Vision-based UAV Detection Models for Small-Edge Devices*

Iryna Yurchuk[1, †], Taras Semenchenko[1,*, †]

[1] Taras Shevchenko National University of Kyiv

Abstract

Unmanned Aerial Vehicles (UAVs) have become very common in modern combat scenarios, making them extremely dangerous weapons that must be effectively detected and eliminated. Traditional detection methods—relying on radio frequencies, radars, and other sensors—are often inefficient due to the low radar visibility and compact size of modern UAVs. This paper introduces a modern UAV detection system with state-of-the-art computer vision models to process video frames in real time.

To support our approach, we developed a custom dataset comprising approximately 2,000 manually annotated images, capturing diverse environmental conditions similar to real-world scenarios where this algorithm can be applied. Additionally, to increase the training dataset size we combined our dataset with several publicly available ones in order to improve the robustness of our detection models. Then we fine-tuned several leading object detection algorithms, including model YOLO, Faster R-CNN, Mask R-CNN, and RT-DETR. We evaluated their performance using mean Average Precision (mAP) metrics and frames per second (FPS). Our findings show that current AI technologies can achieve high accuracy and, at the same time, real-time processing speeds on relatively small devices, which means that they offer a reliable alternative to traditional radar-based detection systems.

We also discuss the trade-offs between UAV detection accuracy and computational efficiency and analyze strategies for deploying these models on small-edge devices. Our results show that computer vision algorithms are mature enough to provide robust UAV detection solutions, potentially improving military operations' situational awareness and response capabilities.

Keywords

UAV, Object Detection, Real-Time, YOLO, RT-DETR, Edge Computing

1. Introduction

Recent events in Ukraine have shown that drones are frequently used in military operations. They play a crucial role in tasks such as intelligence gathering, surveillance, and combat. However, their small size and high speed also make them challenging targets to spot using traditional methods such as radar systems, which increases the risk of unauthorized or adversarial use.

The rapid growth of drone usage in recent times has led to serious security concerns—including illegal spying and even terrorist attacks [1]. Traditional detection systems often struggle because drones have low radar visibility [1][2], and factors such as low light or poor weather make traditional imaging techniques even more complicated to use [3]. Current studies investigated alternative approaches utilizing audio-visual fusion and deep learning-based approaches that analyze acoustic sounds [4][5]. Although these new methods look promising, each has limitations for accuracy, speed, and robustness. Our study focuses on two main objectives: first, we aim to find the most optimal model for UAV detection task based on visual data, which has a good balance between accuracy and speed, and second, we are developing a robust real-time system capable of accurately identifying UAVs under real-world conditions. To tackle these challenges, we rely on the advancements in computer vision and artificial intelligence to build a more precise drone detection solution.

*International Workshop on Computational Intelligence, co-located with the IV International Scientific Symposium "Intelligent Solutions" (IntSol-2025), May 01-05, 2025, Kyiv-Uzhhorod, Ukraine

[1,*] Corresponding author.

i.a.yurchuk@gmail.com (I. Yurchuk); taras.semenchenko@knu.ua (T. Semenchenko);

0000-0001-8206-3395 (I. Yurchuk); 0009-0007-3259-7007 (T. Semenchenko)

The object of our research is the process of drone detection using artificial intelligence techniques. The subject is computer vision algorithms optimized for real-time UAV detection for small-edge devices. The primary aim of the study is to develop a reliable, high-performance detection system capable of identifying UAVs effectively in diverse operational environments. To achieve this aim, we have established the following tasks:

- Develop a comprehensive and representative dataset combining manually annotated data and publicly available sources.
- Evaluate and compare state-of-the-art object detection algorithms to identify the optimal model capable of accurately recognizing UAVs in real time, balancing high detection accuracy and computational efficiency for deployment on edge devices.

Motivated by these challenges, our study aims to identify a state-of-the-art object detection algorithm that can accurately detect UAVs in real time while remaining efficient enough to run on small-edge devices with limited computational resources. Combining a manually annotated dataset with model evaluations using mAP@50 and FPS metrics helps us find an optimal spot between high detection accuracy and operational efficiency, ultimately contributing to developing more reliable UAV recognition systems for military and other critical applications.

Although this study provides valuable insights into UAV detection, it has some limitations. First, while the dataset tries to be as similar to real-world use as possible, it may not include all real-life situations, such as harsh weather or all types of drones. Second, despite using different image augmentation techniques, such as flipping, blurring, and changing color, our evaluations were still conducted under controlled conditions, which might not fully show the challenges of real-world environments.

Finally, the computational performance evaluations were conducted using desktop GPU (RTX 3070 ti specifically), which means that results may vary when deployed on other platforms, especially with much lower computational resources. This setup was chosen to allow efficient testing and ensure fair, consistent comparison across all models. Future research should address these limitations by exploring a broader range of environmental scenarios and testing on portable mini-GPU systems.

The remaining part of this paper is structured as follows. Section 2 reviews the related works, providing an overview of current approaches in UAV and object detection. Section 3 details our methods, starting with a discussion of various object detection algorithms, including Faster-RCNN [14], Mask-RCNN [15], YOLO [16], and RT-DETR [17], followed by a description of our dataset—combining both publicly available data and a manually labeled dataset that includes details on data collection, data splitting, and object characteristics, and at the end of this section, we review the model training pipeline. In Section 4, we present our experimental results through tables of metrics. Finally, Section 5 concludes the paper by summarizing our findings and discussing potential future directions. References are provided at the end.

2. Related Works

The detection of Unmanned Aerial Vehicles (UAVs) has gained much attention due to the increasing use of drones in commercial and security applications. Real-time UAV detection presents challenges like small object sizes, distinguishing objects on complex backgrounds, and varying environmental conditions. The last advances in deep learning-based object detection models, such as YOLO and RT-DETR, have improved UAV detection accuracy. This section reviews previous researches related to UAV detection while focuses on deep learning-based approaches, and briefly discusses alternative vision-based methods.

A comprehensive review by Cao et al. [6] provides an overview of UAV detection methods, covering various detection paradigms, hardware architectures, and optimization strategies. The study highlights that deep learning algorithms are preferred due to their superior accuracy and that GPU-based edge computing platforms are commonly used for real-time detection. It also emphasizes that beyond detection accuracy, speed, latency, and energy efficiency are critical factors in UAV

detection system performance. This review sets the foundation for evaluating specific deep-learning models used in UAV detection.

Among deep learning-based approaches, YOLO has been widely used due to its high-speed processing and accuracy. Barisic et al. [8] developed a YOLO-based UAV detection system, training it on a dataset of 10,000 images to detect various multirotor UAVs in different environments. Their model achieves real-time performance of 20 FPS on an edge computing device, making it suitable for practical deployment. Building on YOLO-based approaches, Zhai et al. [12] introduced YOLO-Drone, an optimized version of YOLOv8 designed explicitly for tiny UAV detection. Their modifications include a high-resolution detection head, reduced network parameters, and feature extraction enhancements, leading to a precision improvement of 11.9%, recall improvement of 15.2%, and mean average precision (mAP) improvement of 9% over the baseline. The model also significantly reduces computational requirements, making it well-suited for real-time UAV detection in resource-limited environments.

Several studies have compared deep learning models for UAV detection. Zhao et al. [7] introduced the DUT Anti-UAV dataset, which consists of manually labeled 10,000 images and 20 tracking videos and used it to train multiple object detection algorithms. Also, their study provides a comprehensive benchmark for estimating the performance of object detection and tracking models.

Beyond deep learning, some researchers have explored template matching and filtering for UAV detection. Opromolla et al. [9] proposed a vision-based detection system that uses template matching and morphological filtering to detect cooperative UAVs. While this approach is computationally efficient, it lacks the adaptability and robustness of deep learning-based models, especially in dynamic environments. For UAV-to-UAV detection applications, Li et al. [10] introduced a "see-and-avoid" system, which combines motion-based target detection and tracking to prevent UAV collisions. Mejias et al. [11] developed a vision-based system designed to prevent collisions by identifying aerial targets within a range of 400m to 900m. Although these studies primarily explore UAV tracking and navigation instead of broad object detection, they offer valuable knowledge on real-time data processing and movement prediction methods.

One of the challenges in UAV detection is low visibility conditions, such as night-time surveillance. Andraši et al. [3] investigated thermal infrared-based UAV detection, showing that infrared cameras can detect slight heat variations emitted by UAVs. However, electrically powered drones generate minimal heat, making thermal-based detection less effective compared to deep learning-based RGB image analysis.

Recent advances in state-of-the-art (SOTA) object detection models, such as RT-DETR and YOLOv10, have significantly improved UAV detection capabilities. These models leverage transformer-based architectures and optimized CNN layers, achieving real-time performance with high detection accuracy.

Overall, while deep learning-based approaches, particularly YOLO variants, demonstrate strong performance in real-time UAV detection, further research is needed to optimize models for deployment on edge computing devices with limited computational power, improve detection accuracy in challenging environments such as night-time surveillance or urban settings with complex backgrounds, and evaluate newer SOTA models like RT-DETR to compare their efficiency with existing deep learning-based UAV detection methods. This study aims to address these research gaps by comparing the performance of YOLOv10, RT-DETR, and other deep learning-based models to determine the most effective approach for real-time UAV detection.

3. Methodology

3.1. Dataset

In order to conduct benchmarks of object detection models, we were using publicly available dataset along with manually labeled one.

3.1.1. Publicly available data

For publicly available data, we were looking for datasets that would include diverse kinds of UAVs from various backgrounds and lightning. An additional requirement was for the UAV, which should have been captured from the ground, and the camera should be directed toward the sky. We considered the following datasets:

DUT-Anti-UAV [7]. This a visible light mode dataset called Dalian University of Technology Anti-UAV dataset (DUT Anti-UAV). It is a detection dataset with 10,000 of manually annotated images, in which the training, testing, and validation sets have 5200, 2200 and 2600 images, respectively.

Drone-vs-Bird Detection Dataset [13]. Developed for the Drone-vs-Bird Detection Challenge (ICASSP 2023), this dataset consists of 77 training video sequences and 30 test sequences recorded in varied environments such as urban, maritime, and woodland areas. It includes eight drone types (e.g., DJI Inspire, Phantom, Mavic) captured with static and moving cameras under different weather and lighting conditions. The dataset presents challenges like small drone sizes, motion blur, and environmental disturbances, with birds frequently appearing as non-annotated objects.

3.1.2. Manually labeled dataset

In addition to the prepared data, we created a dataset that contains manually annotated video frames. We assume this data will be more similar to the data that the model will get during inference.

Figure 1: Examples of the annotated object detection images in our dataset.

Data Collection. For data collection, was used video footage as the primary source. The video files were split into individual frames by extracting one frame approximately every 5 seconds. These frames were then saved as separate image files for further processing, ensuring a dataset that closely resembles real-world inference conditions. When preparing the dataset, recommendations from this article were followed [20]. All extracted frames were then annotated precisely.

Data Splitting. The manually annotated dataset, comprising 2000 images in total, was divided into training and testing sets using an 80%-20% split, resulting in 1600 training images and 400 testing images.

Objects Characteristics. As shown on the Figure 1, the dataset includes UAVs recorded in diverse outdoor settings from ground-to-sky perspectives such as skies with clouds and playgrounds under various lighting and weather conditions. Most UAVs appear as small target objects with area ratios averaging around 0.013 and aspect ratios mostly between 1.0 and 3.0, although some vary significantly. Object positions are mainly centered but exhibit varied motion, ensuring that the dataset presents challenging scenarios for robust object detection.

3.2. Models

For a comprehensive comparison of different approaches, we selected object detection algorithms that can be divided into three categories: single-stage detectors, two-stage detectors, and transformer-based detectors. From each group, we chose widely used models that offer relatively high performance and can be deployed in real-time detection scenarios.

Faster-RCNN. This is a widely used two-stage object detection framework that efficiently generates region proposals using an integrated Region Proposal Network (RPN). As an improved version of Fast R-CNN [19], it shares full-image convolutional features between the RPN and the detection network, enabling nearly cost-free proposals and end-to-end training that effectively directs the network's attention to promising regions. This unified approach accelerates the detection process and has been successfully applied to datasets such as MS COCO [18].

Mask-RCNN. It is a flexible framework for object instance segmentation that detects objects and generates high-quality segmentation masks simultaneously. It extends Faster R-CNN by adding a branch for mask prediction alongside bounding box recognition, with minimal overhead. This unified approach is easy to train and generalizes to tasks like human pose estimation, making it a robust baseline for instance-level recognition.

YOLOv10. It is a one-stage detector that predicts bounding boxes and object classes from a single pass of the input data through the model. This method is known for its high speed and relatively high performance, making it one of the most suitable algorithms for real-time object detection, although it may not be precise enough for detecting small objects or objects close to the camera. In this study, we consider YOLOv10, which reduces reliance on non-maximum suppression (NMS) and improves accuracy with a novel training approach, as they represent distinct yet highly effective models within the YOLO family.

RT-DETR. This model is a state-of-the-art real-time end-to-end object detection framework that addresses the limitations of NMS-based methods and high computational cost in Transformer detectors. It employs an efficient hybrid encoder that decouples intra-scale interactions from cross-scale fusion to rapidly process multi-scale features, along with uncertainty-minimal query selection to provide high-quality decoder inputs. RT-DETR also offers flexible speed tuning by adjusting the number of decoder layers without retraining, achieving competitive performance (e.g., 53.1% AP on COCO at 108 FPS with RT-DETR-R50).

3.3. Training/Evaluation pipeline overview

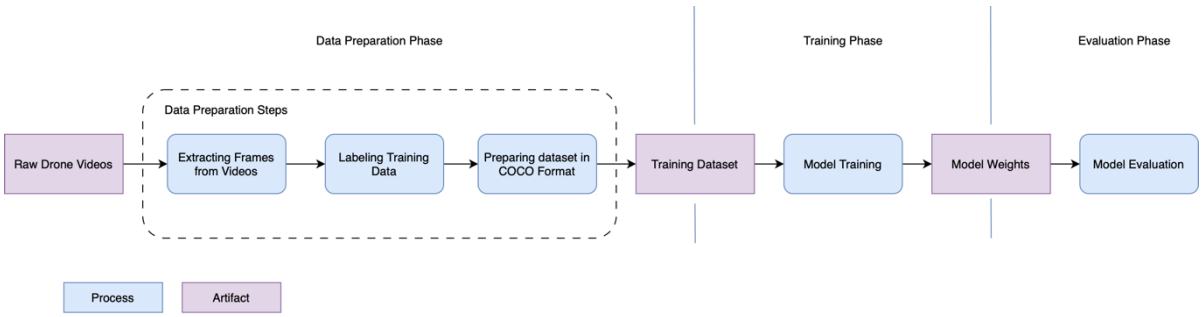

Figure 2: Training and Evaluation Pipeline.

The Figure 2 shows the end-to-end workflow used for training and evaluating of the models in our experiments. The workflow consists of three main phases:

1. **Data Preparation.** The process begins with raw drone videos, which are converted into training data through several steps:
    - Extracting Frames from Videos: Frames are extracted with fixed intervals (every 5 seconds).
    - Labeling Training Data: The extracted frames are manually annotated with bounding boxes around UAVs. For annotation "Label Studio" tool was used.
    - Preparing Dataset in COCO Format: The labeled data is structured into the COCO format which is one of the most common formats.
2. **Training Phase**. The resulting dataset is then used to train object detection models. This step also produces trained weights which are used for evaluations and inference.
3. **Evaluation Phase**. The trained models are used in the evaluation step using performance metrics such as mAP and FPS.

4. Experiments

Table 1
Result comparison of object detection models.

Model	mAP@50	FPS	GFLOPs	Params (M)
Faster R-CNN	0.948	15.85	141.1	48.51
Mask R-CNN	0.927	13.29	141.1	51.16
YOLOv10-n	0.907	56.12	4.20	2.71
YOLOv10-s	0.918	51.82	12.3	8.07
YOLOv10-m	0.939	47.02	31.9	16.49
YOLOv10-l	0.964	40.77	63.6	25.77
YOLOv10-x	0.953	38.92	85.5	31.66
RT-DETR	0.971	17.47	54.0	32.81

To evaluate the performance of the selected object detection models, we used a set of widely recognized metrics in the field of object detection:

- **Accuracy: mean Average Precision (mAP):** We used mean Average Precision (mAP) at IoU thresholds of 0.50 (mAP@50). IoU measures bounding box overlap. Average Precision (AP) is the area under the Precision-Recall curve. mAP is the average AP across all classes, but in our example, we evaluate only one class that is UAV.
- **Speed: Frames Per Second (FPS):** Speed was measured in Frames Per Second (FPS), indicating images processed per second. FPS was calculated by running models on test images, measuring processing time, and averaging. FPS is hardware-dependent, so consistent hardware was used, which is Nvidia RTX 3070 ti GPU.

For the two-stage detectors, Faster R-CNN and Mask R-CNN, we followed a similar training regime. Both models were trained for three epochs using a complete fine-tuning approach, meaning all layers of the pre-trained networks were updated during training. To manage computational resources and ensure stable gradient updates, we used a batch size of 4 for both Faster R-CNN and

Mask R-CNN. This consistent training procedure allowed for a direct comparison of their performance under similar conditions.

In contrast, the YOLO family of models (YOLOv10-n, s, m, l, x) was trained with a different strategy focused on leveraging pre-trained weights while adapting to our specific dataset. We observed that freezing the majority of layers, specifically approximately 80% of the layers, except for the final detection layers, yielded the best performance for these models in our experiments. Consequently, all YOLO variants were trained, with 80% of their layers frozen, and only the last layers were fine-tuned.

Finally, for RT-DETR, the transformer-based detector, we applied full fine-tuning for 10 epochs, using a batch size of 2 due to the high GPU memory requirements of the transformer architecture.

For a more complete model comparison, we also included metrics such as GFLOPs, which indicate the computational complexity of each model, and the number of parameters (in millions), which reflects model size and can impact inference time and memory usage.

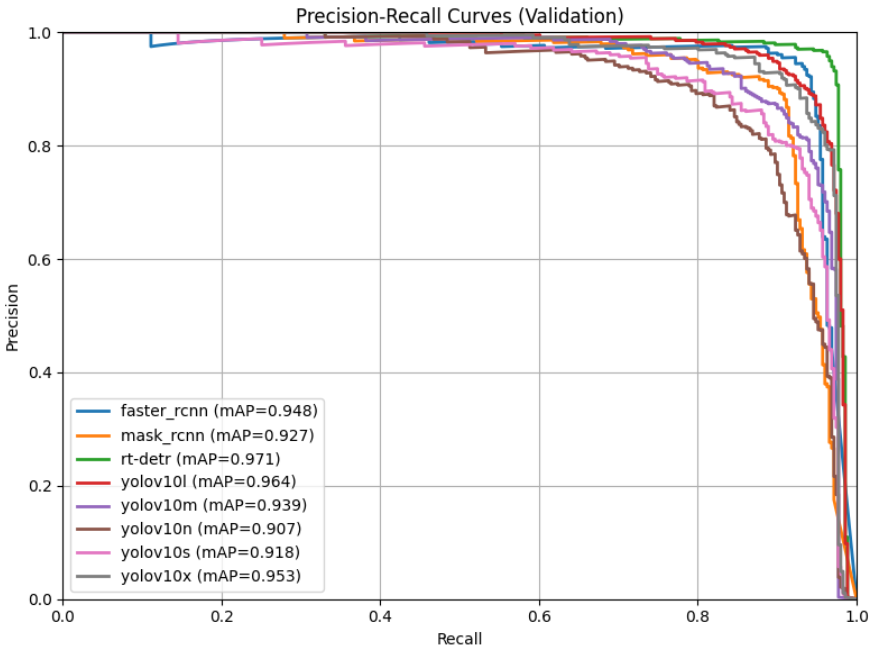

Figure 3: Examples of the annotated object detection images in our dataset.

Figure 3 shows the Precision-Recall (PR) curves for the validation set, using an IoU with a threshold of 0.5 for bounding box matching. In the figure, each curve plots precision (vertical axis) against recall (horizontal axis) at varying confidence thresholds, with the area under each curve corresponding to the mean Average Precision (mAP). Here, RT-DETR achieves the highest overall curve, aligning with its top mAP of 0.971, followed by YOLOv10l (0.964 mAP), which demonstrates the second-best profile. The other YOLO variants (x, m, s, n) maintain strong precision-recall performance but fall slightly behind the top two. Meanwhile, the two-stage detectors (Faster R-CNN and Mask R-CNN) also show relatively high precision until recall approaches its upper limit, though they rank below the best YOLO and RT-DETR results.

Figure 4: Examples of model predictions on test samples.

For error analysis, the YOLOv10l algorithm was selected as it has good balance between accuracy (mAP 0.964) and inference speed (40.8 FPS), making it one of the best options for deployment on resource-constrained devices such as microcomputers.

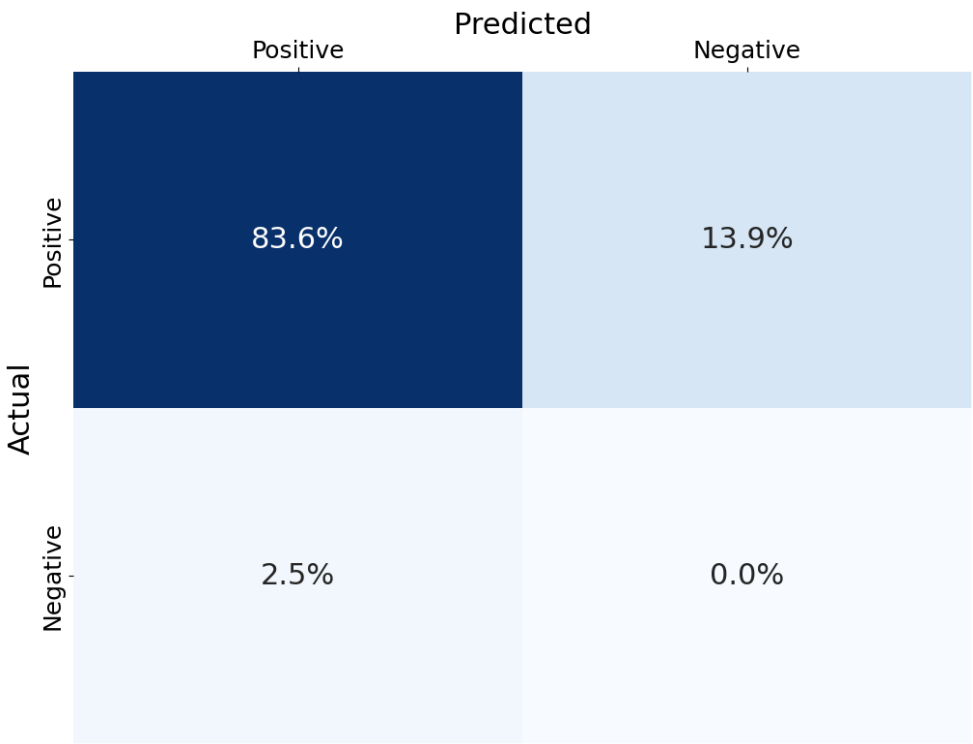

Figure 5: Confusion Matrix for the Validation Dataset.

The confusion matrix for YOLOv10l shows that the algorithm correctly identified drones in 83.6% of cases. Also, it shows two types of errors:

- **False Negative (13.9%)** — cases where the drone was present but not detected. These errors typically arise due to small object sizes, poor visibility conditions (e.g., fog, low lighting), or occlusion by other objects (trees, buildings). To reduce FN errors, it is recommended to increase the dataset size with challenging examples and apply additional augmentation techniques.
- **False Positive (2.5%)** — incorrect detection of drones in images without them. These errors are mainly caused by complex backgrounds and objects resembling drones in shape or size (e.g., birds, antennas, wires). Reducing FP errors can be achieved by adding more negative examples and employing "hard-negative mining".

5. Conclusions

In this study, we aimed to compare top-performing object detection methods for UAV identification and to assess both their accuracy and computational requirements. Our experiments indicated that RT-DETR and YOLOv10l achieved the highest precision on the test dataset (mAP@50 of 0.971 and 0.964, respectively). Nevertheless, smaller YOLO variants proved notably faster in inference while retaining competitive accuracy, suggesting that YOLO-based models strike a good balance for real-time applications on low-power hardware. Interestingly, the YOLOv10l configuration underperformed the YOLOv10x one, possibly due to complexities in training or hyperparameter tuning.

We created our own small-target, ground-to-sky dataset that closely matches real-world scenes and used it to run the first side-by-side test of several modern detectors, including the new transformer-based RT-DETR. The results show which model offers the best mix of accuracy and speed, giving clear guidance on which detector to choose for real-time UAV monitoring on low-power devices.

These findings lay the groundwork for deploying object detection algorithms in drone-related software, particularly for autonomous systems running on resource-constrained edge devices. However, not all of the tested models are able for real-time usage on such devices: RT-DETR, despite its outstanding accuracy, demands substantial computational resources, whereas YOLO's lightweight versions maintain practical throughput and can be readily adopted in edge computing environment.

Future work could involve combining the selected detection approach with tracking modules or incorporating additional modalities (e.g., thermal imaging, acoustic signals) to increase robustness in challenging scenarios such as night operations or heavy background clutter. Moreover, expanding the dataset with more diverse and numerous drone samples would further improve generalization.

Overall, the results show the potential of use either high-accuracy or lightweight CNN architectures—depending on the hardware constraints and real-time requirements—to achieve reliable drone detection. The insights and dataset from this study can help future research on UAV recognition, leading to better and more advanced drone detection and security systems.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] S. A. Musa et al., a review of copter drone detection using radar system, 2019. URL: https://www.researchgate.net/publication/331920623_A_REVIEW_OF_COPTER_DRONE_DETECTION_USING_RADAR_SYSTEM.

[2] A. Coluccia, G. Parisi, A. Fascista, Detection and classification of multirotor drones in radar sensor networks: A review, Sensors 20 (2020) 4172. doi:10.3390/s20154172.

[3] P. Andraši, T. Radišić, M. Muštra, J. Ivošević, Night-time detection of UAVs using thermal infrared camera, Transportation Research Procedia 28 (2017) 183–190. doi:10.1016/j.trpro.2017.12.184.

[4] I. Alla, H. B. Olou, V. Loscri, M. Levorato, From sound to sight: audio-visual fusion and deep learning for drone detection, in: Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM, New York, NY, USA, 2024, pp. 123–133. doi:10.1145/3643833.3656133.
[5] S. Al-Emadi, A. Al-Ali, A. Mohammad, A. Al-Ali, Audio based drone detection and identification using deep learning, in: 2019 15th International Wireless Communications & Mobile Computing Conference, IEEE, Tangier, Morocco, 2019, pp. 459–464. doi:10.1109/IWCMC.2019.8766732.
[6] Z. Cao, L. Kooistra, W. Wang, L. Guo, J. Valente, Real-time object detection based on UAV remote sensing: a systematic literature review, Drones 7 (2023) 620. doi:10.3390/drones7100620.
[7] J. Zhao, J. Zhang, D. Li, D. Wang, Vision-based anti-UAV detection and tracking, 2022. arXiv:2205.10851. doi:10.48550/arXiv.2205.10851.
[8] A. Barisic, M. Car, S. Bogdan, Vision-based system for a real-time detection and following of UAV, in: 2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS), IEEE, 2019, pp. 156–159. doi:10.1109/REDUAS47371.2019.8999675.
[9] R. Opromolla, G. Fasano, D. Accardo, A vision-based approach to UAV detection and tracking in cooperative applications, Sensors 18 (2018) 3391. doi:10.3390/s18103391.
[10] J. Li, D. H. Ye, M. Kolsch, J. P. Wachs, C. A. Bouman, Fast and robust UAV to UAV detection and tracking from video, IEEE Trans. Emerg. Top. Comput. 10 (2022) 1519–1531. doi:10.1109/TETC.2021.3104555.
[11] L. Mejias, S. McNamara, J. Lai, J. Ford, Vision-based detection and tracking of aerial targets for UAV collision avoidance, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 87–92. doi:10.1109/IROS.2010.5651028.
[12] X. Zhai, Z. Huang, T. Li, H. Liu, S. Wang, YOLO-Drone: an optimized YOLOv8 network for tiny UAV object detection, Electronics 12 (2023) 3664. doi:10.3390/electronics12173664.
[13] A. Coluccia, A. Fascista, L. Sommer, A. Schumann, A. Dimou, D. Zarpalas, The drone-vs-bird detection grand challenge at ICASSP 2023: a review of methods and results, IEEE Open J. Signal Process. 5 (2024) 766–779. doi:10.1109/OJSP.2024.3379073.
[14] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, 2015. URL: https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html.
[15] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, arXiv preprint arXiv:1703.06870, 2018. doi:10.48550/arXiv.1703.06870.
[16] A. Wang, et al., YOLOv10: real-time end-to-end object detection, in: Advances in Neural Information Processing Systems, volume 37, Curran Associates, Inc., 2024, pp. 107984–108011. URL: https://proceedings.neurips.cc/paper_files/paper/2024/hash/c34ddd05eb089991f06f3c5dc36836e0-Abstract-Conference.html.
[17] S. Wang, C. Xia, F. Lv, Y. Shi, RT-DETRv3: real-time end-to-end object detection with hierarchical dense positive supervision, arXiv preprint arXiv:2409.08475, 2024. doi:10.48550/arXiv.2409.08475.
[18] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, P. Dollár, Microsoft coco: common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, volume 8693 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, 2014, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.
[19] R. Girshick, Fast R-CNN, arXiv preprint arXiv:1504.08083, September 27, 2015. doi:10.48550/arXiv.1504.08083.
[20] K. Merkulova, Y. Zhabska, Input Data Requirements for Person Identification Information Technology, in: Proceedings of the 1st International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2023), volume 3468 of CEUR Workshop Proceedings, 2023, pp. 24–37. URL: https://ceur-ws.org/Vol-3468/paper3.pdf.