

SynthSwarm: A controllable synthetic dataset for UAV Swarm detection

Chaowen Zheng, Limin Liu, Luyi Zhang , Haojie Yang, Jianyu Liu, Qiang Fu, Qing Yang*, and Xiwei Guo*

Shijiazhuang Campus, Army Engineering University of PLA, Shijiazhuang 050003, PR China

Received 22 February 2026 / Accepted 7 May 2026

Abstract. Reliable detection of unmanned aerial vehicle (UAV) swarms is essential for airspace security and defense applications, however the scarcity of large-scale, densely annotated training data remains a critical bottleneck. Collecting real-world swarm data is costly, logistically challenging, and constrained by airspace regulations, while manual annotation of numerous small, fast-moving targets is time-consuming and prone to errors. To address these challenges, this paper presents SynthSwarm, a large-scale synthetic dataset specifically designed for UAV swarm detection in long-range aerial surveillance scenarios. The dataset is generated through a controllable simulation pipeline built on the Unity engine, enabling precise six-degree-of-freedom (6-DoF) pose specification for each UAV instance and automatic pixel-accurate bounding box annotation without manual labeling. SynthSwarm comprises 7000 high-resolution images (1920×1080) containing 31,542 UAV instances, with systematic variations in swarm density, formation patterns, target scale, and environmental conditions. Statistical analysis reveals that 67.3% of the targets qualify as small objects, reflecting the inherent difficulty of detecting distant UAV swarms. We benchmark several representative deep learning detectors, including one-stage detectors (YOLOX, YOLOv6, YOLOv12, YOLOv13), the two-stage detector Faster R-CNN, and the Transformer-based detector RT-DETR. Experimental results demonstrate that the dataset poses significant challenges for existing methods, particularly in high-density and small-target scenarios. Furthermore, cross-dataset on the MMFW-UAV dataset experiments validate the effectiveness of synthetic data as a pre-training source for improving detection performance on real UAV datasets. The dataset and generation pipeline are publicly available to facilitate further research in UAV swarm detection.

Keywords: Synthetic dataset, UAV swarm detection, Small target detection, Deep learning.

1 Introduction

Unmanned aerial vehicle (UAV) swarms have recently emerged as a key enabling technology in both civilian and defense sectors due to their high flexibility, scalability, and robustness [1]. Cooperative multi-UAV systems are increasingly deployed for applications such as large-area environmental monitoring [2], precision agriculture [3], search and rescue [4], and outdoor firefighting [5]. Compared to single-UAV scenarios, swarm operations involve more complex spatial formations, higher target densities, and more dynamic interactions among agents, significantly increasing the challenges of perception and situational awareness. In particular, reliably detecting multiple small UAVs in cluttered environments remains a difficult task [6]. Beyond these perception challenges, the lack of high-quality training data remains a critical bottleneck limiting

the robustness of data-driven detection algorithms for drone swarms. Publicly available UAV detection datasets either focus on single-UAV or small-scale multi-UAV scenes or contain limited variations in swarm size, formation patterns, and environmental conditions. For instance, multi-sensor and multi-view datasets such as MMFW-UAV [7] provide valuable resources for air-to-air vision tasks involving fixed-wing UAVs, but they mainly target single-platform perception and do not explicitly model swarm formations. Other works have also highlighted the limitations of existing datasets in addressing dense multi-UAV scenarios [8]. In real deployments, the 3D poses and spatial formations of UAV swarms are difficult to precisely control on demand, further hindering the systematic acquisition of diverse training examples, especially for dense or safety-critical configurations. Collecting real swarm UAV data at scale is expensive and logistically challenging, often subject to strict airspace regulations and safety considerations. Moreover, accurately annotating numerous small, fast-moving UAV targets in high-resolution videos is extremely

* Corresponding authors: Qing Yang, yang_qing@aeu.edu.cn; Xiwei Guo, hep0168@aeu.edu.cn.

time-consuming and prone to human error [9]. These factors make it difficult to obtain sufficiently large and diverse labeled datasets for training and evaluating modern deep learning-based detectors. To alleviate these limitations, we construct a synthetic UAV swarm detection dataset using a controllable simulation pipeline. The dataset provides rich variations in swarm configurations and environmental conditions while ensuring precise, automatically generated annotations. It can serve as a primary training source or as an auxiliary domain for methods combining real and synthetic data. The dataset and generation pipeline are publicly available at <https://github.com/marisinpiper/Synthetic-UAV-Swarm-Dataset>. The main contributions of this work can be summarized as follows:

We construct a large-scale synthetic UAV swarm dataset comprising 7000 high-resolution images (1920×1080) with pixel-accurate annotations, featuring systematic variations in swarm density, formation patterns, target scales, and environmental conditions.

We present a controllable simulation pipeline that enables precise 6-DOF pose specification for UAV swarms, effectively bypassing the difficulty of physical swarm coordination and the prohibitive cost of manual annotation.

We benchmark representative deep learning detectors on the proposed dataset and conduct cross-dataset experiments, demonstrating the effectiveness of synthetic data for UAV swarm detection.

2 Related work

2.1 Vision-based UAV detection

With the rapid proliferation of commercial and recreational drones, anti-UAV technologies have attracted increasing attention from both academia and industry. Existing systems typically comprise three key components – detection, tracking, and identification or classification – implemented using heterogeneous sensing modalities such as radio frequency (RF) signals, radar, acoustics, infrared (IR) and visible-light cameras, or their combinations in multi-sensor fusion frameworks. Among these modalities, vision-based approaches have become particularly prominent due to their relatively low deployment cost, rich semantic information, and compatibility with modern deep learning techniques.

Recent surveys provide comprehensive overviews of vision-based and multi-modal anti-UAV methods, and consistently highlight the challenges posed by small, fast-moving aerial targets in complex backgrounds. Wang et al. [10] systematically review vision-based anti-UAV techniques, summarizing deep-learning-based detection and tracking frameworks and emphasizing the difficulty of accurately localizing tiny UAV targets in cluttered scenes. Dong et al. [11] present a broader survey of anti-UAV systems, covering optical, radar, RF, infrared, acoustic, and multi-modal fusion methods, and benchmarking representative approaches on public datasets.

In the vision domain, many works adapt generic object detectors such as Faster R-CNN, SSD and YOLO [12–14] to UAV detection tasks by introducing multi-scale feature

fusion, feature pyramid networks, or attention mechanisms to better handle small targets. However, most existing benchmarks focus on single or sparsely distributed UAVs, and only a few recent studies consider more challenging scenarios such as dense swarms or multiple coordinated targets. For example, multi-sensor and multi-view datasets like MMFW-UAV provide valuable resources for air-to-air vision tasks involving fixed-wing UAVs, but they mainly target single-platform perception and do not explicitly model swarm formations. Overall, while the literature has made substantial progress on anti-UAV detection, no publicly available dataset explicitly addresses swarm scenarios with controllable formation patterns and systematic scale variations, leaving a critical gap for training robust multi-target detectors.

To provide a clearer picture of the current landscape, Table 1 presents a systematic comparison of representative existing UAV detection datasets alongside the proposed SynthSwarm dataset.

As shown in Table 1, existing UAV detection datasets predominantly focus on single-UAV or sparsely distributed scenarios and lack explicit swarm formation modeling. Notably, all three real-world datasets contain no swarm configurations, and their instance counts equal their image counts, indicating that each image contains at most one UAV target. In contrast, SynthSwarm contains an average of 4.5 UAV instances per image, explicitly modeling multi-UAV swarm formations. Furthermore, SynthSwarm adopts an automatic annotation pipeline, eliminating the time-consuming and error-prone manual labeling process required by all real-world counterparts. These characteristics make SynthSwarm uniquely suited for training and evaluating detectors in dense multi-target aerial surveillance scenarios.

Overall, while the literature has made substantial progress on anti-UAV detection, no publicly available dataset explicitly addresses swarm scenarios with controllable formation patterns and systematic scale variations, leaving a critical gap for training robust multi-target detectors.

2.2 Synthetic data and virtual dataset generation

Data-driven perception systems, especially deep learning-based detectors, typically require large-scale, diverse, and well-annotated datasets to achieve robust performance and generalization. However, as discussed in the Introduction, constructing such datasets for UAV swarm detection is particularly difficult due to safety constraints, regulatory restrictions, and the high cost of collecting and labeling real-world swarm data. These limitations have motivated a growing interest in using synthetic data and virtual environments to complement or partially replace real data for training and evaluation in aerial vision tasks.

Thanks to advances in computer graphics and game engines, highly realistic virtual scenes with controllable environmental conditions, sensor configurations, and objects can now be created at relatively low marginal cost. By leveraging simulation platforms and physically based rendering pipelines, researchers can generate large numbers of annotated images or video sequences while maintaining fine-grained control over object appearance, pose, motion

Table 1. Comparison of representative UAV detection datasets.

Dataset	Images	Instances	Swarm	Source
Det-Fly [15]	13,271	13,271	No	Real
DUT Anti-UAV [16]	10,000	10,109	No	Real
MMFW-UAV	147,417	147,417	No	Real
SynthSwarm (Ours)	7000	31,542	Yes	Synthetic

patterns, background complexity, illumination, and weather. Annotations such as bounding boxes, segmentation masks, depth maps, and optical flow can be obtained automatically and accurately, eliminating the need for time-consuming and error-prone manual labeling.

Several simulation platforms have been widely adopted in autonomous driving and aerial vision research. AirSim [17], developed by Microsoft, is a high-fidelity simulator specifically designed for aerial and ground vehicle perception, supporting hardware-in-the-loop simulation with customizable environments. UnrealCV [18] provides a plugin for Unreal Engine that enables the generation of synthetic images with ground-truth depth, surface normal, and semantic segmentation labels. These platforms have demonstrated that synthetic data generated from modern game engines can effectively supplement or even replace real data for training deep learning models, particularly when real data collection is expensive or impractical.

In aerial and remote-sensing applications, several works have demonstrated the effectiveness of synthetic imagery for target recognition and small-object detection. Nasir and Khurshid [19] propose a multiscale attention generator-discriminator framework to produce synthetic remote-sensing aircraft images and show that such synthetic data can significantly improve aircraft recognition performance. Patel et al. [20] develop a CGI-based synthetic data generation and detection pipeline for small objects in aerial imagery, combining synthetic training data with modern object detectors to enhance drone-based image recognition and small-object detection performance. These studies indicate that carefully designed synthetic aerial datasets can serve both as primary training sources and as auxiliary domains for pre-training or data augmentation, especially when real data are scarce or cover only a limited range of operating conditions.

These studies indicate that carefully designed synthetic aerial datasets can serve both as primary training sources and as auxiliary domains for pre-training or data augmentation, especially when real data are scarce or cover only a limited range of operating conditions.

A key challenge in leveraging synthetic data is the domain gap between virtual and real imagery. Differences in texture fidelity, lighting models, noise characteristics, and motion blur patterns can degrade detector performance when models trained on synthetic data are directly applied to real-world images. Common strategies to mitigate this gap include domain randomization [21], style transfer [22], and unsupervised domain adaptation [23]. In the context of UAV detection, understanding and addressing this domain gap is particularly important, as real-world UAV

imagery often exhibits complex atmospheric effects and sensor-specific artifacts that are difficult to fully replicate in simulation.

Despite these advantages, applying synthetic data generation to UAV swarm detection poses several unique challenges. First, realistic modeling of swarm requires coherent control over the relative positions, formations, and motion patterns of multiple UAVs, rather than treating each target as an independent object. Second, to faithfully reflect real-world operating conditions, the virtual environment must capture diverse backgrounds, complex clutter, and various camera viewpoints, including both ground-based and aerial sensors. Third, the synthetic swarm data should include a wide range of target scales and densities, from sparse formations to dense clusters, so that models can learn to detect numerous small UAVs that may occupy only a few pixels in high-resolution images.

In this work, we follow the general paradigm of synthetic data generation for aerial vision, but tailor the simulation and rendering pipeline specifically to the requirements of UAV swarm detection. By explicitly modeling swarm size and inter-UAV spacing, we generate a virtual dataset that emphasizes small-object and multi-target detection in cluttered environments. The automatically generated annotations provide dense, accurate labels for UAV instances in each image, offering a scalable and flexible resource for training, benchmarking, and transferring deep learning-based anti-UAV detectors to real-world swarm scenarios.

3 Dataset generation method

Our training data are generated synthetically in a controllable 3D simulation environment. As illustrated in Figure 1, UAV instances are sampled within a bounded 3D observation volume and projected onto the image plane using a pin-hole camera model. For each rendered frame, 2D bounding boxes are obtained automatically from the projected 3D bounds, providing pixel-accurate annotations without manual labeling. This framework enables systematic variation of swarm size, spatial density and camera configuration for dataset generation.

3.1 6-DOF state parameterization

We model each UAV instance as a rigid body with six degrees of freedom (6-DOF) in three-dimensional Euclidean space. A global right-handed world coordinate frame is defined, and all UAV poses are expressed with respect to

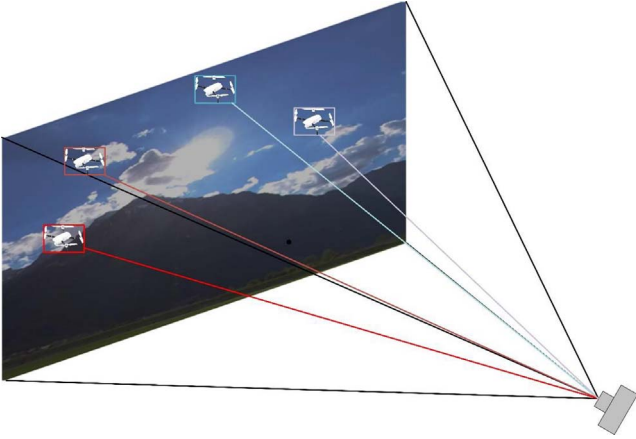


Fig. 1. Overview of the synthetic UAV-swarm dataset generation pipeline.

this frame. For UAV $i \in \{1, \dots, N\}$, the state is parameterized as:

$$\text{titles}_i = (p_i, r_i) \in \mathbb{R}^3 \times SO(3), \quad (1)$$

where the position vector $p_i = (x_i, y_i, z_i)$ specifies the 3D location within the observation volume $W \subset \mathbb{R}^3$, and the orientation $r_i = (\phi_i, \theta_i, \psi_i)$ denotes the Euler angles (roll, pitch, yaw) representing the UAV attitude with respect to the world frame.

The six degrees of freedom can be naturally decomposed into translational and rotational components. The components (x_i, y_i, z_i) correspond to translations of UAV i along the three orthogonal axes of the world frame, as illustrated in Figure 2. The angles $(\phi_i, \theta_i, \psi_i)$ represent rotations around these axes, describing roll, pitch and yaw motions of the UAV body frame, as depicted in Figure 3. Together, these quantities provide a complete and minimal description of the pose of each UAV in space.

For convenience, the full state of a UAV swarm with N agents is written as

$$S = \{s_1, s_2, \dots, s_N\} \in (\mathbb{R}^3 \times SO(3))^N. \quad (2)$$

This representation explicitly separates the per-UAV state into independent pose variables, enabling fine-grained control over both individual trajectories and collective formation structures. In contrast to simplified models that only consider 2D positions or planar motion, the 6-DOF formulation allows us to synthesize complex aerial maneuvers, including out-of-plane rotations, coordinated banking motions, and depth-varying formations, which are essential for faithfully emulating realistic UAV swarm behaviors in three-dimensional space.

Moreover, the use of Euler-angle parameterization $(\phi_i, \theta_i, \psi_i)$ is well aligned with the flight dynamics conventions adopted in most simulation engines and autopilot systems. This facilitates the integration of physically plausible motion primitives and makes it straightforward to map between high-level formation commands and low-level pose specifications in our data generation pipeline. As a result, we can systematically sample diverse yet physically

interpretable swarm configurations by directly manipulating the 6-DOF state variables s_i .

3.2 Trajectory generation and automatic annotation

Building upon the 6-DOF state representation, we generate image sequences by interpolating UAV poses along parameterized trajectories, and derive ground-truth annotations automatically via geometric projection. This subsection describes the trajectory interpolation scheme, the swarm configuration space, the camera model, and the annotation procedure in turn.

For each UAV i , we define initial and terminal states as:

$$s_i^{(0)} = (p_i^{(0)}, r_i^{(0)}), \quad s_i^{(T)} = (p_i^{(T)}, r_i^{(T)}), \quad (3)$$

where the superscripts denote the time indices at the beginning and end of the sequence. Given a total of T interpolation steps, the intermediate state at step $t \in \{0, 1, \dots, T\}$ is computed as follows. The position component is linearly interpolated in Euclidean space:

$$p_i^{(t)} = p_i^{(0)} + \frac{t}{T} \cdot \Delta p_i, \quad \Delta p_i = p_i^{(T)} - p_i^{(0)}. \quad (4)$$

For the orientation component, we employ spherical linear interpolation (SLERP) on unit quaternions to ensure smooth and consistent rotational motion. Let $q_i^{(0)}, q_i^{(T)} \in S^3$ denote the unit quaternion representations of the initial and terminal orientations, respectively. The interpolated orientation at step t is given by

$$\begin{aligned} q_i^{(t)} &= \text{SLERP}(q_i^{(0)}, q_i^{(T)}, \alpha_t) \\ &= \frac{\sin((1 - \alpha_t)\Omega)}{\sin \Omega} q_i^{(0)} + \frac{\sin(\alpha_t \Omega)}{\sin \Omega} q_i^{(T)} \end{aligned} \quad (5)$$

where $\alpha_t = t/T$ is the interpolation parameter and $\Omega = \arccos(q_i^{(0)} \cdot q_i^{(T)})$ is the angular distance between the two orientations. The quaternion result is then converted back to Euler angles $r_{i(t)} = (\phi_i^{(t)}, \theta_i^{(t)}, \psi_i^{(t)})$ for compatibility with the 6-DOF state representation.

This formulation ensures temporal coherence across consecutive frames, mathematically consistent rotational interpolation on $SO(3)$, and full reproducibility given identical boundary conditions.

For a swarm of N UAVs, the collective configuration at step t is defined as

$$S^{(t)} = \{s_1^{(t)}, s_2^{(t)}, \dots, s_N^{(t)}\}. \quad (6)$$

The swarm density is controlled by adjusting the observation volume W and the number of instances N . Let $V(W)$ denote the volume of the observation region; the average spatial density is given by

$$\rho = \frac{N}{V(W)}. \quad (7)$$

By varying ρ , we systematically generate scenes ranging from sparse formations to dense clusters. The apparent

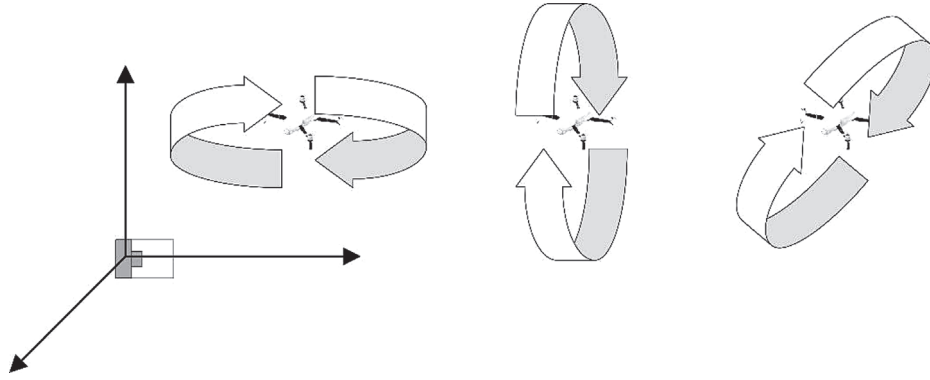


Fig. 2. The arrows indicate roll, pitch and yaw rotations about the body-fixed axes, corresponding to the attitude parameters $(\phi_i, \theta_i, \psi_i)$ in the 6-DOF state representation.

target scale in the image is governed by the depth component z_i ; smaller z_i yields larger bounding boxes, while larger z_i produces small-scale targets.

The imaging sensor is modeled as an ideal pinhole camera with intrinsic matrix $K \in R^{3 \times 3}$ and extrinsic parameters $[R_c|t_c]$, where $R_c \in SO(3)$ is the rotation matrix and $t_c \in R^3$ is the translation vector. A 3D point $v = (X, Y, Z)$ in world coordinates is projected onto the image plane as:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R_c, |t_c] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (8)$$

where (u, v) are pixel coordinates and λ is a scale factor. In our implementation, the camera is positioned at a fixed location within a virtual outdoor scene constructed in Unity, configured to match a standard full-HD resolution of 1920×1080 pixels.

Ground-truth bounding boxes are generated automatically by projecting each UAV's 3D bounding volume onto the image plane. Let $B_i \subset R^3$ denote the axis-aligned bounding box of UAV i in world coordinates, with eight corner vertices $V_i = \{v_1, v_2, \dots, v_8\}$. Each vertex is projected to obtain its 2D image coordinates:

$$(u_k, v_k) = \pi(v_k; K, R_c, t_c), \quad k = 1, \dots, 8 \quad (9)$$

where $\pi(\cdot)$ denotes the perspective projection function. The 2D bounding box $b_i = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$ is then computed as:

$$b_i = (\min_k u_k, \min_k v_k, \max_k u_k, \max_k v_k). \quad (10)$$

This projection-based annotation guarantees pixel-level accuracy and perfect consistency across the dataset, completely eliminating the cost and potential errors associated with manual labeling. Combined with the trajectory-based generation scheme, the proposed pipeline enables efficient synthesis of large-scale, densely annotated UAV swarm datasets with controlled variation in pose, formation and imaging conditions.

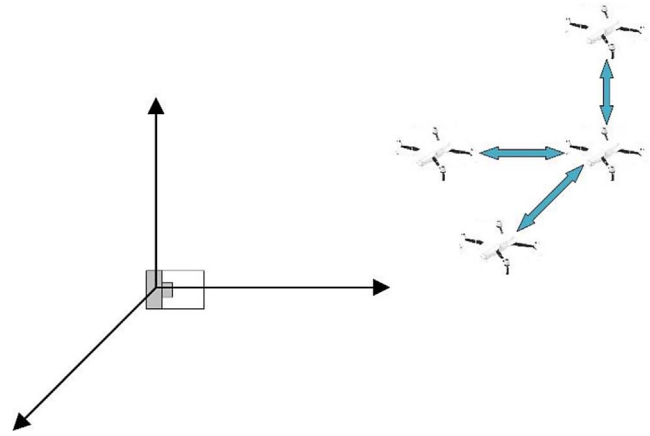


Fig. 3. The arrows indicate translations along the three orthogonal axes of the world coordinate frame, corresponding to the position components (x_i, y_i, z_i) in the 6-DOF state representation.

3.3 Software implementation

The data generation pipeline described above is implemented as a standalone software tool named 3D Model Shots Generator, built on the Unity engine. This tool integrates all components of the proposed methodology, including 6-DOF state initialization, trajectory interpolation, camera configuration, rendering, and automatic annotation, into a unified and user-friendly framework.

The software adopts a parameter-driven architecture in which scene configuration, object placement, camera settings, and rendering control are explicitly decoupled and exposed to the user, as illustrated in Figure 4. This design philosophy enables systematic exploration of the data generation space while maintaining strict reproducibility. Given identical 3D models and parameter settings, the system deterministically produces the same swarm configurations and rendered outputs, which is essential for controlled experimental analysis and fair benchmarking across different detection algorithms.

During initialization, one or more 3D UAV models in common formats are loaded and instantiated multiple times

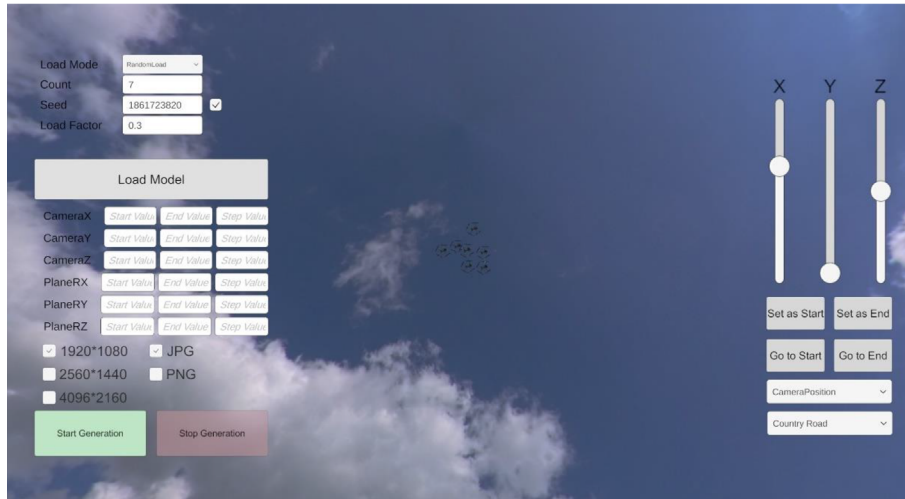


Fig. 4. Overview of the synthetic UAV-swarm dataset generation pipeline.

within the bounded observation volume W . The software supports two complementary placement modes. In structured mode, UAV instances are arranged according to a regular three-dimensional grid, with configurable spacing along each axis (controlled by XCount, YCount, ZCount and a LoadFactor parameter), enabling precise control over formation geometry. In randomized mode, positions are sampled stochastically within W , governed by an explicit random seed that ensures full reproducibility. A global density factor regulates spatial sparsity, preventing unrealistic overlaps while allowing the generation of both sparse and dense swarm configurations. In our experiments, all results are obtained on datasets generated using eight different UAV 3D models, as illustrated in [Figure 5](#).

Camera placement and scene orientation are parameterized independently. The virtual camera is constrained to face the center of the observation volume, guaranteeing that all UAV instances remain within the field of view. Camera translation along the three spatial axes (CameraX, CameraY, CameraZ) and global rotation of the scene (PlaneRX, PlaneRY, PlaneRZ) are specified using start, end, and step values, enabling automated traversal of the parameter space and efficient batch generation of images covering diverse viewpoints. Notably, the CameraZ parameter represents a relative scale factor rather than an absolute distance; the actual camera-to-swarm distance is automatically computed based on the spatial extent of the loaded models. A real-time preview mechanism allows users to inspect parameter effects before committing to large-scale generation, reducing wasted computation and improving dataset quality.

The rendering subsystem supports multiple output resolutions, including full-HD (1920×1080), 2 K (2560×1440), and 4K (4096×2160), as well as both lossy (JPEG, compressed to 85% quality) and PNG image formats. Scene appearance diversity is introduced through seven selectable skybox environments provided by Unity’s rendering pipeline, including Daytime, Sunset, Beautiful Pasture, Snowy Bridge, Small Harbour, Tellsplatte, and Country Road, representing a variety of sky textures, weather conditions, and

lighting scenarios (see [Fig. 6](#)). During batch generation, all parameters are locked to ensure consistency across the produced images.

For each rendered frame, ground-truth bounding boxes are generated automatically using the projection-based annotation procedure described previously. The software outputs both the rendered images and their corresponding annotation files in standard formats compatible with common deep learning frameworks, enabling direct integration into detector training pipelines.

By combining explicit parameterization, deterministic generation, and automated annotation, the proposed software tool provides a flexible and efficient foundation for synthesizing large-scale UAV swarm datasets. The following section presents a quantitative analysis of the dataset produced using this system.

3.4 Dataset analysis and characteristics

This section presents a quantitative analysis of the proposed synthetic UAV swarm dataset, focusing on dataset scale, target size distribution, and swarm density characteristics relevant to long-range multi-UAV detection.

The dataset consists of 7000 RGB images rendered at a resolution of 1920×1080 , containing a total of 31,542 annotated UAV instances. Each image includes between 1 and 15 UAVs, with an average of 4.5 instances per image, emphasizing multi-target swarm scenarios rather than isolated UAV detection. Overall statistics are summarized in [Table 2](#).

Following common practice, we characterize target scale by the square root of bounding box area, \sqrt{wh} , where w and h denote the bounding box width and height in pixels. Based on this measure, UAV instances are categorized into small, medium, and large targets. As shown in [Table 2](#), the dataset is strongly biased toward small objects, with 67.3% of instances having $\sqrt{wh} < 32$ pixels. This distribution closely reflects real-world long-range UAV observation scenarios and poses significant challenges for generic object detectors.

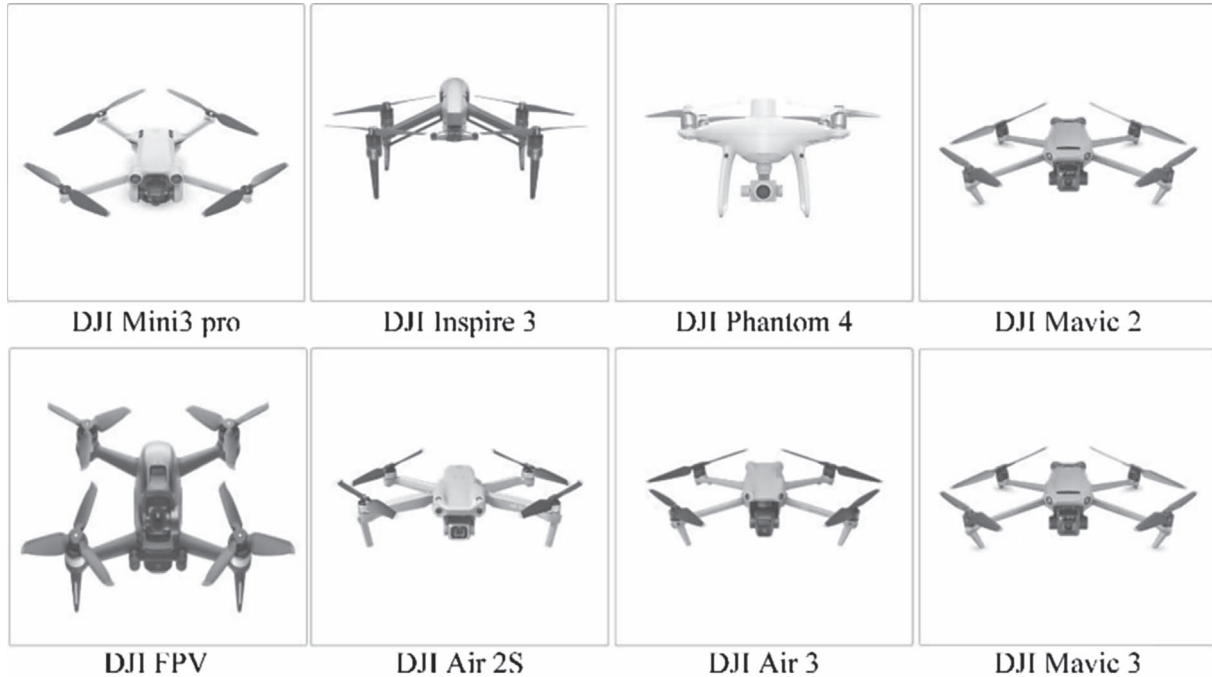


Fig. 5. Eight different UAV 3D models.



Fig. 6. Available skybox environments for scene appearance diversity: Daytime, Sunset, Beautiful Pasture, Snowy Bridge, Small Harbour, Telsplatte, and Country Road.

4 Experimental setup

This section describes the experimental protocol used to evaluate the proposed synthetic UAV swarm dataset. We introduce the implementation details, dataset partition, training configurations, and evaluation metrics.

4.1 Implementation details

All experiments are conducted on an Ubuntu 20.04 LTS system equipped with an Intel Xeon Silver CPU and a single NVIDIA RTX 3090 GPU with 16 GB for reproducibility. The software environment include CUDA 12.1, Python

Table 2. Overall statistics of the proposed synthetic UAV swarm dataset.

Property	Value
Number of images	7000
Image resolution	1920 × 1080
UAV instances	31,542
Average UAVs per image	4.5
Small targets ($\sqrt{wh} < 32\text{px}$)	67.3%
Medium targets ($32 < \sqrt{wh} < 96\text{px}$)	25.1%
Large target ($96 < \sqrt{wh}$)	7.6%

3.9, and PyTorch framework. The models were optimized using the SGD optimizer with a batch size of 16, an initial learning rate of 0.01, a learning rate decay of 0.0005, and an input image size set to 640×640 pixels.

We select six representative object detection architectures as baselines. The proposed synthetic UAV swarm dataset is divided into non-overlapping training, validation, and test subsets. The specific partition is as follows:

Training set: 4940 images (70%), used for learning detector parameters.

Validation set: 706 images (10%), used for model selection and hyperparameter tuning.

Test set: 1412 images (20%), reserved for final performance evaluation.

4.2 Evaluation metrics

In order to validate the performance of the model, the following performance metrics were selected for measurement: P (precision), R (recall), mean average precision (mAP_{50}), mAP_{50-95} , parameters, and Model Size. True Positive (TP) denotes the number of correctly detected targets, False Positive (FP) indicates the number of backgrounds detected as targets, and False Negative (FN) denotes the number of targets detected as backgrounds.

P denotes the proportion of positive samples correctly classified by the model, reflecting the ability of the model to correctly classify. The calculation is as show in equation (11).

$$P = \frac{TP}{TP + FP} \quad (11)$$

R denotes the ratio of the number of correctly predicted samples to the total number of samples, reflecting the ability of the model to detect the target comprehensively, and measuring the number of samples that the model misses in identifying the target. The calculation is as show in equation (12).

$$R = \frac{TP}{TP + FN}. \quad (12)$$

The mAP is the average accuracy of all targets detected by the model, reflecting the ability to generate predictive frames and labels that overlap. The higher the value of this metric, the better the model's detection effect on different categories. The calculation is as show in equation (13).

$$mAP = \frac{1}{n} \sum_{i=1}^n AP \quad (13)$$

where n denotes the average accuracy for each category. In our UAV detection task, $n = 1$. mAP_{50} denotes the average accuracy when the IOU (Intersection over Union) is 50%. mAP_{50-95} denotes that the threshold of mAP ranges from 50% to 95%, and the average of 10 mAP values is obtained at 5% intervals.

Model Size is used to evaluate the complexity of the model. In general, the smaller the Model Size is, the less computing power the model requires, the lower the performance requirements for hardware, and the easier it is to build in low-end devices.

4.3 Experimental results

Among all evaluated models, YOLOv13 achieves the best overall performance, attaining the highest Precision (0.903), Recall (0.897), mAP_{50} (0.934), and mAP_{50-95} (0.612). YOLOX ranks second with a mAP_{50} of 0.912, followed by YOLOv6 ($mAP_{50} = 0.901$) and YOLOv12 ($mAP_{50} = 0.887$). Among the non-YOLO detectors, RT-DETR achieves a mAP_{50} of 0.889, comparable to the mid-range YOLO variants, while Faster R-CNN yields the lowest accuracy ($mAP_{50} = 0.856$), likely due to the limited effectiveness of its region proposal mechanism on predominantly small UAV targets. Overall, the four one-stage YOLO-based detectors consistently outperform both Faster R-CNN and RT-DETR in detection accuracy on this dataset.

A noteworthy observation is the significant drop in mAP_{50-95} across all detectors, with values ranging from 0.487 (Faster R-CNN) to 0.612 (YOLOv13), representing a decrease of approximately 30–35 percentage points relative to their mAP_{50} counterparts. This indicates that precise bounding box regression for small-scale UAV targets remains challenging across all evaluated architectures.

In terms of computational efficiency, the four YOLO variants achieve inference speeds of 75–105 FPS with compact model sizes (17.3–38.0 MB), making them suitable for real-time deployment. RT-DETR operates at 48 FPS with a 64.0 MB model, while Faster R-CNN is the most resource-intensive, with only 22 FPS and a 160.0 MB model size.

Figure 7 illustrates qualitative detection results of four YOLO-based detectors across six representative scene environments. These scenes cover a wide range of background complexities, including dense vegetation, complex cloud distributions, low-texture sky backgrounds, urban street environments, rural landscapes, and snow-covered mountainous areas. Such diversity enables a comprehensive evaluation of detector robustness under realistic and challenging conditions. Note that the qualitative visualization focuses on the four YOLO-based detectors, as all quantitative comparisons including Faster R-CNN and RT-DETR are comprehensively presented in Table 3.

As shown in the figure, all detectors are capable of detecting UAV targets in relatively simple scenes. However, noticeable performance differences emerge in more challenging scenarios. In the Clear Sky and Snowy Mountain scenes, UAVs appear as extremely small objects against low-texture backgrounds, making detection particularly difficult. In these cases, missed detections and low-confidence predictions are frequently observed, especially for earlier-generation models. In contrast, more recent YOLO variants demonstrate improved robustness, benefiting from enhanced feature representation and multi-scale feature aggregation.

Overall, the quantitative results in Table 3 are consistent with the qualitative findings in Figure 7, collectively confirming that YOLOv13 offers the most robust detection performance across diverse and challenging scene conditions, while also highlighting that small object localization accuracy (as reflected by mAP_{50-95}) remains an open challenge for all current detectors.

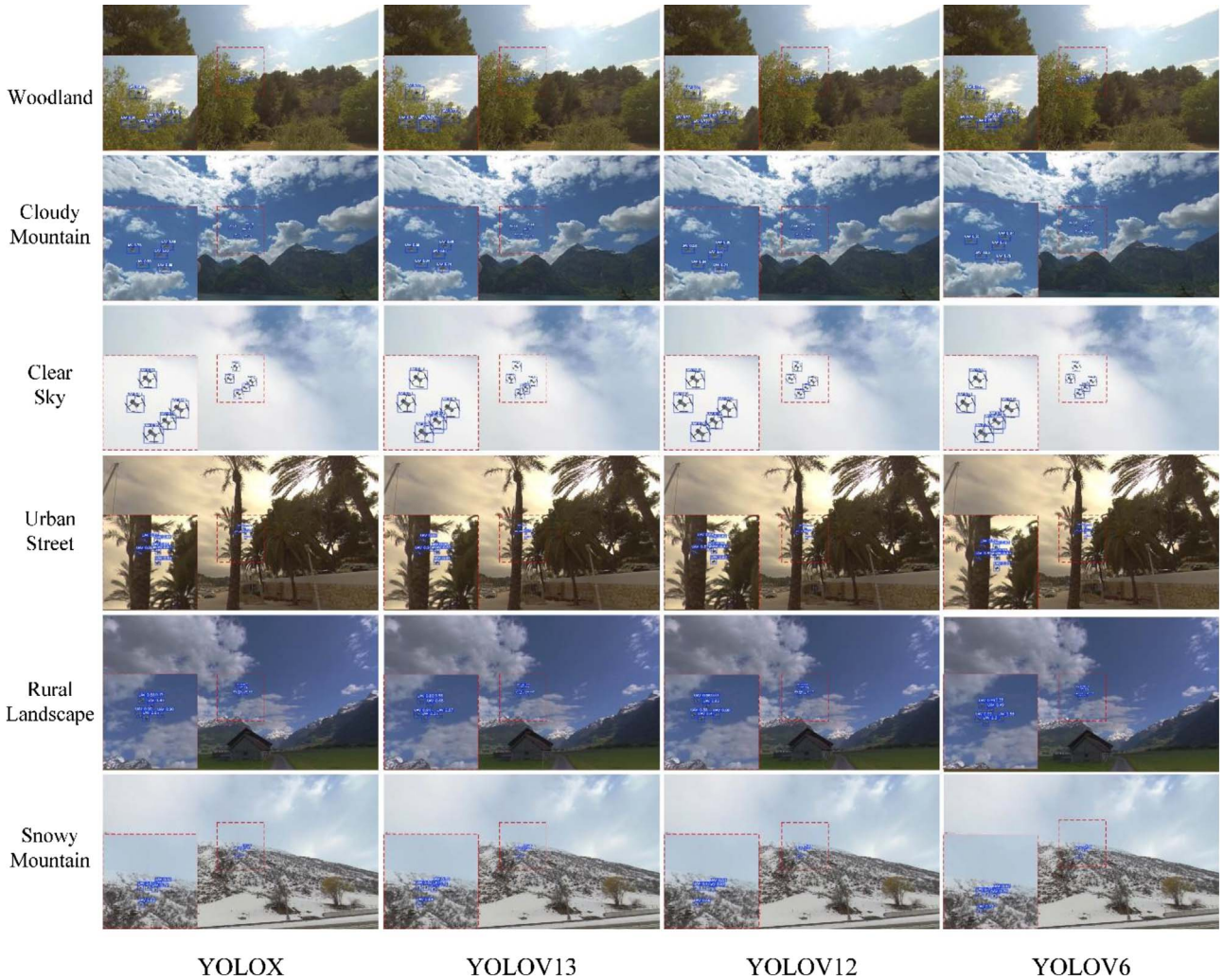


Fig. 7. Qualitative detection results of different YOLO-based detectors under diverse scene conditions. From top to bottom, the scenes correspond to Woodland, Cloudy Mountain, Clear Sky, Urban Street, Rural Landscape, and Snowy Mountain. From left to right, the detection results are generated by YOLOX, YOLOv13, YOLOv12, and YOLOv6, respectively. Red dashed boxes highlight challenging regions, while blue bounding boxes indicate detected UAV targets.

Table 3. Summarizes the quantitative detection performance and model complexity of all six detectors, including the four YOLO-based detectors as well as Faster R-CNN and RT-DETR, on the proposed synthetic UAV swarm dataset.

Detector	mAP ₅₀	mAP ₅₀₋₉₅	P	R	Params	Model Size	FPS
YOLOX	0.886	0.875	0.912	0.573	9.0	17.3	95
YOLOv13	0.903	0.897	0.934	0.612	9.0	18.5	75
YOLOv12	0.871	0.852	0.887	0.542	9.4	18.0	80
YOLOv6	0.879	0.868	0.901	0.558	18.5	38	105
Faster R-CNN	0.842	0.813	0.856	0.487	41.1	160	22
RT-DETR	0.862	0.847	0.889	0.531	32.0	64	48

In cluttered environments such as Woodland and Urban Street, false positives occasionally occur due to background elements such as tree branches or building edges exhibiting visual patterns similar to UAVs. This highlights the importance of background diversity when evaluating detection performance.

Despite the overall satisfactory detection performance, several failure cases are observed across all detectors. The most common failure cases occur when UAV targets are extremely small or partially occluded, particularly in the Clear Sky and Snowy Mountain scenes. In such cases, the lack of discriminative texture information leads to missed detections.

Table 4. Cross-dataset evaluation results.

Training set	Test set	mAP ₅₀	mAP ₅₀₋₉₅
MMFW-UAV	MMFW-UAV	0.856	0.521
SynthSwarm	MMFW-UAV	0.743	0.412

Another typical failure case arises in cluttered environments, such as Woodland and Urban Street, where background structures may cause false positives. These failure cases indicate that small object detection under complex and low-contrast backgrounds remains a challenging problem. Future work will focus on incorporating more diverse training samples and enhancing feature representation to further improve robustness in these scenarios.

4.4 Cross-dataset experiments

To further validate the generalizability of the proposed SynthSwarm dataset, we conduct a cross-dataset evaluation using the MMFW-UAV dataset as an additional test domain. Specifically, the YOLOv13 detector trained on SynthSwarm is directly evaluated on the MMFW-UAV test set without any fine-tuning, and the results are compared against a model trained on MMFW-UAV itself. The results are summarized in Table 4.

As shown in Table 4, the model trained on SynthSwarm achieves a mAP₅₀ of 0.743 and mAP₅₀₋₉₅ of 0.412 on the MMFW-UAV test set without any fine-tuning, compared to 0.856 and 0.521 for the in-domain baseline. While a performance gap is expected due to the inherent domain shift between synthetic rendering and real-world imaging conditions, the SynthSwarm-trained model retains approximately 87% of the in-domain mAP₅₀ performance, demonstrating reasonable cross-domain transferability. These results suggest that SynthSwarm captures sufficient visual diversity and structural characteristics of UAV swarm targets to serve as a viable pre-training or standalone training source for real-world UAV detection tasks.

5 Conclusion

In this paper, we proposed a synthetic UAV swarm dataset specifically designed to study small-object, multi-target detection in long-range aerial surveillance scenarios. By leveraging a controllable virtual environment built in Unity, we modeled a fixed sky-facing camera and instantiated multiple 3D UAV models with randomized positions and orientations in a three-dimensional observation volume. This setup enabled the automatic generation of thousands of high-resolution images together with pixel-accurate 2D bounding box annotations, without the need for manual labeling.

We provided a detailed description of the dataset design and generation pipeline, and analyzed the resulting data in terms of target scale, swarm density, and scene diversity. Extensive experiments with six representative detectors spanning three paradigms – one-stage detectors (YOLOX,

YOLOv13, YOLOv12, YOLOv6), a two-stage detector (Faster R-CNN), and a Transformer-based detector (RT-DETR) – demonstrated that the proposed dataset poses significant challenges due to the predominance of small UAVs and the presence of dense swarms. Among all evaluated models, YOLOv13 achieved the best overall performance, while the four one-stage YOLO-based detectors consistently outperformed Faster R-CNN and RT-DETR in both detection accuracy and inference efficiency, suggesting that lightweight one-stage architectures are better suited for this task. The consistent performance trends across detector paradigms indicate that SynthSwarm supports fair and meaningful cross-architecture comparisons. Furthermore, cross-dataset experiments on the MMFW-UAV dataset confirmed that SynthSwarm possesses reasonable transferability to real-world UAV detection scenarios, validating its potential as a pre-training data source.

In future work, we plan to further enrich the dataset by incorporating additional sensor modalities, more diverse weather and illumination conditions, and multiple object categories such as birds or manned aircraft to better capture real-world confusion scenarios. We also intend to investigate specialized detection architectures tailored to extremely small and crowded UAV targets, as well as domain adaptation techniques that more effectively bridge the gap between synthetic and real imagery. We hope that the release of this dataset will foster further research on robust and scalable UAV swarm detection in the computer vision and remote sensing communities.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped improve the quality of manuscript.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of interest

The authors declare that they have no competing interests.

Data availability statement

The dataset and generation pipeline are publicly available at <https://github.com/marisinpiper/Synthetic-UAV-Swarm-Dataset>.

Author contribution statement

All authors take part in the discussion of the work described in this paper. These authors contributed equally to this work.

Glossary

UAV	Unmanned Aerial Vehicle;
mAP	mean Average Precision;
IoU	Intersection over Union.

References

- Alqudsi Y, Makaraci M, UAV swarms: research, challenges, and future directions, *J. Eng. Appl. Sci.* **72**, 12 (2025). <https://doi.org/10.1186/s44147-025-00582-3>
- Stodola P, Nohel J, Horák L, Dynamic reconnaissance operations with UAV swarms: adapting to environmental changes, *Sci. Rep.* **15**, 15092 (2025). <https://doi.org/10.1038/s41598-025-00201-4>
- Fan Y, Optimized task allocation for unmanned aerial vehicle swarms in smart agriculture, in *Advances in Guidance, Navigation and Control*, Lecture Notes in Electrical Engineering, Vol. **1349** (Springer, Singapore, 2025), pp. 349–359. https://doi.org/10.1007/978-981-96-2248-1_34
- Zhang X, Zhang J, Gao J, et al., A sharding blockchain-based UAV system for search and rescue missions, *Front Comput. Sci.* **19**, 193805 (2025). <https://doi.org/10.1007/s11704-024-3467-8>
- Lattimer BY, Huang X, Delichatsios MA, et al., Use of unmanned aerial systems in outdoor firefighting, *Fire Technol.* **59**, 2961–2988 (2023). <https://doi.org/10.1007/s10694-023-01437-0>
- Ding G, Ren Y, Liu Y, Zhao Q, Li S, Vision-based anti-unmanned aerial technology: opportunities and challenges, *IEEE Geosci. Remote Sens. Mag.* **13**(4), 382–405 (2025). <https://doi.org/10.1109/MGRS.2025.3589763>
- Liu Y, Sun Z, Xi L, et al., MMFW-UAV dataset: multi-sensor and multi-view fixed-wing UAV dataset for air-to-air vision tasks, *Sci. Data* **12**, 185 (2025). <https://doi.org/10.1038/s41597-025-04482-2>
- Xu L, Luo Z, Anti-UAV detection and identification technology: fundamentals, methods and challenges, *Phys. Commun.* **71**, 102676 (2025). <https://doi.org/10.1016/j.phycom.2025.102676>
- Al-Iqubaydhi N, Alenezi A, Alanazi T, et al., Deep learning for unmanned aerial vehicles detection: a review, *Comput. Sci. Rev.* **51**, 100614 (2024). <https://doi.org/10.1016/j.cosrev.2023.100614>
- Wang B, Li Q, Mao Q, et al., A survey on vision-based anti unmanned aerial vehicles methods, *Drones* **8**, 518 (2024). <https://doi.org/10.3390/drones8090518>
- Dong Y, Wu F, Zhang S, et al., Securing the skies: a comprehensive survey on anti-UAV methods, benchmarking, and future directions, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA* (IEEE, New York, 2025), pp. 6661–6675. <https://doi.org/10.1109/CVPRW67362.2025.00663>
- Last M, Early detection of small- and medium-sized drones in complex environments, *Drone Syst. Appl.* **13**, 1–10 (2025). <https://doi.org/10.1139/dsa-2025-0018>
- Cheng Q, Li J, Du J, et al., Anti-UAV detection method based on local-global feature focusing module, in: *Proceedings of the 7th IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China* (IEEE, New York, 2024), pp. 1413–1418. <https://doi.org/10.1109/IAEAC59436.2024.10503882>
- Jiang Y, Jingliang G, Yanqing Z, et al., Detection and tracking method of small-sized UAV based on YOLOv5, in: *Proceedings of the 19th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China* (IEEE, New York, 2022), pp. 1–5. <https://doi.org/10.1109/ICCWAMTIP56608.2022.10016550>
- Zheng Y, Chen Z, Lv D, Li Z, Lan Z, Zhao S, Air-to-air visual detection of micro-UAVs: an experimental evaluation of deep learning, *IEEE Robot. Autom. Lett.* **6**, 1020–1027 (2021). <https://doi.org/10.1109/LRA.2021.3056059>
- Zhao J, Zhang J, Li D, Wang D, Vision-based anti-UAV detection and tracking, *IEEE Trans. Intell. Transp. Syst.* **23**, 25323–25334 (2022). <https://doi.org/10.1109/TITS.2022.3177627>
- Shah S, Dey D, Lovett C, Kapoor A, AirSim: high-fidelity visual and physical simulation for autonomous vehicles, in *Field and Service Robotics*, Springer Proc. Adv. Robot. **Vol. 5** (Springer, Cham, 2018), pp. 621–635. https://doi.org/10.1007/978-3-319-67361-5_40
- Qiu W, Yuille A, UnrealCV: connecting computer vision to Unreal Engine, in *Computer Vision – ECCV 2016 Workshops*, Lecture Notes in Computer Science, **Vol. 9915** (Springer, Cham, 2016), pp. 909–916. https://doi.org/10.1007/978-3-319-49409-8_75
- Nasir FA, Khurshid K, SyntAR: synthetic data generation using multiscale attention generator-discriminator framework (SinGAN-MSA) for improved aircraft recognition in remote sensing images, *Multimed. Tools Appl.* **84**, 42777–42805 (2025). <https://doi.org/10.1007/s11042-025-20825-y>
- Patel R, Chandalia D, Nayak A, et al., CGI-based synthetic data generation and detection pipeline for small objects in aerial imagery, *IEEE Access* **13**, 61192–61206 (2025). <https://doi.org/10.1109/ACCESS.2025.3553530>
- Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P, Domain randomization for transferring deep neural networks from simulation to the real world, in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Vancouver, Canada* (IEEE, New York, 2017), pp. 23–30. <https://doi.org/10.1109/IROS.2017.8202133>
- Zhu JY, Park T, Isola P, Efros AA, Unpaired image-to-image translation using cycle-consistent adversarial networks, in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy* (2017), pp. 2223–2232. <https://doi.org/10.1109/ICCV.2017.244>
- Ganin Y, Ustinova E, Ajakan H, et al., Domain-adversarial training of neural networks, *J. Mach. Learn. Res.* **17**, 1–35 (2016). https://doi.org/10.1007/978-3-319-58347-1_10