SINERGI Vol. 30, No. 2, June 2026: 449-458
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2026.2.011

Vision-guided tracking and detection using the YOLOv5 model on a logistic delivery fixed-wing UAV

Son Ali Akbar[1]*, Muhammad Taufiq Dinar Akbar[1], Anton Yudhana[1], Kamarul Hawari Ghazali[2]
[1]Department of Electrical Engineering, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Indonesia.
[2]Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, Malaysia

Abstract
Deep learning technologies utilizing Convolutional Neural Networks (CNNs) have advanced the development of autonomous systems, particularly in the exploration of hazardous environments. This study integrates the YOLOv5 object detection model with a fixed-wing Unmanned Aerial Vehicle (UAV) to identify simulated 5x5-meter orange marker dropping kits deployed in inaccessible disaster zones. Experiments were conducted at altitudes of 45 m, 75 m, and 100 m above sea level to assess real-time detection accuracy and terrain-mapping efficiency. The system achieved a mean Average Precision (mAP) of 88% across varying altitudes, demonstrating robust performance despite environmental challenges such as false positives from similarly colored rooftops. Computational efficiency tests were performed on the Jetson Nano platform using the TensorRT engine to accelerate object detection model inference on NVIDIA GPUs. Lighting variability significantly impacted detection reliability, resulting in a reduced mAP under suboptimal illumination. To enhance precision, post-processing filters and parameter optimizations were applied, improving the balance between detection sensitivity and specificity. These findings underscore the potential of YOLOv5-enabled UAVs for rapid, high-accuracy aid localization in disaster scenarios, although adaptive threshold tuning remains critical to address environmental variability in operational settings.

Keywords:
Deep Learning;
Fast Responder Disaster;
Object Detection;
UAV Fixed-Wing;
YOLOv5;

Article History:
Received: June 9, 2025
Revised: October 29, 2025
Accepted: November 25, 2025
Published: June 6, 2026

Corresponding Author:
Son Ali Akbar
Department of Electrical Engineering, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Indonesia
Email: sonali@ee.uad.ac.id

INTRODUCTION

Advances in robotics and artificial intelligence (AI) have revolutionized disaster management, enabling rapid, safe, and efficient responses in high-risk environments [1][2]. Unmanned aerial vehicles (UAVs), especially fixed-wing drones equipped with deep learning-based scanning systems, have become indispensable tools for surveying hazardous terrain, monitoring environmental conditions, and delivering aid to disaster zones. These systems leverage convolutional neural networks (CNNs), a technology that has seen significant progress, to process real-time data with high precision [3]. Among CNN-based models, You Only Look Once version 5 (YOLOv5) excels as a state-of-the-art object detector, offering fast and reliable identification of critical targets (e.g., survivors or supply markers) in cluttered settings. This capability is particularly valuable in disaster scenarios where speed and operational safety are paramount, as demonstrated by its potential for rapid emergency response [4].

Currently, YOLOv5 also addresses the challenges of low recognition accuracy and occlusion detection, particularly when identifying small or distant targets, such as unmanned aerial vehicles (UAVs) [4, 5, 6, 7, 8]. Despite its demonstrated capabilities, deploying YOLOv5 on fixed-wing UAV platforms is hindered by substantial implementation barriers [10]. Unpredictable environmental conditions

characteristic of disaster zones - including irregular topography, severe meteorological phenomena, and inconsistent lighting - have been shown to affect detection reliability adversely [11][12]. Additionally, operational factors (altitude, velocity, environmental interference) degrade detection of critical markers, while computational constraints and basic optics further limit high-speed UAV performance [13].

The You Only Look Once (YOLO) framework has undergone continuous development across multiple versions, each introducing architectural and performance improvements. However, these advancements do not always guarantee consistent superiority in practical applications [12][13]. To address specific challenges in real-world scenarios, the study developed an evaluation of an object detection system based on the YOLOv5 model, integrated into a fixed-wing UAV platform to identify logistical drop points in disaster-affected regions [16]. System performance is evaluated under diverse operational parameters and environmental conditions, with focused analysis on the detection accuracy of standardized orange square markers serving as designated drop-zone indicators [17]. Optimization is pursued through rigorous analysis of performance metrics and noise interference patterns to enhance operational reliability under dynamic conditions.

Additionally, the potential for reduced human intervention in high-risk environments is examined, offering a safer paradigm for disaster response operations [18]. While previous research has investigated CNN integration with UAVs, few have thoroughly tested YOLOv5's usefulness in real-world catastrophe scenarios, especially when limited to daytime operations due to lighting constraints. This gap underlines the importance of doing targeted research to optimize such systems for practical deployment. To mitigate these challenges, this work delivers the following key contributions:

- Improving object detection precision on high-altitude imagery with constrained training datasets using YOLOv5 architecture.
- Enhancing computational efficiency via TensorRT framework implementation, enabling real-time performance on resource-constrained systems.

RELATED WORK

Unmanned Aerial Vehicles (UAVs) equipped with vision-based sensing systems have emerged as a powerful tool for surveillance applications, offering real-time, high-resolution monitoring of large and complex environments. These systems leverage onboard cameras, a processing system, and a deep learning algorithm to autonomously [19]. The YOLO algorithm has become particularly prevalent in UAV applications due to its efficient multi-object detection capability and real-time performance, making it ideal for complex scenarios involving dense or overlapping targets [20].

Safonova [21] introduced a YOLOv5 model with multilayer feature fusion to improve small object detection in UAV imagery, highlighting enhanced feature representation as its main contribution. However, its dependence on augmented training data may limit generalization to real-world conditions. Notably, Ajayi [22] omitted evaluations of model performance under high UAV speeds, altitude changes, and optical disturbances such as motion blur and gaps; this study resolves these issues through YOLOv5 enhancements and dynamic condition optimization. While research on YOLOv5 for UAV applications is limited, Zhang [23] enhanced YOLOv5 using attention and multiscale strategies to improve small object detection in complex UAV imagery, effectively addressing challenges such as small object size, cluttered backgrounds, and dense scenes; however, its limitation lies in restricted generalization, as validation is conducted on limited datasets. Ling [24] proposed an improved YOLOv5 for dual-mode UXO detection using visible–infrared data, achieving high accuracy and real-time performance. However, it does not account for variations in data acquisition across different altitudes, which may limit robustness.

Numerous studies have investigated YOLO's operational performance when deployed on embedded minicomputer platforms. Dilmi [25] proposed a vision-based UAV system that employs an Intel camera for depth sensing and a Jetson version for onboard deep learning (YOLOv4-Tiny). The performance was sufficiently robust. Ling [26] processed the 2-D imagery dataset using an onboard Jetson system. The experiment used YOLOv4 to train an object detection model, achieving results demonstrating the system's ability to recognize small objects with low computational overhead efficiently.

METHOD

UAV Design

The fully assembled UAV, along with its integrated electrical and control systems, is depicted in Figure 1. This configuration demonstrates the complete structural layout, including the arrangement of propulsion components, sensor modules, power distribution units, and communication interfaces. The figure

Figure 1. Fixed-wing UAV

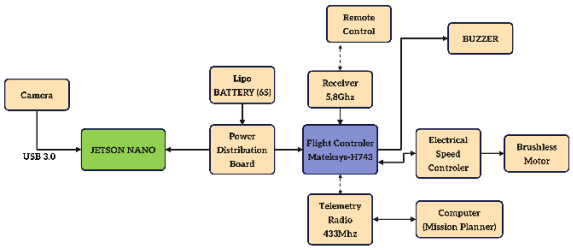

Figure 3. System Workflow

provides a comprehensive visualization. of the UAV's overall design, highlighting the integration between mechanical and electronic subsystems that enable stable flight performance and efficient operational functionality.

Figure 1 shows the aircraft model-based fixed-wing UAV. Development of a 1.3-meter wingspan fixed-wing UAV type T-tail integrates a Jetson Nano mini-PC and an autonomous flight control system powered by the Mateksys H743 Wing V3. The UAV is equipped with a Sony RX0 Mark II camera to enable advanced object detection capabilities. The Jetson Nano, configured with a 64 GB microSD card for storage and booting, runs Ubuntu 20.04 to ensure stable performance during flight missions.

The cameras are mounted on the ventral section of the fuselage in an optimized configuration for aerial observation, with protective cushioning integrated to ensure durability during belly landings. The camera system setup is detailed in Figure 2. As illustrated in Figure 2, the vision system employs sophisticated image processing algorithms to detect targets using visual cues, such as an object's color or shape. Furthermore, the UAV's system architecture is depicted in Figure 3. The proposed architecture demonstrates a modular design integrating three core subsystems: perception, control, and power management. The UAV's operational system architecture, illustrated in Figure 3, integrates multiple subsystems for comprehensive functionality.

A Remote Control unit enables manual operation while an auditory Buzzer provides system notifications. The visual perception system consists of a Sony RX0 Mark II camera interfaced with a Jetson Nano processor, which performs real-time image processing and artificial intelligence-based object detection.

Power management is achieved through a 6S LiPo battery connected to a Power Distribution Board (PDB), which allocates electrical power to three primary subsystems: (1) the Mateksys-H743 Flight Controller for autonomous navigation, (2) Electronic Speed Controllers (ESCs) for motor regulation, and (3) Brushless Motors that generate the necessary propulsion for flight operations. Continuous communication during flight operations is ensured by a full-featured ground control station (GCS), with the Mission Planner software interface depicted in Figure 4.

Figure 4 illustrates that the communication architecture employs a dual-channel wireless system, consisting of a 5.8 GHz receiver for command signal acquisition from the ground station and a 433 MHz telemetry radio for bidirectional transmission of positional and system status information to the Mission Planner software. The system's robust interoperability between hardware components and software interfaces ensures seamless data processing and precise actuation of flight control surfaces throughout operational phases [27].

Dataset handling

This study employs a logistic-ground image dataset as the primary source of visual data for analysis and model evaluation. An overview and a visual representation of the dataset are presented in Figure 5.

Figure 2. Camera Installation

Figure 4. Mission Planner Visualization

Figure 5. An image of the Drop Area

As depicted in Figure 5, to maintain framework compatibility with the Jetson embedded system, all training images were resized to 640×640 pixels. The training dataset was resized to maintain compatibility with the framework deployed on the Jetson platform [28]. The dataset was acquired using a smartphone camera, with the scale adjusted to reflect real field conditions. A simulation of the dataset dimensions was conducted at altitudes of 50 m, 75 m, and 100 m, corresponding to a real-world scale of 1:1000. The dataset comprises a total of 2,669 image frames, consisting of 1,877 images for the training set, 532 images for the validation set, and 260 images for the test set. Figure 6 illustrates the augmentation produced.

Figure 6 presents the image augmentation techniques applied, including rotation, brightness adjustment, and Gaussian transformation. These augmentation methods were employed to expand the dataset and mitigate the risk of overfitting. The image labeling process was conducted using Roboflow, with the COCO128 dataset serving as the reference for the labeling framework. This study used Roboflow annotation software to facilitate object labeling, ensuring consistent and precise bounding box annotations. Figure 7 illustrates the step-by-step labeling procedure performed.

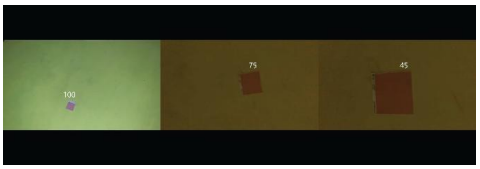

Figure 6. Scaling Image

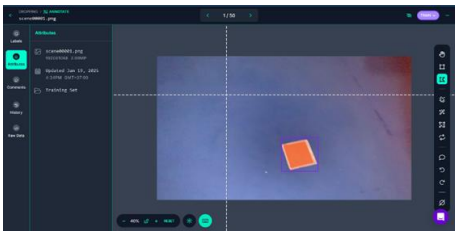

Figure 7. Image Annotation

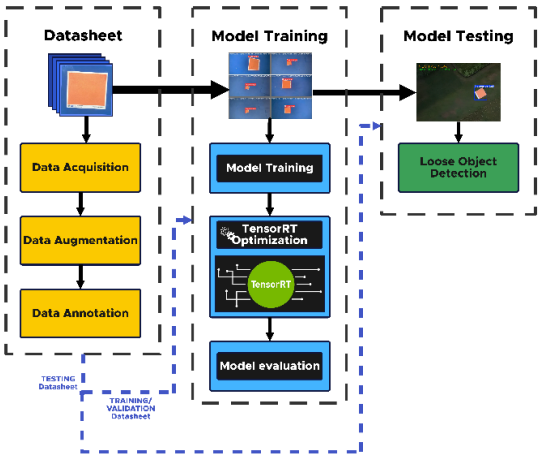

Figure 8. Proposed Method Architecture

Proposed Method

This study presents a computer vision architecture employing the YOLOv5 model, integrated with a fixed-wing UAV platform to accurately detect drop-kit load positions during disaster mitigation simulations. The implementation leverages TensorRT-based optimization to enhance inference speed, enabling real-time processing and efficient deployment in field environments. The overall pipeline architecture is illustrated in Figure 8.

Figure 8 describes the proposed method, which defines the YOLOv5 algorithm utilized for training the model, which is optimized on TensorRT. The architecture provided real-time detection of 5×5 m orange drop kits during

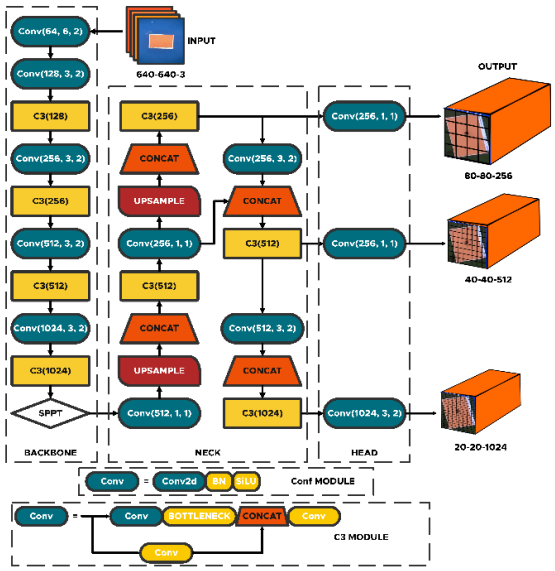

Figure 9. Architecture of YOLOv5

disasters. Additionally, the architecture of the YOLOv5 model is presented in Figure 8.

As depicted in Figure 9, YOLOv5 is a single-stage object detection architecture that employs a Cross-Stage Partial Network (CSPDarknet53) backbone and a Path Aggregation Network (PANet) neck for enhanced multiscale feature fusion [29]. The YOLOv5 architecture employs a multi-component loss function that combines distinct optimization objectives for different detection tasks. The final loss computation integrates these components through a weighted summation approach defined in (1).

$$Loss = \lambda_1 \cdot L_{cls} + \lambda_2 \cdot L_{obj} + \lambda_3 \cdot L_{loc} \quad (1)$$

The total loss combines three weighted terms: classification (L_{cls}) and objectness (L_{obj}) BCE losses, plus a CIoU-based localization loss (L_{loc}). Hyperparameters $\lambda_1 \lambda_2 \lambda_3$ control the balance of their optimization.

YOLOv5 achieves high accuracy even with small datasets while reducing computational demands. However, it requires a high-performance system. To improve computational efficiency, YOLOv5 can be optimized with TensorRT, an NVIDIA-developed inference framework that accelerates model execution on GPU architectures. Demonstrates the integration of the YOLOv5 object detection model accelerated by TensorRT, enabling efficient inference on the Jetson Nano platform while maintaining real-time performance during UAV flight operations [30]. TensorRT implementation also enables model deployment on edge devices like the NVIDIA Jetson Nano by compiling CUDA-architecture-optimized engine files. As a result, YOLOv5-TensorRT achieves an mAP@0.5 of approximately 56.6% on the COCO dataset, making it an effective solution for real-time embedded detection systems [31]. The TensorRT integration workflow with the YOLOv5 object detection system is illustrated in Figure 10.

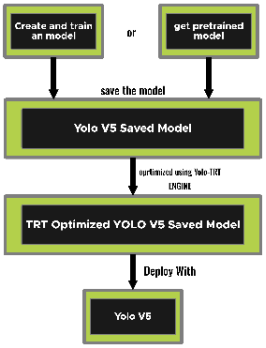

Figure 10. TensorRT Model

Figure 10 describes the optimization pipeline, which consists of: (1) converting the PyTorch model to the ONNX intermediate representation, followed by (2) precision quantization from 32-bit floating-point (FP32) to either 16-bit (FP16) or 8-bit integer (INT8) formats. This process achieves up to a 3× reduction in model size and inference latency through advanced techniques such as layer fusion and kernel optimization, while preserving near-baseline detection accuracy.

Experimental evaluation of the optimized YOLOv5 model quantified detection performance through precision, recall, mAP (mean average precision), and F1-score metrics computed during training and validation [32]. Precision quantifies the ratio of correctly identified positive predictions to all positive predictions made by the model. This metric, presented in the metrics/precision column, is calculated using (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

The recall metric, recorded in the metrics/recall column, quantifies the model's ability to identify all relevant positive cases, as defined by (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Mean Average Precision (mAP) evaluates detection accuracy across different confidence thresholds, with values reported in both metrics/mAP_0.5 and metrics/mAP_0.95 columns. The computation follows (4).

$$mAP = \frac{1}{n}\sum_{1=0}^{n} AP_i \quad (4)$$

F1-Score represents the harmonic mean of precision and recall, providing a balanced measure that accounts for both false positives and false negatives - a critical consideration in object detection tasks. This metric is derived using Equation (5).

$$F1 - Score = 2x\frac{Precision\ x\ Recall}{Precision\ x\ Recall} \quad (5)$$

The accuracy metric, recorded in the metrics/recall column, assesses the model's overall prediction correctness, computed in (6).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

RESULTS AND DISCUSSION

Experimental results

The model was trained using the train.py script with the following parameters: 640×640 input resolution, 150 epochs, and a batch size of 16. Computational constraints necessitated CPU-based training (enabled via the cpu parameter), which substantially increased the training duration. The training results, including bounding box predictions, are shown in Figure 11.

Figure 11 presents a visualization of the system's detection capability despite hardware limitations, with annotations indicating the model's precision under the specified conditions. Furthermore, the training metrics are demonstrated in Figure 12. As illustrated in Figure 12, the graph shows consistent improvement: train/box_loss decreased from 0.125 to 0.016, train/obj_loss from 0.031 to 0.009, and cls_loss remained stable at 0, indicating effective localization and object detection. Precision rose from 0.25 to 0.95, while recall achieved 1.0 by epoch 99, reflecting near-perfect detection capability. mAP@50 improved from 0.074 to 0.95, confirming robust accuracy. Validation losses (box: 0.125→0.011, obj: 0.024→0.005) declined steadily without overfitting, and minor fluctuations in val/obj_loss (epochs 45–50) had no significant impact. The model converged optimally with a near-zero learning rate by epoch 99.

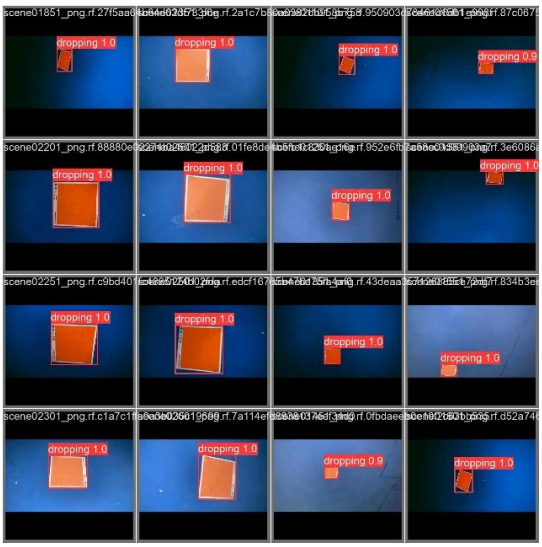

Figure 11. Object detected using the proposed model

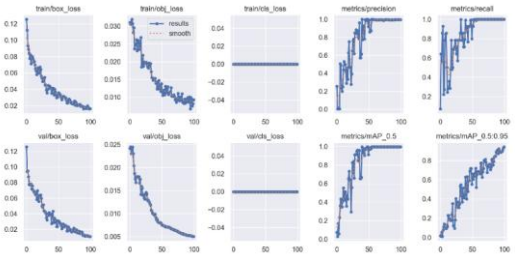

Figure 12. Training Graph

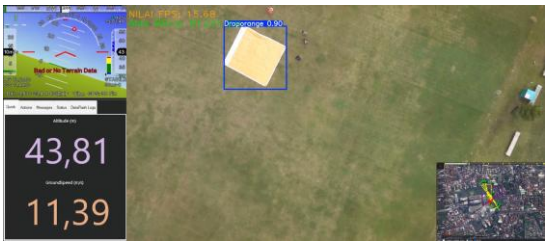

Figure 13. Implementation Test

Test Distance is the horizontal distance (in meters) between the sensor/camera and the target object during testing. Base Speed indicates the object's movement speed relative to the ground surface, displayed in a minimum–maximum range to illustrate variations in measurement results. The system's testing results, visualized through Mission Planner's interface, are shown in Figure 13. As shown in Figure 13, the image representation included spatial detection accuracy (as measured by the test altitude distance metric) and bounding-box tracking capability under maneuvering conditions.

Table 1 provides a comprehensive performance evaluation from flight tests assessing detection reliability across multiple operational parameters. The dataset captures quantitative metrics that demonstrate system robustness across diverse scenarios.

Table 1. Summary of Detection Performance Across Flight Trials

Flight Test	Ground Speed Range (m/s)	Altitude Average (m)	Average mAP
1	7.73–17.51	45.52	0.82
	6.10–14.92	76.21	0.86
	6.10–12.15	96.21	0.78
2	7.76–10.04	45.90	0.79
	13.18–13.47	72.15	0.80
	11.70–12.71	95.02	0.79
3	11.22–12.76	45.50	0.79
	20.95–22.11	78.01	0.77
	5.58–6.12	100.40	0.75
4	15.20–16.12	45.15	0.85
	15.16–15.48	76.79	0.84
	12.88–13.07	101.87	0.77

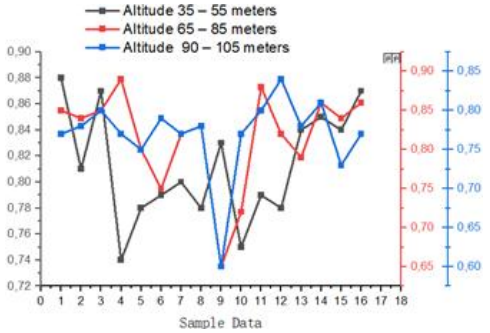

Figure 14. Altitude Data Trial

As shown in Table 1, the altitude referred to the vertical distance of an object from the ground surface, calculated as the average across all trials within the same distance category. mAP (Mean Average Precision) reflects the model's detection accuracy, computed as the average score (0–1 scale) across all iterations, with values ≥0.8 indicating reliable detection. Default Setpoint (e.g., 45 m, 75 m). Along with Table 1, the average at each flight altitude is illustrated in a graph shown in Figure 14.

Figure 14 presents the system's detection performance across experimental flight tests. The dataset comprises 16 measurements grouped into 4 distinct flight trials, each representing one of 4 predefined sample conditions. The graphical representation employed a color-coded scheme: blue (35–55 m test range), red (65–85 m), and yellow (90–105 m). The performance average evaluation in the test flight of the detection system is conducted through quantitative analysis using a set of standard metrics listed in Table 2. Accuracy measures the overall percentage of correct detections using the Recall, Precision, and F1 Score.

As shown in Table 2, the flight tests showed strong but varying detection performance. The flight test results demonstrated consistent detection performance across four trials, with accuracy remaining high (0.89–0.92) despite variations in other metrics. Recall showed stability (0.80–0.87), while precision exhibits greater fluctuation (0.71–0.89), particularly in Test 3, where a precision drop to 0.71 corresponded with a reduced F1-score (0.84).

Table 2. Detection Accuracy Evaluation

Flight test	Recall	Precision	F1-Score	Accuracy
1	0.82	0.89	0. 90	0.91
2	0.87	0.83	0.77	0.89
3	0.80	0.71	0. 84	0.90
4	0.84	0.88	0.81	0.92

Figure 15. Detection of Another Object Test

Notably, Test 2 showed an inverse relationship between precision (0.83) and recall (0.87), yielding the lowest F1-score (0.77), suggesting potential trade-offs between detection sensitivity and false positives. Test 4 achieved the most balanced performance with recall (0.84), precision (0.88), and peak accuracy (0.92), indicating optimal parameter tuning. The average detection value accumulation from each test flight will be used to analyze the appearance of uncounted objects in Figure 15. (those not originally included in the detection objects). The table above displays average detection values representing each test flight.

The mean Average Precision (mAP) is a standard metric in object detection and information retrieval tasks. It quantifies the overall detection accuracy by averaging the Average Precision (AP) scores across distinct classes or queries. In the provided table, each row likely corresponds to a specific class or query, with Precision and Recall values either computed at predefined thresholds or averaged across multiple thresholds. Furthermore, this study compared the performance of the proposed model with baseline models (e.g., YOLOv3, YOLOv4). Table 3 shows the performance comparison.

Based on Table 3, the evaluation was conducted by testing the pre-trained weights from the baseline model. The YOLOv5 model demonstrated superior performance, achieving an mAP@50 of 0.95 and the lowest average loss value of 0.10. This remarkably high mAP score indicates that the model is not only highly accurate at object identification but also effective at detecting the majority of objects in the dataset.

Comprehensive analysis

This study investigates the processing efficiency of a Jetson Nano-based CPU system for high-altitude object detection.

Table 3. Comparison Performance with Baseline Model

Model	mAP@50	Avg Loss	FPS (Inference Speed)
YOLOv3	0.80	0.25	70
YOLOv4	0.88	0.15	50
Proposed (YOLOv5)	0.95	0.10	60

The proposed YOLOv5 implementation demonstrates robust performance under elevated operational conditions, maintaining detection accuracy despite computational constraints. To assess the proposed method, a benchmark comparison was conducted against leading state-of-the-art models using similar experimental setups. The comparative analysis is defined in Table 4.

As shown in Table 4, the proposed YOLOv5-based architecture achieves enhanced detection performance, with a mean average precision (mAP) of 0.88, representing significant improvements over comparable state-of-the-art approaches. This 0.01 mAP enhancement over the baseline YOLOv5s implementation (mAP 0.87) may appear modest, but it proves statistically significant ($p < 0.05$).

CONCLUSION

This study presents critical findings from a systematic evaluation of a YOLOv5-based detection system implemented on a fixed-wing UAV platform for identifying aerial-dropped orange relief kits in disaster scenarios. The system was specifically developed to facilitate search-and-rescue operations by reliably detecting orange-colored aid packages in affected areas. When deployed on a Jetson Nano embedded system, the solution demonstrated robust detection capabilities, with the YOLOv5 model successfully identifying rectangular orange targets while achieving an 88% mAP score, 90.5% accuracy, 83.2% recall, 82.7% precision, and an 83% F1 score. This research was constrained by its experimental implementation within an urban environment, where the designated drop-off areas were intentionally positioned among residential structures. The challenges posed by environmental interference affecting detection performance were effectively mitigated. Future work should expand the experimental scope to diverse environments, including rural, forested, and coastal regions, and further develop the proposed method to enhance the reliability of UAV technology in supporting disaster mitigation efforts.

Table 4. Summary of A Comparison with an Existing Study

References	Dataset	Method	Performance
[33]	Self collected	YOLOv5	mAP = 0.73
[25]	Self collected	YOLOv5	mAP = 0.76
[34]	Self collected	YOLOv5s	mAP = 0.87
Proposed study	Self collected	YOLOv5	mAP = 0.88

ACKNOWLEDGMENT

The authors gratefully acknowledge the technical support and expert contributions provided by the Robotics Research Group at Universitas Ahmad Dahlan, Indonesia.

REFERENCES

[1] M. El Adawy *et al.*, “Design and fabrication of a fixed-wing Unmanned Aerial Vehicle (UAV),” *Ain Shams Eng. J.*, vol. 14, no. 9, p. 102094, 2023, doi: 10.1016/j.asej.2022.102094.

[2] M. Shahrieel, M. Aras, P. Ponusamy, and M. R. Nawawi, “Real-time Unmanned Surface Robot (USR) for river quality monitoring system,” *SINERGI*, vol. 30, no. 1, pp. 1–10, 2026, doi: 10.22441/sinergi.2026.1.001.

[3] M. H. Bashir, M. Ahmad, D. R. Rizvi, and A. A. A. El-Latif, “Efficient CNN-based disaster events classification using UAV-aided images for emergency response application,” *Neural Comput. Appl.*, vol. 36, no. 18, pp. 10599–10612, 2024, doi: 10.1007/s00521-024-09610-4.

[4] J. Su, X. Zhu, S. Li, and W. Chen, “AI meets UAVs : A survey on AI empowered UAV perception systems for precision agriculture,” *Neurocomputing*, vol. 518, pp. 242–270, 2023, doi: 10.1016/j.neucom.2022.11.020.

[5] Y. Li, Y. Xia, G. Zheng, X. Guo, and Q. Li, “YOLO-RWY: A Novel Runway Detection Model for Vision-Based Autonomous Landing of Fixed-Wing Unmanned Aerial Vehicles,” *Drones*, vol. 8, no. 10, 2024, doi: 10.3390/drones8100571.

[6] S. Cao, T. Wang, T. Li, and Z. Mao, “UAV small target detection algorithm based on an improved YOLOv5s model,” *J. Vis. Commun. Image Represent.*, vol. 97, p. 103936, 2023, doi: 10.1016/j.jvcir.2023.103936.

[7] Z. Liu, X. Gao, Y. U. Wan, J. Wang, and H. A. O. Lyu, “An Improved YOLOv5 Method

for Small Object Detection in UAV Capture Scenes," *IEEE Access*, vol. 11, no. February, pp. 14365–14374, 2023, doi: 10.1109/ACCESS.2023.3241005.

[8] H. Zhang, F. Shao, X. He, Z. Zhang, Y. Cai, and S. Bi, "Research on Object Detection and Recognition Method for UAV Aerial Images Based on Improved YOLOv5," *Drones*, vol. 7, no. 06:402, 2023, doi: 10.3390/drones7060402.

[9] Y. Yang *et al.*, "Automatic terminal guidance for small fixed-wing unmanned aerial vehicles," *J. F. Robot.*, vol. 40, no. 1, pp. 3–29, Jan. 2023, doi: 10.1002/rob.22113.

[10] S. Zhou *et al.*, "Improved YOLO for long-range detection of small drones," *Sci. Rep.*, vol. 15:12280, 2025, doi: 10.1038/s41598-025-95580-z 1.

[11] Y. Song *et al.*, "Distributed swarm system with hybrid-flocking control for small fixed-wing UAVs: Algorithms and flight experiments," *Expert Syst. Appl.*, vol. 229, p. 120457, 2023, doi: 10.1016/j.eswa.2023.120457.

[12] A. Purwanto *et al.*, "Image Segmentation in Aerial Imagery : A Review," *SINERGI*, vol. 27, no. 3, pp. 343–360, 2023, doi: 10.22441/sinergi.2023.3.006.

[13] S. Sambolek and M. Ivasic-Kos, "Person Detection and Geolocation Estimation in Drone Images," *SN Comput. Sci.*, vol. 6, no. 4, p. 358, 2025, doi: 10.1007/s42979-025-03869-7.

[14] A. Vijayakumar and S. Vairavasundaram, "YOLO-based Object Detection Models: A Review and its Applications," *Multimed. Tools Appl.*, vol. 83, no. 35, pp. 83535–83574, 2024, doi: 10.1007/s11042-024-18872-y.

[15] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, 2024, doi: 10.3390/computers13120336.

[16] Y. Liu *et al.*, "Detection method of the seat belt for workers at height based on UAV image and YOLO algorithm," *Array*, vol. 22, no. March, p. 100340, 2024, doi: 10.1016/j.array.2024.100340.

[17] J. Di, K. Xi, H. Niu, X. Wu, and Y. Yang, "Enhanced YOLOv8 Framework for Precise Small Object Detection in UAV Imagery," *IEEE Access*, vol. 13, no. August, pp. 157811–157827, 2025, doi: 10.1109/ACCESS.2025.3604772.

[18] B. Xu, B. Gao, and Y. Li, "Improved Small Object Detection Algorithm Based on YOLOv5," *IEEE Intell. Syst.*, vol. 39, no. 5, pp. 57–65, 2024, doi: 10.1109/MIS.2024.3399053.

[19] C. Xu, W. Zhao, J. Zhao, Z. Guan, X. Song, and J. Li, "Uncertainty-Aware Multiview Deep Learning for Internet of Things Applications," *IEEE Trans. Ind. Informatics*, vol. 19, no. 2, pp. 1456–1466, 2023, doi: 10.1109/TII.2022.3206343.

[20] L. Jiao and M. I. Abdullah, "YOLO series algorithms in object detection of unmanned aerial vehicles: a survey," *Serv. Oriented Comput. Appl.*, vol. 18, no. 3, pp. 269–298, 2024, doi: 10.1007/s11761-024-00388-w.

[21] S. Zeng, W. Yang, Y. Jiao, L. Geng, and X. Chen, "SCA-YOLO: a new small object detection model for UAV images," *Vis. Comput.*, vol. 40, no. 3, pp. 1787–1803, 2024, doi: 10.1007/s00371-023-02886-y.

[22] O. G. Ajayi, J. Ashi, and B. Guda, "Smart Agricultural Technology Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images," *Smart Agric. Technol.*, vol. 5, no. April, p. 100231, 2023, doi: 10.1016/j.atech.2023.100231.

[23] J. Zhang, G. Wan, M. Jiang, G. Lu, X. Tao, and Z. Huang, "Small object detection in UAV image based on improved YOLOv5," *Syst. Sci. Control Eng.*, vol. 11, no. 1, p. 2247082, 2023, doi: 10.1080/21642583.2023.2247082.

[24] Z. Ling, H. Zhao, X. Zhao, Z. Liu, and W. Chen, "A Dual Mode Detection Method for Unexploded Ordnance Based on YOLOv5 for Low Altitude Unmanned Aerial Vehicle," *IEEE Access*, vol. 13, no. January, pp. 42634–42649, 2025, doi: 10.1109/ACCESS.2025.3537058.

[25] W. Dilmi, S. El Ferik, F. Ouerdane, M. K. Khaldi, and A.-W. A. Saif, "Technical Aspects of Deploying UAV and Ground Robots for Intelligent Logistics Using YOLO on Embedded Systems," *Sensors*, vol. 25, no. 08:2572, 2025, doi: 10.3390/s25082572.

[26] M. A. M. Alhassan and E. Yılmaz, "Evaluating YOLOv4 and YOLOv5 for Enhanced Object Detection in UAV-Based Surveillance," *Processes*, vol. 13, no. 01:254, 2025, doi: 10.3390/pr13010254.

[27] S. Kusmirek *et al.*, "Dynamic Flight Tracking : Designing System for Multirotor UAVs With Pixhawk Autopilot Data Verification," *IEEE Access*, vol. 12, no. June, pp. 109806–109821, 2024, doi: 10.1109/ACCESS.2024.3441115.

[28] H. Ijaz, R. Ahmad, R. Ahmed, W. Ahmed, Y. Kai, and W. Jun, "A UAV-Assisted Edge Framework for Real-Time Disaster

Management," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–13, 2023, doi: 10.1109/TGRS.2023.3306151.
[29] S. Feng, H. Qian, H. Wang, and W. Wang, "Real-time object detection method based on YOLOv5 and efficient mobile network," *J. Real-Time Image Process.*, vol. 21, no. 2, p. 56, 2024, doi: 10.1007/s11554-024-01433-9.
[30] I. Ben Rouighi, H. Chtioui, I. Jegham, I. Alouani, and A. Ben Khalifa, "FRD-YOLO: a faster real-time object detector for aerial imagery," *J. Real-Time Image Process.*, vol. 22, no. 5, p. 169, 2025, doi: 10.1007/s11554-025-01750-7.
[31] Z. Hua *et al.*, "A Benchmark Review of YOLO Algorithm Developments for Object Detection," *IEEE Access*, vol. 13, no. June, pp. 123515–123545, 2025, doi: 10.1109/ACCESS.2025.3586673.
[32] Q. Fan, Y. Li, M. Deveci, K. Zhong, and S. Kadry, "LUD-YOLO: A novel lightweight object detection network for unmanned aerial vehicle," *Inf. Sci. (Ny).*, vol. 686, no. August 2024, p. 121366, 2025, doi: 10.1016/j.ins.2024.121366.
[33] S. Li, X. Yang, X. Lin, Y. Zhang, Wu, and Jiahui, "Real-Time Vehicle Detection from UAV Aerial Images Based on Improved YOLOv5," *Sensors*, vol. 23, no. 12:5634, 2023, doi: 10.3390/s23125634.
[34] Y. Sun *et al.*, "Enhancing UAV Detection in Surveillance Camera Videos through Spatiotemporal Information and Optical Flow," *Sensors*, vol. 23, no. 13, 2023, doi: 10.3390/s23136037.