electronics

Article

A Novel Anti-UAV Detection Method for Airport Safety Based on Style Transfer Learning and Deep Learning

Ruiheng Zhang [1,2], Yitao Song [2], Ruoxi Zhang [1], Yang Lei [1], Hanglin Cheng [2] and Jingtao Zhong [3,*]

[1] Luoyang Flight College, Civil Aviation Flight University of China, Luoyang 471132, China; 220231710@seu.edu.cn (R.Z.)
[2] School of Transportation, Southeast University, Nanjing 211189, China
[3] Department of Civil and Environmental Engineering, The University of Tennessee Knoxville, Knoxville, TN 37996, USA
* Correspondence: jzhong7@vols.utk.edu

Abstract

Unmanned aerial vehicle (UAV) intrusions cause flight delays and disrupt airport operations, so accurate monitoring is essential for safety. To address the scarcity and mismatch of real-world training data in small-target detection, an anti-UAV approach is proposed that integrates style transfer learning (STL) with deep learning. An airport monitoring platform is established to acquire a real UAV dataset, and a Cycle-Consistent Generative Adversarial Network (CycleGAN) is employed to synthesize multi-scene images that simulate diverse airport backgrounds, thereby enriching the training distribution. Using these simulated scenes, a controlled comparison of YOLOv5/YOLOv6/YOLOv7/YOLOv8 is conducted, in which YOLOv5 achieves the best predictive performance with AP values of 93.95%, 98.09%, and 97.07% across three scenarios. On public UAV datasets, the STL-enhanced model (YOLOv5_STL) is further compared with other small-object detectors and consistently exhibits superior performance, indicating strong cross-scene generalization. Overall, the proposed method provides an economical, real-time solution for airport UAV intrusion prevention while maintaining high accuracy and robustness.

Keywords: CycleGAN; object detection; UAV; YOLO; airport safety

check for updates

Academic Editor: Chiman Kwan

Received: 9 October 2025
Revised: 16 November 2025
Accepted: 24 November 2025
Published: 25 November 2025

Citation: Zhang, R.; Song, Y.; Zhang, R.; Lei, Y.; Cheng, H.; Zhong, J. A Novel Anti-UAV Detection Method for Airport Safety Based on Style Transfer Learning and Deep Learning. *Electronics* **2025**, *14*, 4620. https://doi.org/10.3390/electronics14234620

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

UAVs are becoming smaller, automated, and intelligent [1]. They are widely used in military [2] and civilian fields [3]. However, due to inadequate regulation, incidents of unauthorized or excessive unmanned aerial vehicle flights may occur, leading to instances of UAV-induced air disruption, and causing serious negative effects [4]. UAV monitoring is therefore necessary. Existing anti-UAV methods at airports include radio, acoustic, and radar detection. These approaches primarily provide target localization, but they are susceptible to environmental noise, strong electromagnetic interference in airport settings, and adverse weather, which degrades detection accuracy.

Image-based sensing is immune to acoustic and radio-frequency noise and, beyond determining target presence, can provide semantic cues such as shape and texture, thereby offering a viable solution for anti-UAV detection in airport environments. In recent years, computer vision technology has advanced rapidly, particularly in the domain of deep learning-based object detection. The available algorithms for deep learning-based object detection can be divided into two categories: two-stage detection and one-stage detection.

The two-stage detection algorithm divides the detection problem into two stages [5]: candidate region extraction and region classification, including algorithms such as R-CNN [6], SPP-Net [7], Fast R-CNN [8], and Faster R-CNN [9]. For one-stage detection algorithm, the You Only Look Once (YOLO) series are commonly used [10]. Instead of extracting of regions of interest (ROIs), the algorithm unifies image classification and localization as a regression problem. Combining the anchor idea of R-CNN with the regression idea of YOLO, the Single Shot Multi-box Detector (SSD) was proposed [11].

The application of deep learning algorithms in anti-UAV detection has gained momentum. Existing research has shown that single-stage algorithms demonstrate superior detection performance in anti-UAV detection tasks. Zhao [12] proposed a simple fusion algorithm by integrating detection into tracking for the UAV tracking task. Experiment results show that the tracking performance is improved significantly. Shi [13] compared the detection performance of YOLOv4, YOLOv3, and SSD for small UAV targets. The research findings indicate that YOLOv4 exhibits the most optimal detection efficacy. Hu [14] enhanced the YOLOv3 model for anti-UAV detection by utilizing the last four scales of feature maps instead of the last three scales. This modification enables the model to capture additional texture and contour information, enhancing its ability to detect small objects. Current studies have made advancements in algorithmic detection performance. However, the image-based object detection algorithms suffer from limited dataset quantity and weak model generalization capability, leading to limited detection accuracy for UAVs. Zhu [15] proposed YOLOv9-CAG, a multimodal UAV detection framework that fuses visible, infrared, and audio inputs and upgrades YOLOv9 with CAM, GAM, and AKConv modules, yielding markedly higher mAP and recall across datasets and real-world videos compared to the baseline YOLOv9.

Although these methods improve the model at the algorithmic level, anti-UAV detection in airport environments faces multiple challenges, including extreme weather, overexposure under intense sunlight, and image degradation caused by rapid UAV motion. Furthermore, the detection of UAVs at airports based on object detection requires a large amount of data to train the model. Currently, there is a lack of UAV data collection platforms, and it is difficult to collect UAV datasets under various weather conditions. Single-scene dataset may lead to poor performance of the model in other scenes and inferior generalization performance.

Style transfer techniques use image generation networks to transform input images while preserving their content and changing their style to a predefined one [16]. Li [17] mitigated data imbalance in bridge crack detection by proposing Tiny CycleGAN and Multi-CycleGAN with spectral normalization to synthesize paired labels and augment from unpaired labels. This strategy improved segmentation performance. Arezoomandan [18] proposed a two-phase data augmentation pipeline that first generates synthetic drone images in Unreal Engine and then translates them to a realistic style with CycleGAN. This pipeline was used to train YOLO models, and the results show significant improvements on multiple real-world datasets. In conclusion, STL algorithm provides a solution for multi-scene simulation.

This study proposes an anti-UAV detection method for different airport environments, which processes training data through style transfer learning to train YOLOv5 and is denoted as YOLOv5_STL for descriptive convenience. The CycleGAN network [19] is used to generate different scene datasets with small UAV targets under foggy, overexposed, and blurred conditions, taking into account the effects of lighting, noise, and weather changes. Different YOLO models including YOLOv5 [20], YOLOv6 [21], YOLOv7 [22], and YOLOv8 are used for detection. Moreover, comparative analyses are conducted to select the most suitable algorithmic model for this detection task.

The research framework is outlined as follows and illustrated in Figure 1. (1) Monitoring platform: cameras are deployed on both sides of airport runways to establish a UAV monitoring platform. (2) Multi-scenario data generation: three generative adversarial networks (GANs) are compared, and the CycleGAN network is used to generate three scene datasets that simulate foggy, overexposed, and blurred airport backgrounds. (3) Model training and comparison: YOLOv5, YOLOv6, YOLOv7, and YOLOv8 are trained on the simulated scene datasets, and their detection performance is analyzed and compared. (4) Validation on open datasets and field data: the proposed approach is evaluated on both open datasets and on-site detection data.

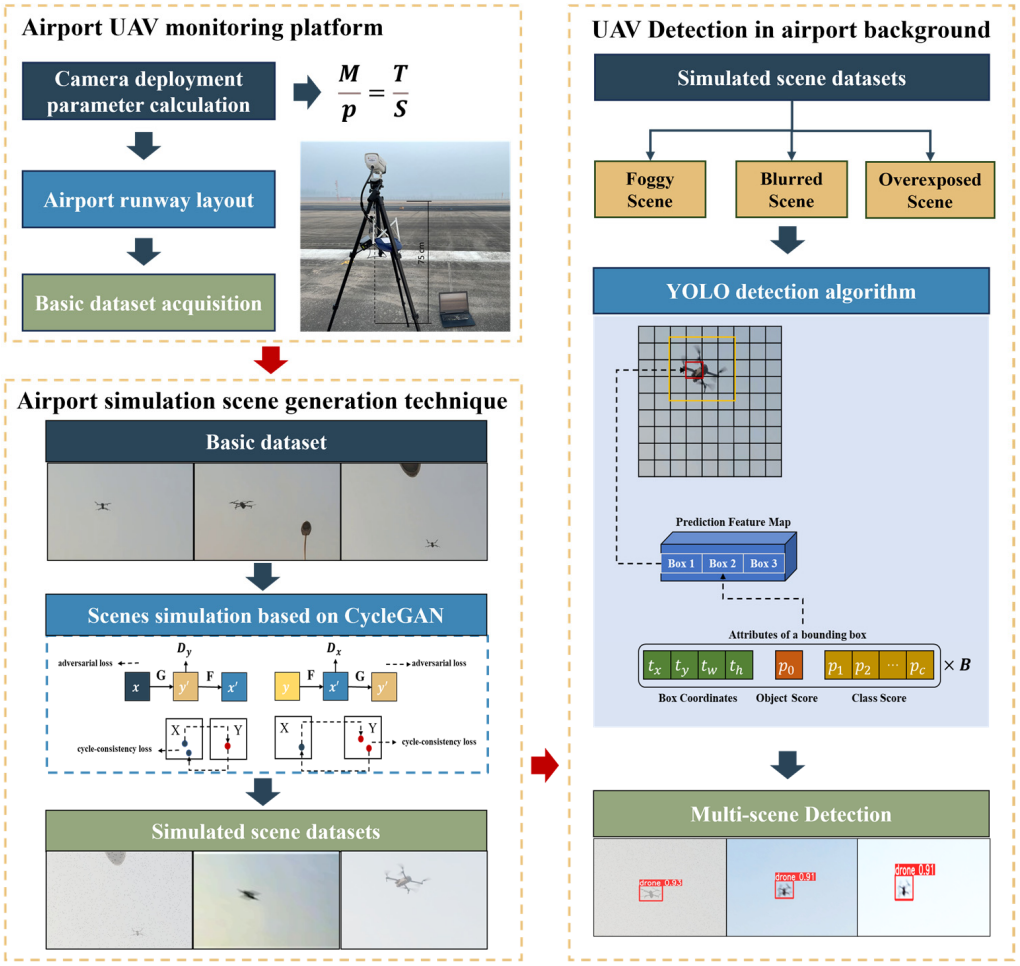


Figure 1. The framework of multi-scene anti-UAV detection method.

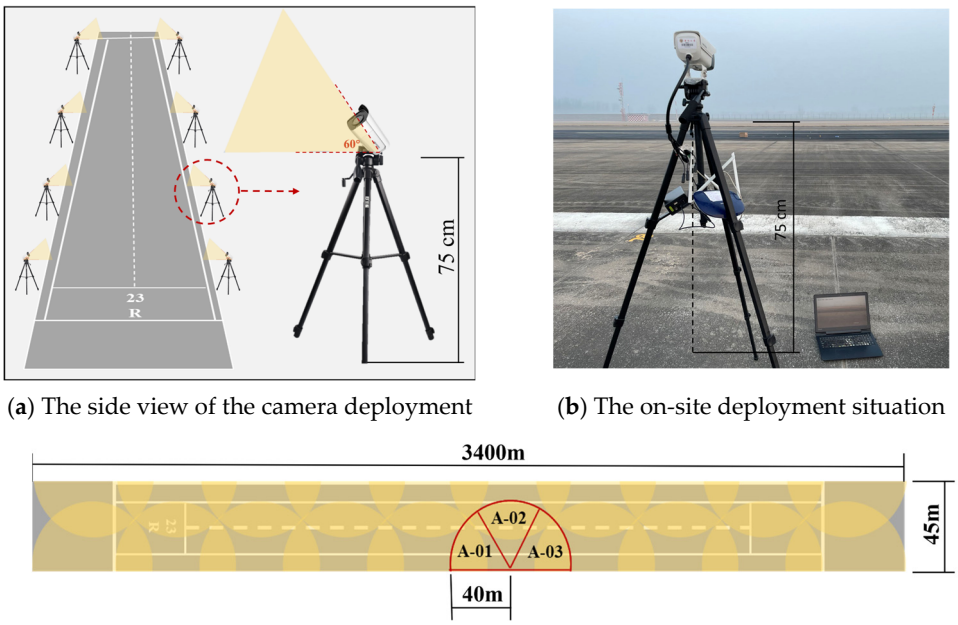
2. Establishment of a UAV Monitoring Platform

A UAV monitoring platform was established to guarantee airport flight safety. Photographic devices were selected as image acquisition devices for monitoring the airspace conditions by placing photographic units on both sides of the airport runway. Each unit comprised three cameras equipped with infrared assistance to enable continuous shooting during nighttime. These units can monitor a 180° range on one side of the runway, with each camera covering a 60° range. Overlapping monitoring areas were established between adjacent photographic units to improve accuracy.

$$\frac{M}{p} = \frac{T}{S} \tag{1}$$

where M represents the minimum number of pixels required for detection, p represents the number of pixels in the camera device, T represents the minimum size of the UAV, S represents the detection area, $S = \pi \times \frac{l^2}{4}$, and l represents the diameter of photographic unit.

The distance can be calculated by Equation (1). The cameras were positioned at a height of 75 cm with an inclination angle of 60°. To ensure accurate UAV recognition, a camera with 3840 × 2160 pixels was selected. The UAV should occupy a minimum of 8 × 8 pixels in the captured image. The detection range of each photoelectric unit is approximated as a semicircle. Taking into account the overlap of the capture area and the parameters of the airport runway, the coverage area formed by each photoelectric unit was set as a semicircle with a radius of 40 m. The specific placement is shown in Figure 2.



(**a**) The side view of the camera deployment (**b**) The on-site deployment situation

(**c**) The top-down view of the camera deployment

Figure 2. Equipment layout of airport UAV monitoring platform.

3. Airport Simulation Scene Generation Technique Based on CycleGAN

The experiments were conducted from January to February 2025 at Xuzhou Airport in China. Using frame extraction from video streams captured by the UAV monitoring platform under clear-sky conditions, a baseline dataset of 1143 images was created. To ensure robust anti-UAV detection across diverse scenes, a simulation scene generation technique based on CycleGAN was developed, and three additional datasets were constructed for fog, overexposure, and blur, each containing 1143 images.

3.1. Airport Simulation Scene Generation Based on CycleGAN

To achieve multi-scene detection, this study proposed an image augmentation technique based on CycleGAN to simulate various airport environments. CycleGAN, WGAN [23], and InfoGAN [24] were compared for the task of style transfer. Wasserstein Generative Adversarial Network (WGAN) introduces the Wasserstein distance as the loss function for GAN. By minimizing the Wasserstein distance between the generated distribution and the real distribution, the generated samples approach the real distribution [25]. Based on feature decoupling, InfoGAN achieves feature clarification and normalization. By learning independent features, it generates images with distinct styles. Ultimately, CycleGAN was selected for style generation. The CycleGAN network comprises three components: a generator network, a discriminator network, and a loss network. The structure diagram of the network is shown in Figure 3.

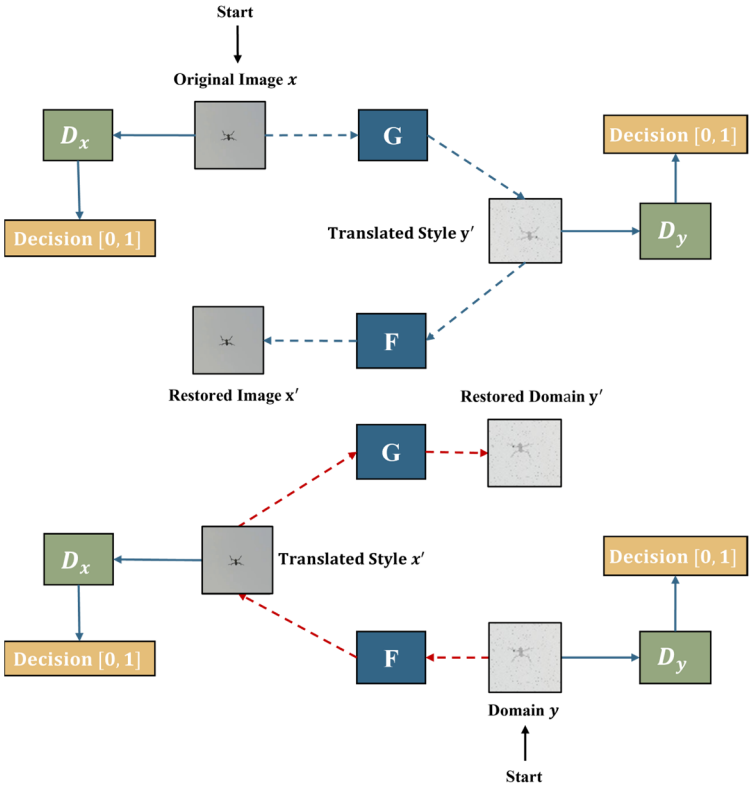


Figure 3. Principle diagram of CycleGAN network.

3.1.1. Principle of CycleGAN

The network learns mapping functions between two domains X and Y given training samples $\{x_i\}_{i=1}^{N}$, where $x_i \in X$ and $\{y_i\}_{j=1}^{M}$, where $y_j \in Y$. The model consists of two generative networks, where generator G transforms the X style into the Y style, i.e., $G(x) = y\prime$, $x \in X$, and generator F transforms the Y style back into the X style, i.e., $F(y) = x\prime$, $y \in Y$. At the same time, the model introduces adversarial discriminators D_x and D_y, where D_x aims to distinguish between images $\{x\}$ and translated style $\{x\prime\}$; D_y aims to discriminate between $\{y\}$ and $\{y\prime\}$.

The generative networks [26] adopt feed-forward transformation networks which use perceptual loss functions that depend on high-level features from a pre-trained loss network. This network contains 3 convolutions, several residual blocks [27], 2 fractionally strided convolutions with 1/2 stride, and 1 convolution that maps features to RGB. The discriminator networks adopt PatchGANs [28–30], which have fewer parameters than a full-image discriminator and can work on arbitrarily-sized images in a fully convolutional fashion [28].

3.1.2. Generation of Three Simulated Scene Datasets

Using video sequences can realize target tracking. By extracting contextual information from the video, the prediction of target motion states and positions is achieved. However, it results in higher computational demands and extended training durations. Moreover, the incorporation of temporal information may cause susceptibility to noise or tracking errors, especially in challenging scenarios characterized by occlusion or abrupt motion changes.

This paper focuses on addressing the challenge of limited datasets through the application of STL to simulate diverse environmental conditions. Therefore, a novel multi-scenario object detection method was proposed.

Based on the video sequences captured in Section 2, the basic dataset was obtained using frame sampling technique. The open-source datasets O-HAZE dataset [31], Exposure-

Errors dataset [32], and GoPro dataset [33] were collected. The O-HAZE dataset contains outdoor images with varying levels of fog concentration, the Exposure-Errors dataset consists of overexposed outdoor images, and the GoPro dataset comprises outdoor images with blurriness. The basic dataset acquired from the airport is denoted as the original image x in the CycleGAN network structure, while the three open-source datasets serve as the target domains y to be generated.

Each dataset was trained for 1000 epochs. The networks were trained from the beginning, using a learning rate of 0.0002. The learning rate remained constant for the initial 500 epochs and was then linearly decreased to zero over the next 500 epochs.

The generated simulated scene datasets are illustrated in Figure 4. From the figures, it can be observed that in the foggy scene, the visibility of the UAV is reduced, and the image contrast is decreased, presenting a bluish or greyish tone. In the blurry scene, details such as the UAV's outline are lost, resulting in unclear image edges and a considerable amount of noise. In the overexposed scene, the bright areas lose all detail, leading to the loss of critical information about the UAVs.

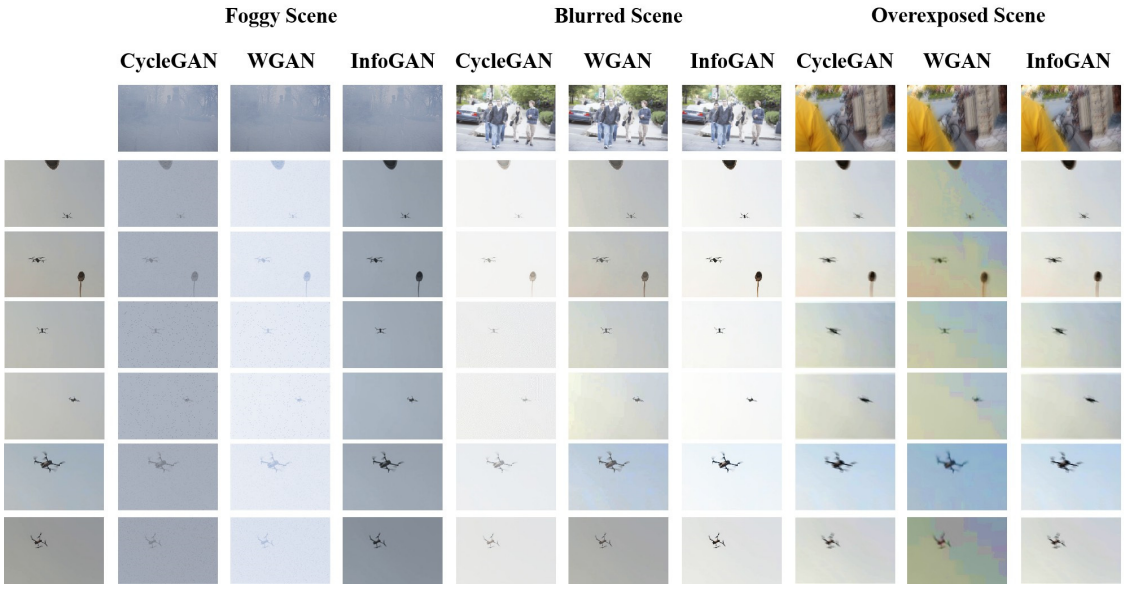


Figure 4. Three simulated scene datasets: foggy scene, blurred scene, overexposed scene.

3.2. Quality Evaluation of Simulated Scene Datasets

To evaluate the quality of the generated images, this study adopted PSNR and SSIM metrics under the RGB channels [34,35]. By calculating the PSNR and SSIM values between the simulated scene images and the transfer style, as well as between the initial UAV images and the transfer style, the values of both were compared.

PSNR [36] is a metric based on the errors between corresponding pixels in two images. It is the logarithmic ratio of the mean squared error between the original and processed images to $2^n - 1$ (the square of the maximum signal value, where n is the number of bits per sample).

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{2}$$

where MAX_I^2 represents the maximum value of the image, and MSE denotes the mean square error of the image, as shown in Equation (3).

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{m-1}[I(i,j) - K(i,j)]^2 \tag{3}$$

where I represents a clean image of size $m \times n$, K represents a noisy image of size $m \times n$, i represents the i-th pixel, and j represents the j-th pixel.

SSIM [37] is used to evaluate the similarity between the original and processed images. It mainly evaluates the luminance, contrast, and structure of the two images.

$$SSIM(x,y) = \left[l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot (x,y)^{\gamma} \right] \tag{4}$$

where $l(x,y)$ denotes luminance comparison, $c(x,y)$ represents contrast comparison, and $s(x,y)$ signifies structural comparison, as shown in Equations (5)–(7).

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{5}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{6}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{7}$$

where μ_x represents the mean of x, μ_y denotes the mean of y, σ_x represents the variance of x, σ_y signifies the variance of y, and σ_{xy} denotes the covariance between x and y. c_1, c_2 and c_3 are constant values introduced to avoid division by zero. In practical engineering calculations, it is common to set $\alpha = \beta = \gamma = 1$, and $c_3 = c_2/2$.

The principle of SSIM is based on the comparison of brightness, contrast, and structural information between images. When the SSIM value approaches 1, it indicates a higher similarity to the original image. The principle of PSNR is based on the MSE between pixels in images. During the image transformation process, adjustments to the luminance, color, and structure of images may result in noise amplification and distortion in the generated images. Therefore, a higher PSNR indicates better quality.

The image quality metrics are presented in Table 1. In general, CycleGAN achieves the best performance among the compared models (18.7436, 18.5922, 20.8549), while WGAN shows the poorest quality, with pronounced color distortions and noticeable degradation in the generated images.

Table 1. Quality comparison of images generated by different GAN models.

Method	Images	Fog	Blur	Overexposure
Mean PSNR	Initial UAV images	20.5428	21.0485	23.6723
	Model I CycleGAN [19]	18.7436	18.5922	20.8549
	Model II WGAN [23]	17.7436	16.5922	15.5922
	Model III InfoGAN [24]	18.4273	18.7983	19.9579
Mean SSIM	Initial UAV images	0.7515	0.7355	0.7183
	Model I CycleGAN [19]	0.9655	0.7983	0.9281
	Model II WGAN [23]	0.7721	0.6491	0.6278
	Model III InfoGAN [24]	0.9232	0.8283	0.9027

Furthermore, it can be observed that the simulated scene images exhibit higher SSIM values and demonstrate a greater resemblance to the transfer style, as compared to the initial UAV images. The PSNR of the simulated scene images is lower compared to the initial UAV images. However, the difference in PSNR values between the two is relatively small, indicating a relatively low degree of loss.

4. Anti-UAV Detection Experiment Setup

4.1. Deep Learning Object Detection Algorithms

Two-stage object detection algorithms, which are also known as region-based detection methods, divide the detection problem into two stages: region proposal and object classification. Although two-stage algorithms have significantly improved detection performance, they cannot meet real-time requirements [38]. Therefore, an end-to-end algorithm was proposed. This algorithm does not require ROIs extraction and instead unifies classification and localization as a regression problem on the image. Compared to two-stage methods, the one-stage algorithm has a faster detection speed but lower detection accuracy [39]. Representative algorithms include the YOLO series. The target detection model structure is shown in Figure 5.

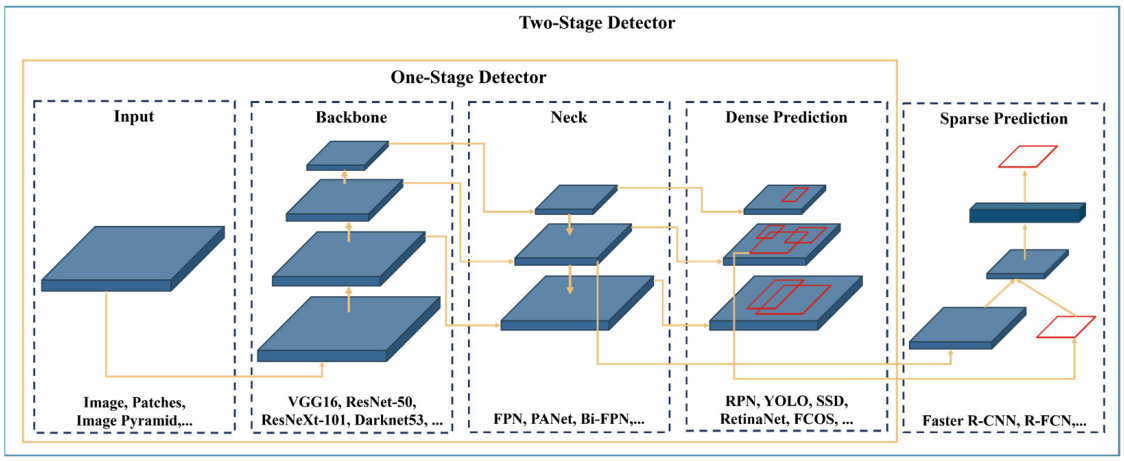


Figure 5. Object detection model structure based on deep learning.

For real-time anti-UAV detection, this paper adopted the YOLO series algorithm. The YOLO algorithm incorporates Feature Pyramid Networks (FPN) for multi-scale feature extraction, effectively capturing small objects. It widely employs data augmentation techniques such as random cropping, color transformation, and mirroring. By adopting a multi-scale training strategy, it ensures the precision of small object detection. As a single-stage object detection algorithm, it guarantees real-time detection, meeting the requirements of drone detection tasks. Therefore, four algorithms, YOLOv5, YOLOv6, YOLOv7, and YOLOv8, were used to detect small UAV targets in different airport scenes. Table 2 presents a comparison of four YOLO algorithmic network architectures.

Through comparative experiments, the optimal model YOLOv5 was obtained. The model architecture diagram of YOLOv5_STL is illustrated in Figure 6. YOLOV5_STL first applies a style transfer model to map raw UAV images to distributionally matched samples for multiple target scenes. The original data and the style-transferred data are then used jointly for training, which enhances robustness to distribution shifts across scenes. Compared with the baseline YOLOv5, YOLOV5_STL expands the effective training distribution through style transfer and improves the detectability and stability of small targets under fog, low illumination, and overexposure. The approach emphasizes domain generalization from the data-source perspective, and the detection head can be interchanged according to task requirements without altering the overall framework.

Table 2. Difference in architectural components of the YOLO models.

Model	Anchor	Data Augmentation	Backbone	Neck	Predict	Param (M)
YOLOv5 [20]	Anchor based	Mosaic	CSPDarknet-53, SPP, Focus, CSP	FPN + PAN	Classification loss: BCE, Bounding box regression loss: CIoU	21.2
YOLOv6 [21]	Anchor free	Mosaic augmentation was disabled for the final 10 epochs	CSPDarknet-53	Rep-PAN, CSP-Block in YOLOv5 was replaced by The RepBlock	Classification loss: VFL Loss, Regression loss: DFL Loss + CIOU Loss, Efficient Decoupled Head	34.9
YOLOv7 [22]	Anchor based	Mosaic	Darknet-53	SPP, PAN	CloU_Loss, DIOU_Nms, Auxiliary head	36.9
YOLOv8	Anchor free	Mosaic augmentation was disabled for the final 10 epochs	Darknet-53	PAFPN	Classification loss: VFL Loss, Regression loss: DFL Loss + CIOU Loss, DIOU_Nms	25.9

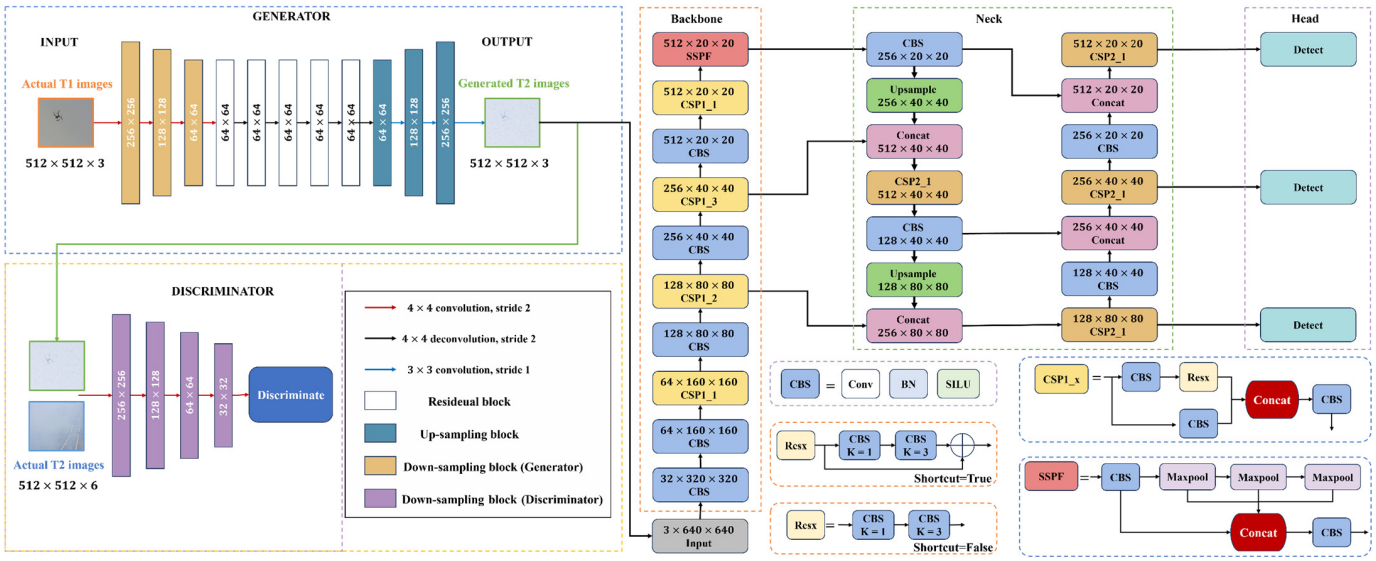


Figure 6. Architecture of YOLOv5_STL.

4.2. Test Environment and Parameter Setting

The experimental computer hardware configuration for this study includes a 64-bit Ubuntu 20.04 operating system, PyTorch 1.11.0 deep learning platform, Python 3.8, CUDA 11.3, GPU RTX 3090 (24 GB), and a CPU with 14 vCPUs Intel(R) Xeon(R) Gold 6330 CPU@2.00 GHz.

The images used in this study have a size of 1920 × 1080 pixels. The datasets are randomly divided into training, validation, and testing sets with a ratio of 8:2:2. For each simulated scene, there are 881 images in the training set, 262 images in the validation set, and 278 images in the test set. The test set consists exclusively of original real images without any CycleGAN-enhanced data or simulated images.

The initial learning rate was set to 0.002, and a cosine annealing schedule was employed to adjust the learning rate. The momentum was set to 0.9, weight decay was set to

0.0005, the number of training iterations was set to 400, and the batch size was set to 16. The learning rate was reduced to 1/10 of the original value at iterations 200 and 300.

4.3. Evaluation Metrics

Interaction over union (IoU)

IoU measures the overlap between the predicted bounding box and the ground truth bounding box, which is the ratio of their intersection to their union.

$$IoU = \frac{Area(B_p \cap B_g)}{Area(B_p \cup B_g)} \tag{8}$$

where B_p is the predicted bounding box, and B_g is the ground truth bounding box.

In object detection tasks, the IoU threshold is set to 0.5. TP (True Positive) refers to the number of detection boxes whose IoU with the ground truth box is greater than 0.5; FP (False Positive) refers to the number of detection boxes whose IoU with the ground truth box is less than 0.5; FN (False Negative) refers to the number of missed ground truth boxes.

Recall and precision

Recall refers to the proportion of correctly predicted positive samples out of the total positive predictions made by the model.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

Precision refers to the proportion of correctly predicted samples among all the predicted samples.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

AP (Average Precision)

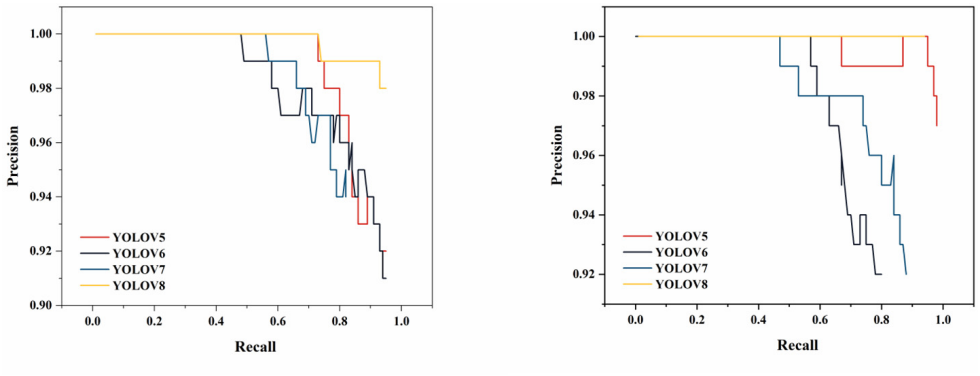
Average Precision (AP) is the area under the Precision-Recall (PR) curve, which is formed by plotting Precision against Recall, with Recall on the horizontal axis and Precision on the vertical axis.

5. Result and Analysis

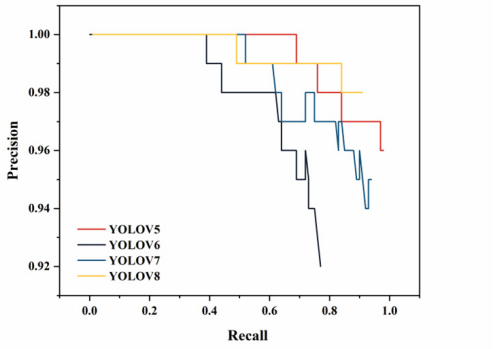
5.1. Performance Comparison and Analysis of 4 Detection Models

In this experiment, single object detection was conducted, and other aerial objects and birds were treated as background. Because missed detections can affect airport operations, the study prioritizes recall while maintaining competitive AP. Accordingly, the IoU threshold was set to 0.8. To satisfy real-time detection requirements, models with higher FPS are preferred.

Using the best training weights, predictions were made on the test set, and Average precision was used to evaluate overall model performance. Figure 7 shows the P-R curves for different scenes using different models. It can be seen from Figure 7 that the P-R curves of YOLOv5 and YOLOv8 cover a larger area than the curves of YOLOv6 and YOLOv7, indicating that YOLOv5 and YOLOv8 demonstrate comparable outstanding performance and achieve higher AP values, while YOLOv6 exhibits the poorest performance. YOLOv6 focuses on model lightweighting to improve detection speed. By reducing the number of feature layers, YOLOv6 fails to adequately extract detailed information from small targets. Furthermore, employing ReLU as the activation function can lead to neuron deactivation and consequently a decline in detection accuracy as the network depth and width expand. Additionally, YOLOv6 utilizes single-frame prediction, resulting in relatively lower detection accuracy in scenarios where target boundaries are unclear or objects overlap. As a result, it exhibits subpar performance in both blurred and overexposed scenes.



(**a**) Foggy scene (**b**) Blurred scene



(**c**) Overexposed scene

Figure 7. Precision–Recall curves of YOLO models on 3 simulated scene datasets.

The comparison of the performances of the various models on the prediction dataset is shown in Table 3 and Figure 8. It can be observed that the YOLOv8 model has the best AP values (96.95%, 98.9%, 97.47%) with the lowest FP numbers (5, 2, 4) among all the models. YOLOv5 performs well, with AP values only slightly lower than YOLOv8, and the difference between the two is minimal. The Friedman test was performed on the calculated AP values, and the results indicate that $p_{value} = 0.04205 < \alpha = 0.05$, indicating a significant difference among the measurement results of multiple models. YOLOv5 utilizes Mosaic data augmentation and adaptive anchor box calculation at the input end to address the issue of large-scale variations in small UAV targets. It adopts the CSPDarknet53 backbone architecture to enhance feature extraction capabilities and implements the P5 neck and PANet path-aggregation neck, enabling the fusion of high-level and low-level features for improved object detection. YOLOv8 adopts similar data augmentation and training strategies as YOLOv5, with some improvements. Specifically, YOLOv8 replaces the C3 structure of YOLOv5 with the C2f structure, which features a richer gradient flow. In the Head section, YOLOv8 transitions from the coupled head to decoupled head structure. Regarding the positive and negative sample allocation strategy, YOLOv8 introduces a dynamic allocation strategy using the TaskAlignedAssigner, compared to the static allocation strategy of YOLOv5. This results in superior performance. However, the Anchor-Free approach is not as effective as the Anchor-Based approach, thus yielding comparable detection results for small UAV targets.

Table 3. Detection performances comparison of 4 YOLO Models.

Model Name		Average Precision (AP)	True Positives (TP)	False Negatives (FN)	False Positives (FP)	Precision Rate	Recall Rate	FPS
YOLOv5 [20]	Fog	93.95%	169	60	15	0.918	0.738	119
	Blur	98.09%	226	22	6	0.974	0.911	94
	Overexposure	97.07%	209	15	8	0.963	0.933	94
YOLOv6 [21]	Fog	93.89%	148	187	12	0.925	0.442	82
	Blur	91.72%	150	168	11	0.912	0.572	89
	Overexposure	90.36%	178	90	18	0.908	0.664	94
YOLOv7 [22]	Fog	94.39%	155	284	10	0.939	0.353	88
	Blur	92.79%	155	127	13	0.923	0.550	100
	Overexposure	93.73%	182	81	23	0.888	0.692	93
YOLOv8	Fog	96.95%	188	69	5	0.974	0.732	86
	Blur	98.9%	195	56	2	0.989	0.777	84
	Overexposure	97.47%	184	53	4	0.979	0.776	85

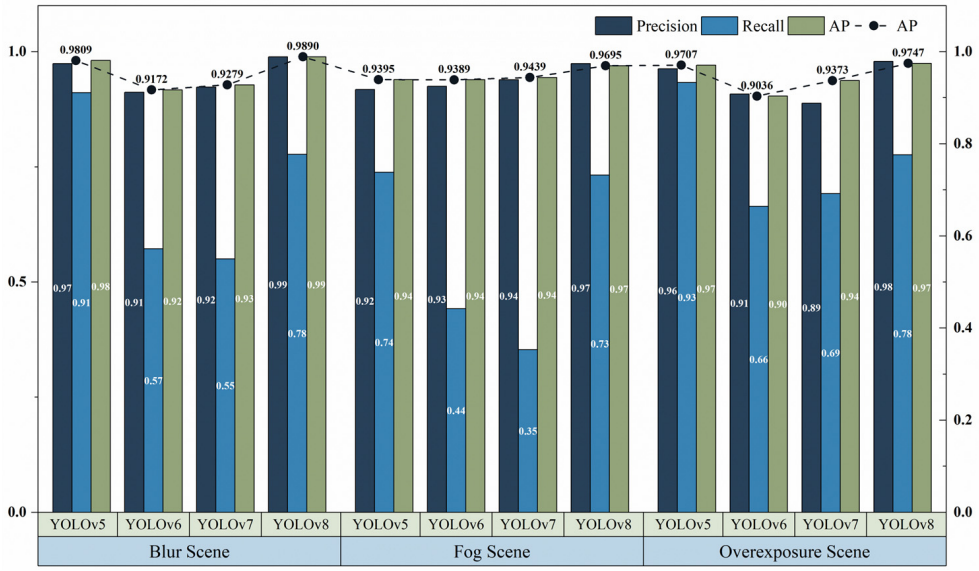


Figure 8. Comparison of evaluation metrics of 4 YOLO models on 3 simulated scene datasets.

Considering the purpose of airport UAV monitoring, a larger number of false negatives (FN) indicates a greater impact of undetected UAVs on airport operations. This detection task emphasizes the importance of minimizing missed detections rather than false alarms. Other flying objects and birds can equally impact airport flight safety. Their misidentification does not affect the detection result. Hence, it is crucial to have superior performance in terms of the recall rate metric, indicated by a lower FN number. A comparison of the recall values of the models is presented in Figure 8. It can be observed that YOLOv5 achieves the highest recall rates in three scenes (0.738, 0.911, 0.933), while YOLOv7 exhibits the lowest recall rates (0.353, 0.550, 0.692). Performing the Friedman test on the recall rate, the results indicate that $p_{value} = 0.04205 < \alpha = 0.05$, indicating a significant difference among the measurement results of multiple models. Considering the real-time requirements of UAV monitoring in airports, in addition to the detection accuracy metric, the model's inference time needs to be evaluated. It can be observed that YOLOv5 can process the highest number of images per second among all the models (119, 94, 94). On the contrary, the FPS of YOLOv8 is the lowest with 86, 84, 85.

YOLOv6 and YOLOv7 reduce the number of feature layers, resulting in inadequate extraction of detailed information from small targets. The lack of effective data augmentation strategies lead to a decrease in model performance and accuracy. Additionally, YOLOv6 employs single-frame prediction, which fails to adapt to the detection requirements of object transformations in various scenes, affecting the learning ability and detection performance of models.

Based on the aforementioned results and model analysis, YOLOv5 is the optimal choice as the UAV detector in the proposed framework due to its high AP value (with high precision and recall rates) and relatively faster model inference time.

YOLOv5 utilizes CSPDarknet-53 as its backbone network which exhibits better feature extraction capabilities and increased receptive field compared to traditional Darknet networks. The backbone network employs the CSP (Cross Stage Partial) structure, which divides the input feature map into two branches. One branch performs convolution operations while the other branch is connected through residual connections, enhancing the enhanced representation of features [40].

Moreover, YOLOv5 incorporates channel attention mechanism, which adaptively adjusts the weights of different channels to highlight the ones that contribute more to object detection. This reduces redundant computations and improves the effectiveness of features and the computational efficiency of the network. In terms of data augmentation strategies, YOLOv5 incorporates a series of techniques, including random scaling, random cropping, and color jittering, enhancing the ability to recognize objects under diverse scales, angles, and lighting conditions.

Therefore, in diverse scenarios involving changes in features, YOLOv5 achieves a faster detection speed while ensuring the preservation of detection efficacy.

5.2. Validation on Open Datasets and Field Test Data

To confirm the validity of the proposed method, YOLOv5_STL was compared with small object enhancement algorithms on public datasets, as shown in Table 4. Dataset 1 and Dataset 3 comprise UAV datasets of outdoor environments, encompassing complex backgrounds such as buildings and forests. Dataset 2 comprises indoor UAV data. Improvements in small object detection involve multi-scale feature learning, data augmentation, and context-based detection. Based on these different enhancement measures, the small object detection algorithms R-FCN [41], RefineDet [42], and MPNet [43] were selected for comparison with YOLOv5_STL. The specific improvement methods and types are detailed in Table 5.

Table 4. Open datasets.

Dataset Number	Dataset
Dataset 1	Drone Dataset: Amateur Unmanned Air Vehicle Detection [44]
Dataset 2	Drone Detection Dataset [45]
Dataset 3	Anti-UAV Dataset [46]

Table 5. Small object detection algorithms comparison.

Method	Solution	Improvement Type
R-FCN [41]	Single feature map	Multi-scale feature learning
RefineDet [42]	Variants of FPN	Multi-scale feature learning
MPNet [43]	Local context	Context-based detection
YOLO26 [47]	Multi-scale aggregation	P3 branch
YOLOv5_STL	Style Transfer Learning	Data augmentation

The PR curves of the detection results are shown in Figure 9. It can be observed that the small-object detectors exhibit broadly similar behavior, with MPNet and R-FCN performing poorly on Dataset 3. The evaluation metrics are further computed as the area under the PR curves in Figure 10. YOLOv5_STL attains higher AP than YOLOv5, confirming the effectiveness of the proposed optimization. On Dataset 1 and Dataset 3, the AP of YOLOv5_STL also exceeds that of MPNet and R-FCN, which validates the method's effectiveness. In addition, RefineDet achieves the best AP values on the three datasets, namely 0.879, 0.608, and 0.819, which are slightly higher than those of YOLOv5_STL at 0.820, 0.400, and 0.784. On the newly collected dataset, YOLOv5_STL delivers performance comparable to the recent YOLO26, with differences within normal experimental variance, indicating that the gains observed here are primarily attributed to STL-driven domain generalization rather than the specific detector version. This result is consistent with the task characteristics: YOLO26 focuses on general-purpose multiscale aggregation, whereas long-range tiny-object detection at airports benefits most from diversified training data. Once STL broadens the data distribution, the choice of detector head plays a secondary role under the same latency budget, and the two models converge to similar accuracy while maintaining real-time efficiency.

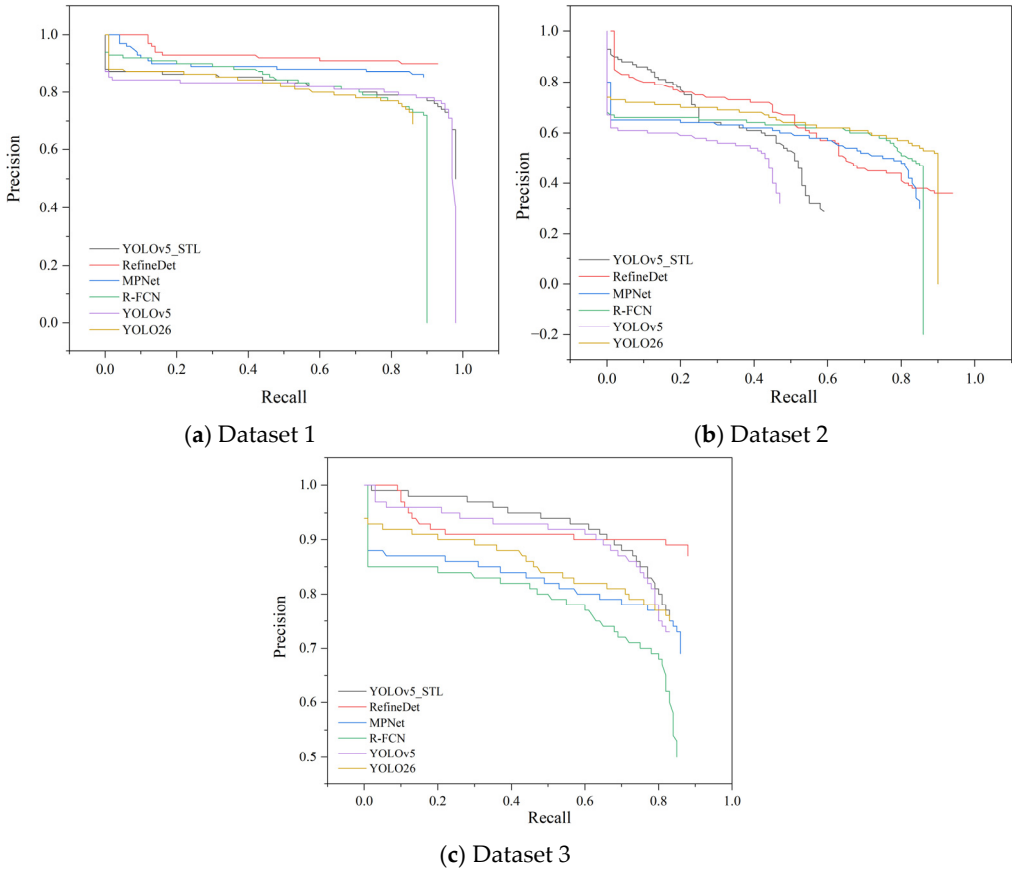


Figure 9. Precision–Recall curves of different models on open datasets.

Moreover, for airport anti-UAV detection tasks, it is crucial to ensure that no targets are missed while focusing on detection accuracy. Therefore, Recall is another key evaluation metric. As shown in Figure 10, although RefineDet has higher AP compared to YOLOv5_STL in Dataset 3, many UAV targets are missed, resulting in lower Recall across the three datasets. Overall, YOLOv5_STL demonstrates a well-balanced detection performance in all aspects. It is worth noting that in indoor scenarios (Dataset 2), where the features are significantly different from the training dataset, the detection performance is

relatively low. The proposed method is oriented to airport background and is not suitable for indoor scenarios. However, due to the improved data diversity, it ensures generalization capabilities on other datasets, and its detection performance still surpasses that of YOLOv5.

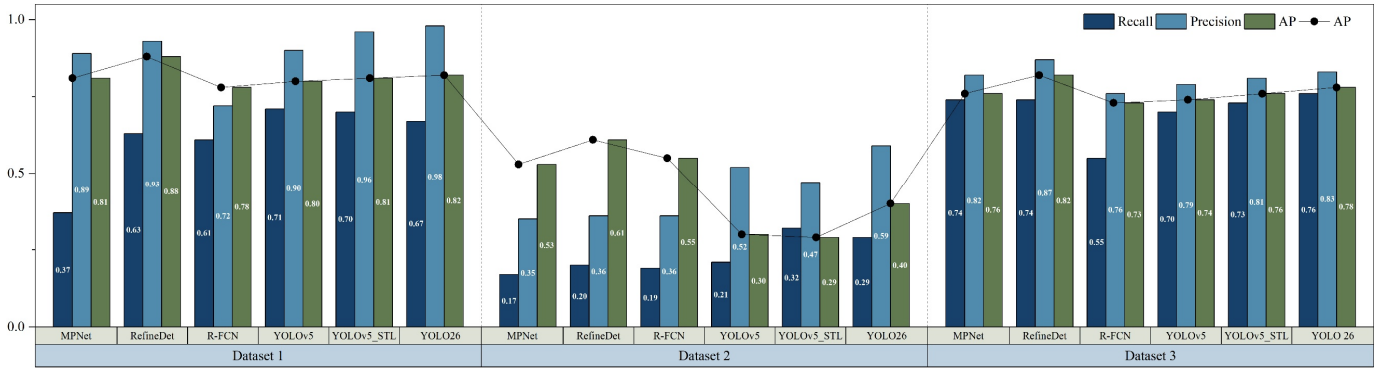


Figure 10. Comparison of evaluation metrics of small object detection algorithms on open datasets.

In conclusion, through comparison tests with other small target detection algorithms on UAV public datasets, it can be concluded that the integration of style transfer algorithms enriches the dataset, thereby enabling YOLOv5_STL to achieve higher generalization capabilities and superior detection performance.

To further verify the accuracy of the trained model, a field test was conducted at Xuzhou Guanyin International Airport in China. Monitoring cameras were deployed on both sides of the runway to capture surveillance videos, which covered multiple days and various weather conditions. Frames were extracted from the collected videos, and 1000 images were selected for performance evaluation. UAV object detection was then implemented on dedicated processing equipment. The layout of the field test is illustrated in Figure 11, and the detection performance results are presented in Figure 12. The test results show that the model achieved an AP of 95.37%, IoU values all greater than 0.8, and the total processing time for 1000 images was only 10.53 s. These findings demonstrate that the proposed monitoring method can accurately identify UAVs in different airport scenarios, providing effective technical support for ensuring flight safety.



Figure 11. Equipment setup for field test of anti-UAV detection.



(**a**) Foggy scene (**b**) Blurred scene (**c**) Overexposed scene

Figure 12. Anti-UAV detection result using YOLOv5_STL.

6. Conclusions

The prevention of UAV intrusion is an important task in airport operation management. This study employed the CycleGAN network to generate style transfer images for simulating various airport scenes. Additionally, a simple, efficient, and cost-effective anti-UAV detection method based on YOLOv5 was developed for airport surveillance. The findings of the present research study are summarized as follows:

(1) The UAV monitoring platform proposed in this study has demonstrated UAV surveillance capabilities. The proposed deployment scheme, which ensures full coverage of the anti-UAV detection range, has been proven to be an effective data acquisition approach.

(2) Based on a generative adversarial network, this study established the first simulated dataset of UAVs for various backgrounds at airports, which can be utilized for anti-UAV detection training.

(3) The research indicated that YOLOv5 exhibited the best prediction performance with high recall rates and relatively faster model inference time, achieving AP values of 93.95%, 98.09%, and 97.07% on three scenes.

(4) YOLOv5_STL demonstrates superior detection performance on open datasets, showcasing the efficacy of the proposed approach in augmenting the training dataset. This augmentation enhances the model generalization capabilities, enabling adaptability to diverse scenarios.

(5) During on-site testing, YOLOv5_STL achieved an AP value of 95.37%, which meets the surveillance requirement of airports.

This research demonstrates good robustness of the new method in anti-UAV detection under different airport backgrounds, but there are some limitations that should be further investigated.

This study only focuses on training the detection of small UAV targets in the airspace of airports, without considering anti-UAV detection in other backgrounds. Future research directions could concentrate on complex backgrounds to extend warning coverage. Multi-category differentiation could be explored in subsequent research to address classification needs. Additionally, the robustness of the model in real complex environments requires further validation with real-world data. And the potential for overfitting due to augmented data or the impact of data quality on the results could be further addressed.

Author Contributions: Conceptualization, H.C. and R.Z. (Ruiheng Zhang); methodology, R.Z. (Ruiheng Zhang) and Y.S.; software, R.Z. (Ruoxi Zhang) and Y.S.; validation, R.Z. (Ruiheng Zhang); formal analysis, R.Z. (Ruoxi Zhang) and H.C.; investigation, Y.S. and J.Z.; resources, Y.L.; data curation, Y.S. and H.C.; writing—original draft preparation, R.Z. (Ruiheng Zhang); writing—review and editing, R.Z. (Ruiheng Zhang); visualization, Y.L.; supervision, J.Z.; project administration, R.Z. (Ruoxi Zhang); funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Youth Fund Project of Civil Aviation Flight University of China (Project No.: 24CAFUC05020), and 2025 Sichuan Provincial Civil Aviation Flight Technology and Flight Safety Engineering Technology Research Center Project (Project No.: GY2025-13C).

Data Availability Statement: Please contact the corresponding author to request access to the data mentioned in the article, but note that it cannot be used for commercial activities.

Conflicts of Interest: The authors declare no conflicts of interest. The funder was responsible for the visualization and supervision in the article.

References

1. Zhu, J.; Zhong, J.; Ma, T.; Huang, X.; Zhang, W.; Zhou, Y. Pavement distress detection using convolutional neural networks with images captured via UAV. *Autom. Constr.* **2022**, *133*, 103991. [CrossRef]
2. Siddiqi, M.A.; Iwendi, C.; Jaroslava, K.; Anumbe, N. Analysis on security-related concerns of unmanned aerial vehicle: Attacks, limitations, and recommendations. *Math. Biosci. Eng.* **2022**, *19*, 2641–2670. [CrossRef] [PubMed]
3. Zhou, Y.; Rui, T.; Li, Y.; Zuo, X. A UAV patrol system using panoramic stitching and object detection. *Comput. Electr. Eng.* **2019**, *80*, 106473. [CrossRef]
4. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
5. Cheng, H.; Li, Y.; Zhang, R.; Zhang, W. Airport-FOD3S: A Three-Stage Detection-Driven Framework for Realistic Foreign Object Debris Synthesis. *Sensors* **2025**, *25*, 4565. [CrossRef]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
7. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *Ieee Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
8. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; Volume 28.
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
12. Zhao, J.; Zhang, J.; Li, D.; Wang, D. Vision-based anti-uav detection and tracking. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25323–25334. [CrossRef]
13. Shi, Q.; Li, J. Objects detection of UAV for anti-UAV based on YOLOv4. In Proceedings of the 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Weihai, China, 14–16 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1048–1052.
14. Hu, Y.; Wu, X.; Zheng, G.; Liu, X. Object detection of UAV for anti-UAV based on improved YOLO v3. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 8386–8390.
15. Zhu, J.; Rong, J.; Kou, W.; Zhou, Q.; Suo, P. Accurate recognition of UAVs on multi-scenario perception with YOLOv9-CAG. *Sci. Rep.* **2025**, *15*, 27755. [CrossRef]
16. Zhong, J.; Huyan, J.; Zhang, W.; Cheng, H.; Zhang, J.; Tong, Z.; Jiang, X.; Huang, B. A deeper generative adversarial network for grooved cement concrete pavement crack detection. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105808. [CrossRef]
17. Li, B.; Guo, H.; Wang, Z. Data augmentation using CycleGAN-based methods for automatic bridge crack detection. *Structures* **2024**, *62*, 106321. [CrossRef]
18. Arezoomandan, S.; Klohoker, J.; Han, D.K. Data augmentation pipeline for enhanced uav surveillance. In Proceedings of the International Conference on Pattern Recognition, Hammamet City, Tunisia, 25–27 September 2025; Springer: Berlin/Heidelberg, Germany, 2025; pp. 366–380.
19. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
20. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 2778–2788.
21. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
22. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
23. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
24. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2016**, *29*. [CrossRef]
25. Wang, S. A hybrid SMOTE and Trans-CWGAN for data imbalance in real operational AHU AFDD: A case study of an auditorium building. *Energy Build.* **2025**, *348*, 116447. [CrossRef]
26. Johnson, J.; Alahi, A.; Li, F.-F. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9906, pp. 694–711. [CrossRef]
27. Shafiq, M.; Gu, Z. Deep Residual Learning for Image Recognition: A Survey. *Appl. Sci.* **2022**, *12*, 8972. [CrossRef]
28. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
29. Li, C.; Wand, M. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9907, pp. 702–716. [CrossRef]
30. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 105–114. [CrossRef]
31. Jin, Y.; Yan, W.; Yang, W.; Tan, R.T. Structure representation network and uncertainty feedback learning for dense non-uniform fog removal. In *Asian Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 155–172.
32. Afifi, M.; Derpanis, K.G.; Ommer, B.; Brown, M.S. Learning multi-scale photo exposure correction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9157–9167.
33. Park, D.; Kim, J.; Chun, S.Y. Down-scaling with learned kernels in multi-scale deep neural networks for non-uniform single image deblurring. *arXiv* **2019**, arXiv:1903.10157.
34. Huang, S.-C.; Yeh, C.-H. Image contrast enhancement for preserving mean brightness without losing image features. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1487–1492. [CrossRef]
35. Li, L.; Liu, Z.; Li, Y. Modeling and Simulation of Image Quality Evaluation. *Comput. Simul.* **2012**, *29*, 284–287.

36. Kurban, R.; Durmus, A.; Karakose, E. A comparison of novel metaheuristic algorithms on color aerial image multilevel thresholding. *Eng. Appl. Artif. Intell.* **2021**, *105*, 104410. [CrossRef]
37. Ye, S.-N.; Su, K.-N.; Xiao, C.-B.; Duan, J.J.A.E.S. Image quality assessment based on structural information extraction. *Acta Electron. Sin.* **2008**, *36*, 856.
38. Wu, X.; Sahoo, D.; Hoi, S.C.H. Recent advances in deep learning for object detection. *Neurocomputing* **2020**, *396*, 39–64. [CrossRef]
39. Li, K.; Wang, X.; Lin, H.; Li, L.; Yang, Y.; Meng, C.; Gao, J. Survey of One-Stage Small Object Detection Methods in Deep Learning. *J. Front. Comput. Sci. Technol.* **2022**, *16*, 41–58.
40. Ren, S.; He, K.; Girshick, R.; Zhang, X.; Sun, J. Object Detection Networks on Convolutional Feature Maps. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1476–1481. [CrossRef]
41. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*. [CrossRef]
42. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.
43. Zagoruyko, S.; Zagoruyko, S.; Lerer, A.; Lin, T.Y.; Pinheiro, P.O.; Gross, S.; Chintala, S.; Dollár, P. A multipath network for object detection. *arXiv* **2016**, arXiv:1604.02135. [CrossRef]
44. Pawełczyk, M.Ł.; Wojtyra, M. Real World Object Detection Dataset for Quadcopter Unmanned Aerial Vehicle Detection. *IEEE Access* **2020**, *8*, 174394–174409. [CrossRef]
45. Svanström, F.; Englund, C.; Alonso-Fernandez, F. Real-time drone detection and tracking with visible, thermal and acoustic sensors. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Virtual, 10–15 January 2021; IEEE: Piscataway, NJ, USA, 2020; pp. 7265–7272.
46. Zhang, Z.; Wang, J.; Li, S.; Jin, L.; Wu, H.; Zhao, J.; Zhang, B. Review and Analysis of RGBT Single Object Tracking Methods: A Fusion Perspective. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *20*, 259. [CrossRef]
47. Sapkota, R.; Cheppally, R.H.; Sharda, A.; Karkee, M. Yolo26: Key architectural enhancements and performance benchmarking for real-time object detection. *arXiv* **2025**, arXiv:2509.25164. [CrossRef]

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