

A real-time drone detection framework for variable environmental conditions based on web-based deep learning

Yuliani Indrianingsih¹, Astika Ayuningtyas¹, Anton Yudhana², Haruno Sajati¹,
Anggraini Kusumaningrum¹, Fajar Khanif Rahmawati³, Imam Riadi²

¹Department of Informatics, Institut Teknologi Dirgantara Adisutjipto, Yogyakarta, Indonesia

²Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

³Department of Aerospace Engineering, Institut Teknologi Dirgantara Adisutjipto, Yogyakarta, Indonesia

Article Info

Article history:

Received Dec 15, 2025

Revised Jul 23, 2026

Accepted Aug 12, 2026

Keywords:

Deep learning

Drone detection

FastAPI

Real-time object detection

Web-based framework

ABSTRACT

The rapid proliferation of unmanned aerial vehicles (UAVs) in civilian, industrial, and military domains has intensified challenges related to airspace security, particularly in restricted and sensitive areas. Unauthorized drone operations pose risks including illegal surveillance, disruption of air traffic, and threats to critical infrastructure. This research proposes a lightweight, web-based real-time drone detection framework utilizing a convolutional neural network (CNN) integrated with a fast application programming interface (FastAPI) server to support continuous visual surveillance. The system is trained and evaluated using a custom dataset of 23,118 annotated drone images, enhanced through data augmentation to improve robustness under varying environmental conditions. Experimental results demonstrate real-time inference performance of 20–25 frames per second across multiple camera resolutions, including 1080p, 2K, and digital single-lens reflex (DSLR) Canon 600D. Detection accuracy varies between 53% and 74%, revealing a strong dependency on illumination and imaging conditions, with optimal performance achieved under high illumination levels (45,000 lux). Performance degradation is observed in low-light environments, particularly for DSLR-based detection. The proposed system enables live video streaming and static image analysis through a user-friendly web interface without reliance on external services. Overall, this study presents a practical and deployable prototype for automated drone monitoring, emphasizing the trade-offs between accuracy, illumination, and camera resolution in real-world airspace security applications.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Astika Ayuningtyas

Department of Informatics, Institut Teknologi Dirgantara Adisutjipto

St. Majapahit Blok R Lanud Adisutjipto, Banguntapan, Bantul, D.I. Yogyakarta 55198

Email: astika@itda.ac.id

1. INTRODUCTION

The popularity of drones, which are also referred to as unmanned aerial vehicles (UAVs), has increased rapidly. They are capable of monitoring, cultivating, and transporting products. However, accessing prohibited areas, privacy, and security are significant challenges when they are employed without authorization [1], [2]. Traditional radar or sensor-based systems may require additional hardware to accommodate the fluctuating conditions in metropolitan areas. It is now feasible to detect objects in real time with remarkable precision, thanks to recent developments in deep learning. The convolutional neural network (CNN) is one of the most critical forms of neural networks for this purpose [3], [4].

Models such as you only look once (YOLO), single shot multibox detector (SSD), and faster region-based convolutional neural network (Faster R-CNN) have succeeded in detecting objects in the air. Conversely, most of the research remains dependent on offline or GPU-intensive conditions, which can be challenging to implement in web-based or lightweight environments [5]–[8]. A lightweight web-based deep learning framework for real-time drone detection is proposed by integrating a CNN-based model with fast application programming interface (FastAPI). The proposed framework enables real-time object detection through a web interface using central processing unit (CPU)-only resources. Its primary contributions are threefold: i) developing a lightweight real-time drone detection framework without GPU dependency, ii) evaluating detection performance under different illumination conditions and camera resolutions, and iii) demonstrating practical deployment through FastAPI-based web integration. Unlike conventional UAV detection systems that typically rely on GPU-intensive environments, the proposed framework emphasizes deployment feasibility on standard hardware while maintaining reliable detection performance under diverse environmental conditions. The remainder of this paper is organized as follows. Section 2 reviews related work, section 3 describes the proposed methodology, section 4 presents the experimental results and discussion, and section 5 concludes the paper.

2. METHOD

The research methodology utilized in this study adheres to a systematic experimental workflow, as depicted in Figure 1 flowchart of the experimental methodology. The proposed methodology comprises four primary stages. These stages are dataset preparation, model development, system integration, and performance assessment. The initial phase involves dataset preparation, emphasizing the development of a high-caliber dataset for deep learning applications in computer vision. Image data are obtained from pertinent sources and subsequently undergo data cleansing to eliminate corrupted or substandard samples. Each image is labeled in accordance with established classifications. Pre-processing methods such as image resizing, normalization, and data augmentation (including rotation, flipping, scaling, and brightness modification) are employed to increase data variability and enhance the model's generalization capabilities. The dataset is subsequently partitioned into training, validation, and testing subsets according to predetermined proportions. The second stage involves constructing the model, during which a deep learning architecture for computer vision is systematically designed and developed. A CNN-based architecture is chosen, either through a bespoke design or by utilizing transfer learning with pre-trained models. Model training is performed utilizing the prepared training dataset, with hyperparameters including learning rate, batch size, number of epochs, and optimizer meticulously adjusted to attain optimal performance. The validation dataset is employed to oversee training progress and mitigate overfitting. The third stage involves system integration, wherein the trained model is incorporated into a computer vision system or application. This phase entails implementing the model within the designated environment, such as a desktop application, web-based system, or embedded platform. Functional testing is conducted to verify that the integrated system accurately processes input images and generates precise and dependable outputs in real-world environments. The final stage involves performance evaluation, which seeks to assess the efficacy and resilience of the proposed deep learning model and the integrated system.

Model performance is assessed through standard metrics widely utilized in computer vision research, such as accuracy, precision, recall, F1-score, and confusion matrix analysis. Furthermore, inference time and computational efficacy can be assessed to determine the system's viability for practical implementation. The evaluation results serve as the foundation for performance assessment, discussion, and conclusion of the study.

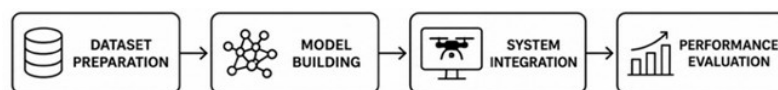


Figure 1. Flowchart of experimental methodology

2.1. Deep learning

Deep learning has made a considerable contribution to studies on the detection of drones. Due to their ability to detect in a single stage and their real-time capabilities, YOLO-based techniques, for example, have been utilized for surveillance drones. The strong detection performance that was reported by studies such as [9], [10] necessitated the use of GPU acceleration and was mostly evaluated under controlled

settings. Edge computing and web-based deployment were the subjects of exploration in other studies. For example, lightweight CNNs have been used on mobile or embedded systems, although they generally exchange accuracy for speed [8], [11]. In a similar vein, there are object identification frameworks that are housed in the cloud [12], [13]. YOLOv5s was selected because it provides an appropriate compromise between inference speed, computational complexity, and detection accuracy for CPU-based deployment. Larger variants such as YOLOv5m or YOLOv5l generally provide higher accuracy but require significantly greater computational resources. The suggested framework makes use of a CNN-based detector that is comprised of three primary components: a backbone, a neck, and a detection head. Through a sequence of convolution operations, the backbone oversees extracting both low-level and high-level information from the input image, as in (1) illustrates this process mathematically.

$$(f * g)(x, y) = \sum_{i=-m}^m \sum_{j=-n}^n f(i, j) * g(x - i, y - j) \quad (1)$$

Where g stands for the input picture and f for the kernel. The network can capture spatial hierarchies, which are crucial for drone recognition, thanks to these convolutional layers.

The neck module enhances multi-scale feature fusion through the feature pyramid network (FPN) and path aggregation network (PANet), enabling robust drone detection across different object sizes and viewing distances. The detection head subsequently predicts bounding box coordinates, objectness scores, and class probabilities, where the objectness score represents the probability of an object's presence, as defined in (2).

$$P_{obj} = \sigma(t_o) \quad (2)$$

Intersection over union (IoU) metric used to optimize bounding box localization, as in (3).

$$L_{obj} = 1 - I_o U(B_p, B_{gt}) \quad (3)$$

Where B_{gt} is the ground truth, and B_p is the predicted box. The multi-task loss function in (4) represents the overall training objective.

$$L = \lambda_{box} L_{box} + \lambda_{cls} L_{cls} + \lambda_{obj} L_{obj} \quad (4)$$

The CNN model that has been trained is integrated with FastAPI as the backend framework for deployment. Through a simple web interface created using Jinja2 templates, the system can now execute asynchronous inference on both image and video streams, providing real-time results. The framework thus fills the gap between sophisticated deep learning models and useful, user-friendly web-based drone detection applications.

2.2. Dataset preparation

The dataset comprises 23,118 drone images acquired using a digital single-lens reflex (DSLR) camera, a 1080p webcam, and a 2K webcam under daytime, afternoon, and nighttime conditions. Images were annotated in Roboflow and augmented through flipping, rotation, and brightness adjustment to improve model generalization. All images were resized to 416×416 pixels and normalized to the range $[0, 1]$, as expressed in (5), where $I(x, y)$ denotes the pixel intensity, and I_{max} and I_{min} represent the maximum and minimum intensity values, respectively.

$$I_{norm}(x, y) = \frac{I(x, y) - I_{min}}{I_{max} - I_{min}} \quad (5)$$

Images were chosen to guarantee diversity in quality, perspective, background intricacy, and lighting conditions. The dataset was partitioned into three subsets: 70% allocated for training, 20% for testing, and 10% for validation as shown in Table 1.

Table 1. Dataset drone

Dataset split	Number of images	Percentage (%)
Training set	16,183	70
Test set	4,624	20
Validation set	2,311	10
Total	23,118	100

To improve model generalization, all images were annotated, resized, and augmented using the Roboflow platform. Figure 2 presents representative drone images that illustrate variations in illumination, object scale, viewing angle, and background complexity. These include morning illumination (20–37k lux) in Figure 2(a), afternoon illumination (42–46k lux) in Figure 2(b), evening low-light (0 lux) in Figure 2(c), small-scale distant view in Figure 2(d), medium-scale mid-distance view in Figure 2(e), large-scale close-range view in Figure 2(f), cloudy background in Figure 2(g), complex urban background in Figure 2(h), and backlight or partial occlusion conditions in Figure 2(i).

The corresponding annotations were exported as YOLO-compatible bounding box labels for model training. The dataset was then randomly divided using a fixed random seed to preserve class balance and illumination diversity across the training, validation, and testing subsets. This strategy minimizes sampling bias by ensuring that each subset adequately represents the variability encountered during image acquisition. Consequently, the experimental results provide a more reliable assessment of the proposed framework's generalization capability under diverse environmental conditions.



Figure 2. Representative drone image showing variations: (a) morning illumination (20–37k lux), (b) afternoon illumination (42–46k lux), (c) evening low-light (0 lux), (d) small-scale for distant, (e) medium-scale mid-distance, (f) large-scale close-range, (g) cloudy background, (h) complex background, and (i) backlight or partial occlusion

2.3. Model training

The detection model employed the YOLOv5s backbone, trained for 250 epochs with a batch size of 16, using the adaptive moment estimation (Adam) optimizer with a cosine annealing learning rate scheduler. Anchor boxes were refined using k-means clustering to better match drone object sizes. The convolution operation in the backbone is defined as shown in (1). The loss function combines multiple components as

shown in (4), where L_{box} is the bounding box regression loss based on IoU, L_{cls} is the classification loss, and L_{obj} is objectness loss.

2.4. System deployment

An environment that was built on the web was used to deploy the trained model. Inference was processed asynchronously by FastAPI, which functioned as the backend. Jinja2 templates were utilized for the design of the frontend during the development process. To perform real-time video processing, OpenCV was incorporated. The formula for the inference process is as in (6), where f_θ denotes the trained YOLOv5s model parameterized by θ , and y represents the output consisting of bounding box coordinates, objectness scores, and class predictions.

$$y = f_\theta(x), x \in \mathbb{R}^{416 \times 416 \times 3} \quad (6)$$

2.5. Evaluation metrics

There were three primary criteria that were utilized to conduct an exhaustive evaluation of the proposed system. The first method is known as IoU, and it is used to quantify the amount of overlap that exists between the predicted bounding box and the accurate ground truth. In (7) is defined as follows: where B_p is the predicted bounding box, and B_{gt} is the ground truth. The exact location of the drone can be determined with higher accuracy if IoU.

$$IoU = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \quad (7)$$

The second metric is the mean average precision (mAP), which is shown in (8). It gives a general idea of how well the detection works across all classes. To get it, you take the average precision for each class and divide it by the number of classes (N). N is the number of classes, and AP_i is the average precision for class i . This metric evaluates both precision and recall, providing a balanced assessment of the model's detection performance.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (8)$$

Frames per second (FPS) statistics are used to assess system performance in real-time. FPS in (9) is defined as N_f , representing the number of frames processed, and T being the total time in seconds.

$$FPS = \frac{N_f}{T} \quad (9)$$

A higher FPS number indicates the system's capability to execute inference in real time with minimal delay. IoU, mAP, and FPS together offer a comprehensive evaluation methodology, allowing for the assessment of drone detection accuracy, efficiency, and appropriateness for real-time implementation.

3. RESULTS AND DISCUSSION

This section presents the experimental results and evaluates the performance of the proposed framework under different environmental conditions. Detection accuracy, real-time processing capability, and the effects of camera resolution and illumination are analyzed. These factors are compared with previous studies to demonstrate the effectiveness and practical applicability of the proposed approach.

3.1. Offline evaluation on validation dataset

The preliminary evaluation on the validation set assessed the proposed model using 30 representative test images. These images had varying object sizes, background complexity, and environmental conditions as shown in Figure 3. Detection accuracy ranged from 53% to 81% for successfully detected drone instances, while several challenging scenarios resulted in missed detections or misclassification. Higher accuracy was consistently achieved when drones were larger and clearly distinguishable from the background, whereas small, partially occluded, or visually ambiguous targets reduced detection performance. Despite these challenging cases, the proposed framework demonstrated satisfactory generalization capability across diverse validation scenarios. Quantitative evaluation further yielded an mAP@0.5 of 68.2%, precision of 72.4%, and recall of 65.1%, indicating competitive performance compared with previous lightweight CNN-based drone detection approaches [14].

3.2. Real-time testing under environmental variations

The proposed framework was evaluated under three illumination conditions using three camera configurations (Canon 600D DSLR, 1080P webcam, and 2K webcam). The FastAPI-based system maintained real-time performance at approximately 24 FPS, and the results of ten repeated experiments are summarized in Table 2. The highest average detection accuracy was achieved under afternoon (high illumination) using the 1080P camera (63.7%), followed by the 2K camera during the morning condition (62.6%). Under moderate illumination, the average accuracy ranged from 57.8% to 62.6%, whereas detection performance declined during evening (0 lux), with average accuracies of 55.9% (1080P) and 48.1% (2K). The Canon 600D failed to detect drones under complete darkness. Statistical analysis further showed that the standard deviation ranged from 16.1% to 27.5%, indicating moderate variability primarily caused by illumination changes, drone position, and background complexity. The relatively wide variation observed across repeated experiments reflects realistic operational conditions, where differences in drone orientation, flight distance, object scale, background clutter, and illumination affect detection confidence rather than indicating instability of the proposed model. These findings confirm that illumination remains the dominant factor influencing vision-based drone detection performance and support the applicability of the proposed framework for real-time surveillance in restricted airspace. These findings are consistent with previous studies reporting that illumination variability and small-object characteristics remain major challenges in YOLO-based drone detection systems [15].

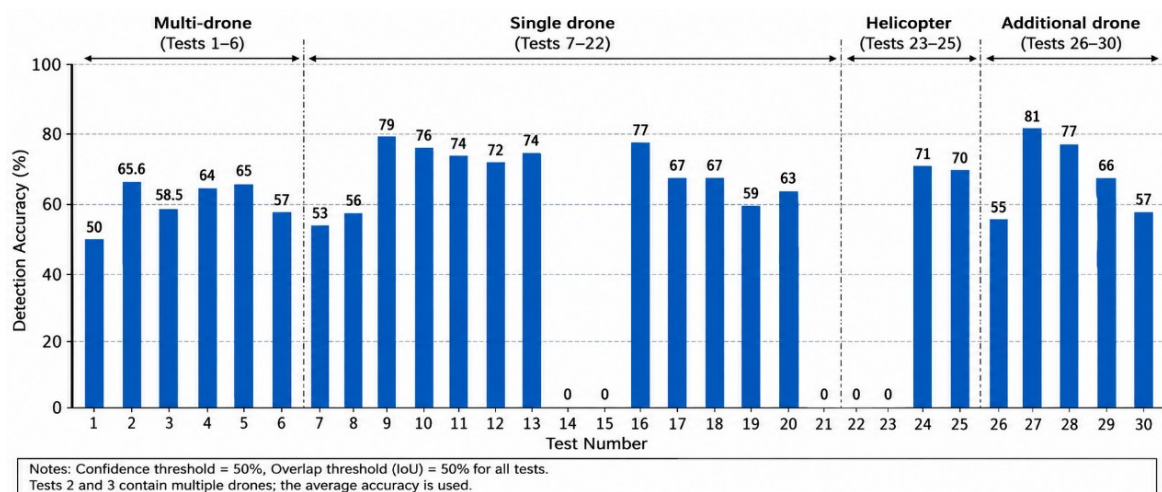


Figure 3. Results in Roboflow

Table 2. Real-time drone detection performance under different environmental conditions

Times	Camera type	Frames per second	Light intensity	Test 1 (%)	Test 2 (%)	Test 3 (%)	Test 4 (%)	Test 5 (%)	Test 6 (%)	Test 7 (%)	Test 8 (%)	Test 9 (%)	Test 10 (%)	Mean ± SD (%)
Morning (9:00-10:00)	Camera 1080P	24.75 fps	20k	55	80	20	90	98	20	35	48	70	62	57.8
	Camera 2K	24.64 fps	28k	28	77	30	86	42	90	95	51	59	68	62.6
	Canon 600D	24.70 fps	37k	65	20	73	25	88	38	82	49	92	57	58.9
Afternoon (13.00-14.00)	Camera 1080P	24.86 fps	45k	67	30	58	49	71	90	25	85	74	88	63.7
	Camera 2K	24.79 fps	42k	58	74	25	89	32	90	52	48	62	83	61.3
	Canon 600D	24.80 fps	46k	26	70	56	35	77	64	44	84	88	47	59.1
Evening (18.00-19.00)	Camera 1080P	24.07 fps	0	55	27	65	59	31	68	60	49	70	75	55.9
	Camera 2K	24.02 fps	0	38	75	25	58	45	27	63	34	49	67	48.1
	Canon 600D	24.02 fps	0	0	0	0	0	0	0	0	0	0	0	0.0

3.3. Key trade-offs

The experimental results reveal three major trade-offs. First, detection accuracy increased under higher illumination, confirming that lighting remains the primary factor influencing vision-based drone detection. Second, higher image resolution did not necessarily improve performance, as the 1080P webcam consistently outperformed both the 2K webcam and Canon 600D despite its lower resolution. Third, the FastAPI-based implementation maintained real-time performance on CPU-only hardware, demonstrating an effective balance between computational efficiency, deployment cost, and detection accuracy without requiring GPU acceleration. Compared with previous studies, the proposed framework emphasizes practical deployment rather than architectural complexity. While approaches such as YOLO-ARIA [10], YOLO D-NET [16], convolutional block attention module (CBAM)-enhanced YOLOv5 [9], hybrid bounding-box optimization [17], and 3D deep learning models [18] have reported improved performance in domain-specific applications, they generally require more complex architectures or higher computational resources [19]. Similarly, federated learning-based surveillance systems [20] focus on distributed industrial environments rather than lightweight real-time drone detection. In contrast, the proposed framework demonstrates that reliable drone detection can be achieved using a lightweight web-based architecture suitable for CPU-based deployment in practical surveillance applications.

3.4. Comparative analysis with previous studies

The proposed framework demonstrates the feasibility of deploying drone detection models on a lightweight, web-based platform without sacrificing processing performance, in contrast to earlier approaches that necessitated high-end GPU resources for real-time inference [21]. Although the obtained mAP@0.5 remains under GPU-oriented state-of-the-art models [22], [23], the proposed framework prioritizes deployment feasibility on CPU-based systems. This design choice balances detection accuracy with computational efficiency and accessibility for real-world surveillance applications. According to the information that is shown in Table 3, the proposed model was able to successfully achieve a harmonious equilibrium between the accuracy of its predictions and the computational efficiency of its results.

Table 3. Real-time test results

Model	Year	Dataset	mAP@0.5 (%)	FPS	Platform
Proposed (YOLOv5s + FastAPI)	2025	Custom UAV	68.2	24 FPS	CPU
Light-YOLOv5 [14]	2023	Drone Dataset	65.8	25 FPS	Edge
FE-YOLOv5 [15], [23]–[25]	2023	UAV Dataset	73.5	20 FPS	GPU
HIC-YOLOv5 [15]	2024	COCO UAV	78.4	21 FPS	GPU
YOLOv5 mamba [26], [27]	2024	UAV Object	80.3	22 FPS	GPU

It is nevertheless able to reach real-time performance on conventional CPU hardware (24 frames per second), even though its mAP@0.5 (68.2%) is not as high as when GPU-accelerated models are used. The fact that this is the case demonstrates that the framework is suitable for application in different situations that have minimal resources. Figure 4 provides an overall comparison of drone detection performance under different environmental conditions.

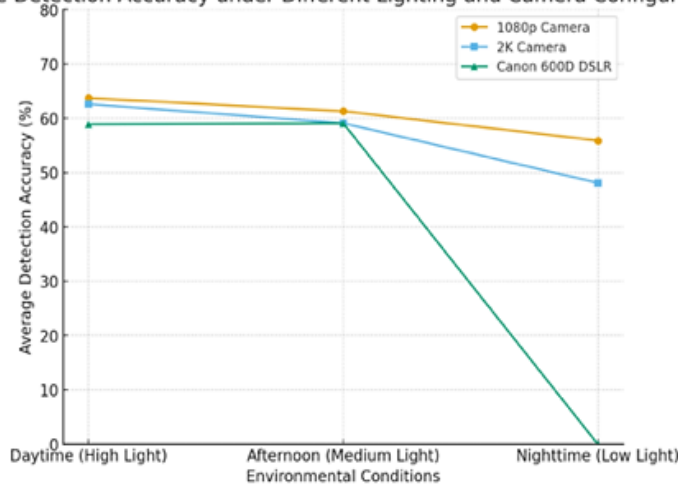
As can be seen in Figure 4(a), lighting was identified as the most significant element influencing the accuracy of detection. The 1080p camera achieved the highest average accuracy of 63.7%, which was better than the performance of both the DSLR camera and the 2K camera. All three camera types performed best under daytime conditions. This conclusion is consistent with prior research that has demonstrated that illumination conditions have a major influence on UAV detection performance, particularly in outdoor and surveillance contexts [21]. The accuracy of the DSLR camera decreased significantly in settings with low light, and it completely failed to function. However, the 1080p and 2K cameras were able to retain a moderate level of accuracy, with 55.9% and 48.1%, respectively. The findings of are consistent with the results that were obtained in this study, which support the conclusion that illumination quality is more important than resolution benefits, it's found that models based on YOLO had difficulty detecting tiny objects and objects in low light [15].

Figure 4(b) shows that the average processing speed for all settings was between 24 and 25 frames per second. This means that the speed of processing didn't change much, even when the camera resolution or the amount of light varied. This kind of behavior demonstrates that the framework that was put in place works, and it also shows that the framework is good for applications that need real-time performance. This is in line with previous research findings, which showed that real-time inference rates can be maintained without high-end GPUs by using lightweight YOLO based designs [28]. The main goal of this study is to establish a balance between how well the system works and how easy it is to use. This makes sure that

performance stays the same in online environments that can be accessed with regular computer hardware. This contrasts with earlier research, which mostly concentrated on the accuracy of controlled datasets [27].

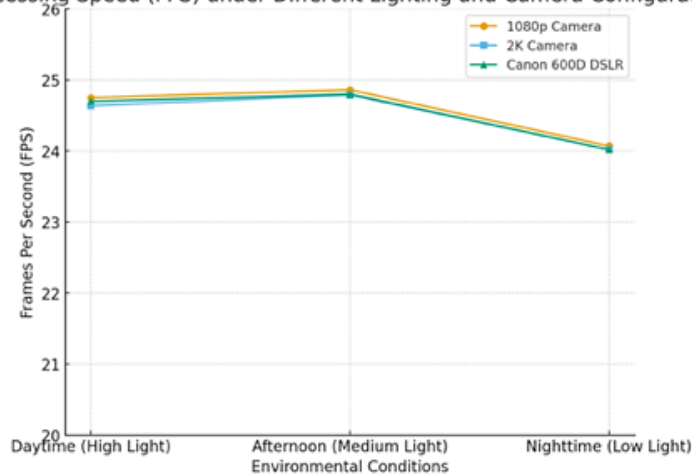
In general, these comparative results add something new in two ways. First, they demonstrate that mid-resolution cameras (1080p) with sufficient illumination surpass higher-resolution configurations in practical drone detection tasks, a viewpoint not thoroughly examined in previous studies that generally presume high-resolution input. Second, by demonstrating consistent real-time FPS across varied conditions, this study extends beyond algorithm-centric evaluations and emphasizes system-level integration through FastAPI, which ensures accessibility and deployability in real-world surveillance systems. Moreover, the integration with FastAPI allows web-based real-time deployment, fulfilling a critical requirement that prior research was limited to offline datasets, GPU-intensive environments, or specialized applications. Our balance of high accuracy, efficiency, and accessibility underscores the importance of our research in linking sophisticated deep learning with practical drone surveillance applications. Operationally, the proposed framework is compatible with existing airport and military surveillance infrastructures, utilizing conventional CCTV networks and edge computing devices. Instead of replacing radar or radiofrequency (RF)-based counter-drone technologies, the vision-based detector enhances current systems by enabling real-time visual verification of low-altitude airborne threats. Integrating this approach with radar, RF, and acoustic sensors in the future could create a multilayered airspace-monitoring architecture, thereby improving situational awareness and facilitating rapid decision-making by airport authorities and defense agencies.

Drone Detection Accuracy under Different Lighting and Camera Configurations



(a)

Processing Speed (FPS) under Different Lighting and Camera Configurations



(b)

Figure 4. Performance comparison under different environmental conditions: (a) average drone detection accuracy and (b) processing speed (FPS) across camera types and lighting conditions

3.5. System resources profiling

System profiling was carried out on a CPU-only environment (Intel i7-10750H, 16 GB RAM) to validate the practical applicability of the framework that was suggested. Table 4 displays the results of the test, which included measurements of CPU load, RAM use, and inference delay. The purpose of the study was to establish whether the model maintained real-time performance under conditions that did not involve a GPU.

These recommendations were derived from earlier research that highlighted the significance of computing efficiency and deployment practicality in lightweight artificial intelligence-based detection systems [29]. This evaluation is a result of those recommendations. These results confirm that the proposed framework maintains an average inference latency of 43 ms per frame (approximately 24 FPS) on a CPU-only platform, satisfying the real-time requirements for continuous visual surveillance in airport perimeter monitoring and restricted airspace security applications.

Table 4. System profiling results on a CPU-only environment

Metric	Average value	Platform
CPU Load	68%	Intel i7-10750H
RAM Usage	2.4 GB	16 GB
Average inference latency	0.043 s/frame	CPU only
Average FPS	23.8	CPU only

Nevertheless, the experimental results indicate that detection accuracy decreased substantially under low-light conditions (0 lux), particularly for the Canon 600D camera, due to reduced image contrast and increased sensor noise. Moreover, this study employed only the YOLOv5s architecture and focused primarily on technical performance evaluation. Consequently, usability assessment, operator interaction with the web-based interface, and comparative evaluation against other lightweight detectors, such as MobileNet-SSD and EfficientDet, were beyond the scope of this work. Future work will investigate low-light image enhancement techniques, such as histogram equalization and Retinex-based preprocessing, together with multimodal sensing (e.g., RGB and infrared fusion), while extending the framework through lightweight model comparisons and human-centered usability evaluation to further improve robustness and deployment readiness.

4. CONCLUSION

This investigation presented a web-based deep learning framework for the real-time identification of drones, employing a CNN-based YOLOv5s model integrated with FastAPI. The system demonstrated the capability to identify objects with an accuracy ranging from 53% to 74% (mAP@0.5 = 68.2%) while processing at 20 to 25 frames per second on standard CPU hardware. This demonstrates that it can function effectively without relying on GPU-intensive resources. The findings indicate that illumination is the most critical environmental factor affecting detection performance, with 1080p camera configurations demonstrating the highest consistency in accuracy under different lighting conditions. Despite a decline in performance during low-light conditions, particularly in detecting small or partially obscured drones, the suggested framework demonstrates that real-time identification of UAVs can be achieved even with restricted computing resources. Future investigations will focus on the integration of attention-based processes, multimodal sensor fusion (RGB + IR), and optimization for edge computing devices to enhance robustness and flexibility. This study introduces a streamlined and easily implementable system that bridges the gap between advancements in deep learning and practical applications for overseeing airspace and conducting autonomous aerial surveillance.

5. FURTHER WORKS

Although the proposed web-based deep learning framework demonstrated effective real-time drone detection using CPU-only resources, several challenges remain. Future work will focus on improving low-light detection through image enhancement techniques, such as adaptive histogram equalization and Retinex-based preprocessing, together with multimodal sensor fusion using RGB and infrared imagery. In addition, lightweight attention mechanisms or transformer-based modules will be investigated to enhance small-object detection while preserving real-time performance. Further evaluation on embedded edge-AI platforms, larger datasets containing diverse drone types, weather conditions, and aerial viewpoints, as well as broader field validation in operational surveillance environments, is expected to improve the robustness, scalability, and practical deployment of the proposed framework for autonomous airspace monitoring.

FUNDING INFORMATION

This research is supported by the Institut Teknologi Dirgantara Adisutjipto and Ahmad Dahlan University. This research is also part of the Fundamental Research Grant Scheme under grant number 126/C3/DT.5.00/PL/2025 from the Directorate General of the Ministry of Education, Culture, Research, and Technology

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Yuliani Indrianingsih	✓					✓	✓			✓		✓		
Astika Ayuningtyas	✓	✓			✓			✓	✓					
Anton Yudhana		✓			✓					✓		✓		
Haruno Sajati				✓		✓	✓			✓	✓			
Anggraini			✓			✓		✓		✓	✓		✓	
Kusumaningrum														
Fajar Khanif Rahmawati				✓		✓				✓			✓	
Imam Riadi		✓								✓				

C : Conceptualization	I : Investigation	Vi : Visualization
M : Methodology	R : Resources	Su : Supervision
So : Software	D : Data Curation	P : Project administration
Va : Validation	O : Writing - Original Draft	Fu : Funding acquisition
Fo : Formal analysis	E : Writing - Review & Editing	

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY

The dataset used in this research was compiled from multiple sources. Part of the dataset was obtained from the publicly available Mendeley Data repository at <https://doi.org/10.17632/6s449dpwv6.1>, while additional images were collected and annotated by the authors to support the experimental evaluation under diverse environmental conditions. Accordingly, the Mendeley repository represents only one component of the complete dataset used in this study.

REFERENCES

[1] B. Kotkova, "Airport defense systems against drones attacks," in *2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, Jul. 2022, pp. 85–90, doi: 10.1109/CSCC55931.2022.00025.

[2] K. Al-Dosari, Z. Hunaiti, and W. Balachandran, "Systematic review on civilian drones in safety and security applications," *Drones*, vol. 7, no. 3, Mar. 2023, doi: 10.3390/drones7030210.

[3] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, Sep. 2023, doi: 10.1016/j.aiia.2023.07.001.

[4] D. L. Shanthi, K. Vinutha, N. Ashwini, and S. Vashistha, "Tomato leaf disease detection using CNN," *Procedia Computer Science*, vol. 235, pp. 2975–2984, 2024, doi: 10.1016/j.procs.2024.04.281.

[5] H. R. Alsanad, A. Z. Sadik, O. N. Ucan, M. Ilyas, and O. Bayat, "YOLO-V3 based real-time drone detection algorithm," *Multimedia Tools and Applications*, vol. 81, no. 18, pp. 26185–26198, Jul. 2022, doi: 10.1007/s11042-022-12939-4.

[6] T. Saidani, "Deep learning approach: YOLOv5-based custom object detection," *Engineering, Technology and Applied Science Research*, vol. 13, no. 6, pp. 12158–12163, Dec. 2023, doi: 10.48084/etasr.6397.

[7] S. Singha and B. Aydin, "Automated drone detection using YOLOv4," *Drones*, vol. 5, no. 3, Sep. 2021, doi: 10.3390/drones5030095.

[8] P. A. K. Goud, G. M. Raj, K. Rahul, and A. V. Lakshmi, "Military aircraft detection using YOLOv5," in *Intelligent Communication Technologies and Virtual Mobile Networks*, 2023, pp. 865–878, doi: 10.1007/978-981-99-1767-9_63.

[9] M. A. E. Alkhalisy and S. H. Abid, "Students behavior detection based on improved YOLOv5 algorithm combining with CBAM attention mechanism," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 5, pp. 473–487, Oct. 2023, doi: 10.22266/ijies2023.1031.41.

[10] A. Hendrawan, R. Gernowo, O. D. Nurhayati, and C. Dewi, "A novel YOLO-ARIA approach for real-time vehicle detection and classification in urban traffic," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 1, pp. 428–446, Feb. 2024, doi: 10.22266/ijies2024.0229.38.

[11] D. G. Park, T. N. Jung, J. G. Kim, S. H. Lee, E. S. Oh, and D. H. Kim, "DBSCAN and YOLOv5 based 3D object detection and its adaptation to a mobile platform," *Mechatronics*, vol. 103, Nov. 2024, doi: 10.1016/j.mechatronics.2024.103238.

- [12] J. Liu and X. Dai, "Real-time object detection for UAVs images based on improved YOLOv5," in *Third International Conference on Computer Vision and Pattern Analysis (ICCPA 2023)*, Aug. 2023, doi: 10.1117/12.2684359.
- [13] S. Abba, A. M. Bizi, J.-A. Lee, S. Bakouri, and M. L. Crespo, "Real-time object detection, tracking, and monitoring framework for security surveillance systems," *Heliyon*, vol. 10, no. 15, Aug. 2024, doi: 10.1016/j.heliyon.2024.e34922.
- [14] J. Han, R. Cao, A. Brighente, and M. Conti, "Light-YOLOv5: a lightweight drone detector for resource-constrained cameras," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 11046–11057, Mar. 2024, doi: 10.1109/JIOT.2023.3329221.
- [15] S. Tang, S. Zhang, and Y. Fang, "HIC-YOLOv5: improved YOLOv5 for small object detection," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 6614–6619, doi: 10.1109/ICRA57147.2024.10610273.
- [16] D. Jegatheesan and C. Arumugam, "Intelligent traffic management support system unfolding the machine vision technology deployed using YOLO D-NET," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 25–34, Oct. 2021, doi: 10.22266/ijies2021.1031.03.
- [17] N. T. Allo and Z. Zainuddin, "A novel approach of hybrid bounding box regression mechanism to improve convergency rate and accuracy," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 2, pp. 715–727, Apr. 2024, doi: 10.22266/ijies2024.0430.57.
- [18] E. Mofreh, A. Abozeid, H. Farouk, and K. A. El-Dahshan, "Multi-object semantic video detection and indexing using a 3D deep learning model," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 3, pp. 268–280, 2022, doi: 10.22266/ijies2022.0630.23.
- [19] H. M. Ali and N. K. El Abbadi, "Optic disc localization in retinal fundus images based on you only look once network (YOLO)," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 2, pp. 332–342, Feb. 2023, doi: 10.22266/ijies2023.0430.27.
- [20] I. A. Alnajjar, L. Almazaydeh, A. A. Odeh, A. A. Salameh, K. Alqarni, and A. A. B. Atta, "Anomaly detection based on hierarchical federated learning with edge-enabled object detection for surveillance systems in Industry 4.0 Scenario," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 4, pp. 649–665, Aug. 2024, doi: 10.22266/ijies2024.0831.49.
- [21] B. Aydin and S. Singha, "Drone detection using YOLOv5," *Eng*, vol. 4, no. 1, pp. 416–433, Feb. 2023, doi: 10.3390/eng4010025.
- [22] A. Hendrawan, R. Gernowo, O. D. Nurhayati, and C. Dewi, "A novel YOLO-ARIA approach for real-time vehicle detection and classification in urban traffic," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 1, pp. 428–446, Feb. 2024, doi: 10.22266/ijies2024.0229.38.
- [23] B. Jiang, H. Jiang, H. Zhang, Q. Zhang, Z. Li, and L. Huang, "4AC-YOLOv5: an improved algorithm for small target face detection," *EURASIP Journal on Image and Video Processing*, vol. 2024, no. 1, May 2024, doi: 10.1186/s13640-024-00625-4.
- [24] B. Mahaur and K. K. Mishra, "Small-object detection based on YOLOv5 in autonomous driving systems," *Pattern Recognition Letters*, vol. 168, pp. 115–122, Apr. 2023, doi: 10.1016/j.patrec.2023.03.009.
- [25] M. Wang *et al.*, "FE-YOLOv5: feature enhancement network based on YOLOv5 for small object detection," *Journal of Visual Communication and Image Representation*, vol. 90, Feb. 2023, doi: 10.1016/j.jvcir.2023.103752.
- [26] M. Huang, W. Yan, W. Dai, and J. Wang, "EST-YOLOv5s: SAR image aircraft target detection model based on improved YOLOv5s," *IEEE Access*, vol. 11, pp. 113027–113041, 2023, doi: 10.1109/ACCESS.2023.3323575.
- [27] S. Wu, X. Lu, and C. Guo, "YOLOv5_mamba: unmanned aerial vehicle object detection based on bidirectional dense feedback network and adaptive gate feature fusion," *Scientific Reports*, vol. 14, no. 1, Sep. 2024, doi: 10.1038/s41598-024-73241-x.
- [28] F. Dadboud, V. Patel, V. Mehta, M. Bolic, and I. Mantegh, "Single-stage UAV detection and classification with YOLOv5: mosaic data augmentation and PANet," in *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Nov. 2021, pp. 1–8, doi: 10.1109/AVSS52988.2021.9663841.
- [30] J. S. Murthy, G. M. Siddesh, W.-C. Lai, B. D. Parameshachari, S. N. Patil, and K. L. Hemalatha, "ObjectDetect: a real-time object detection framework for advanced driver assistant systems using YOLOv5," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, Jan. 2022, doi: 10.1155/2022/9444360.

BIOGRAPHIES OF AUTHORS



Yuliani Indrianingsih    received her Bachelor of Science degree in Informatics and her Master of Computer Science degree (specific institutions and years not publicly listed). She currently serves as a lecturer in the Department of Informatics at Institut Teknologi Dirgantara Adisutjipto, Yogyakarta, Indonesia. Her research and teaching interests encompass decision sciences, computational theory, discrete mathematics and combinatorics, statistics, probability, and uncertainty modeling. She is actively involved in developing theoretical frameworks that support decision-making processes and mathematical modeling in Informatics. Her academic presence is further affirmed by her SINTA profile under ID 6001930. She can be contacted at email: yuliani@itda.ac.id.



Astika Ayuningtyas    is a lecturer and researcher at the Department of Informatics, Institut Teknologi Dirgantara Adisutjipto, Yogyakarta, Indonesia. She has an academic background in informatics and is actively involved in research in the fields of artificial intelligence, computer vision, and intelligent systems. Her primary research focus lies in object detection, intelligent surveillance, and the application of deep learning techniques for drone and aircraft monitoring systems. She has authored and co-authored several scientific publications in national and international journals and conference proceedings, including outlets indexed by SINTA and Scopus. Her academic identity and research outputs are registered and accessible through international scholarly platforms, enhancing the visibility and traceability of her work. She can be contacted at email: astika@itda.ac.id.



Prof. Anton Yudhana    received his Bachelor of Engineering in Electrical Engineering from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2001, his Master of Engineering in Electrical Engineering from Universitas Gadjah Mada, Yogyakarta, Indonesia, in 2005, and his Ph.D. degree in Electrical Engineering from Universiti Teknologi Malaysia in 2011. He also holds a professional engineering certification from Universitas Muhammadiyah Yogyakarta. He is currently a full professor in the Department of Electrical Engineering at Universitas Ahmad Dahlan, Yogyakarta, where he serves as the head of the Institute for Research and Community Service (LPPM). His scholarly identity and research outputs are indexed and accessible through international academic platforms, enhancing the visibility and impact of his work. He can be contacted at email: eyudhana@ee.uad.ac.id.



Haruno Sajati    received his Master of Engineering degree from Universitas Gadjah Mada, Yogyakarta. He is a lecturer in the Department of Informatics at Institut Teknologi Dirgantara Adisutjipto, Yogyakarta, and serves as head of the IT Service Center. He specializes in image processing and computer vision, focusing on image quality analysis and pre-processing techniques. His research interests span computer networks and communications, information security, pattern recognition, and software engineering. His scholarly output is reflected in his Google Scholar profile (214 citations overall, h-index of 8), along with profiles on SINTA (ID: 153727) and Garuda (ID: 368732). He can be contacted at email: haruno@itda.ac.id.



Anggraini Kusumaningrum    is a lecturer in the Department of Informatics at Institut Teknologi Dirgantara Adisutjipto, Yogyakarta. Her academic interests span decision sciences, renewable energy, sustainability, social sciences, and multidisciplinary domains. Her scholarly presence is recognized through multiple platforms: SINTA (ID: 6004642), and Garuda (ID: 623210). Her research contributions include the development of decision support systems. She has also worked on object detection, biometric detection, and simulation modeling. Her multidisciplinary publications reflect a commitment to applied informatics solutions addressing industry and community challenges. She can be contacted at email: anggraini@itda.ac.id.



Fajar Khanif Rahmawati    is a lecturer in the Department of Aerospace Engineering at Institut Teknologi Dirgantara Adisutjipto, Yogyakarta. She specializes in aviation system reliability, aircraft maintenance, and aerospace engineering. Her scholarly output includes analyses of aircraft assemblies, engine performance, helicopter rotor reliability, and maintenance planning. Her academic presence is highlighted by profiles on SINTA (ID: 6150998) and Garuda (ID: 893282). She can be reached via his institutional channels for collaboration, student consultation, or seminar participation. She can be contacted at email: fajarkhanif@itda.ac.id.



Prof. Imam Riadi    received his bachelor's degree in Electrical Engineering from Universitas Negeri Yogyakarta, a professional engineering qualification from Universitas Muhammadiyah Yogyakarta, and his master's and Ph.D. degrees in Computer Science from Universitas Gadjah Mada. Currently, he is the secretary of the Doctoral Program in Informatics (S3 DIFA) at Universitas Ahmad Dahlan, Yogyakarta. He previously served as head of the computer laboratory (FMIPA UAD), head of networking and communications (BISKOM UAD), assessor for LAM INFOKOM and BKD nasional, and as internal quality auditor at UAD. He stands among the leading academics in his field both at UAD and nationally, as reflected in Google Scholar and AD Scientific Index rankings. He is also registered with SINTA (ID: 22981) for his contributions in cybersecurity and forensics. He can be contacted at email: imam.riadi@is.uad.ac.id.