visual. Commun. Image Represent. **34** (2016) 187. https://doi.org/10.1016/j.jvcir.2015.11.002

38 A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese: Computer Vision-ECCV 2016 14th European Conference. (Springer International Publishing) 549.
39 M. B. Jensen, A. Møgelmose, and T. B. Moeslund: 23rd Int. Conf. Intelligent Transportation Systems. (IEEE, 2020) 1.
40 K. Mu, F. Hui, and X. Zhao: J. Inf. Process. Syst. **12** (2016) 196. https://doi.org/10.3745/JIPS.03.0037
41 J. Li, Z. Xu, L. Fu, X. Zhou, and H. Yu: Transp. Res. Part C Emerging Technol. **124** (2021) 102946. https://doi.org/10.1016/j.trc.2020.102946
42 X. Zhang, E. Izquierdo, and K. Chandramouli: Proc. IEEE/CVF Int. Conf. Computer Vision Workshops. (IEEE, 2019) 118.
43 D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian: Proc. European Conf. Computer Vision. (Springer International Publishing) 370.
44 P. Kim and J. Youn: Sensors **24** (2024) 6347. https://doi.org/10.3390/s24196347
45 M. K. Khoshboresh Masouleh, and R. Shah-Hosseini: ISPRS J. Photogramm. Remote Sens. **155** (2019) 172. https://doi.org/10.1016/j.isprsjprs.2019.07.009
46 J. Zhu, K. Sun, S. Jia, Q. Li, X. Hou, W. Lin, B. Liu, and G. Qiu: IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **11** (2018) 4968. https://doi.org/10.1109/JSTARS.2018.2879368
47 L. Wang, J. Liao, and C. Xu: Proc. Of the 2018 11th Int. Conf. Machine Learning and Computing (ACM, 2019) 466.
48 J. Li, Z. Zhang, Y. Tian, Y. Xu, Y. Wen, and S. Wang: IEEE Geosci. Remote Sens. Lett. **19** (2021) 1. https://doi.org/10.1109/LGRS.2021.3112172
49 C. Xing, X. Liang, and Z. Bao: 7th Int. Conf. Computer Science and Network Technology (IEEE, 2019) 313.
50 Y. Chen, J. Li, Y. Niu, and J. He: Chinese Control and Decision Conf. (IEEE, 2019) 4610.
51 S. Yoo, P. Kim, and J. Youn: J. Korean Soc. Surv. Geod. Photogramm. Cartogr. **41** (2023) 651. https://doi.org/10.7848/ksgpc.2023.41.6.651.
52 K. Fan, M. Hu, M. Zhao, L. Qi, W. Xie, H. Zou, B. Wu, S. Zhao, and X. Wang: Forest **15** (2024) 859. https://doi.org/10.3390/f15050859
53 K. El amraoui, Z. Pu, L. Koutti, L. Masmoudi, and J. Valente de Oliveira: Neurocomputing **577** (2024) 127346. https://doi.org/10.1016/j.neucom.2024.127346
54 Z. Ren, J. Zhao, C. Chen, Y. Lou, and X. Ma: Appl. Sci.-Basel **13** (2023) 1245. https://doi.org/10.3390/app13031245
55 K. Karwowska, and D. Wierzbicki: Remote Sens. **16** (2024) 1926. https://doi.org/10.3390/rs16111926
56 B. Pang, S. Zhao, and Y. Liu: Remote Sens. **15** (2023) 5064. https://doi.org/10.3390/rs15205064
57 J. Zhao, Y. Ma, F. Chen, E. Shang, W. Yao, S. Zhang, and J. Yang: Remote Sens. **15** (2023) 1391. https://doi.org/10.3390/rs15051391
58 S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du, and B. Lai: arXiv:2203.16250 (accessed May 2025).