Improving Small Drone Detection Through Multi-Scale Processing and Data Augmentation

Rayson Laroca*,†, Marcelo dos Santos†, and David Menotti†

*Pontifical Catholic University of Paraná, Curitiba, Brazil
†Federal University of Paraná, Curitiba, Brazil

*rayson@ppgia.pucpr.br   †{msantos, menotti}@inf.ufpr.br

Abstract–**Detecting small drones, often indistinguishable from birds, is crucial for modern surveillance. This work introduces a drone detection methodology built upon the medium-sized YOLOv11 object detection model. To enhance its performance on small targets, we implemented a multi-scale approach in which the input image is processed both as a whole and in segmented parts, with subsequent prediction aggregation. We also utilized a copy-paste data augmentation technique to enrich the training dataset with diverse drone and bird examples. Finally, we implemented a post-processing technique that leverages frame-to-frame consistency to mitigate missed detections. The proposed approach attained a top-3 ranking in the 8th WOSDETC Drone-vs-Bird Detection Grand Challenge, held at the 2025 International Joint Conference on Neural Networks (IJCNN), showcasing its capability to detect drones in complex environments effectively.**

Fig. 1. The similarity in size and appearance often makes it challenging to differentiate a drone (blue box) from birds (red box), especially at a distance.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have experienced a surge in popularity across diverse civil sectors [1], [2]. Their autonomy, flexibility, and affordability have driven their widespread adoption in applications such as search and rescue, package delivery, and remote sensing [3]. However, this rapid expansion presents significant security and privacy challenges [4], [5].

The versatility of drones also makes them susceptible to malicious uses, such as smuggling contraband, conducting intrusive surveillance, and executing physical attacks [6], [7]. Additionally, unauthorized drone operations can violate aviation safety regulations, posing direct threats to civilian aircraft and passengers while causing disruptions at airports, including flight delays [8], [9]. As a result, there is an escalating and urgent demand for advanced drone detection systems to address these growing security and privacy concerns [10], [11].

A major challenge in drone detection is accurately distinguishing drones from birds, given their similarities in size and appearance (see Fig. 1). Moreover, practical systems must be capable of identifying drones at long distances to allow for timely responses. This requires detecting very small objects and differentiating them from structured backgrounds and other complex visual elements. To spur innovation in this area, the International Workshop on Small-Drone Surveillance, Detection and Counteraction Techniques (WOSDETC) consortium has launched the 8th WOSDETC Drone-vs-Bird Detection Grand Challenge at the 2025 International Joint Conference on Neural Networks (IJCNN) [12]. The challenge

seeks to attract research focused on developing novel signal-processing solutions for the problem of distinguishing birds from drones at long distances. To support participants, the consortium provides a video dataset, which is inherently difficult to acquire given the specific conditions and permissions required for drone flights. The dataset is progressively expanded with each challenge installment [13], [14] and subsequently released to the research community.

Considering the preceding discussion, this work presents a drone detection methodology that leverages the state-of-the-art YOLOv11 object detection model [15]. To enhance the detection of small-scale drones, we implemented a multi-scale approach that processes the input image both as a whole and in segmented components (simulating a zoom effect). Additionally, we employed extensive data augmentation, including a copy-paste technique, to increase the representation of drones and birds in the training images. A post-processing stage, utilizing adjacent frame predictions, was also employed to reduce missed detections. Our approach achieved a top-3 ranking (exact position not disclosed at the time of writing) in the 8th WOSDETC Drone-vs-Bird Detection Grand Challenge [12].

The rest of this paper is organized as follows: Section II briefly reviews related work. Section III describes the experiments conducted, including the datasets explored, the proposed method, and the experimental results. Finally, Section IV concludes the paper and suggests directions for future research.

II. RELATED WORK

To contextualize our work within the emerging yet rapidly growing field of drone detection, this section focuses on reviewing the works that introduced the datasets employed in our experiments, as well as the top-performing methods from the previous installment of the Drone-vs-Bird challenge [14].

Bosquet et al. [16] highlighted the lack of networks specifically designed for detecting small targets, defining small objects as those smaller than 16×16 pixels. They then introduced STDnet, a region-proposal-based network that incorporates a visual attention mechanism to select the most promising regions in the feature map while disregarding less relevant areas. Evaluated on their proposed USC-GRAD-STDdb dataset, STDnet demonstrated superior performance compared to Faster R-CNN [17] and FPN [18].

Svanström et al. [19] explored sensor fusion for drone detection, an area with limited research at the time. They developed a multi-sensor system that employed then state-of-the-art models, such as YOLOv2 [20], and combined class outputs and confidence scores from multiple sensors, including infrared cameras, visible cameras, and audio. Their analysis of camera data identified insects and clouds as the primary sources of false positives. Remarkably, implementing a two-sensor confirmation protocol significantly reduced these errors.

Zhao et al. [21] advanced anti-UAV research by evaluating state-of-the-art object detectors and trackers on their proposed DUT Anti-UAV dataset, which comprises 10,000 images and 20 sequences captured in diverse scenarios. They explored combinations of 14 detectors, including Faster-RCNN [17] and YOLOX [22], along with 8 trackers, such as SiamFC [23] and TransT [24]. Additionally, they proposed a fusion strategy that improves tracking robustness by integrating detection. When tracker confidence falls below a threshold, the detector refines the bounding box, ensuring more accurate tracking.

OBSS AI (OBSS Teknoloji, Ankara, Turkey), the winning team of the previous Drone-vs-Bird challenge [14], proposed a drone detection framework comprising an object detector, a sequence classifier, and a template matching module. They employed YOLOv5m6 [25] for object detection, followed by an object tracker to generate trajectories for detected objects. The sequence classifier then evaluated eight instances from each track to estimate the probability of a drone. To address missed detections in complex scenes, the framework employed template matching around the last known location within a small search region, producing bounding boxes that were subsequently evaluated by the sequence classifier. Although their approach was not formally published, it was heavily inspired by the team's earlier work [26].

Mistry et al. [27], the runner-up team in the previous installment of the Drone-vs-Bird challenge [14], developed a drone detection algorithm featuring three key stages. First, YOLOv7 [28] performed initial detection. Second, false positives were reduced using a heuristic based on confidence score thresholds. Specifically, the algorithm estimates the number of drones, n, in each sequence and retains only the n bounding boxes with the highest confidence scores in each frame. Finally, a CSRT tracker [29] was employed to reduce missed detections in complex environments.

Kim et al. [30], who placed third in the previous Drone-vs-Bird challenge [14], chose the YOLOv8 model [31] for drone detection. Recognizing the challenge of detecting small drones, they enhanced YOLOv8 by incorporating the P2 layer from its backbone into the feature pyramid, thereby leveraging its detailed spatial information. They further improved inference performance by scaling input images to 1280 pixels and employing extensive data augmentation.

We developed our drone detection method by combining key analytical insights from prior research. Specifically, we employ a state-of-the-art iteration of the YOLO family, given its strong track record. Our approach includes multi-scale processing and extensive data augmentation techniques to enhance model generalization and robustness, particularly in detecting small-scale drones. Lastly, we integrate temporal information to reduce missed detections and improve overall performance.

III. EXPERIMENTS

This section describes the experiments conducted for this work. Section III-A provides details on the datasets used for training and validation. Section III-B elaborates on the design and implementation of the proposed approach. Lastly, Section III-C presents the experimental results.

All experiments were performed on a PC equipped with an AMD Ryzen Threadripper 1920X CPU (3.5GHz), 64 GB of RAM, and an NVIDIA Quadro RTX 8000 GPU (48 GB).

A. Datasets

The competition organizers provided the Drone Data Set (DDS), consisting of 77 annotated video sequences to support method development. These videos, averaging 1,384 frames each, feature one or more drones, with an average of 1.12 annotated drones per frame. Annotations include the frame number and bounding box for each drone. Although birds frequently appear in the footage, they were left unlabeled. The dataset exhibits high variability in resolution, ranging from 720×576 to 3840×2160 pixels. It also presents sequences with backgrounds of sky or vegetation, diverse weather conditions (cloudy, sunny), direct sun glare, and varying camera characteristics, as shown in Fig. 2.

Given the absence of predefined guidelines for splitting the DDS dataset into training and validation subsets, a pragmatic, empirically driven approach was adopted. We initially allocated 70 videos for training and 7 for validation, with a deliberate focus on constructing a relatively diverse validation set. Specifically, we manually curated the dataset to ensure the validation set included videos with diverse drone distances, image resolutions, backgrounds, and environmental conditions. Recognizing the potential biases introduced by the initial split, we implemented an iterative refinement process [32]. This involved strategically swapping videos between the training and validation sets, guided by empirical observations of model performance while maintaining the intended diversity of the

Fig. 2. Representative images from the DDS dataset. Each image contains at least one drone, but bounding boxes were intentionally omitted to emphasize the challenge of visually identifying them in certain scenarios.

validation set. This methodology, grounded in careful observation and iterative adjustment, aimed to enhance the robustness and generalizability of the trained model. For reproducibility, the final composition of the validation set is as follows:

- `2019_10_16_C0003_3633_inspire.MP4;`
- `dji_mavick_mountain.avi;`
- `dji_phantom_4_hillside_cross.avi;`
- `GOPR5843_002.mp4;`
- `gopro_002.mp4;`
- `parrot_disco_distant_cross_3.avi;`
- `swarm_dji_phantom4_2.avi.`

During our analysis, we uncovered annotation errors within the dataset. The iterative refinement process of the training and validation splits revealed that some poor results stemmed from annotation inaccuracies rather than detection failures. While sporadic errors appeared across several videos, suggesting a semi-automated annotation process, two specific videos, `2019_10_16_C0003_3633_inspire.MP4` and `two_distant_phantom.avi`, exhibited significant inconsistencies. We hypothesize this resulted from desynchronization between video frames and drone position labels, potentially caused by uncorrected frame deletions at the beginning of the videos. These errors have since been corrected, and the updated annotations are now publicly available[1].

We employed OpenCV to extract individual frames from the videos, saving them as JPG images with a quality setting of 90. Although the PNG format would have eliminated compression artifacts, it would have substantially increased the dataset size. To further optimize processing time and storage, we implemented a frame subsampling strategy, selecting every fifth frame, following [30]. This subsampling process, while potentially introducing a reduction in performance, considerably decreased processing time, thereby enabling a greater number of experimental iterations and a more thorough validation of the annotations, as previously discussed.

To improve the robustness of our model across a wider range of scenarios, we incorporated three additional datasets

[1] https://raysonlaroca.github.io/supp/drone-vs-bird/

into our training data, as permitted by the competition rules. These datasets are the *Small Target Detection database* (USC-GRAD-STDdb) [16], *Dataset2* [19] (which lacks a formal name in its original publication, hence our designation), and *DUT Anti-UAV* [21]. A brief description of these datasets, along with representative images, can be found in the subsequent paragraphs and Fig. 3, respectively. Note that while these datasets include their own subdivisions for training, testing, and validation, we utilized all available images from them for training our model. As a result, the validation images consisted exclusively of scenarios from the target competition.
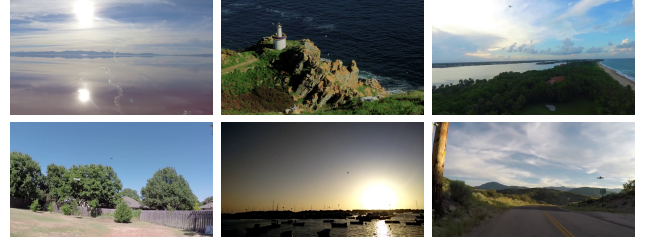
(a) USC-GRAD-STDdb [16]

(b) Dataset2 [19]
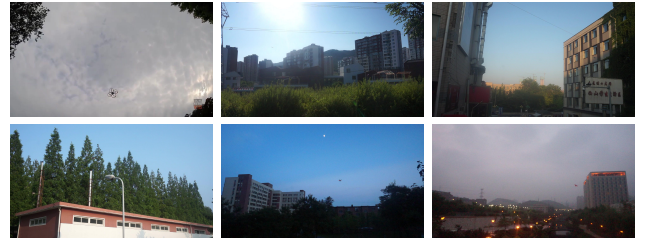
(c) DUT Anti-UAV [21]

Fig. 3. Samples from the three public datasets integrated into our training set.

USC-GRAD-STDdb [16] comprises 115 video segments collected from YouTube, totaling over 25,000 annotated frames. The videos, captured primarily by drones or from a bird's-eye perspective, have a resolution of approximately 1280×720 pixels and cover three main landscapes: air, sea, and land. The dataset is characterized by small object instances, with pixel areas ranging from 16 ($\approx 4 \times 4$) to 256 ($\approx 16 \times 16$) pixels. Considering our target application, we explored only the 2,263 frames that feature drones or birds.

Dataset2 contains 365 infrared and 285 visible ten-second videos, totaling 203,328 frames. We selected 51 visible videos depicting birds and 114 visible videos featuring drones, excluding infrared videos as well as visible videos depicting

other objects, such as airplanes and helicopters. All selected videos have a resolution of 640×512 pixels. A major challenge encountered during the processing of Dataset2 was the provision of annotations in MATLAB format (.mat), which we could not access due to licensing restrictions. Efforts to use alternative tools, such as Octave, were unsuccessful and mirrored issues reported by other users on GitHub. As a result, we resorted to manually selecting and labeling every tenth frame from the selected videos, resulting in 4,516 frames. These annotations have also been made publicly available.

The DUT Anti-UAV [21] dataset comprises 10,000 images with 10,109 manually annotated drone positions. It is distinguished by its high variability, encompassing resolutions from 240×160 to 5616×3744 pixels, over 35 drone models, and a wide array of backgrounds (sky, clouds, jungles, urban landscapes, farmland, and playgrounds). Furthermore, it accounts for various lighting conditions (day, night, dawn, and dusk) and weather scenarios (sunny, cloudy, and snowy days). The drones typically occupy less than 5% of the image area, though there are cases where the drone nearly covers the entire image.

While we considered several other datasets from the literature, we ultimately decided not to use them for various reasons. These included major differences between the datasets and our target scenarios, as well as challenges in accessing the datasets, such as non-functional websites or unresponsive authors regarding the download links, among other factors.

B. Proposed Approach

Given the proven track record of YOLO models not only in drone detection [10], [11], [19], [30] but also across a broad spectrum of computer vision applications [33]–[36], we selected this architecture for our investigation. Specifically, we leveraged YOLOv11 [15], a state-of-the-art iteration that builds on the strengths of its predecessors while incorporating enhancements to further improve performance and flexibility.

This model employs a single-stage, anchor-free architecture optimized for efficiency and ease of deployment. Its key components include: (i) an enhanced backbone that reduces computational costs while maintaining strong feature extraction capabilities; (ii) a refined neck structure that improves feature aggregation with fewer parameters, enabling faster inference; and (iii) an optimized detection head, designed to reduce latency and enhance speed in the final prediction layers. For more details on YOLOv11, please refer to [15].

YOLOv11 comes in five variants: YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x. These models vary in size and complexity, directly influencing their speed and accuracy trade-offs. YOLO11n (nano) is the smallest and fastest, making it ideal for resource-constrained environments, though with reduced accuracy. As the models scale up – YOLO11s (small), YOLO11m (medium), YOLO11l (large), and YOLO11x (extra-large) – they become progressively deeper and wider, enhancing accuracy at the cost of increased computational demands and slower inference speeds. Ultimately, selecting a variant involves balancing detection performance and speed to meet specific application requirements.

We started by conducting experiments using the YOLO11m model as a baseline. Nevertheless, we quickly encountered a major challenge: detecting distant drones. Specifically, when processing high-resolution images, such as 4K or even Quad HD, and resizing them to the model's 640-pixel input, distant objects became very small, often approaching or falling below the detection threshold.

Taking this into account, we proceed to investigate several alternative models, including but not limited to the variants previously listed in this section, each offering unique configurations and capabilities. YOLO11m-p2, for example, is a specialized variant with a finer stride configuration optimized for small object detection; however, this comes at the cost of increased computational overhead. We also experimented with doubling the models' input size (1,280 instead of 640) to capture finer details. While such variations did lead to some improvements in detection performance, the training times became prohibitively long, and we concluded that the [relatively small] gains did not justify the added computational cost.

Recognizing the need to improve detection performance, we decided to process the input images by dividing them into four segments. Each segment covers 55% of the original image's width and height, resulting in a small overlap between them. For example, a 1920×1080 image is split into four segments of 1056×594 pixels (contrasting with non-overlapping 960×540 segments). To handle drones occupying large portions of the image, the entire image is also processed. Finally, after processing both the segments and the full image, non-maximum suppresion is applied to eliminate redundant detections. This strategy, illustrated in Fig. 4 (left), yields substantial detection performance gains, as detailed in the next section, thus justifying the increased processing overhead.

Based on the preceding analysis, the YOLO11m variant was selected over its larger counterparts. We opted for fine-tuning rather than training from scratch to leverage pre-trained weights and accelerate convergence. This process was conducted using the well-established Ultralytics framework [15], with the following hyperparameter configuration: Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 and a momentum of 0.9, a batch size of 16, an early stopping patience value of 15, and an Intersection over Union (IoU) threshold for NMS of 0.1.

One of the core challenges of the competition lies in accurately distinguishing drones from birds, hence its name. In this sense, we trained our model to detect two distinct classes: drone and bird. Although the DDS dataset contains images with birds, it does not provide corresponding labels. In contrast, the USC-GRAD-STDdb and Dataset2 datasets, which we incorporated into our training data, include annotations for both classes. During inference, however, we focused exclusively on drone detection, ignoring predictions for the bird class. We also experimented with training a model solely for drone detection (i.e., without the bird class), but this approach resulted in slightly worse results.

To further enhance performance, we extensively applied data augmentation. Table I details the specific parameters used
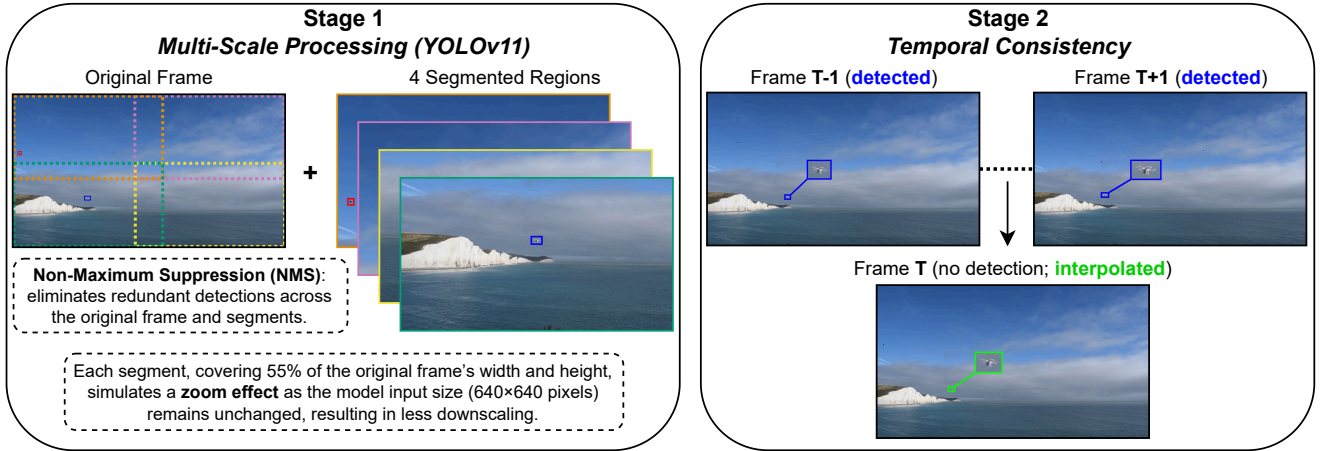
Fig. 4. Overview of the proposed approach. First, the medium-sized YOLOv11 model (YOLO11m) [15] is applied to both the full input frame and its segmented regions (simulating a zoom effect). Detections across the original frame and segments are then aggregated, with redundant bounding boxes removed via NMS. Lastly, temporal consistency and robustness to missing detections are achieved by tracking drones across a temporal window and applying linear interpolation.

for each transformation, adhering to the naming conventions of the Ultralytics framework. These transformations, standard practice in object detection, introduce variability into the training data, thereby improving model generalization. We conducted multiple iterations with varying parameter settings to determine the optimal values, evaluating their impact empirically. For instance, while the default values for scale and perspective transformations are 0.5 and 0, respectively, we found that 0.3 and 0.0001 yielded better results in our target scenario. For an explanation of each parameter and its specific effects, refer to the official Ultralytics documentation[2].

TABLE I
DATA AUGMENTATION PARAMETERS, FOLLOWING THE NAMING CONVENTIONS EMPLOYED IN THE ULTRALYTICS FRAMEWORK.

Parameter	Value	Parameter	Value
bgr	0.05	hsv_v	0.4
degrees	10	mosaic	0.1
erasing	0.15	perspective	0.0001
flipud	0.0	scale	0.3
fliplr	0.5	shear	10
hsv_s	0.7	translate	0.1

Beyond standard transformations, we applied a copy-paste technique to improve the training set with additional drone and bird instances. This approach involved randomly selecting patches of drones and birds and placing them in new, randomly chosen locations within the existing training images. This process is detailed in the following paragraph.

The images used for pasting were collected from both the training set and various online sources, all featuring transparent backgrounds to ensure seamless integration. To enhance variability, these patches were augmented using the Albumentations library [37], applying transformations such as blurring, pixel dropout, noise addition, and adjustments to brightness, contrast and gamma. Each patch instance was resized based on randomly selected scale factors within predefined ranges and

[2] https://docs.ultralytics.com/usage/cfg/

positioned realistically within the image. Overlap checks and color difference (ΔE) evaluations were performed to maintain visual consistency. Once a suitable insertion location was identified, the augmented instance was incorporated into the image, and its corresponding bounding box information was stored. Fig. 5 provides a visual representation of this process.

Fig. 5. Demonstration of the applied copy-paste data augmentation technique: original images (left) and their augmented counterparts (right), where patches of drones and birds were inserted to enhance variability in the training set.

To optimize detection performance, a confidence threshold of 0.375 was experimentally established. To mitigate missed detections, a post-processing stage was implemented. Specifically, this routine examines each frame's detected objects and searches for matching objects within a temporal window spanning six frames both preceding and following the target frame. The matching criteria require objects to share the same classification label and have a minimum degree of overlap (IoU ≥ 0.1). Upon identifying a potential match, the bounding box of the interpolated object is computed by linearly interpolating the bounding box coordinates between the matched objects. To ensure reliability, objects near image boundaries are discarded to prevent extrapolating detections for objects that have left the frame. Furthermore, the interpolated object is accepted only if it does not significantly overlap with existing detections in the target frame, avoiding

redundant predictions. The confidence score of the interpolated object is computed as the mean confidence of the matching objects, reduced by a factor of 2 to account for the uncertainty introduced by interpolation. As illustrated in Fig. 4 (right), this process helps bridge detection gaps, improving temporal consistency and overall accuracy, albeit with a minor increase in the number of false positives.

Another post-processing strategy we explored involved incorporating a classifier to refine the regions detected by YOLO11m, aiming to further reduce false positives. We experimented with various models, ranging from lightweight networks designed for specific tasks [38], [39] to well-established architectures such as MobileNetV3 [40]. To minimize the impact on inference time, we prioritized smaller models. However, regardless of the model used, performance declined. While this approach effectively filtered out many false positives, it also resulted in the loss of many true positives (in other words, several drones correctly detected by YOLO11m were erroneously discarded by the classifier).

We chose not to implement a maximum-drone-count heuristic, as utilized in [27], due to its limited real-world applicability. While such heuristics can be effective in competition environments with short, pre-defined videos, they struggle to adapt to real-time video streams' dynamic and unpredictable nature.

C. Results

We evaluated our approach on the validation set, which consists of seven videos from the Drone-vs-Bird challenge dataset, covering diverse conditions as described in Section III-A (the challenge organizers maintained exclusive access to the test set annotations). Performance was assessed using the mean Average Precision (mAP) at an IoU threshold of 50% (mAP_{50}). This metric calculates the average precision across recall values, considering a detection successful if the IoU between the predicted and ground truth bounding boxes is at least 50%.

The results, presented in Table II, show the mAP_{50} achieved for each video. We compare two approaches: our full proposed method, which employs a multi-scale input processing strategy analyzing both whole images and segmented components, and a simplified variant processing only whole images.

TABLE II
RESULTS OBTAINED FOR EACH VIDEO IN THE VALIDATION SET.

Video Name	mAP_{50}	$mAP_{50}^{\dagger}$
dji_mavick_mountain	0.9891	0.6431
2019_10_16_C0003_3633_inspire	0.9421	0.9219
parrot_disco_distant_cross_3	0.8684	0.5550
GOPR5843_002	0.7175	0.3371
swarm_dji_phantom4_2	0.7077	0.6566
dji_phantom_4_hillside_cross	0.4992	0.7406
gopro_002	0.4491	0.0121
Average	0.7390	0.5523

† using the simplified variant that processes only whole images.

Table II clearly shows that processing the input images both as a whole and in segmented components improves detection performance, particularly for smaller drones. However, the dji_phantom_4_hillside_cross video stands as an exception, where processing only whole images resulted in better outcomes. Upon further analysis, we determined that the reason for this was the video's complex hillside background, characterized by distant rocks and vegetation that closely mimic small-scale drones. When examining cropped bounding boxes from these regions, we occasionally mistook them for drones as well. In this particular video, the drone is white, providing a strong contrast against the background, and is relatively large. As a result, dividing the input image into segments was not strictly necessary. While incorporating segmented components in the processing still improved Recall, it resulted in a disproportionately larger decrease in Precision.

Examples of successfully detected drones are shown in Fig. 6. It is important to observe that the proposed approach is capable of detecting drones across various backgrounds, weather conditions, scales, among other factors. Of particular interest is the left image in the third row of Fig. 6 (also featured on the first page of this work), where none of the birds were erroneously identified as drones.
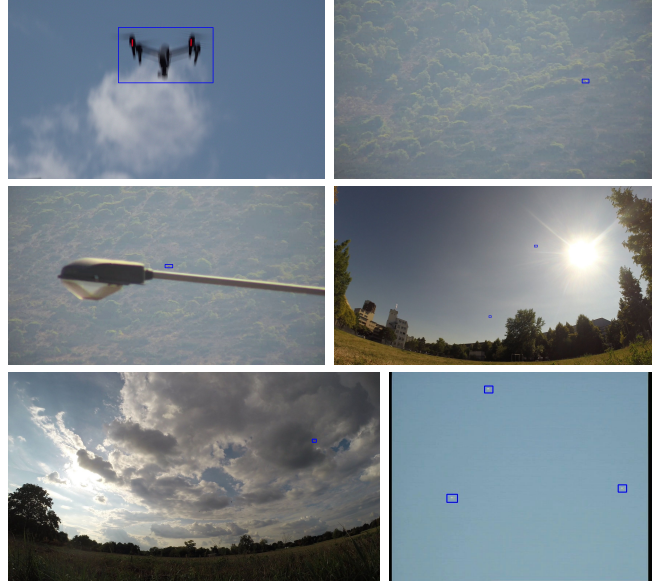

Fig. 6. Examples of drones successfully detected by the proposed approach. In this figure, we have intentionally refrained from providing zoomed-in views of distant, small drones to highlight the difficulty of detecting them.

IV. CONCLUSIONS

In this article, we describe our approach for drone detection, which ranked among the top 3 teams in the 8th WOSDETC Drone-vs-Bird Detection Grand Challenge [12].

We utilized the medium-sized YOLOv11 model [15] for drone detection. To overcome the model's struggles with small-scale drone detection, we implemented a multi-scale approach in which the input image is processed both as a whole and in segmented components. This strategy significantly boosted detection performance, especially for distant drones.

To further enhance performance, we employed extensive data augmentation. In addition to the standard transformations

applied during training, we utilized a copy-paste technique to increase the number of drone and bird instances in the training images. This involved randomly placing cropped and scaled instances into new locations, ensuring they did not overlap with existing instances. Finally, a post-processing stage was incorporated to mitigate missed detections.

A key performance bottleneck in our current approach is the computational overhead associated with the detector's repeated execution. To mitigate this, we propose three primary avenues for improvement: optimizing the detection pipeline to reduce detector calls, examining faster model alternatives, and investigating the use of parallel processing or hardware acceleration to handle multiple detections concurrently.

We aim to improve detection performance through the following investigations: (i) leveraging a state-of-the-art tracker to enhance multi-frame predictions; (ii) developing a classifier that analyzes multi-frame image patches to accurately differentiate drones from similar objects, such as birds, thereby reducing false positives [26]; and (iii) investigating latest architectures, such as D-FINE [41] and RT-DETRv2 [42], while incorporating a wider variety of datasets, including the recently introduced SynDroneVision [43], into the training process.

ACKNOWLEDGMENTS

REFERENCES

[1] Y. Liu, H.-N. Dai, Q. Wang, M. K. Shukla, and M. Imran, "Unmanned aerial vehicle for internet of everything: Opportunities and challenges," *Computer Communications*, vol. 155, pp. 66–83, 2020.

[2] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends," *Intelligent Service Robotics*, vol. 16, no. 1, pp. 109–137, 2023.

[3] J. Zhao *et al.*, "The 3rd anti-UAV workshop & challenge: Methods and results," *arXiv preprint*, vol. arXiv:2305.07290, pp. 1–12, 2023.

[4] M. Yahuza, M. Y. I. Idris, I. B. Ahmedy, A. W. A. Wahab, T. Nandy, N. M. Noor, and A. Bala, "Internet of drones security and privacy issues: Taxonomy and open challenges," *IEEE Access*, vol. 9, pp. 57 243–57 270, 2021.

[5] V. Chamola, P. Kotesh, A. Agarwal, Naren, N. Gupta, and M. Guizani, "A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques," *Ad Hoc Networks*, vol. 111, p. 102324, 2021.

[6] D. Kunertova, "The Ukraine drone effect on European militaries," *CSS Policy Perspectives*, vol. 10, no. 15, pp. 1–5, 2022.

[7] V. V. Ghaleb Krame and A. Davies, "Narco drones: tracing the evolution of cartel aerial tactics in Mexico's low-intensity conflicts," *Small Wars & Insurgencies*, vol. 34, no. 6, pp. 1095–1129, 2023.

[8] P. Wendt, A. Voltes-Dorta, and P. Suau-Sanchez, "Estimating the costs for the airport operator and airlines of a drone-related shutdown: an application to Frankfurt international airport," *Journal of Transportation Security*, vol. 13, no. 1, pp. 93–116, 2020.

[9] S. Park, H. T. Kim, S. Lee, H. Joo, and H. Kim, "Survey on anti-drone systems: Components, designs, and challenges," *IEEE Access*, vol. 9, pp. 42 635–42 659, 2021.

[10] D. Gao, J. Tang, H. Li, B. Wang, J. Qiu, S. Chen, and X. Mei, "A novel dataset and detection method for unmanned aerial vehicles using an improved YOLOv9 algorithm," *Sensors*, vol. 24, no. 23, 2024.

[11] A. Munir, A. J. Siddiqui, and S. Anwar, "Investigation of UAV detection in images with complex backgrounds and rainy artifacts," in *IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2024, pp. 232–241.

[12] A. Coluccia, A. Fascista, L. Sommer, A. Schumann, A. Dimou, D. Zarpalas, and E. Mele, "8th WOSDETC Drone-vs-Bird Detection Grand Challenge @ IJCNN25," 2025, accessed: 2025-03-08. [Online]. Available: https://wosdetc2025.wordpress.com/

[13] A. Coluccia, A. Fascista, A. Schumann, L. Sommer, A. Dimou, D. Zarpalas, F. C. Akyon, O. Eryuksel, K. A. Ozfuttu, S. O. Altinuc, F. Dadboud, V. Patel, V. Mehta, M. Bolic, and I. Mantegh, "Drone-vs-bird detection challenge at IEEE AVSS2021," in *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2021, pp. 1–8.

[14] A. Coluccia, A. Fascista, L. Sommer, A. Schumann, A. Dimou, and D. Zarpalas, "The drone-vs-bird detection grand challenge at ICASSP 2023: A review of methods and results," *IEEE Open Journal of Signal Processing*, vol. 5, pp. 766–779, 2024.

[15] Ultralytics, "YOLOv11," 2025, accessed: 2025-03-04. [Online]. Available: https://docs.ultralytics.com/models/yolo11/

[16] B. Bosquet, M. Mucientes, and V. M. Brea, "STDnet: A convnet for small target detection," in *British Machine Vision Conference (BMVC)*, Sept 2018, pp. 1–12.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[18] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 936–944.

[19] F. Svanström, C. Englund, and F. Alonso-Fernandez, "Real-time drone detection and tracking with visible, thermal and acoustic sensors," in *International Conference on Pattern Recognition (ICPR)*, 2021, pp. 7265–7272.

[20] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6517–6525.

[21] J. Zhao, J. Zhang, D. Li, and D. Wang, "Vision-based anti-UAV detection and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 323–25 334, 2022.

[22] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," *arXiv preprint*, vol. arXiv:2107.08430, pp. 1–7, 2021.

[23] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision Workshops (ECCVW)*, 2016, pp. 850–865.

[24] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8122–8131.

[25] Ultralytics, "YOLOv5," 2020, accessed: 2025-03-08. [Online]. Available: https://github.com/ultralytics/yolov5

[26] F. C. Akyon, E. Akagunduz, S. O. Altinuc, and A. Temizel, "Sequence models for drone vs bird classification," *arXiv preprint*, vol. arXiv:2207.10409, pp. 1–5, 2022.

[27] S. K. Mistry, S. Chatterjee, A. K. Verma, V. Jakhetiya, B. N. Subudhi, and S. Jaiswal, "Drone-vs-bird: Drone detection using YOLOv7 with CSRT tracker," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–2.

[28] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7475.

[29] A. Lukežič, T. Vojíř, L. Čehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter tracker with channel and spatial reliability," *International Journal of Computer Vision*, vol. 126, no. 7, pp. 671–688, 2018.

[30] J.-H. Kim, N. Kim, and C. S. Won, "High-speed drone detection based on Yolo-V8," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–2.

[31] Ultralytics, "YOLOv8," 2023, accessed: 2025-03-08. [Online]. Available: https://docs.ultralytics.com/models/yolov8/

[32] R. Laroca, V. Estevam, A. S. Britto Jr., R. Minetto, and D. Menotti, "Do we train on test data? The impact of near-duplicates on license plate recognition," in *International Joint Conference on Neural Networks (IJCNN)*, June 2023, pp. 1–8.

[33] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 483–503, 2021.

[34] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, 2023.

[35] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[36] R. Laroca, L. A. Zanlorensi, V. Estevam, R. Minetto, and D. Menotti, "Leveraging model fusion for improved license plate recognition," in *Iberoamerican Congress on Pattern Recognition*, Nov 2023, pp. 60–75.

[37] A. Buslaev *et al.*, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[38] R. Laroca, A. B. Araujo, L. A. Zanlorensi, E. C. De Almeida, and D. Menotti, "Towards image-based automatic meter reading in unconstrained scenarios: A robust and efficient approach," *IEEE Access*, vol. 9, pp. 67 569–67 584, 2021.

[39] R. Laroca, M. Santos, V. Estevam, E. Luz, and D. Menotti, "A first look at dataset bias in license plate recognition," in *Conference on Graphics, Patterns and Images (SIBGRAPI)*, Oct 2022, pp. 234–239.

[40] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324.

[41] Y. Peng, H. Li, P. Wu, Y. Zhang, X. Sun, and F. Wu, "D-FINE: Redefine regression task in DETRs as fine-grained distribution refinement," *arXiv preprint*, vol. arXiv:2410.13842, pp. 1–18, 2024.

[42] W. Lv, Y. Zhao, Q. Chang, K. Huang, G. Wang, and Y. Liu, "RT-DETRv2: Improved baseline with bag-of-freebies for real-time detection transformer," *arXiv preprint*, vol. arXiv:2407.17140, pp. 1–4, 2024.

[43] T. R. Lenhard, A. Weinmann, K. Franke, and T. Koch, "SynDroneVision: A synthetic dataset for image-based drone detection," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025, pp. 7637–7647.