Lightweight vs. Advanced Architectures: Performance Analysis of YOLOv8n and YOLOv9T in RF Spectrogram-Based UAV Detection *

Feten Slimeni[1], Tijeni Delleji[2] and Ahmed Siala[3]

Abstract— **The rapid adoption of Unmanned Aerial Vehicles (UAVs) has introduced critical security risks to sensitive infrastructures such as airports and power plants. This paper presents a real-time system for UAV detection and identification by leveraging radio frequency (RF) spectrogram analysis and deep learning. We propose a Software-Defined Radio (SDR)-based framework using the USRP X310 to capture RF signals from drone remote control (RC) link, WiFi, and Bluetooth, generating a custom dataset of spectrogram images. The You Only Look Once (YOLO) architectures, YOLOv8n and YOLOv9t, are trained and benchmarked for drone detection.**

A comprehensive evaluation reveals the following trade-offs: YOLOv8n achieves superior inference speed (1.6 ms vs. 3.0 ms), making it suitable for latency-sensitive applications, while YOLOv9t offers a lighter parameter footprint despite comparable computational complexity (7–8 GFLOPs). Both models exhibit similar accuracy, with mean Average Precision (mAP50-95) stabilizing at 0.8–0.85 and matching classification/box loss trajectories. However, YOLOv8n slightly outperforms in bounding box refinement (22% better DFL loss) and precision.

Our findings underscore YOLOv8n's dominance in real-time anti-drone systems, balancing speed and accuracy, whereas YOLOv9t's compact design suits resource-constrained environments. This work bridges RF signal processing and AI-driven threat mitigation, offering actionable insights for deploying scalable drone surveillance solutions.

I. INTRODUCTION

The rapid adoption of mini-UAVs has revolutionized industries ranging from precision agriculture to disaster management, offering unprecedented efficiency in tasks like crop monitoring, flood response, and logistics [1]–[3]. However, their proliferation also introduces significant security risks, including unauthorized surveillance, contraband smuggling, and threats to critical infrastructure such as airports [4], [5]. These dual-use challenges underscore the urgent need for robust, real-time detection systems capable of mitigating risks while preserving legitimate UAV applications.

Traditional drone detection methods—such as radar, acoustic sensors, and optical cameras—often struggle with limitations like high cost, environmental sensitivity, or line-of-sight requirements. In contrast, RF-based detection has gained prominence due to its ability to identify drones via their communication signals (e.g., remote control, WiFi, or Bluetooth) even in obstructed environments. Yet, existing RF approaches face two critical gaps:

- Computational inefficiency: Raw RF signal processing (e.g., I/Q sample analysis) requires resource-intensive transformations (e.g., STFT) for feature extraction, hindering real-time deployment.
- Limited adaptability: Statistical learning methods struggle with nonlinear signal variations (e.g., multipath interference, dynamic modulation), reducing accuracy in complex environments.

To address these challenges, this paper proposes a YOLO-based spectrogram analysis framework that bridges RF signal processing and real-time deep learning. Our work makes three key contributions:

- First comparative study of YOLOv8n and YOLOv9t for RF spectrogram-based drone detection, highlighting trade-offs between speed (1.6 ms inference for YOLOv8n) and lightweight design (YOLOv9t's parameter efficiency).
- A custom dataset of RF spectrograms capturing drone control links, WiFi, and Bluetooth signals, annotated for signal type/duration and optimized for YOLO architectures.
- Comprehensive benchmarks (mAP50-95: 0.8–0.85) demonstrating superior robustness to noise and multipath effects compared to raw RF processing, with actionable insights for deployable anti-drone systems.

The rest of the paper is organized as follows: Section 2 reviews related work in drone detection, RF signal analysis, and YOLO architectures. Section 3 details dataset construction, Section 4 presents the YOLO training methodology and discusses results, and Section 5 concludes with future directions.

II. RELATED WORKS

Mini-UAVs have become indispensable in precision agriculture, disaster management, and logistics, offering efficiency gains in crop monitoring, flood response, and last-mile delivery [1]–[3]. However, their proliferation raises critical security concerns, including airspace violations near airports, privacy breaches, and illicit activities [4], [5]. These risks necessitate reliable detection systems that balance security with legitimate UAV use.

RF signal analysis has emerged as a robust solution for drone identification, leveraging emitted control and commu-

*This work was supported by the Military Research Center

[1]Feten Slimeni is an Assistant Professor of Telecommunications Engineering at the Military Academy of Fondouk Djedid, 8012 Nabeul, Tunisia and a Researcher at the Science and Technology for Defense Laboratory of the Military Research Center (CRM), Aouina, 2045 Tunis, Tunisia feten.slimeni@gmail.com

[2]Tijeni Delleji is an Assistant Professor in Avionics Engineering at the Aviation School of Borj El Amri (Tunisia), he serves as Anti-Drone Project Manager and Head of the Research Valorization Unit at Tunisia's Military Research Center tijeni_dalleji@yahoo.fr

[3]Ahmed Siala, Professor in civil engineering, is the General Manager of the Military Research Center ahmed.siala.71@gmail.com

nication signals (e.g., RC, WiFi, Bluetooth) even in non-line-of-sight conditions. Early work by [6], [7] established RF fingerprinting for UAV classification, while [8], [9] demonstrated adaptive RF methodologies for active/passive detection. Multi-modal approaches combining RF with acoustic or visual sensors further improve accuracy [10], though at increased computational cost.

Convolutional Neural Networks (CNNs) have dominated vision-based UAV detection, with YOLO architectures excelling in real-time performance. Studies like [11] benchmarked YOLOv7 for drones, achieving sub-10ms latency, while [12] optimized YOLOv9 for tiny UAVs via enhanced feature extraction. However, environmental factors (e.g., weather, occlusion) remain challenges [13]. Prior work by our team addressed visual detection limitations through improved YOLOv5 models for no-fly zones [14] and low-visibility conditions [15].

Bridging RF and visual domains, recent efforts focus on spectrogram-based deep learning. Our earlier work introduced an MLP for RF fingerprinting [16] and RF-YOLO, a hybrid model treating spectrograms as visual inputs [17]. While these studies validated spectrograms' compatibility with CNNs, gaps persist in:

- Architecture benchmarking: No prior study compares YOLOv8n and YOLOv9t for RF spectrograms.
- Real-time trade-offs: Raw RF processing struggles with latency (e.g., STFT overhead), unlike end-to-end spectrogram models.
- Edge deployment: Lightweight YOLO variants (e.g., YOLOv9t) remain unexplored for resource-constrained SDR systems.

III. Dataset construction

We have constructed our customized dataset of spectrogram images related to the three types of RF signals: RC-drone, WiFi, and Bluetooth. The drone signals were generated by the remote control of the DJI Phantom 4 Pro, and the WiFi and Bluetooth signals were generated by types of equipment like AirPods, AirBox, and mobile phones. These signals were collected by the software-defined radio (SDR) platform USRP X310, which generates the related spectrograms. All of these signals operate within the 2.4GHz band, which can lead to false alarms classifying WiFi or Bluetooth as RC-drone signals.

The USRP X310 [18], standing for Universal Software Radio Peripheral X310, is a powerful and flexible hardware platform for capturing RF signals. We meticulously configured the USRP X310 to capture the signals, harnessing its substantial hardware capabilities for precise and exhaustive data acquisition. The USRP X310 was connected to a PC via dual 10 Gigabit Ethernet links, essential for supporting high-speed data transfers required for extensive RF signal capture. We relied exclusively on a custom Python script for software configuration, enabling direct control of the hardware and precise management of the data collection process. The script utilized the USRP Hardware Driver (UHD) API, providing direct access to configure and control the USRP devices.

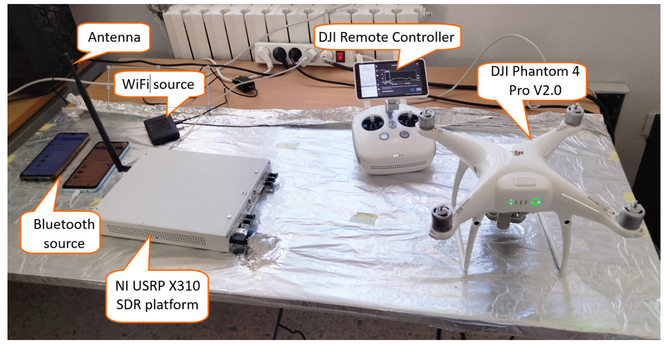

Fig. 1. Experimental setup for data collection

This approach allowed us to tailor the configuration to our needs, controlling parameters such as sampling rates, center frequency, and bandwidth settings. By bypassing additional software like GNU Radio, our streamlined, efficient data capture process was directly managed through our Python code, ensuring our system was optimized for capturing and processing RF signals. This setup provided a robust and flexible platform for our RF classification project. Fig. 1 provides the setup used to collect Bluetooth, WiFi, and RC-drone signals. The RF spectrogram is a visual representation of the frequency spectrum of signals over time. The dataset of these images was generated from raw I/Q samples using a reproducible Python pipeline based on Short-Time Fourier Transform (STFT). It displays how the power of different frequency components varies with time, helping to analyze and identify various signals in a given frequency range. The time axis represents the duration of the signal being analyzed. The frequency axis shows the frequency components, typically in Hertz (Hz). The power is represented by color intensity or brightness, indicating the strength of the signal at different frequencies over time.

The WiFi typically operates on channels ranging from 2.412 GHz to 2.472 GHz. The Bluetooth uses a frequency-hopping spread spectrum, primarily around 2.402 GHz to 2.480 GHz. The RC Drone operates in the same band but often has distinct modulation patterns and bandwidths. Therefore, each signal can be visualized as distinct peaks or clusters in the spectrogram, allowing for identification based on object detection. Fig. 2, 3, and 4 illustrate the spectrograms related to Bluetooth, WiFi, and RC-drone signals, respectively. The axis labels (frequency in GHz and time in ms) were intentionally omitted from the spectrograms to avoid perturbing the YOLO model's training and inference, as raw visual patterns were the primary input features. The spectrogram of a Bluetooth signal displays different small frequency bands over time, showing a spread across multiple channels due to the frequency hopping technique. Differently, the spectrogram of a WiFi signal displays a wider bandwidth and a different modulation pattern. It shows a dynamic frequency spectrum with varying amplitudes over time and a more complex pattern due to the higher data rates used for communication. Otherwise, the behavior of the RC-drone signal over time looks different from Bluetooth and WiFi

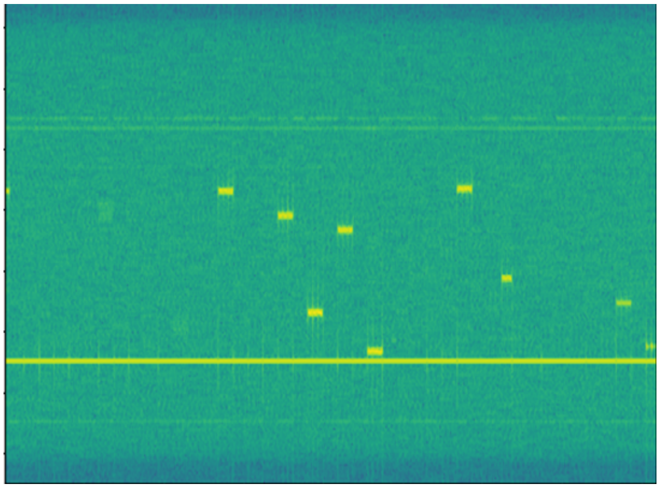

Fig. 2. The spectrogram of a Bluetooth signal

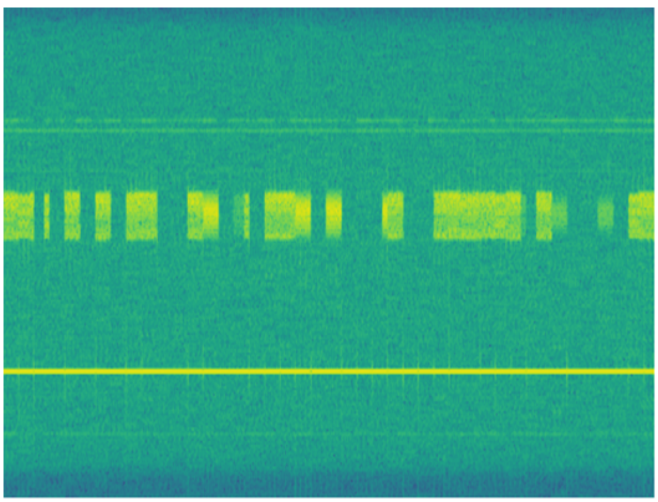

Fig. 3. The spectrogram of a WiFi signal

signals, it has a more continuous and stable frequency pattern due to the constant communication between the drone and the controller. The distinct characteristics of Bluetooth, WiFi, and RC-drone signals as reflected in their spectrograms can make them distinguishable using an object detection model trained on these spectrogram images. Before training the model, we annotated the spectrograms of WiFi, Bluetooth, and RC-drone signals using the LabelImg tool.

IV. YOLO MODELS TRAINING

The evolution of YOLO versions highlights important advances in real-time detection for various applications, including the detection of RF drones [13]. The dataset was divided into a training set of 1296 spectrograms and a test set of 325 spectrograms. We have trained and compared YOLOv9t and YOLOv8n models on the same dataset under identical conditions, considering a batch size of 16, a learning rate of 0.01, and 200 epochs.

The specs given in Table I provide a snapshot of the models' complexity, computational requirements, and performance. Param(M) stands for the number of parameters in

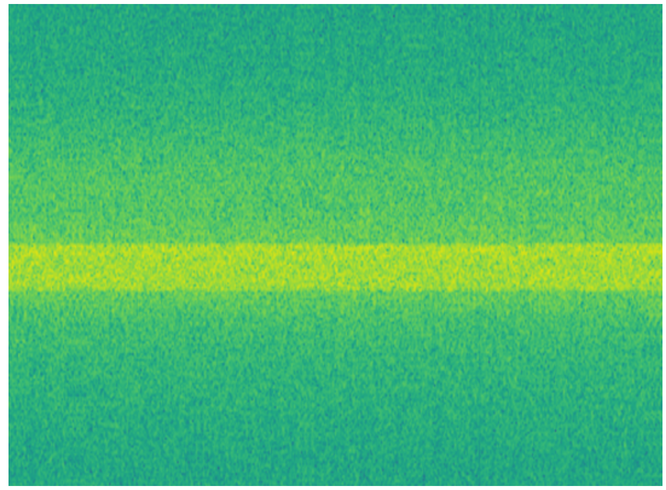

Fig. 4. The spectrogram of an RC-drone signal

millions. It indicates the size of the model in terms of the number of trainable parameters. FLOPS(G) stands for Floating Point Operations per Second, and the "G" means billions. This metric indicates the computational complexity of the model, i.e., how many floating-point operations are needed to make a prediction. Layers refer to the number of layers in the neural network. More layers allow learning more complex patterns, but it also means more computational requirements. Training Time (hours) is the amount of time needed to train the model on a given dataset. It provides an estimate of the computational cost and time required to achieve a trained model. Finally, the metric inference (ms) stands for the time it takes to make a single prediction (in milliseconds). The comparison between YOLOv9t and YOLOv8n models reveals significant differences in their architecture and performance metrics. YOLOv9t, with 1,765 million parameters and 6.7 GFLOPS (Giga Floating Point Operations per Second), comprises 941 layers. It has a training time of 1.157 hours and an inference speed of 3.0 milliseconds. In contrast, YOLOv8n boasts 3,011 million parameters and 8.2 GFLOPS, consisting of 225 layers. YOLOv8n's training time is quicker at 0.831 hours, demonstrating a faster inference speed of 1.6 milliseconds. The specs indicate that while YOLOv8n is more resource-intensive with more parameters and GFLOPS, it also offers faster training and inference times compared to YOLOv9t. This suggests that YOLOv8n might be better suited for applications requiring real-time performance, whereas YOLOv9t could be preferred for its efficiency in less resource-intensive environments.

TABLE I
SPECS OF YOLO MODELS

Model	Param(M)	FLOPS(G)	Layers	Train(h)	Inf(ms)
YOLOv9t	1,765	6.7	941	1.157	3.0
YOLOv8n	3.011	8.2	225	0.831	1.6

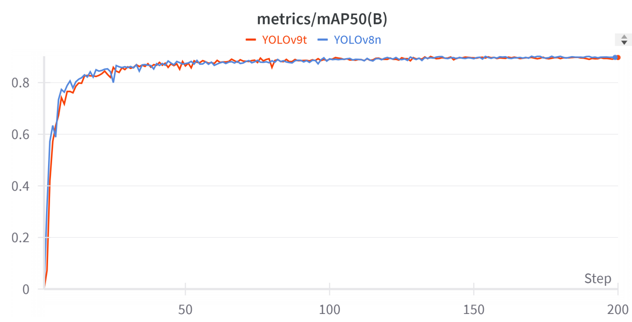

Fig. 5. mAP@50 curves

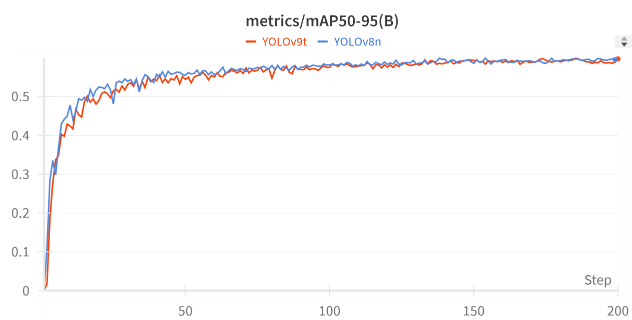

Fig. 6. mAP@50-95 curves

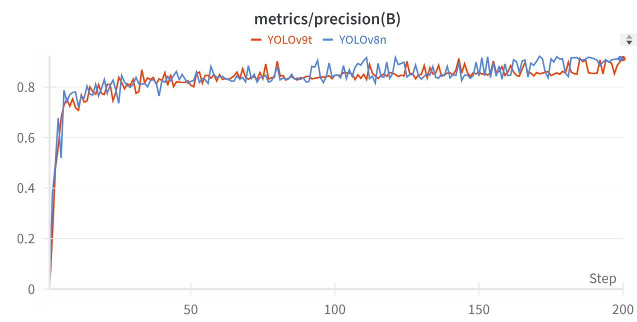

Fig. 7. The precision curves

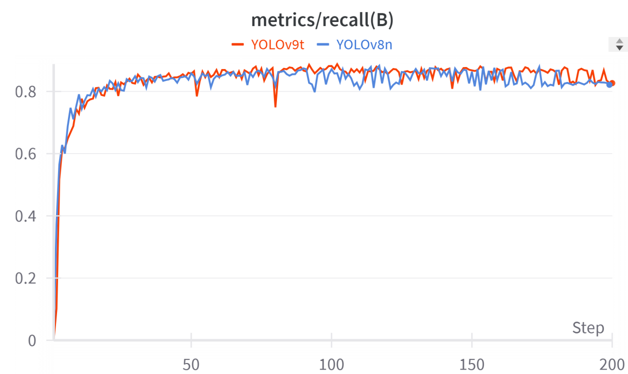

Fig. 8. The recall curves

V. Results and Discussion

We have considered the following evaluation metrics to evaluate and compare the implemented YOLO models. 1) The mAP@50 measures the average precision of the model at an intersection over a union (IoU) threshold of 0.50. The precision here is calculated by comparing the overlap between predicted bounding boxes and ground truth boxes. 2) The mean Average Precision mAP@50-95, which is a more comprehensive metric that averages the precision over multiple IoU thresholds, ranging from 0.50 to 0.95 in increments of 0.05. This gives a more detailed assessment of the model's performance across different levels of strictness. 3) The precision metric indicates the proportion of positive identifications that were correct. In simpler terms, it measures the accuracy of the positive predictions. 4) The recall represents the proportion of actual positives that were correctly identified by the model. It measures the model's ability to find all the relevant instances in the dataset.

Both YOLOv9t and YOLOv8n models achieve similar mAP performance, ranging from approximately 0.8 to 0.85, as given in Fig.6 and Fig.5. YOLOv8n shows a slight advantage over YOLOv9t in the early training steps, demonstrating higher initial accuracy. However, as the training progresses, both models converge to similar levels of accuracy, indicating no clear advantage of YOLOv9t over YOLOv8n. While both models exhibit comparable overall accuracy, YOLOv8n is preferable for UAV detection requiring high-speed inference.

Based on the comparison of recall given in Fig.8 and precision given in Fig.7, we can state that both models achieve a recall in the range of approximately 0.8 to 0.85, indicating that they detect most objects with similar efficacy. Precision, while more variable, stabilizes around 0.8 for both models, showcasing their ability to correctly identify objects. There is no significant difference observed between YOLOv9t and YOLOv8n in terms of these metrics. Ultimately, both models exhibit similar accuracy and recall in object detection tasks, but YOLOv8n remains the preferred option due to its faster inference speed.

We have also compared the two models in terms of loss. The box_loss measures the error in predicting the coordinates of the bounding boxes around objects. A lower value is better because it means the predicted boxes are closer to the ground truth. The box loss analysis, Fig.9, reveals that both YOLOv9t and YOLOv8n models start with a high loss of approximately 5-6 and converge quickly, with the loss stabilizing around 0.2-0.3 after roughly 50 steps. This similarity in the loss curves indicates that both models learn object bounding box localization equally well, with no significant advantage in terms of box regression accuracy.

The dfl_loss (distribution focal loss) is a specific loss function used in object detection models to improve localization accuracy.

The cls_loss (classification loss) measures the error in predicting the class of the detected objects. A lower value is better because the predicted classes are more accurate. In the classification loss analysis, Fig.10, both models exhibit a

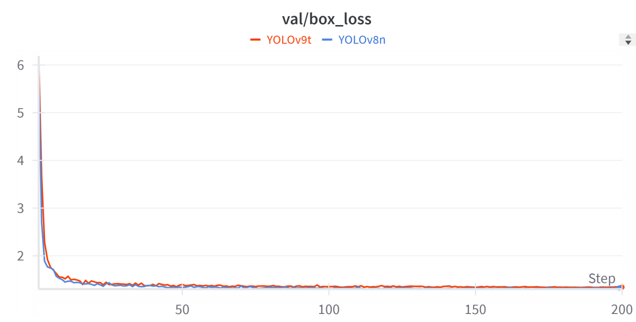

Fig. 9. The box-loss curves

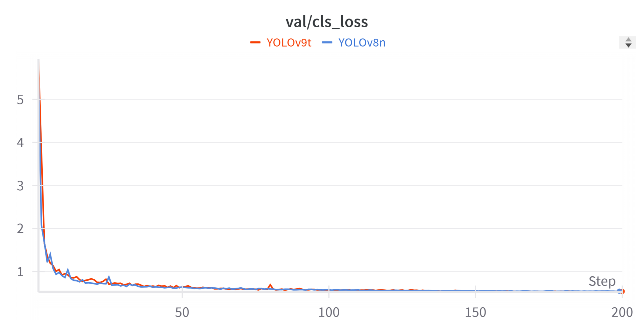

Fig. 10. The cls-loss curves

similar loss trajectory, with a significant drop in loss within the first 50 steps, followed by stabilization. This suggests that both YOLOv9t and YOLOv8n classify objects with similar efficiency, without any major difference in classification loss between the two models.

Lastly, Fig.11 shows that DFL loss, which measures the precision of predicted bounding boxes, is slightly lower for YOLOv8n compared to YOLOv9t. This indicates that YOLOv8n refines bounding box predictions slightly better than YOLOv9t. Overall, both models perform similarly in terms of box regression and object classification, but YOLOv8n slightly outperforms YOLOv9t in precise bounding box predictions. Therefore, if high-precision object detection is important, YOLOv8n is an appropriate choice.

Our current study focused on validating core detection

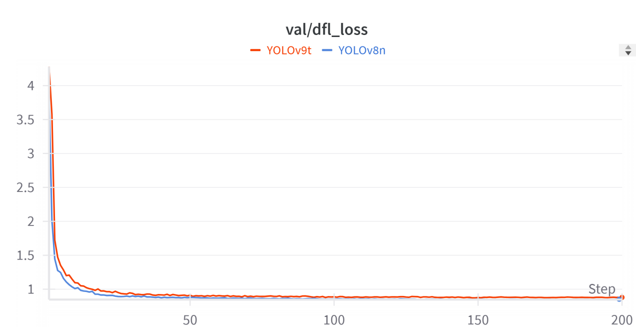

Fig. 11. The dfl-loss curves

capabilities in controlled single-UAV scenarios, and we have not yet conducted formal testing with multiple simultaneous UAVs or in complex electromagnetic environments. However, based on the architectural properties of YOLOv8n/YOLOv9t, particularly their proven multi-object detection capabilities in computer vision domains, our system would maintain proportional performance degradation when scaled to moderate UAV densities (3-5 drones), as the spectrogram representation naturally preserves spatial relationships between distinct RF emitters. Regarding environmental complexity, while our current dataset includes basic WiFi/Bluetooth interference, we agree that urban multipath and adversarial jamming scenarios require dedicated investigation. These scalability questions shape our future work, and we are currently working on drone swarm RF detection, considering a hybrid vision-RF testbed for complex environment validation.

VI. Conclusion

This research introduces a real-time system for UAV detection and identification using the YOLO model and the SDR platform USRP X310. The models YOLOv8n and YOLOv9t were trained on a constructed dataset of RF spectrogram images generated from the captured RF signals (RC, WiFi, and Bluetooth).

Based on our comprehensive evaluation, we conclude that both YOLOv8n and YOLOv9t demonstrate robust capabilities in drone detection and distinction from WiFi and Bluetooth signals. However, YOLOv8n exhibits distinct advantages, particularly in latency-sensitive, real-time applications. It achieves significantly faster inference speeds (1.6 ms vs. 3.0 ms) and sharper edge probability distributions (critical for accurately capturing small UAV signatures in spectrograms), as evidenced by its lower DFL loss, which enhances detection precision. Both models deliver comparable accuracy, with mean Average Precision (mAP), precision, and recall consistently ranging between 0.8 and 0.85. While YOLOv9t offers a lighter parameter footprint and similar computational complexity (7–8 GFLOPs), YOLOv8n's larger parameter count suggests greater model capacity, contributing to its enhanced performance in key areas. Consequently, YOLOv8n is an adequate choice for real-time drone detection systems where high-speed inference is critical.

In summary, our findings underscore YOLOv8n's ability to effectively balance speed and accuracy, making it suitable for real-time anti-drone systems. This study bridges the gap between RF signal processing and AI-driven threat mitigation, offering practical insights for deploying scalable and efficient drone surveillance solutions in diverse operational contexts.

VII. Future Work

While this study demonstrates the efficacy of YOLOv8n and YOLOv9t for single-UAV detection in controlled environments, several directions warrant further investigation. First, scaling to multi-UAV scenarios (e.g., swarms or dense urban airspace) requires evaluating model performance under overlapping RF signatures and higher interference levels, leveraging the spatial preservation properties of spectrograms. Second, environmental robustness—such as adversarial jamming, severe multipath effects, or dynamic modulation schemes—could be addressed through hybrid vision-RF fusion or adversarial training of YOLO architectures. Third, though YOLOv8n's lower DFL loss (22% better than YOLOv9t) ensures precise bounding box refinement, its computational efficiency on edge devices merits empirical validation for field deployment. Finally, expanding the dataset to include diverse UAV models and emerging communication protocols would enhance generalizability. These advancements would bridge the gap between lab-scale validation and real-world anti-drone systems, further solidifying RF spectrogram analysis as a cornerstone for AI-driven UAV threat mitigation.

REFERENCES

[1] R. Tyagi and P. C. Pandey, "Chapter 21 - applications of drones in precision agriculture: future of smart and sustainable farming," in *Remote Sensing in Precision Agriculture*, ser. Earth Observation, S. Lamine, P. K. Srivastava, A. Kayad, F. Muñoz-Arriola, and P. C. Pandey, Eds. Academic Press, 2024, pp. 429–453. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323910682000035

[2] Q. Feng, J. Liu, and J. Gong, "Urban flood mapping based on unmanned aerial vehicle remote sensing and random forest classifier—a case of yuyao, china," *Water*, vol. 7, no. 4, pp. 1437–1455, 2015. [Online]. Available: https://www.mdpi.com/2073-4441/7/4/1437

[3] E. Ackerman and E. Strickland, "Medical delivery drones take flight in east africa," *IEEE Spectrum*, vol. 55, no. 1, pp. 34–35, 2018.

[4] G. Lykou, D. Moustakas, and D. Gritzalis, "Defending airports from uas: A survey on cyber-attacks and counter-drone sensing technologies," *Sensors*, vol. 20, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3537

[5] E. Vattapparamban, Güvenç, A. Yurekli, K. Akkaya, and S. Uluağaç, "Drones for smart cities: Issues in cybersecurity, privacy, and public safety," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, pp. 216–221.

[6] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen, "Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 68–74, 2018.

[7] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak, "Detection, tracking, and interdiction for amateur drones," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 75–81, 2018.

[8] M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, "Rf-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database," *Future Generation Computer Systems*, vol. 100, pp. 86–97, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18330760

[9] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138 669–138 682, 2019.

[10] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective rf-based detection of drones," in *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ser. DroNet '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 17–22. [Online]. Available: https://doi.org/10.1145/2935620.2935632

[11] G. K and S. Kanmani, "A comparative study of various versions of yolo algorithm to detect drones," *Recent Research Reviews Journal*, vol. 2, pp. 54–61, 06 2023.

[12] S. Al-Emadi and F. Al-Senaid, "Drone detection approach based on radio-frequency using convolutional neural network," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 29–34.

[13] R. Hakani and A. Rawat, "Edge computing-driven real-time drone detection using yolov9 and nvidia jetson nano," *Drones*, vol. 8, no. 11, 2024. [Online]. Available: https://www.mdpi.com/2504-446X/8/11/680

[14] T. Delleji and Z. Chtourou, "An improved yolov5 for real-time mini-uav detection in no fly zones," in *International Conference on Image Processing and Vision Engineering*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248866411

[15] T. Delleji, F. Slimeni, H. Fekih, A. Jarray, W. Boughanmi, A. Kallel, and Z. Chtourou, "An upgraded-yolo with object augmentation: Mini-uav detection under low-visibility conditions by improving deep neural networks," *Operations Research Forum*, vol. 3, 09 2022.

[16] T. Delleji, F. Slimeni, A. Ayadi, M. Lafi, and Z. Chtourou, "Electro-optical monitoring system for early-warning detection of mini-uavs," *SN Computer Science*, vol. 6, no. 227, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s42979-025-03756-1

[17] T. Delleji and F. Slimeni, "Rf-yolo: a modified yolo model for uav detection and classification using rf spectrogram images," *Telecommunication Systems*, vol. 88, no. 33, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s11235-025-01264-4

[18] NI, "Usrp x310," 2025. [Online]. Available: https://www.ettus.com/all-products/X310-KIT/