ADG-YOLO: A Lightweight and Efficient Framework for Real-Time UAV Target Detection and Ranging

Hongyu Wang *, Zheng Dang, Mingzhu Cui, Hanqi Shi, Yifeng Qu, Hongyuan Ye, Jingtao Zhao, Duosheng Wu

Posted Date: 25 August 2025

doi: 10.20944/preprints202508.1730.v1

Keywords: UAV detection; monocular ranging; edge computing; YOLOv11; real-time tracking

 doi:10.20944/preprints202508.1730.v1

Article

ADG-YOLO: A Lightweight and Efficient Framework for Real-Time UAV Target Detection and Ranging

Hongyu Wang *, Zheng Dang, Mingzhu Cui, Hanqi Shi, Yifeng Qu, Hongyuan Ye, Jingtao Zhao and Duosheng Wu

Shanghai Jiao Tong University

* Correspondence: redrain@sjtu.edu.cn

Abstract

The rapid evolution of UAV technology has increased the demand for lightweight airborne perception systems. This study introduces ADG-YOLO, an optimized model for real-time target detection and ranging on UAV platforms. Building on YOLOv11n, we integrate C3Ghost modules for efficient feature fusion and ADown layers for detail-preserving downsampling, reducing the model's parameters to 1.77M and computation to 5.7 GFLOPs. The Extended Kalman Filter (EKF) tracking improves positional stability in dynamic environments. Monocular ranging is achieved using similarity triangle theory with known target widths. Evaluations on a custom dataset, consisting of 5,343 images from three drone types in complex environments, show that ADG-YOLO achieves 98.4% $mAP_{0.5}$ and 85.2% recall at 27 FPS on Lubancat4 edge devices. Distance measurement tests indicate an average error of 4.18% in the 0.5–5 m range for the DJI NEO model, and an average error of 2.40% in the 2–50 m range for the DJI 3TD model. This work effectively bridges the gap between accuracy and efficiency for resource-constrained applications.

Keywords: UAV detection; monocular ranging; edge computing; YOLOv11; real-time tracking

1. Introduction

With the rapid evolution of unmanned aerial vehicle (UAV) technology, its significance has been steadily rising across military, civilian, and commercial domains. In modern warfare especially, UAVs have transitioned from auxiliary roles to primary combat platforms, profoundly reshaping combat models and battlefield dynamics [1–3]. Breakthroughs in artificial intelligence (AI), low-cost manufacturing, and stealth flight technology have endowed various unmanned systems with enhanced perception, autonomous decision-making, and strike capabilities. A representative case is the "Operation Spider's Web" launched by Ukraine in June 2025: Ukrainian forces covertly deployed 117 FPV kamikaze drones deep within Russian strategic zones, coordinating the destruction of multiple high-value targets. This operation inflicted severe damage on Russia's airborne nuclear deterrent system — at a cost of merely a few thousand dollars per drone — and exposed the critical failure of existing defense systems against attacks employing "low-slow-small UAVs + AI swarms + covert delivery" tactics. This incident highlighted both the increasingly complex combat capabilities of unmanned systems and the urgent demand for advancements in target detection, path planning, and low-latency response mechanisms.

Currently, the development of unmanned systems technology exhibits two prominent trends. On one hand, AI-empowered intelligent systems are breaching the traditional OODA (Observe–Orient–Decide–Act) response loop, demonstrating AI swarm-driven closed-loop operational capabilities in real combat scenarios. On the other hand, adversaries are developing more sophisticated counter-UAV strategies, including communication jamming, deception interference, and terrain-based evasion tactics [4,5]. Within this technological contest, enhancing the perception capability of unmanned systems—ensuring high efficiency, stability, and low power consumption—has become a pivotal tactical breakthrough point.

 doi:10.20944/preprints202508.1730.v1

Compared to conventional ground-based fixed observation platforms, airborne perception systems integrate visual ranging functions directly onto UAV platforms, achieving mobile and real-time sensing. Airborne platforms allow closer proximity to targets, enable dynamic close-range tracking, and support complex tasks such as autonomous obstacle avoidance, formation coordination, and precision strikes—significantly enhancing autonomous operational capabilities [2]. However, airborne deployment also imposes stringent demands on system lightweight design, low power consumption, and real-time responsiveness, driving researchers to develop visual ranging and target detection algorithms optimized for embedded edge computing platforms. Advancing efficient and reliable airborne visual ranging systems is not only a key technology for UAV intelligence upgrades but also a core enabler for enhancing future unmanned combat effectiveness and survivability.

At present, mainstream ranging methods include radar [6], laser [7], ultrasonic [8], and visual ranging [9]. While radar and laser offer high precision and long detection ranges, their bulky size, high power consumption, and cost render them unsuitable for small UAV applications. Visual ranging, with its non-contact nature, low cost, and ease of integration, has become a research focus for lightweight perception systems. Based on camera configurations, visual ranging can be categorized into binocular and monocular systems. Binocular vision offers higher accuracy but requires extensive calibration and disparity computation resources [10,11]. In contrast, monocular vision—with its simple hardware structure and algorithmic flexibility—has emerged as the preferred solution for edge and embedded platforms [12,13].

Monocular visual ranging methods fall into two categories: depth map estimation and physical distance regression. The former relies on convolutional neural networks to generate relative depth maps, which are suitable for scene modeling but lack direct physical distance outputs [14–19]. The latter directly outputs target distances based on geometric modeling or end-to-end regression, offering faster and more practical responses. Geometric modeling approaches, such as perspective transformation, inverse perspective mapping (IPM), and similar triangle modeling, estimate depth by mapping pixel data to physical parameters [20–25]. In recent years, deep learning techniques have been introduced to monocular ranging models to enhance adaptability in unstructured environments [26]. Traditional two-stage detection algorithms such as RCNN [27], Fast R-CNN [28], and Faster R-CNN [29] excel in detection accuracy but are limited by their complex architectures and high computational demands—making them less suitable for edge platforms that require low power and real-time performance, thus restricting their practical applications in UAV scenarios.

In the field of target detection, the YOLO (You Only Look Once) family of algorithms has gained widespread use in unmanned systems due to its fast detection speed and compact architecture, ideal for edge deployment [30–41]. To enhance small-object detection and long-range recognition on embedded platforms, researchers have proposed various improvements based on YOLOv5, YOLOv8, and YOLOv11, incorporating lightweight convolutions, attention mechanisms, and feature fusion modules [42–45]. However, most existing studies focus on ground-based or general-purpose deployments and lack systematic research on lightweight airborne deployment and integrated perception-ranging systems. Particularly on ultra-low-power, computation-constrained mobile platforms, balancing detection accuracy, ranging stability, and frame rate remains a critical unresolved challenge. Reference [46] explored a low-power platform based on Raspberry Pi combined with YOLOv5 for real-time UAV target detection. Although it did not deeply address deployment efficiency and response latency, it provided a valuable reference for lightweight applications.

To this end, this study expands the research perspective and application scope of UAV visual perception. Transitioning from traditional ground-based observation models to onboard autonomous sensing, we propose a lightweight airborne perception system mounted directly on UAV platforms. This system is designed to meet the integrated requirements of target detection and distance measurement, achieving real-time, in-flight optimization of both functionalities. The main innovations and contributions of this study are summarized as follows:

 doi:10.20944/preprints202508.1730.v1

1. Lightweight Detection Architecture Design: Based on the YOLOv11n model, this study introduces the C3GHOST and ADown modules to construct an efficient detection architecture tailored for edge computing platforms. The C3GHOST module reduces computational overhead through lightweight feature fusion while enhancing feature representation capability. The ADown module employs an efficient down-sampling strategy that lowers computational cost without compromising detection accuracy. Systematic evaluation on a custom-built dataset demonstrates the model's capability for joint optimization in terms of frame rate and ranging precision.

2. Target Tracking Optimization: To further improve the stability of UAV target tracking, this study incorporates the Extended Kalman Filter (EKF) approach. EKF performs target position estimation and trajectory prediction in dynamic environments, significantly reducing position jitter and sporadic false detections during the tracking process, thereby enhancing robustness and consistency.

3. Dataset Expansion: Based on a publicly available dataset from CSDN, this study conducts further expansion by constructing a comprehensive dataset that covers a wide range of UAV models and complex environments. The dataset includes image samples captured under varying flight altitudes, viewing angles, and lighting conditions. This expansion enables the proposed model to adapt not only to different types of UAV targets, but also to maintain high detection accuracy and stability in complex flight environments.

4. Model Conversion and Deployment on Edge Devices: To facilitate practical deployment, the trained model was converted from its standard format to a format compatible with edge computing devices based on the RK3588S chipset. The converted model was successfully deployed onto the edge platform, ensuring efficient operation on resource-constrained hardware.

2. Methodology

2.1. ADG-YOLO

2.1.1. Basic YOLOv11 Algorithm

YOLOv11n is the most lightweight nano version within the Ultralytics YOLOv11 series, implemented using the Ultralytics Python package (e.g., v11.0). In this study, we adopt the official release version 8.3.31. The network architecture is illustrated in Figure 1. To enhance detection performance in lightweight configurations, YOLOv11n comprehensively replaces the C2f modules used in YOLOv8n with unified C3k2 modules (Cross-Stage Partial with kernel size 2) across the entire architecture. This modification enables efficient feature extraction via a series of small-kernel convolutions while maintaining parameter efficiency, thereby significantly improving feature representation and receptive field capacity.

In the backbone, after extracting high-level semantic features, an SPPF (Spatial Pyramid Pooling–Fast) module is introduced to integrate multi-scale contextual information. Subsequently, a C2PSA (Cross-Stage Partial with Spatial Attention) module is appended to enhance feature responses in critical spatial regions via a spatial attention mechanism, thereby improving recognition of small and partially occluded targets. The neck adopts a standard upsampling and feature concatenation strategy, with additional C3k2 modules embedded to reinforce cross-scale feature fusion. Finally, in the detection head, the architecture combines C3k2 and conventional layers to generate bounding box and class predictions. The overall model maintains a compact size of approximately 6.5M parameters, achieving both efficient inference and high detection accuracy.This architectural design endows YOLOv11n with enhanced multi-scale and spatial awareness capabilities while preserving its lightweight nature. Notably, it delivers significant improvements in small object detection accuracy compared to YOLOv8n.

 doi:10.20944/preprints202508.1730.v1

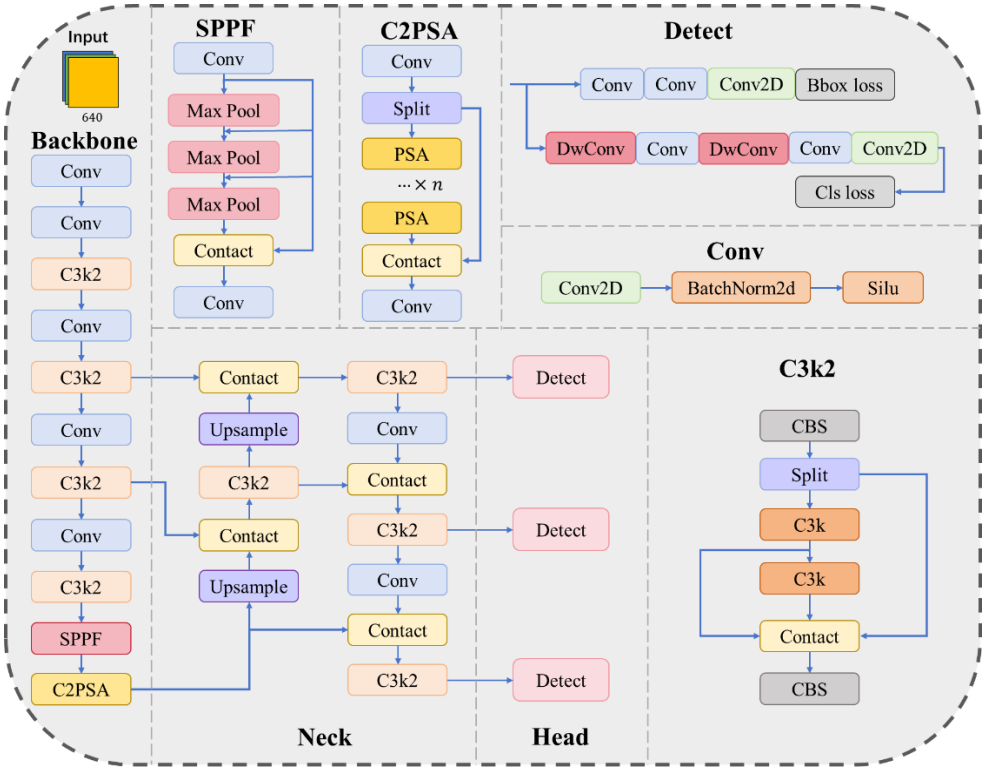
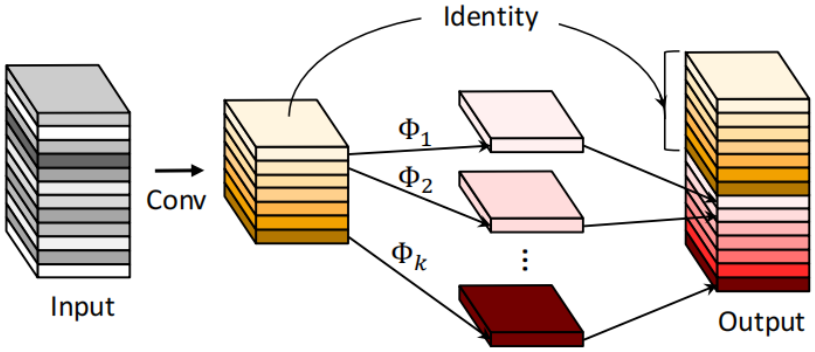


Figure 1. YOLOv11n algorithm structure.

2.1.2. C3Ghost

In this study, the standard C3k2 modules are replaced with C3Ghost modules to achieve a more lightweight design and improved computational efficiency in feature fusion. The C3Ghost module consists of a series of GhostConv layers integrated within a Cross-Stage Partial (CSP) architecture, combining efficient information flow with compact network structure [47]. As shown in Figure 2, GhostConv divides the input feature map into two parts: the first part generates primary features using standard convolution, while the second part produces complementary "ghost" features through low-cost linear transformations such as depthwise separable convolutions. These two parts are then concatenated to form the final output. By exploiting the inherent redundancy in feature representations, this design significantly reduces the number of parameters and floating-point operations, while retaining a representational capacity comparable to conventional convolutions.

On this basis, C3Ghost integrates multiple GhostConv layers into the CSP structure to form a lightweight feature fusion unit. As illustrated in Figure 3, the input is split into two parallel branches. One branch extracts higher-level features through a series of stacked GhostBottleneck layers, and the other directly passes the input via a shortcut connection to preserve original information. The outputs from both branches are then concatenated and fused using a 1×1 convolution. This design enhances feature representation while keeping computational cost and model complexity to a minimum [48].



 doi:10.20944/preprints202508.1730.v1

Figure 2. The Ghost module [47].

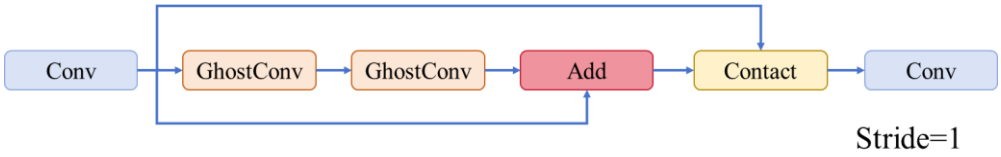


Figure 3. The C3Ghost module [48].

2.1.3. ADown

In this study, an ADown module is introduced to replace conventional convolution-based downsampling operations, aiming to improve efficiency while preserving fine-grained feature information. As illustrated in Figure 4, the core design of ADown consists of the following stages: the input feature map is first processed through average pooling and downsampled to half the spatial resolution. It is then split into two branches along the channel dimension. The first branch applies a 3×3 convolution to extract local detail features, while the second branch undergoes max pooling for downsampling, followed by a 1×1 convolution for channel compression and nonlinear transformation. Finally, the outputs from both branches are concatenated along the channel axis to form the downsampled output [49].

Compared to standard convolution or conventional pooling-based downsampling, ADown offers several distinct advantages. Its multi-path structure allows for the integration of both global and local information, mitigating the severe loss of detail often caused by traditional downsampling methods. Additionally, by reducing spatial resolution through pooling before applying lightweight convolutions, ADown achieves efficient feature extraction with significantly fewer parameters and lower FLOPs, without sacrificing representational power.

Moreover, ADown can be seamlessly integrated with existing multi-scale feature fusion modules, such as SPPF or FPN, enhancing the capacity of both the backbone and neck components to retain fine-grained features. This is particularly beneficial for small object detection tasks. In UAV imagery, where targets tend to be small and highly resolution-sensitive, the hierarchical detail preserved by ADown proves critical for accurately detecting small-scale objects. Its design not only improves fine-feature retention but also enhances the overall robustness of the model in multi-scale environments.

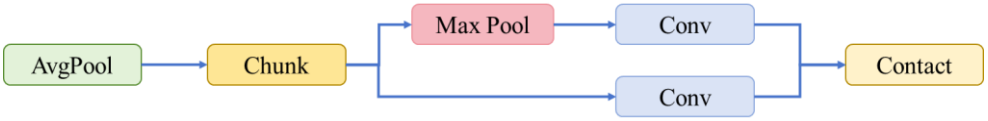


Figure 4. The ADown module [49].

2.1.4. Proposed ADG-YOLO

To enhance the detection capability of lightweight networks for low-altitude, low-speed, and small-sized UAVs—commonly referred to as "low-slow-small" targets—under resource-constrained environments, this study proposes a novel architecture named ADG-YOLO, based on the original YOLOv11n framework. While maintaining detection accuracy, ADG-YOLO significantly reduces model parameter size and computational complexity. The architecture incorporates systematic structural optimizations in three key areas: feature extraction, downsampling strategy, and multi-scale feature fusion. These improvements collectively strengthen the model's perception capability for low-altitude UAV targets in ground-based scenarios, thereby better meeting the practical demands of UAV detection from aerial perspectives. The overall network architecture of ADG-YOLO is illustrated in Figure 5.

 doi:10.20944/preprints202508.1730.v1

Firstly, in both the backbone and neck of the network, the original C3k2 modules are systematically replaced with C3Ghost modules. C3Ghost is a lightweight residual module constructed using GhostConv, initially introduced in GhostNet, and incorporates a Cross Stage Partial (CSP) structure to enable cross-stage feature fusion. Its core design concept lies in generating primary features through standard convolution, while reusing redundant information by producing additional "ghost" features through low-cost linear operations such as depthwise separable convolutions. This approach significantly reduces the number of parameters and the overall computational cost (FLOPs), making it especially suitable for deployment on edge devices with limited computing resources. Additionally, the stacked structure of GhostBottleneck layers further enhances the network's ability to represent features across multiple semantic levels.

Secondly, all stride=2 downsampling operations in the network are replaced with the ADown module. Instead of conventional 3×3 strided convolutions, ADown adopts a dual-path structure composed of average pooling and max pooling for spatial compression. Each path extracts features at different scales through lightweight 3×3 and 1×1 convolutions, and the outputs are concatenated along the channel dimension. This asymmetric parallel design allows ADown to preserve more rich texture and edge information while reducing feature map resolution. Such a design is particularly beneficial in UAV-based detection scenarios where objects are small and captured from high-altitude viewpoints against complex backgrounds. Compared to standard convolutions, ADown effectively reduces computational burden without compromising detection accuracy, while improving the flexibility and robustness of the downsampling process.

Thirdly, the Spatial Pyramid Pooling - Fast (SPPF) module is retained at the end of the backbone to enhance the modeling of long-range contextual information. Meanwhile, in the neck, a series of alternating operations—upsampling, feature concatenation, and downsampling via ADown—are introduced for feature fusion. This design facilitates the precise supplementation of low-level detail with high-level semantic information and improves alignment and interaction across multi-scale feature maps. As a result, the model's ability to detect small objects and capture boundary-level details is significantly enhanced. Combined with the lightweight feature extraction capability of the C3Ghost modules at various stages, the entire network achieves high detection accuracy while substantially reducing deployment cost and system latency.

In summary, the improvements of ADG-YOLO presented in this study are threefold: the C3Ghost modules enable efficient lightweight feature representation; the ADown module reconstructs a more effective downsampling pathway; and the SPPF module, together with multi-scale path interactions, strengthens fine-grained feature aggregation, particularly for small object detection. The complete network architecture of ADG-YOLO is shown in Figure 5, where the overall design seamlessly integrates lightweight structure with multi-scale feature enhancement. This model achieves a well-balanced trade-off among accuracy, inference speed, and computational resource consumption, offering high adaptability and practical value for real-world deployment.

 doi:10.20944/preprints202508.1730.v1

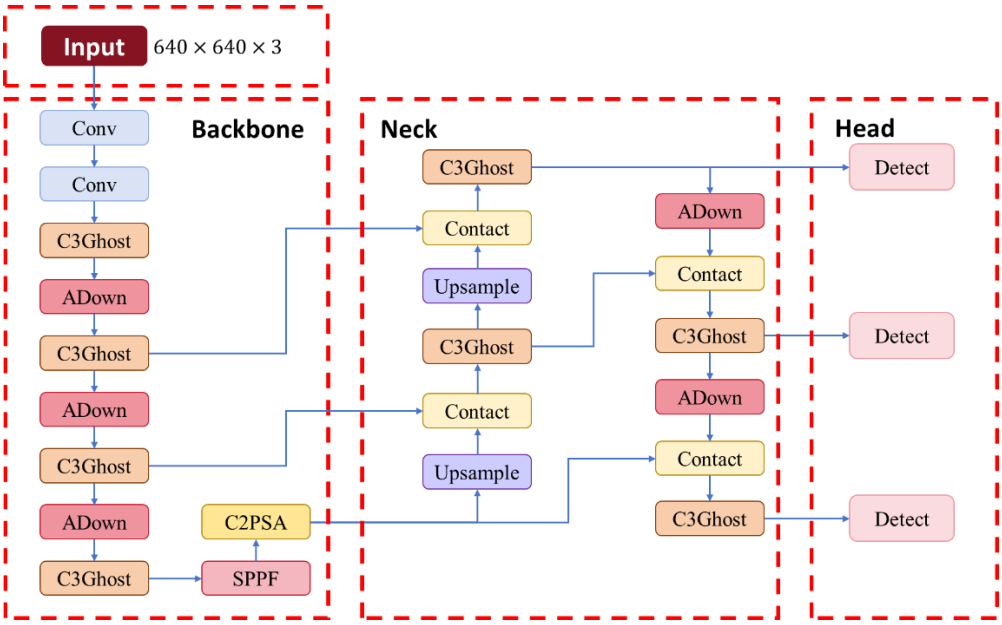


Figure 5. ADG-YOLO algorithm structure.

2.2. Model Conversion and Edge Deployment

Considering factors such as device size, weight, scalability, and cost, this study selects the LubanCat 4 development board as the deployment platform for the ADG-YOLO model. The board is equipped with the Rockchip RK3588S processor and integrates an AI acceleration NPU capable of INT4, INT8, and INT16 mixed-precision computing, with a peak performance of up to 6 TOPS. It includes 4 GB of onboard memory and supports peripheral interfaces such as mini HDMI output and USB camera input, with an overall weight of approximately 62 grams.

Given the limited computing capacity of the CPU, it is necessary to maximize inference efficiency by converting the model from its original .pt format (trained with PyTorch) to the .rknn format compatible with the NPU. The conversion pipeline proceeds as follows: first, the trained model is exported to the ONNX format using the torch.onnx.export interface in PyTorch; then, the RKNN Toolkit is used to convert the ONNX model into RKNN format. The overall conversion process is illustrated in Figure 6.

After conversion, the model is deployed onto the development board to enable real-time detection of UAV targets from live video input via the connected USB camera. With the aid of hardware acceleration provided by the NPU, the system is capable of maintaining a high frame rate and fast response speed while ensuring detection accuracy, thereby fulfilling the dual demands of real-time performance and lightweight deployment in practical application scenarios.

Figure 6. ADG-YOLO Model Conversion Process Diagram.

2.3. Target Monitoring Based on YOLOv11 Detection and EKF Tracking

In this study, we propose a method that integrates the YOLO object detection algorithm with the Extended Kalman Filter (EKF) for target monitoring in dynamic scenarios. The YOLO model is employed to extract bounding box information from consecutive image frames in real time, including the center position and size parameters of the detected targets. To enable temporal filtering and motion trajectory prediction of the detected objects, the target state is modeled as a six-dimensional

 doi:10.20944/preprints202508.1730.v1

vector $x = [c_x, c_y, v_x, v_y, w, h]^T$, consisting of the center coordinates (c_x,c_y), the horizontal and vertical velocity components (v_x,v_y), and the width w and height h of the bounding box. Considering that targets typically follow a constant velocity linear motion within short time intervals and that their size changes are relatively stable, a state transition model is formulated under this assumption. The corresponding state transition matrix is defined as follows:

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The observations provided by YOLO are the bounding box parameters of the detected target in the image, represented as $[c_x, c_y, w, h]^T$. The correspondence between these observations and the system state vector is modeled through an observation matrix, which is expressed as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The execution process of the Extended Kalman Filter (EKF) consists of two stages: prediction and update. In the prediction stage, the target state and its covariance are estimated based on the current state and the state transition model, as expressed by:

$$\begin{cases} x' = F \cdot x, \\ P' = F \cdot P \cdot F^T + Q \end{cases} \tag{3}$$

Here, Q denotes the process noise covariance matrix, and P represents the observation noise covariance matrix.Upon receiving a new observation zzz from the YOLO algorithm, the update stage is performed as follows:

Residual computation:

$$y = z - H \cdot x' \tag{4}$$

Kalman gain computation:

$$K = P' \cdot H^T \cdot (H \cdot P' \cdot H^T + R)^{-1} \tag{5}$$

State update:

$$x = x' + K \cdot y \tag{6}$$

Covariance update:

$$P = (I - K \cdot H) \cdot P' \tag{7}$$

Here, I denotes the identity matrix.

The integration of the EKF module helps to mitigate the potential localization fluctuations and occasional false detections that may occur in YOLO's single-frame inference. This facilitates smoother

 doi:10.20944/preprints202508.1730.v1

target position estimation and further enhances the tracking consistency and robustness of the system in multi-frame processing scenarios.

2.4. Monocular Ranging for UAVs Using Similar Triangles

Figure 7 illustrates the UAV target detection results based on the YOLO model, where the red bounding boxes accurately locate and outline the position and size of the targets in the monocular images. The pixel width of the bounding box is denoted as p, representing the projected size of the target in the image, which serves as a key parameter for subsequent distance estimation. Neglecting lens distortion, and based on the principle of similar triangles, when the actual width of the target is W, the camera focal length is f, and the physical size of a single pixel on the image sensor is s, the actual projected width w of the target on the imaging plane can be expressed as:

$$w = p \cdot s \tag{8}$$

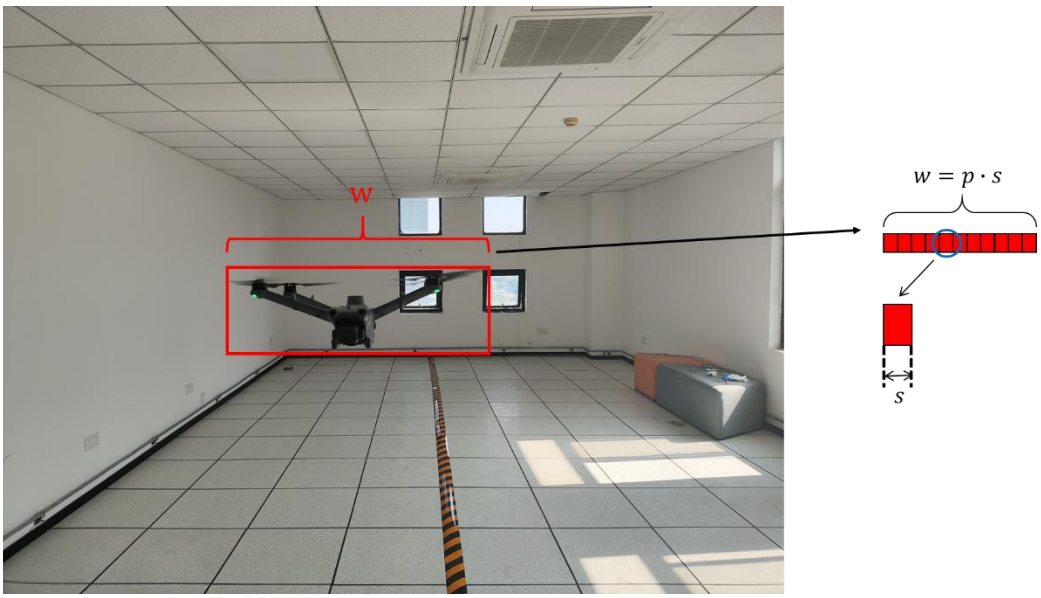


Figure 7. Drone Projection Width w Diagram.

As shown in Figure 8, when the target plane Ω_1 is perpendicular to the optical axis of the camera, the imaging process can be abstracted as two similar triangles, which satisfy the following proportional relationship:

$$\frac{W}{D} = \frac{w}{f} \tag{9}$$

Here, D denotes the distance from the target to the camera along the optical axis. Based on this relationship, the formula for computing the target distance is derived as follows:

$$D = \frac{W \cdot f}{p \cdot s} \tag{10}$$

In this study, the training dataset comprises three different types of UAVs, each associated with a known physical width W_n. The YOLO model not only outputs the bounding box coordinates but also possesses target classification capability, enabling precise identification of the specific UAV type. Once the target type is detected, the corresponding W_n is automatically selected and substituted into the distance estimation formula (10), thereby enhancing the accuracy and generalizability of the distance measurement.

 doi:10.20944/preprints202508.1730.v1

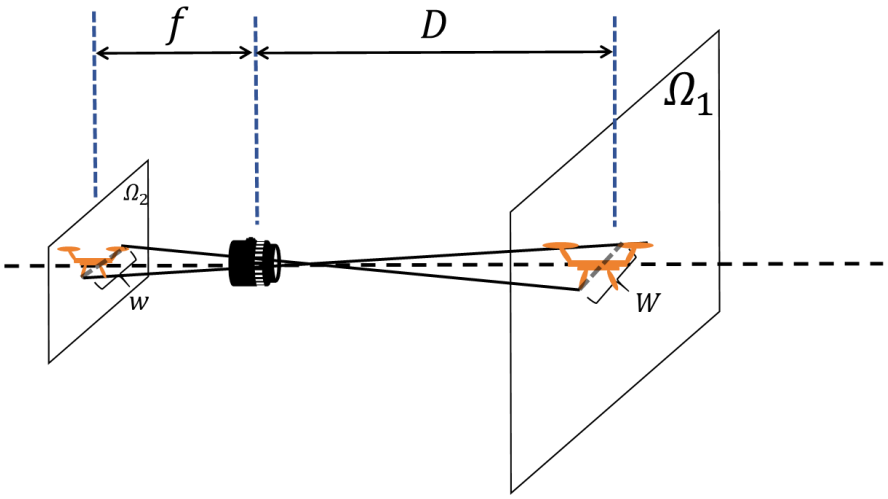


Figure 8. UAV Projection Width w Diagram.

3. Experiment

3.1. Dataset

A comprehensive UAV detection dataset was constructed for this study, comprising a total of 5,343 high-resolution images. This dataset integrates two subsets: a custom-target subset with 2,670 images and a generalization subset with 2,664 images. The custom subset focuses on three specific UAV models: DJI 3TD, with 943 training and 254 testing images (labeled as drone1); DJI NEO, with 739 training and 170 testing images (drone2); and DWI-S811, with 454 training and 110 testing images (drone3). All images in this subset were captured under strictly controlled conditions, with target distances ranging from 5 to 30 meters and 360-degree coverage, to reflect variations in object appearance under different perspectives and distances.

The generalization subset was collected from publicly available multirotor UAV image resources published on the CSDN object detection platform. It contains quadrotor and hexarotor UAVs from popular brands such as DJI and Autel, appearing in diverse environments including urban buildings, rural landscapes, highways, and industrial areas. Additionally, the images cover challenging weather conditions such as bright sunlight, fog, and rainfall. All images were annotated using professional tools in compliance with the YOLOv11 format, including normalized center coordinates (x,y) and relative width w and height h of each bounding box.

Figure 9 shows representative images of the three specific UAV models from the custom subset, highlighting multi-angle and multi-distance variations. Figure 10 presents sample images from the generalization subset, illustrating environmental and visual diversity. Table 1 provides an overview of the dataset composition, including the number of training and testing images for each subset and the corresponding label formats.

A differentiated sampling strategy was used to partition the dataset. The custom subset contains 2,136 training images and 534 testing images, while the generalization subset includes 2,363 training and 301 testing images. To enhance detection performance on specific UAV types, the test set was intentionally supplemented with additional samples of DJI 3TD, DJI NEO, and DWI-S811. This allows the model to better learn and evaluate fine-grained appearance features of these UAVs, contributing to improved detection accuracy and robustness.

 doi:10.20944/preprints202508.1730.v1

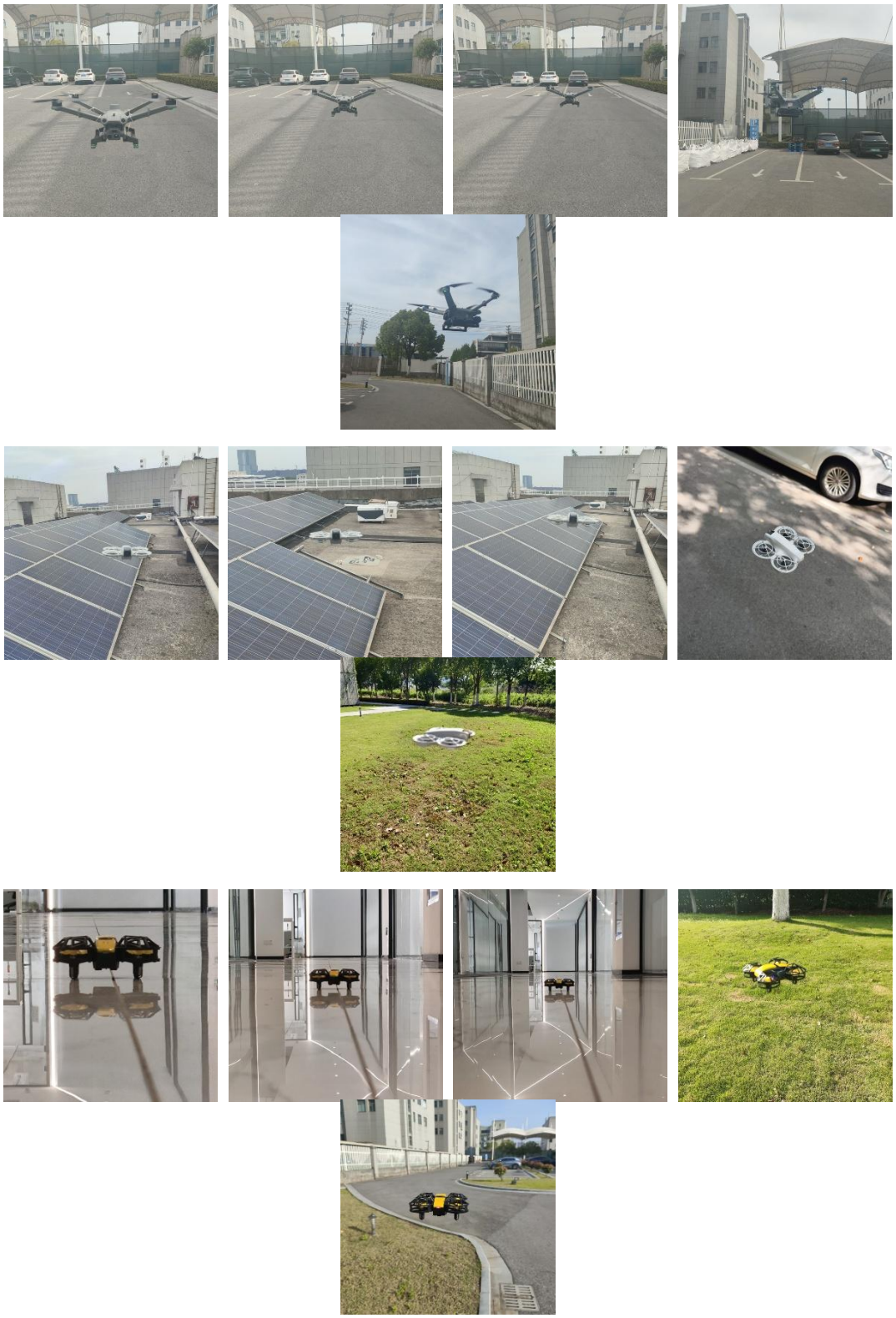


Figure 9. Typical Samples of Custom Subset UAVs.

 doi:10.20944/preprints202508.1730.v1

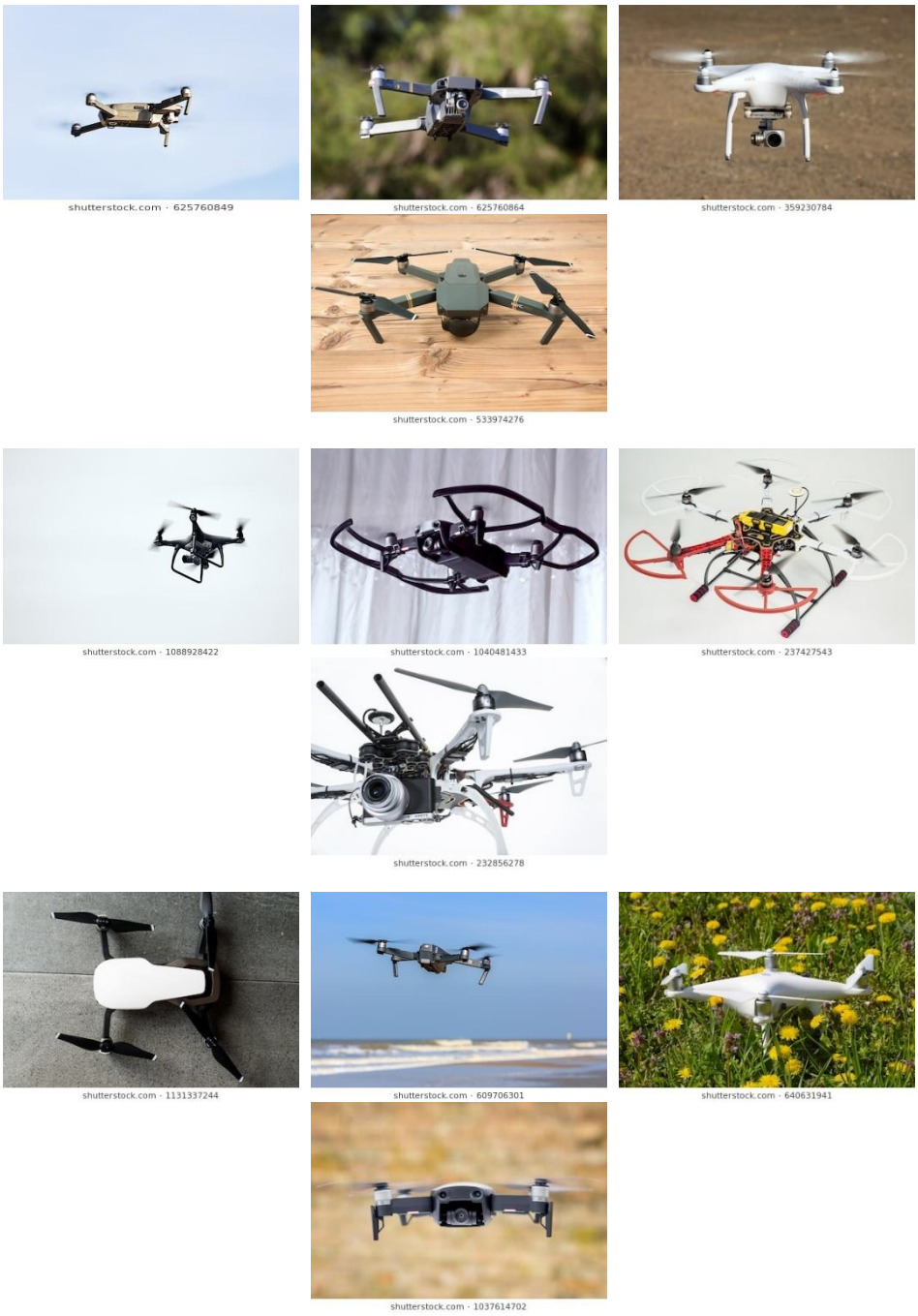


Figure 10. Samples of Multirotor UAVs in Generalization Subset.

Table 1. This is a table. Tables should be placed in the main text near to the first time they are cited.

Subset	Total Images	Training Set	Testing Set	Drone Models	Annotation Standard
Custom-Target	2670	2136	534	DJI 3TD / NEO / DWI-S811	drone1/drone2/drone3 labels
Generalization	2664	2363	301	Multi-brand quad/hexa-rotor	drone label
Total	5334	4499	835	-	YOLO format

3.2. Experimental Environment and Experimental Parametes

 doi:10.20944/preprints202508.1730.v1

To evaluate the performance of the proposed ADG-YOLO model in UAV object detection tasks, the model was trained on the custom dataset described in Section 3.1. Comparative experiments were conducted against several representative algorithms under identical training configurations. To ensure the reproducibility and fairness of the results, the experimental environment settings and training parameters are summarized in Tables 2 and 3, respectively.

Table 2. Configuration experimental environment.

Environment	Parameters
CPU	Intel(R) Xeon(R) Platinum 8358P
GPU	RTX 3090
GPU memory size	90GB
Operating system	ubuntu18.04
Language	Python 3.8
Frame	PyTorch 1.8.1
CUDA version	CUDA 11.1

Table 3. Training parameters setting.

Parameters	Setup
Epochs	500
Input image size	640×640
Batch size	16
Optimizer	SGD
Initial learning rate	0.01

3.3. Evaluation Metrics

In target detection tasks, the Mean Average Precision (*mAP*) is widely employed to evaluate the detection performance of a model. Based on the model's prediction results, two key metrics can be further computed: Precision (*P*) and Recall (*R*). Precision measures the proportion of correctly predicted targets among all samples identified as targets by the model, whereas Recall reflects the model's ability to detect actual targets, defined as the ratio of correctly detected targets to all true targets. Typically, there exists a trade-off between Precision and Recall, where improving one may lead to the reduction of the other. Therefore, a Precision–Recall (*PR*) curve is plotted to comprehensively analyze the detection performance of the model. For a single category, the Average Precision (*AP*) is defined as the area under the *PR* curve, which is calculated as follows:

$$AP = \int_0^1 P(R)dR \tag{11}$$

In practical computations, a discrete approximation method is typically employed:

$$AP = \frac{1}{m}\sum_{i=1}^{m} P(R_i) \tag{12}$$

Here, R_i represents the sampled recall values, and $P(R_i)$ denotes the corresponding precision at each recall point. The calculation of *AP* varies slightly across different datasets. For instance, the PASCAL VOC dataset adopts an interpolation method based on 11 fixed recall points, whereas the COCO evaluation protocol computes the mean over all recall points. For multi-class object detection, the Mean Average Precision (*mAP*) is defined as the mean *AP* across all categories:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{13}$$

 doi:10.20944/preprints202508.1730.v1

Here, N denotes the total number of categories, and AP_i represents the Average Precision of the *i-th* target category.

In practical object detection scenarios, in addition to model accuracy, the actual runtime speed of the model is also of significant concern. Frames Per Second (*FPS*) is a key metric for evaluating the runtime efficiency of a model, representing the number of image frames the model can process per second. The *FPS* can be calculated as follows:

$$FPS = \frac{N_s}{T} \tag{14}$$

Here, N_s denotes the total number of processed frames, and T represents the total processing time in seconds. A higher *FPS* indicates that the model can process input images more rapidly, thereby enhancing its capability for real-time detection.

3.4. Model Performance Analysis

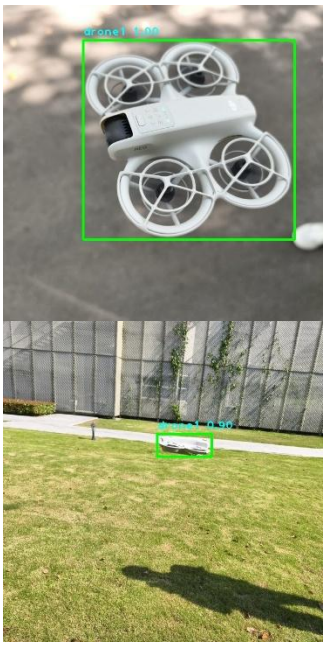
To evaluate the overall performance of the ADG-YOLO model in UAV target detection, three mainstream lightweight models—YOLOv5s, YOLOv8n, and YOLOv11n—are selected as baseline comparisons. The comparison dimensions include model parameters, computational complexity (GFLOPs), precision, recall, and FPS. All FPS values are obtained through real-world measurements on the Lubancat4 edge computing development board. The experimental results are summarized in Table 4.

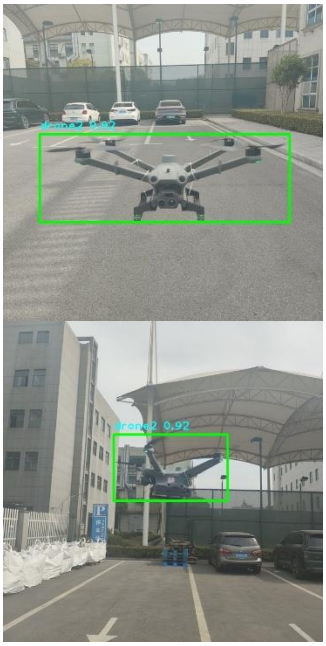
ADG-YOLO contains only 1.77M parameters and requires 5.7 GFLOPs, representing a substantial simplification compared to YOLOv5s, which has 7.02M parameters and 15.8 GFLOPs. It is also more lightweight than YOLOv8n (3.00M parameters, 8.1 GFLOPs) and YOLOv11n (2.58M parameters, 6.3 GFLOPs), making it well-suited for deployment on resource-constrained edge platforms. In terms of detection accuracy, ADG-YOLO achieves 98.4% mAP0.5 and 85.2% mAP0.5:0.95, slightly outperforming the other models. Notably, its mAP0.5:0.95 is significantly higher than that of YOLOv5s and YOLOv8n (both 84.2%), and comparable to YOLOv11n (85.3%), demonstrating strong robustness. For inference speed, ADG-YOLO reaches 27 FPS, which significantly exceeds YOLOv5s (15 FPS), YOLOv8n (12 FPS), and YOLOv11n (10 FPS), thereby achieving an effective balance between model compactness and real-time performance.

As illustrated in Figure 11, the proposed ADG-YOLO model demonstrates robust detection capability for UAV targets under complex backgrounds, various viewing angles, and different lighting conditions.

In summary, ADG-YOLO achieves an optimal trade-off among accuracy, inference speed, and resource consumption, making it particularly suitable for real-time UAV detection tasks in computationally constrained environments. The model also exhibits strong engineering adaptability for practical deployment.

 doi:10.20944/preprints202508.1730.v1





 doi:10.20944/preprints202508.1730.v1

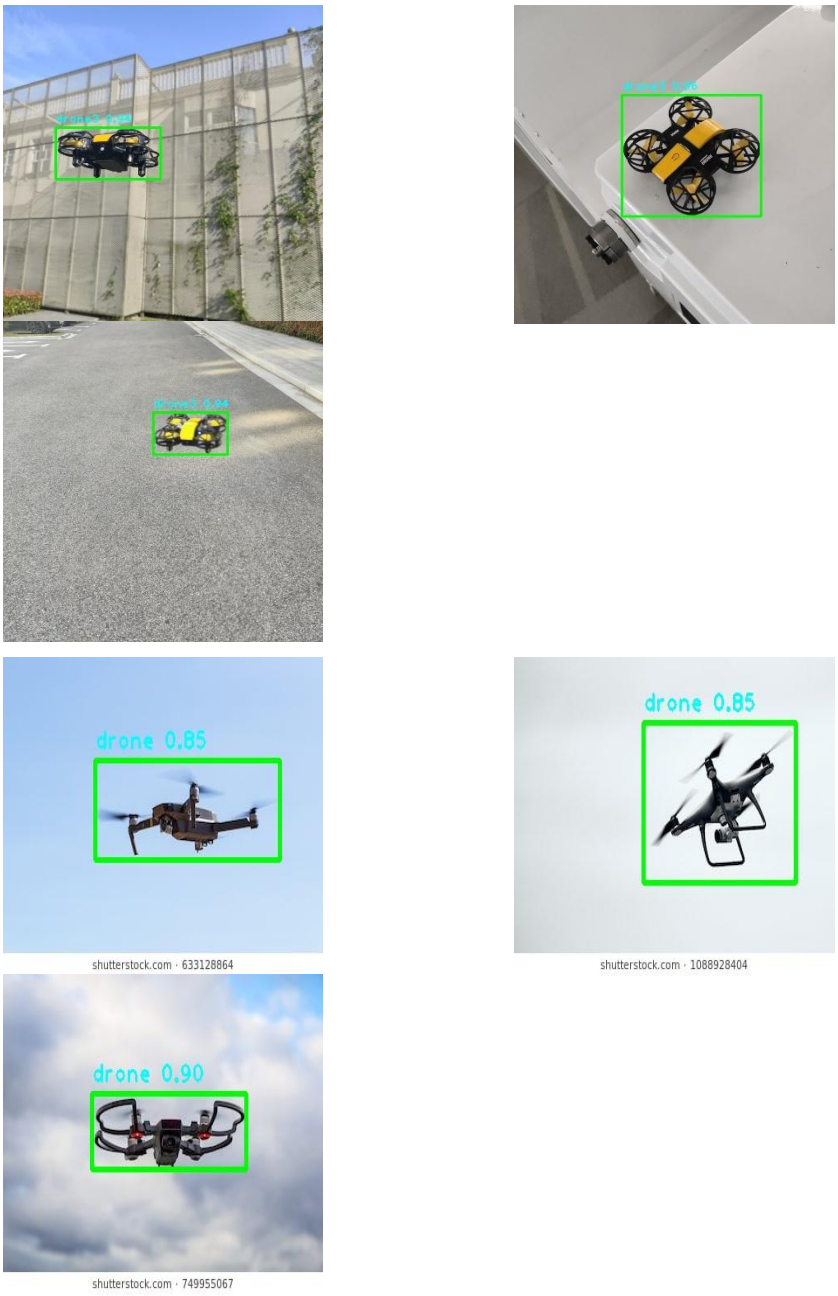


Figure 11. Detection Results of the ADG-YOLO Model.

Table 4. Comparison of Lightweight Detection Models in Terms of Model Size, Accuracy, and Inference Speed.

Model	Params(M)	GFLOPs(G)	$mAP_{0.5}$	$mAP_{0.5:0.95}$	FPS
YOLOv5s	7.02	15.8	98.2%	84.2%	15
YOLOv8n	3.00	8.1	98.2%	84.2%	12
YOLOv11n	2.58	6.3	98.2%	85.3%	10
ADG-YOLO	1.77	5.7	98.4%	85.2%	27

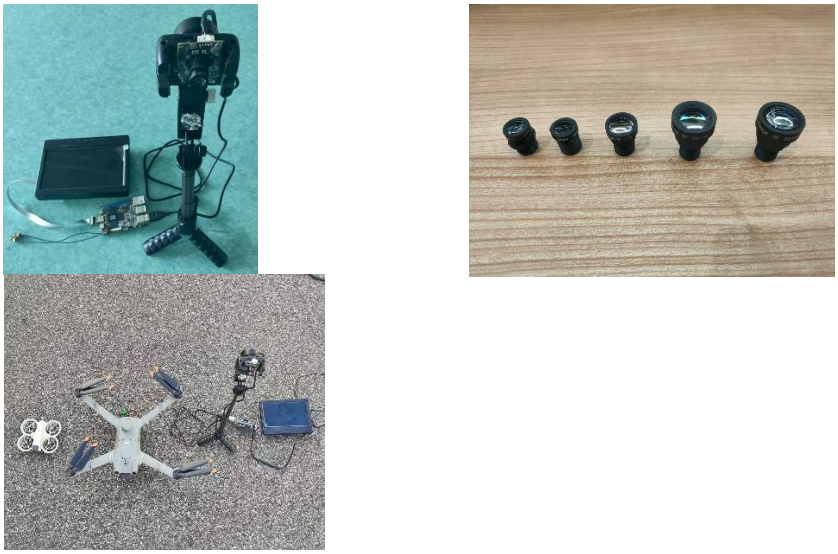
4. Target Distance Estimation Experiment

4.1. Experimental Setup

To verify the accuracy of UAV altitude measurement, the ADG-YOLO model was converted into the .rknn format and deployed on the Lubancat 4 development board. The target UAVs used in the distance measurement experiments were DJI 3TD and DJI NEO, both known for their flight stability. Visual data were captured using a Raspberry Pi USB camera module, which was connected to the

 doi:10.20944/preprints202508.1730.v1

development board via a USB cable. As shown in Figure 9(b), five lenses with focal lengths of 12 mm, 16 mm, 25 mm, 35 mm, and 50 mm were selected for distance measurement experiments at various ranges. The captured images and corresponding distance information were displayed on a YCXSQ-10 display screen, which features a 10-inch size and a resolution of 1920×1080 pixels. As illustrated in Figure 12, the camera was mounted on a tripod, while the display screen was connected to the development board via an HDMI cable for real-time visualization of detection results and distance measurements.



(a) (b) (c)

Figure 12. Experimental Setup. (a) Distance Measurement Platform. (b) The lens used in this experiment. (c) UAVs used for Experiment.

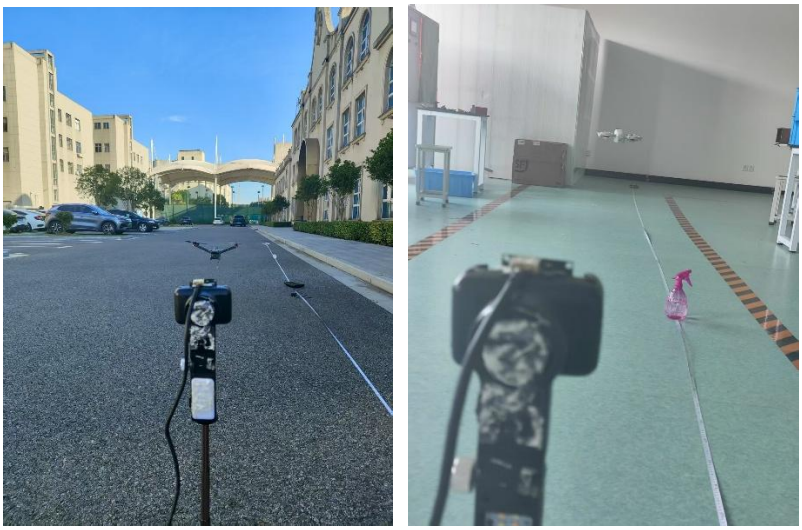
4.2. Distance Measurement for UAV Targets

In our UAV distance measurement experiments, two drone models were selected as test targets: the DJI 3TD and the DJI NEO, with rotor spans of 62 cm and 15 cm, respectively. To assess the adaptability of the proposed measurement method across different UAV sizes and operational environments, each model was tested under distinct conditions.

The DJI 3TD, featuring strong wind resistance, was used for outdoor experiments. During testing, the UAV was manually flown along a straight path aligned with the camera's optical axis, maintaining a level attitude to ensure stable visual features and reduce interference from yaw or pitch. As shown in Figure 11(a), the test was conducted in an open outdoor area, where a standard measuring tape was laid along the flight path to mark reference distance points. The camera system was fixed on a stationary tripod, and images were captured at each distance for subsequent evaluation.

In contrast, the DJI NEO, due to its smaller size and lower wind tolerance, was tested indoors to ensure stable hovering. As shown in Figure 13(b), the indoor experiment was conducted in a closed room, with the measuring tape placed along a straight line. The UAV hovered at various predefined points to collect image samples at known distances. The controlled indoor environment—with stable lighting and minimal airflow—provided favorable conditions for high-precision distance calibration.

 doi:10.20944/preprints202508.1730.v1



(a) (b)

Figure 13. Distance Measurement Experimental Scene. (a)DJI 3TD. (b)DJI NEO.

In this study, distance estimation was performed using the principle of similar triangles. Based on the known physical width of the UAV and the width of the corresponding bounding box output by the detection model, the distance between the UAV and the camera was calculated. All estimated distances were compared with ground-truth values obtained from physical measurements using a tape measure. This comparison was conducted to evaluate the effectiveness and stability of the proposed distance estimation method under real-world conditions.

To further quantify the accuracy of the system in UAV distance estimation, relative error was introduced as a performance evaluation metric. By calculating the ratio between the prediction error and the ground-truth distance, this metric reflects the overall precision of the proposed distance measurement method. The formula for computing the relative error e_{mea} is given in Eq. (11).

$$\mathrm{e}_{\mathrm{mea}} = \frac{\left|d_{real} - d_{mea}\right|}{d_{real}} \times 100\% \qquad (15)$$

The distance estimation results for the DJI NEO are summarized in Table 5, obtained using a 12 mm lens. The experimental results for the DJI 3TD are presented in Table 6, based on tests conducted with five different lens focal lengths. In addition, representative experimental images are provided to visually demonstrate the distance measurement process and outcomes—Figure 14 shows the measurement setup for the DJI NEO, while Figure 15 presents the measurement setup for the DJI 3TD.

The UAV distance estimation method based on the principle of similar triangles demonstrated excellent performance in real-world scenarios. As shown in Tables 5 and 6, the DJI NEO achieved an average relative error of 4.18% across 10 test cases within the range of 0.5 to 5 meters. For the DJI 3TD, a total of 45 measurements across various focal lengths and distances ranging from 2 to 50 meters resulted in a combined average relative error of only 2.40%. High accuracy was maintained across all test distances, with the maximum error not exceeding 12.33%. Notably, even at a distance of 50 meters, the method achieved a minimal error of 0.26%, further validating the effectiveness and stability of the proposed approach.

 doi:10.20944/preprints202508.1730.v1

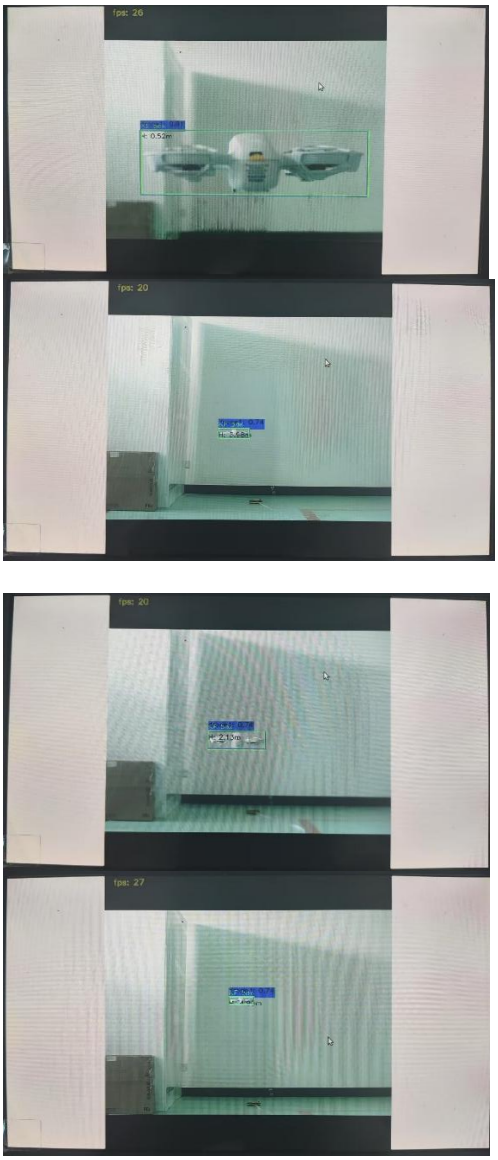


Figure 14. Distance Measurement Scene Of DJI NEO.

 doi:10.20944/preprints202508.1730.v1

Figure 15. Distance Measurement Scene Of DJI 3TD.

 doi:10.20944/preprints202508.1730.v1

Table 5. Distance Measurement of DJI NEO(f=12mm).

Experiment ID	$d_{real}(m)$	$d_{mea}(m)$	e_{mea}
1	0.5	0.52	4%
2	1	0.95	5%
3	1.5	1.64	9.3%
4	2	2.13	6.5%
5	2.5	2.36	5.6%
6	3	3.05	1.7%
7	3.5	3.61	3.1%
8	4	3.98	0.5%
9	4.5	4.36	3.1%
10	5	4.90	2%

Table 6. Distance Measurement of DJI 3TD.

Experiment ID	focal length (㎜)	$d_{real}(m)$	$d_{mea}(m)$	e_{mea}
1	12	2	2.15	7.50%
2	12	3	3.37	12.33%
3	12	4	4.28	7.00%
4	12	5	4.99	0.20%
5	12	6	5.85	2.50%
6	12	7	7.58	8.29%
7	12	8	8.07	0.88%
8	12	9	8.81	2.11%
9	16	11	10.59	3.73%
10	16	12	11.62	3.17%
11	16	13	13.01	0.08%
12	16	14	13.73	1.93%
13	16	15	14.79	1.40%
14	16	16	16.35	2.19%
15	16	17	16.87	0.76%
16	16	18	18.02	0.11%
17	16	19	18.33	3.53%
18	25	21	21.06	0.29%
19	25	22	22.36	1.64%
20	25	23	22.59	1.78%
21	25	24	23.66	1.42%
22	25	25	25.39	1.56%
23	25	26	26.78	3.00%
24	25	27	27.66	2.44%
25	25	28	28.21	0.75%
26	25	29	29.33	1.14%
27	35	31	30.52	1.55%
28	35	32	31.78	0.69%
29	35	33	33.26	0.79%
30	35	34	33.70	0.88%
31	35	35	34.96	0.11%
32	35	36	35.78	0.61%
33	35	37	38.21	3.27%
34	35	38	38.62	1.63%
35	35	39	39.45	1.15%
36	50	41	40.73	0.66%

 doi:10.20944/preprints202508.1730.v1

37	50	42	42.56	1.33%
38	50	43	43.34	0.79%
39	50	44	43.97	0.07%
40	50	45	45.26	0.58%
41	50	46	45.69	0.67%
42	50	47	46.91	0.19%

5. Discussion

The proposed ADG-YOLO framework demonstrates significant advancements in real-time UAV detection and distance estimation on edge devices. Nonetheless, several challenges remain that warrant further investigation to enhance its scalability and real-world applicability. First, although the current custom dataset (5,343 images) includes three UAV models across diverse backgrounds, its limited scope hinders generalization. Future work should focus on building a large-scale, open-source UAV dataset covering various drone types (e.g., quadrotors, hexarotors, fixed-wing), sizes (micro to commercial), and environmental conditions (e.g., night, adverse weather, swarm operations), particularly under low-SNR settings prone to false positives. Collaborative data collection across platforms may expedite this process.

Second, current distance estimation depends on known UAV dimensions (e.g., DJI 3TD: 62 cm, DJI NEO: 15 cm), which limits its flexibility in handling unknown models. Future research should explore multi-model support through an onboard UAV identification module containing pre-calibrated physical parameters. Additionally, geometry-independent approaches, such as monocular depth estimation fused with detection outputs, offer promising alternatives that remove dependency on prior shape knowledge. Adaptive focal length calibration should also be considered to reduce measurement errors in long-range scenarios, particularly beyond 50 meters.

Third, while ADG-YOLO achieves 27 FPS on the Lubancat4 edge device, its practical deployment on UAVs introduces additional challenges. These include optimizing the model for ultra-low-power processors, ensuring efficient thermal dissipation during extended operation, and compensating for dynamic motion via IMU and EKF integration to stabilize detection during rapid pitch or yaw movements. Moreover, expanding the system to air-to-air detection, such as in drone swarm environments, requires altitude-invariant ranging models and training strategies that are robust to occlusion. Finally, achieving sub-20 ms end-to-end latency is essential for enabling closed-loop tasks such as autonomous interception and cooperative formation flight.

6. Conclusions

This study proposes ADG-YOLO, a lightweight and efficient framework for real-time UAV target detection and distance estimation on edge devices. The framework integrates multiple key innovations: (1) a computationally optimized architecture that incorporates C3Ghost modules and ADown layers, reducing model parameters to 1.77 M and GFLOPs to 5.7, while maintaining high detection accuracy with 98.4% $mAP_{0.5}$; (2) an EKF-based tracking mechanism that significantly improves detection stability in dynamic environments; (3) a monocular distance estimation method based on similarity triangle theory, which achieves average relative errors ranging from 2.40% to 4.18% over distances of 0.5–50 meters; and (4) successful real-time deployment on the Lubancat4 edge platform (RK3588S NPU) at 27 FPS, demonstrating its practical applicability in resource-constrained settings.

Overall, ADG-YOLO effectively balances detection accuracy and computational efficiency, bridging the gap between advanced perception and edge deployment for UAV-based applications. Future work will focus on expanding large-scale UAV datasets, enabling generalized ranging for unknown UAV models, and facilitating deployment in autonomous aerial systems to support next-generation capabilities in both military and commercial UAV operations..

References

 doi:10.20944/preprints202508.1730.v1

1. Lin Y H, Joubert D A, Kaeser S, et al. Field deployment of Wolbachia-infected Aedes aegypti using uncrewed aerial vehicle [J]. Science Robotics, 2024, 9(92): eadk7913.
2. Xin Zhou, et al., "Swarm of micro flying robots in the wild," Science Robotics, vol. 7, eabm5954, 2022. DOI: 10.1126/scirobotics.abm5954.
3. Han J, Yan Y, Zhang B. Towards Efficient Multi-UAV Air Combat: An Intention Inference and Sparse Transmission Based Multi-Agent Reinforcement Learning Algorithm [J]. IEEE Transactions on Artificial Intelligence, 2025.
4. Karimov C Y. THE ROLE OF UNMANNED AIRCRAFT VEHİCLES IN THE RUSSIAN-UKRAINIAN WAR [J]. Endless light in science, 2025 (30 апрель ELB): 83-89.
5. Wennerholm D. Above the trenches: Russian military lessons learned about drone warfare from Ukraine [J]. 2025.
6. Tang Z, Ma H, Qu Y, et al. UAV Detection with Passive Radar: Algorithms, Applications, and Challenges [J]. Drones, 2025, 9(1): 76.
7. Seidaliyeva U, Ilipbayeva L, Utebayeva D, et al. LiDAR Technology for UAV Detection: From Fundamentals and Operational Principles to Advanced Detection and Classification Techniques [J]. Sensors, 2025, 25(9): 2757.
8. Qiu Z, Lu Y, Qiu Z. Review of ultrasonic ranging methods and their current challenges [J]. Micromachines, 2022, 13(4): 520.
9. Rahmaniar W, Wang W J, Caesarendra W, et al. Distance measurement of unmanned aerial vehicles using vision-based systems in unknown environments [J]. Electronics, 2021, 10(14): 1647.
10. Tian X, Liu R, Wang Z, et al. High quality 3D reconstruction based on fusion of polarization imaging and binocular stereo vision [J]. Information Fusion, 2022, 77: 19-28.
11. Tang Y, Zhou H, Wang H, et al. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision [J]. Expert systems with applications, 2023, 211: 118573.
12. Bao D, Wang P. Vehicle distance detection based on monocular vision [C]//2016 International Conference on Progress in Informatics and Computing (PIC). IEEE, 2016: 187-191.
13. Ali A A, Hussein H A. Distance estimation and vehicle position detection based on monocular camera [C]//2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA). IEEE, 2016: 1-4.
14. Liu F, Shen C, Lin G. Deep convolutional neural fields for depth estimation from a single image [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 5162-5170.
15. Li J, Klein R, Yao A. A two-streamed network for estimating fine-scaled depth maps from single rgb images [C]//Proceedings of the IEEE international conference on computer vision. 2017: 3372-3380.
16. Eigen D, Fergus R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture [C]//Proceedings of the IEEE international conference on computer vision. 2015: 2650-2658.
17. Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network [J]. Advances in neural information processing systems, 2014, 27.
18. Jiao J, Cao Y, Song Y, et al. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss [C]//Proceedings of the European conference on computer vision (ECCV). 2018: 53-69.
19. Zhe T, Huang L, Wu Q, et al. Inter-vehicle distance estimation method based on monocular vision using 3D detection [J]. IEEE transactions on vehicular technology, 2020, 69(5): 4907-4919.
20. Mallot H A, Bülthoff H H, Little J J, et al. Inverse perspective mapping simplifies optical flow computation and obstacle detection [J]. Biological cybernetics, 1991, 64(3): 177-185.
21. Tuohy S, O'Cualain D, Jones E, et al. Distance determination for an automobile environment using inverse perspective mapping in OpenCV [C]//IET Irish signals and systems conference (ISSC 2010). IET, 2010: 100-105.

 doi:10.20944/preprints202508.1730.v1

22. Wongsaree P, Sinchai S, Wardkein P, et al. Distance detection technique using enhancing inverse perspective mapping [C]//2018 3rd International Conference on Computer and Communication Systems (ICCCS). IEEE, 2018: 217-221.
23. Huang L, Zhe T, Wu J, et al. Robust inter-vehicle distance estimation method based on monocular vision [J]. IEEE Access, 2019, 7: 46059-46070.
24. Qi S H, Li J, Sun Z P, et al. Distance estimation of monocular based on vehicle pose information [C]//Journal of Physics: Conference Series. IOP Publishing, 2019, 1168(3): 032040.
25. Jiafa M, Wei H, Weiguo S. Target distance measurement method using monocular vision [J]. IET Image Processing, 2020, 14(13): 3181-3187.
26. Yang R, Yu S, Yao Q, et al. Vehicle Distance Measurement Method of Two-Way Two-Lane Roads Based on Monocular Vision [J]. Applied Sciences, 2023, 13(6): 3468.
27. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
28. Girshick R. Fast r-cnn [C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
29. Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks [J]. Advances in neural information processing systems, 2015, 28.
30. Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
31. Redmon J, Farhadi A. YOLO9000: better, faster, stronger [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.
32. Redmon J, Farhadi A. Yolov3: An incremental improvement [J]. arXiv preprint arXiv:1804.02767, 2018.
33. Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection [J]. arXiv preprint arXiv:2004.10934, 2020.
34. Wu W, Liu H, Li L, et al. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image [J]. PloS one, 2021, 16(10): e0259283.
35. Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications [J]. arXiv preprint arXiv:2209.02976, 2022.
36. 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 [C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475.
37. Sohan M, Sai Ram T, Rami Reddy C V. A review on yolov8 and its advancements [C]//International Conference on Data Intelligence and Cognitive Informatics. Springer, Singapore, 2024: 529-545.
38. Wang C Y, Yeh I H, Mark Liao H Y. Yolov9: Learning what you want to learn using programmable gradient information [C]//European conference on computer vision. Cham: Springer Nature Switzerland, 2024: 1-21.
39. Wang A, Chen H, Liu L, et al. Yolov10: Real-time end-to-end object detection [J]. Advances in Neural Information Processing Systems, 2024, 37: 107984-108011.
40. Khanam R, Hussain M. Yolov11: An overview of the key architectural enhancements [J]. arXiv preprint arXiv:2410.17725, 2024.
41. Tian Y, Ye Q, Doermann D. Yolov12: Attention-centric real-time object detectors [J]. arXiv preprint arXiv:2502.12524, 2025.
42. Cheng Q, Wang Y, He W, et al. Lightweight air-to-air unmanned aerial vehicle target detection model [J]. Scientific Reports, 2024, 14(1): 2609.
43. Su J, Qin Y, Jia Z, et al. MPE-YOLO: enhanced small target detection in aerial imaging [J]. Scientific Reports, 2024, 14(1): 17799.
44. Wang C, Han Y, Yang C, et al. CF-YOLO for small target detection in drone imagery based on YOLOv11 algorithm [J]. Scientific Reports, 2025, 15(1): 1-18.
45. Zhou S, Yang L, Liu H, et al. Improved YOLO for long range detection of small drones [J]. Scientific Reports, 2025, 15(1): 12280.

 doi:10.20944/preprints202508.1730.v1

46. Kanjalkar P, Kinhikar S, Zagade A, et al. Intelligent Surveillance Tower for Detection of the Drone from the Other Aerial Objects Using Deep Learning [C]//International Conference on Information Science and Applications. Singapore: Springer Nature Singapore, 2023: 39-51.
47. Han K, Wang Y, Xu C, et al. GhostNets on heterogeneous devices via cheap operations [J]. International Journal of Computer Vision, 2022, 130(4): 1050-1069.
48. Ji C L, Yu T, Gao P, et al. Yolo-tla: an efficient and lightweight small object detection model based on YOLOv5 [J]. Journal of Real-Time Image Processing, 2024, 21(4): 141.
49. Fang S, Chen C, Li Z, et al. YOLO-ADual: A Lightweight Traffic Sign Detection Model for a Mobile Driving System [J]. World Electric Vehicle Journal, 2024, 15(7): 323.