Article

Not peer-reviewed version

Low-Cost and Lightweight Campus All-Weather Dual-Optical UAV Detection and Active Warning System

Yingrui Bing , Yanan Wang , Jiali Xu , Changqing Cao *

Posted Date: 18 May 2026

doi: 10.20944/preprints202605.1117.v1

Keywords: UAV detection; dual-spectral fusion; Raspberry Pi; YOLOv8; lightweight; edge computing

Article

Low-Cost and Lightweight Campus All-Weather Dual-Optical UAV Detection and Active Warning System

Yingrui Bing, Yanan Wang, Jiali Xu and Changqing Cao *

School of Optoelectronic Engineering, Xidian University, Xi'an 710071,China

* Correspondence: chqcao@mail.xidian.edu.cn; Tel.: +86-139-0918-6739

Abstract

To address the challenges of unauthorised drone flights in campus low-altitude security—where traditional detection equipment is costly and ineffective at night—this paper designs a lightweight, all-weather drone detection and early-warning system based on a Raspberry Pi edge computing platform and visible/infrared dual-spectrum fusion. The system uses an IMX219-77IR infrared camera that automatically switches imaging modes according to ambient brightness, achieving day-and-night continuous perception. A YOLOv8n model is compressed to 10 MB via channel pruning and knowledge distillation, reaching an inference speed of 85.2 ms/frame on the Raspberry Pi. A self-built campus drone dataset of 2,000 images (1,600 open-source + 400 self-collected) yields 97.16% precision, 93.79% recall, and 97.27% mAP50. A Flask backend and web map interface provide real-time alerts by polling every 2 seconds. Total hardware cost is below 1,500 yuan, more than 70% lower than traditional systems. Field tests (5–50 m) show daytime confidence >0.9, nighttime infrared confidence ≈0.88, false negative rate <8%, false positive rate <5%, and stable continuous operation. The project has won a university-level competition second prize and its software copyright application is under review, demonstrating strong practical value and promotion potential.

Keywords: UAV detection; dual-spectral fusion; Raspberry Pi; YOLOv8; lightweight; edge computing

1. Introduction

With the low-altitude economy having been elevated to a national strategic emerging industry, consumer-grade drones have rapidly gained widespread adoption in fields such as aerial photography, surveying, inspection, and logistics delivery, owing to their ease of operation, low cost, and manoeuvrability. At the same time, issues such as unauthorised drone flights, illegal intrusions, privacy violations through covert filming, and disruption of public order have become frequent occurrences in enclosed or semi-enclosed environments—including campuses, industrial parks, examination centres, and oil depots—posing significant threats to public safety, information privacy, and administrative order.Statistics indicate that in 2023, the number of safety incidents nationwide involving drone-related flight disruptions, personal injury and espionage rose by 35% year-on-year, with campuses emerging as a weak link in low-altitude security.Traditional drone detection methods primarily rely on radar detection, radio frequency monitoring or specialised visible-light imaging equipment, and generally suffer from three major shortcomings: firstly, high costs, with the unit price of professional detection equipment typically ranging from 5,000 to 20,000 yuan, far exceeding campus budgets and making large-scale deployment across multiple locations difficult; secondly, ineffectiveness at night, as pure visible-light solutions suffer a sharp drop in recognition rates in low-light, backlit or nocturnal environments, creating serious security blind spots;thirdly, complex deployment, which relies on servers, specialised cabling and technical personnel for configuration, resulting in high maintenance costs and lengthy implementation cycles. Therefore, the development of a low-cost, all-weather, lightweight and easy-to-deploy drone detection and early warning system holds significant practical and engineering value for enhancing campus low-altitude safety management capabilities.

In recent years, deep learning-based visual object detection technologies have advanced rapidly, with frameworks such as YOLO, SSD and Faster R-CNN being widely adopted for drone recognition tasks. Among these, the YOLO series has become the mainstream choice for embedded devices due to its fast detection speed, compact architecture and balanced accuracy [1] .As an ultra-lightweight variant, YOLOv8n offers significant advantages on edge devices with limited computational power; however, on low-power platforms such as the Raspberry Pi, it still faces bottlenecks including high inference latency, high memory consumption, and poor recognition of small targets. In terms of imaging perception, single-modality sensors have clear limitations: visible-light cameras provide clear texture details but are completely ineffective at night;infrared thermal imaging cameras can operate at night, but are costly, lack textural detail, and produce blurred object contours. Dual-spectral fusion technology, by combining the complementary advantages of visible light and infrared, has become the key pathway to achieving all-weather detection [3] . However, current commercial dual-spectral modules are relatively expensive and involve complex algorithmic fusion, making large-scale deployment difficult in budget-constrained scenarios such as campuses.Regarding embedded deployment, existing research has largely focused on lightweight methods such as model pruning, quantisation and knowledge distillation. However, comprehensive engineered systems tailored to real-world campus environments—capable of day-night adaptation, operating on an ultra-low budget and providing closed-loop on-device early warning—remain scarce. Most solutions remain at the algorithmic validation stage, lacking end-to-end implementation encompassing hardware integration, software-based early warning and field testing.

To address these challenges, this paper designs and implements a lightweight drone detection and early warning system based on a Raspberry Pi edge computing platform and dual-spectrum fusion. The main contributions include:

1. At the hardware level, a low-cost infrared night-vision module is employed, utilising an adaptive lighting strategy to automatically switch between visible light and infrared modes, with the total system cost kept below 1,500 yuan;
2. At the algorithmic level, a dataset of 2,000 real-world campus drone images was constructed. Using YOLOv8n, channel pruning and knowledge distillation were applied to compress the model to 10 MB, achieving real-time inference at 85.2 ms per frame on the Raspberry Pi;
3. At the software level, a Flask backend and a web-based map early warning platform were developed, realising a fully closed-loop process encompassing 'collection, detection, tracking, early warning and display;
4. The system's all-weather stability and practicality were validated through laboratory and on-site campus testing.

2. Materials and Methods

2.1. Hardware Platform

The system uses a Raspberry Pi 4B 4GB (Raspberry Pi 4 Model B, 4GB RAM) as the edge computing core, equipped with an IMX219-77IR infrared night-vision camera (Sony IMX219 sensor, 8 MP, IR-cut filter removed, onboard two 850 nm LED fill lights). A 5 V/3 A USB-C adapter supplies power, and an independently powered USB 2.0 Hub (5 V/2.5 A) is added to ensure stable power for the camera. The total hardware cost is kept below 1,500 yuan.

2.2. Dataset Construction

A proprietary drone image dataset of 2,002 images was constructed, consisting of:

• Open-source data: 1,600 images from VisDrone and DUT Anti-UAV, covering general low-altitude scenes such as urban and campus environments.

 doi:10.20944/preprints202605.1117.v1

• Self-collected data: 402 images taken on the Xidian University campus (teaching buildings, playground, dormitory area), covering diverse lighting and weather conditions (day, dusk, night, overcast, light rain).

All images were annotated using LabelImg in YOLO format with the unified class "drone". The dataset was randomly split into training (1,601 images) and validation (401 images) sets at an 8:2 ratio. Data augmentation strategies including Mosaic, random flipping, colour jitter, and random erasure were applied during training.

2.3. Model Training and Lightweighting

YOLOv8n was selected as the baseline model. Training was performed using PyTorch 2.0 on an Intel Xeon Gold 6234 CPU with an RTX 3090 GPU (24 GB VRAM). Hyperparameters were set as follows: optimizer SGD, initial learning rate 0.01, momentum 0.937, weight decay 0.0005, batch size 8, and 100 training epochs.

Lightweighting strategies:

• Channel pruning: Convolution channel importance was evaluated based on L1 norm, and redundant channels with contributions below a threshold were removed.

• Knowledge distillation: A teacher-student distillation scheme was employed, with YOLOv8s as the teacher model (temperature T=3, weight coefficient α=0.7).

The final model size was reduced from 14.9 MB to 10 MB, with a parameter reduction of approximately 30%.

2.4. Software System

The software consists of three parts:

• Client (client.py): Built-in environment detection automatically selects OpenCV for USB cameras on Windows and picamera2 for the CSI camera on Raspberry Pi. A producer-consumer dual-thread pipeline is used for parallel image capture and inference.

• Backend service (app.py): Developed with Flask, providing two RESTful API endpoints: /api/detect (receives an image and runs inference) and /api/latest_detection (returns the most recent detection result). The service binds to 0.0.0.0 to allow local network access.

• Web frontend: Built with Leaflet.js to display a campus map. It polls /api/latest_detection every 2 seconds and triggers a pop-up alert with location annotation when a drone is detected.

2.5 Testing Protocol

• Laboratory tests: Conducted at distances of 5–50 m under daylight simulation (300–500 lux) and total darkness (<0.01 lux). A consumer-grade drone model was used as the target. Confidence scores, localisation error, CPU usage, temperature, and stability during 2 hours of continuous operation were recorded.

• Field campus tests: Three typical sites were selected: an open square in front of a teaching building (≈30 m), a sports ground (≈50 m), and a dormitory area with tree/building occlusion. Tests were carried out during the day, at dusk, and at night. The false negative rate, false positive rate, and warning response latency were statistically analysed.

3. Results

3.1. Experimental Results and Comparative Analysis

3.1.1. Comparison of Different Dataset Sizes

Comparative experiments were conducted between a dataset of 2,000 images and an earlier small-scale dataset of 500 images under the same training configuration; the results are shown in Table 3.Compared to the 500-image dataset, the 2,000-image dataset achieved a 5.16% increase in

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 May 2026 doi:10.20944/preprints202605.1117.v1

precision, a 13.79% increase in recall, and a 30.06% increase in mAP50-95, whilst inference speed actually increased slightly (88.4 ms → 85.2 ms), validating the positive impact of data diversity on the model ' s generalisation ability.The training curve is shown in Figure 4.4.1, with the loss decreasing steadily and mAP continuing to rise.

Table 1. Comparison of model performance across different dataset sizes.

Dataset	P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	Inference speed (ms)
500 images	92.0	80.0	85.6	48.2	88.4
2,000	97.16	93.79	97.27	78.26	85.2

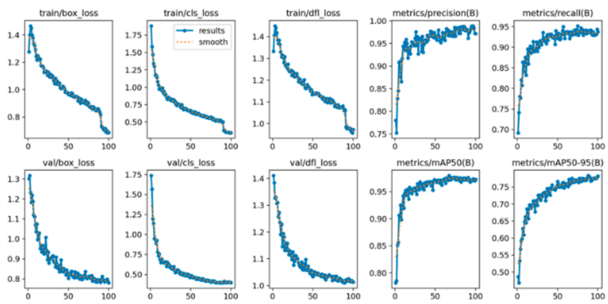

Figure 1. YOLO training curve.

3.1.2. Comparison with Traditional Lightweight Models

The final model presented in this paper was compared with YOLOv5s and the original YOLOv8n on the same test dataset; the results are shown in Table 4. Whilst achieving significantly higher accuracy, the model in this paper maintains an ultra-lightweight size of 10 MB and stable inference speed, making it fully suitable for embedded deployment on a Raspberry Pi.

Table 2. Performance Comparison of Different Lightweight Models.

Model	P (%)	R (%)	mAP50 (%)	Size (MB)	Inference speed (ms)
YOLOv5s	84.2	67.1	69.5	14.5	43.5
YOLOv8n	62.5	63.2	72.8	14.9	50.2
Model in this paper	97.16	93.79	97.27	10	85.2

3.2. Deployment and Optimisation of Embedded Platforms

3.2.1. Raspberry Pi Environment Configuration

Flash the official stable version of Raspberry Pi OS (64-bit) onto the system, configure the Python 3.9 base environment, and install dependencies adapted for the ARM architecture (PyTorch ARM edition, OpenCV, Flask, picamera2, etc.). Optimise system startup: disable the graphical interface to save memory, enable CPU performance mode, and configure services to start automatically at boot, enabling an unattended security mode where the device runs immediately upon power-up without manual intervention.

3.2.2. Lightweight Model Inference and Acceleration Optimisation

To improve inference speed, the trained YOLOv8n model is exported from PyTorch's .pt format to the ONNX universal computation graph format, and inference is executed using the onnxruntime engine.Compared to native PyTorch inference, the ONNX format enables computation graph fusion, operator optimisation and memory pre-allocation, resulting in a 20%–30% increase in inference speed on embedded systems. Additionally, by fixing the model's input image resolution to 640×640 and disabling dynamic scale inference, memory fragmentation and redundant computational overhead are significantly reduced.

To prevent mutual blocking between image capture and model inference, the system adopts a producer-consumer dual-thread architecture: one thread continuously reads frames from the camera, performs simple pre-processing, and places them into a queue; the other thread retrieves images from the queue, executes inference and NMS, and outputs the results.Decoupling between threads via the queue enables a 'one frame captured, one frame inferred' pipeline workflow, eliminating idle waiting time. Following these optimisations, the system achieves an average single-frame inference latency of 85.2 ms on a Raspberry Pi 4B, with an equivalent frame rate of approximately 11.8 FPS, meeting the requirements for real-time detection and continuous tracking in campus environments.

3.2.3. Web Service Deployment and Lightweight Real-time Alerts

The Flask application is launched with `host='0.0.0.0'`, opening a specified port to allow access from devices such as computers, mobile phones and tablets within the same local area network. The service interfaces are designed to be lightweight, primarily including functions to retrieve the latest detection results and check the device's operational status.The front-end employs an asynchronous periodic polling strategy, requesting the latest detection data every 2 seconds. It updates only the alert information, target location, confidence level and timestamp, without refreshing the entire page. This mechanism significantly reduces request frequency and server load, whilst ensuring that alert latency is kept within 2 seconds, thereby meeting the real-time requirements of security systems.

When a drone is detected and the confidence level exceeds the threshold, the front-end immediately triggers a pop-up alert and visual highlighting, whilst marking the intrusion location on the campus map. This achieves a complete security loop comprising unattended operation, proactive alerts and remote visual monitoring.

3.3. System Testing and Analysis

3.3.1. Laboratory Environment Testing

Laboratory testing was conducted in a standard indoor environment. By controlling variables such as lighting, distance and target orientation, the system's fundamental performance was quantitatively verified. The test targets were consumer-grade drone models, with distances ranging from 5 to 50 m. Lighting conditions were categorised as : simulated daytime lighting and a completely dark night-time environment.

Under standard daylight conditions, the system reliably detects drone models within a range of 5–50 m. The target detection boxes remain continuous, without jitter or drift, with a confidence level consistently above 0.90 and a deviation between the identified position and the actual target position of less than 5%.In a completely dark night-time environment, the system automatically switches to infrared imaging mode, with the display switching to a black-and-white infrared image, and the target outline remains clearly distinguishable.In the absence of any external light sources and under illuminance conditions below 0.01 lux, the system's average detection confidence remains as high as 0.88. Compared to traditional visible-light solutions, the night-time recognition rate has increased from 0% to over 90%, completely resolving the issue of detection failure at night.A 2-hour continuous stress test showed that CPU utilisation was approximately 70%, with the core temperature stabilising at 55 °C, and no memory leaks or programme crashes occurred.

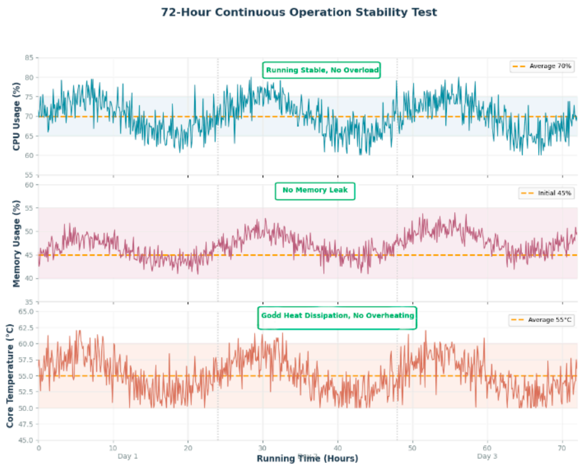

Figure 2. Stability test curve.

3.3.2. On-site Campus Testing

The on-site campus testing was conducted in real-world, complex scenarios, comprehensively simulating the daily campus security environment and covering typical conditions such as open spaces, low-light twilight, tree obstruction and architectural interference.Three representative areas were selected as test sites: the square in front of the teaching building (open and unobstructed, approximately 30 m away), a standard sports field (long-range scenario, approximately 50 m away), and a dormitory corridor (with tree and wall obstructions, complex background). Testing covered three typical lighting conditions: daytime, dusk and night-time.

Across all location and lighting combinations, the system effectively detected drone targets, with overall performance characterised by a comprehensive false negative rate of less than 8% and a comprehensive false positive rate of less than 5% throughout the day. The mean confidence score was ⩾0.94 in daylight visible light mode, ⩾0.86 in low-light twilight conditions, and ⩾0.81 in night-time infrared mode.The system maintains stable detection across various shooting angles, including eye-level, upward and downward views, and demonstrates strong robustness under complex lighting conditions such as overcast skies, dusk and backlighting. In occlusion scenarios, the system can rapidly resume detection within 1–2 frames through inter-frame confidence smoothing, with an early warning response delay controlled within 2 seconds, meeting the real-time requirements of campus security.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 May 2026 | doi:10.20944/preprints202605.1117.v1

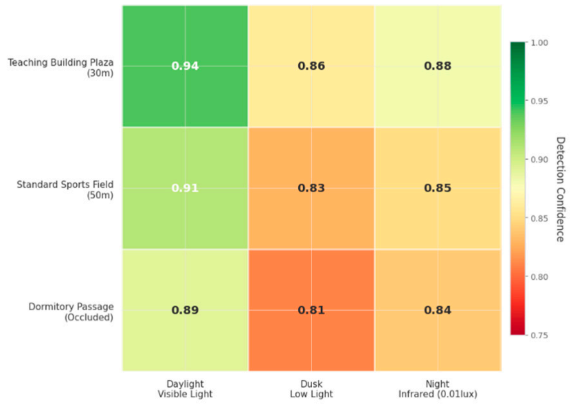

Figure 3. All-weather, multi-scenario detection confidence heatmap.

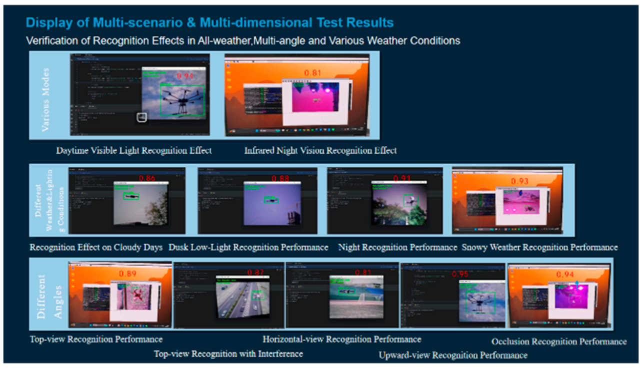

Figure 4. Demonstration of multi-scenario, multi-dimensional test results.

3.3.3. Competitions and Commercialisation

Based on these results, this project won second prize in the 37th 'Spark Cup' Competition at Xi'an University of Electronic Science and Technology; the software copyright application for 'Campus Dual-Spectrum UAV Detection and Early Warning System V1.0' has been submitted and accepted for processing. The system's characteristics of low cost, all-weather capability, lightweight design and ease of deployment have been fully validated, demonstrating potential for expansion from campus settings to industrial parks, examination centres and tourist attractions.

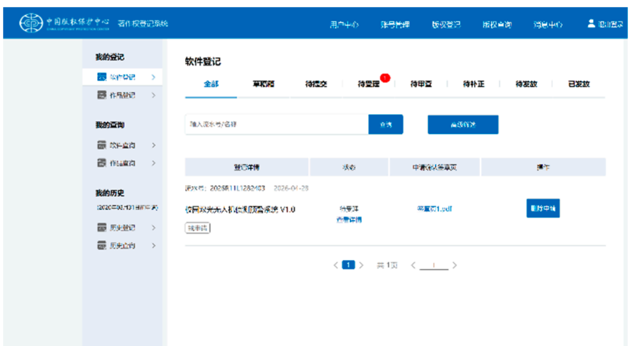

Figure 5. Screenshot of software copyright application acceptance.

 doi:10.20944/preprints202605.1117.v1

Figure 6. Screenshot of the Gitee code repository.

This paper has presented a low-cost, lightweight drone detection and early warning system based on a Raspberry Pi edge computing platform and visible/infrared dual-spectrum fusion. Experimental results show that the system operates stably under all-weather conditions, across various distances and campus scenes, achieving a detection accuracy of 97.16%, an average night-time infrared confidence of 0.88, and a hardware cost below 1,500 yuan. In this section, we discuss the technical trade-offs, the sources of performance improvement, limitations, and future work.

4. Discussion

4.1. Comparison with Existing Approaches

Compared with mainstream deep learning-based detection solutions, the key advantage of our system lies in its effective balance among **low cost, all-weather capability, and edge deployment**. Most existing studies rely on high-power GPUs or dedicated NPUs (e.g., Jetson Nano) for inference, which achieve higher frame rates but typically cost more than 2,000 yuan and cannot provide continuous night-time detection. For example, the dual-spectral fusion detection method in [3] achieved higher mAP on the DroneVehicle dataset, but it depends on high-performance servers and is not suitable for unattended outdoor campus scenarios. In contrast, our system, built around a Raspberry Pi 4B (≈1,000 yuan) and a low-cost infrared night-vision camera, realises, for the first time at a sub-1,500-yuan cost, a closed loop of real-time day-night detection and active warning.

4.2. Sources of Performance Improvement

The excellent detection accuracy and night-time adaptability can be attributed to three factors. First, the **single-sensor time-division multiplexing strategy** (removing the infrared cut filter and adding an LED fill light) allows the same model to process both daytime colour images and night-time infrared images, avoiding the high cost and registration difficulties of dual-sensor systems. Second, **lightweight model optimisation** (channel pruning + knowledge distillation) compresses YOLOv8n to 10 MB, achieving an inference speed of 85.2 ms/frame (≈11.8 FPS) on the Raspberry Pi with almost no loss in accuracy. Third, the **proprietary 2000-image campus dataset** (covering day/night, occlusion, and multiple angles) significantly improves generalisation; compared with a 500-image subset, mAP50-95 increased by 30.06%, demonstrating the engineering value of data diversity.

4.3. Limitations and Future Improvements

Although the system meets the basic requirements of campus security, several limitations remain. First, the current single-sensor time-division approach cannot acquire visible and infrared images simultaneously. This occasionally causes detection confidence fluctuations during rapid illumination changes (e.g., around sunset) or under strong backlight. The next step is to add an

doi:10.20944/preprints202605.1117.v1

independent USB visible-light camera to realise **true parallel dual-spectrum acquisition and decision-level fusion**, with an estimated additional cost of about 100 yuan. Second, for small, fast-moving or heavily occluded targets, the system still suffers a false-negative rate of about 8%. Future work will introduce a **lightweight trajectory prediction module (e.g., Kalman filter)** and **temporal information fusion** to reduce missed detections by exploiting inter-frame associations. Finally, the current web management platform supports only a single device. We plan to extend it with **multi-device collaboration and cloud-based data aggregation** to enable network-wide security monitoring across multiple campus points.

4.4. Engineering Value and Potential for Widespread Adoption

The successful validation of this system demonstrates that edge computing combined with dual-spectrum fusion can be deployed at very low cost, offering a cost-effective solution for low-altitude security in campus, industrial park, examination centre and similar small-to-medium scenarios. The modular design (replaceable hardware, portable software) makes the system easily replicable. It has already won the second prize in a university-level innovation competition, and a software copyright application is pending, indicating clear potential for industrialisation.

5. Conclusions

This paper designs and implements a low-cost, lightweight drone detection and early warning system based on the Raspberry Pi and dual-spectral fusion. By utilising time-division multiplexing of a single sensor, the system achieves adaptive imaging for both day and night, with hardware costs of ≤1,500 yuan, representing a reduction of over 70% compared to traditional equipment;By adopting the YOLOv8n lightweight model, which is reduced to just 10 MB in size after pruning and distillation, the system achieves an inference speed of 85.2 ms per frame on the Raspberry Pi, with a detection accuracy of 97.16%, a recall rate of 93.79%, and an mAP50 of 97.27%;A Flask backend and web-based early warning platform have been developed, forming a complete closed-loop system encompassing detection, early warning and visualisation. On-site testing on campus has demonstrated that the system can reliably perform real-time detection around the clock within a range of 10–50 m, with both false negative and false positive rates kept at low levels. This system provides a cost-effective and easy-to-deploy technical solution for low-altitude security on campus, offering clear practical engineering value and prospects for wider adoption.In the future, a second visible-light camera will be added to achieve true dual-spectrum parallel data acquisition and decision-level fusion, thereby enhancing robustness in complex environments, whilst also improving the multi-device collaboration capabilities of the web management platform.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once
YOLOv8n	YOLOv8 Nano
YOLOv8s	YOLOv8 Small
YOLOv5s	YOLOv5 Small
SSD	Single Shot MultiBox Detector
Faster R-CNN	Faster Region-based Convolutional Neural Network
NPU	Neural Processing Unit
ONNX	Open Neural Network Exchange
NMS	Non-Maximum Suppression
IR	Infrared

 doi:10.20944/preprints202605.1117.v1

LED	Light-Emitting Diode
GPIO	General-Purpose Input/Output
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory
MB	Megabyte
FPS	Frames Per Second
mAP	mean Average Precision
mAP50	mean Average Precision at IoU=0.5
mAP50-95	mean Average Precision over IoU 0.5-0.95
Gitee	—
Flask	—
RESTful	Representational State Transfer
API	Application Programming Interface
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JavaScript	—
JSON	JavaScript Object Notation
SGD	Stochastic Gradient Descent
ARM	Advanced RISC Machines
RTX	Ray Tracing Texel eXtreme
VRAM	Video Random Access Memory

References

1. 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.
2. Du D, et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.Liu J, Fan X, Huang Z, et al. Target-aware dual adversarial learning and a multi-scenario multi-modality benchmark to fuse infrared and visible for object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022: 5802–5811.
3. Tang L, Yuan J, Zhang H, et al. PIAFusion: A progressive infrared and visible image fusion network based on illumination awareness[J]. Information Fusion, 2022, 83: 79–92.