

Advanced Machine Learning
BA-64061

DEEP LEARNING FOR DRONE DETECTION

Final Project Report

Submitted by

Dileep Kumar Pasala
811350727

Table of Contents

- 1. Introduction**
- 2. Background: Deep Learning for Object Detection**
- 3. Deep Learning for Drone Detection: Literature Review**
- 4. Industry Applications of Deep Learning–Based Drone Detection**
- 5. Problem Formulation and Dataset**
- 6. Methodology**
- 7. Experimental Results**
- 8. Discussion**
- 9. Future Directions and Potential Developments**
- 10. Conclusion**
- 11. References**

1. Introduction

1.1 Motivation for Drone Detection

Once considered recreational gadget drones have become widely used in sectors such as delivery, filmmaking inspections, agriculture and defense. Their affordability, agility and ease of operation make them ideal for tasks, yet these same features also render them vulnerable to misuse. They can interfere with airport operations that transport cargo or illegal items, cross zones, enter no-fly areas and conduct unauthorized surveillance.

Security personnel and site/event managers need to detect and track drones instantly using sensors such as cameras, radar or thermal devices. Cameras are especially attractive since CCTV systems are already prevalent and cost-effective to expand. Nevertheless, drones are often tiny fast-moving and sometimes obscured. Can easily merge with busy or low-contrast scenes, like the sky, structures or foliage which complicates spotting them in conventional RGB video. Due to the variation in drones' appearances conventional vision methods relying on handcrafted features struggle to handle them effectively. It is logical to evaluate how well current learning detectors (like YOLO and transformer-based architectures) recognize drones and to compare their accuracy, against model size and real-time performance.

1.2 Problem Statement

This project addresses vision-based drone detection: given a single RGB image, we use a single-class setup with the label "drone" to determine whether a drone is present and draw a bounding box around it. The goal is to achieve high accuracy with few false alarms.

We evaluate the flexibility of detectors (YOLOv8, YOLO11 and RT-DETR) on a labeled drone dataset identify which lightweight/medium models achieve the optimal balance between accuracy and complexity and contrast the transformer-based RT-DETR with CNN-based YOLO models using identical supervised training, with standard train/val/test splits and YOLO-format annotations.

1.3 Objectives of this Study

The two primary aims of this study are a literature-centered objective and an exploratory analysis.

a. Literature and Conceptual Objective

- Examine current deep learning techniques for drone detection, concentrating on transformer-based models, YOLO-style detectors, and UAV-specific strategies.
- Put drone identification in the broader context of computer vision and list the primary challenges: small targets, distant settings, crowded backgrounds, and time constraints. Highlight open research gaps and possible future paths while discussing present and growing industry uses (commercial, defense, aviation, and security).

b. Investigational and Analytical Objective

- To build a practical drone detection pipeline using a large, labeled drone image dataset in YOLO format.
- To train and compare several state-of-the-art detectors on this dataset, specifically:
 1. *YOLOv8n (lightweight, nano-scale model)*
 2. *YOLOv8s (larger YOLOv8 variant)*
 3. *YOLO11n (latest-generation YOLO nano model)*
 4. *RT-DETR-l (transformer-based detector)*
- To evaluate these models quantitatively using standard metrics such as mAP@0.5, mAP@0.5:0.95, precision, and recall on a held-out test set.
- Explain why some models perform better for drone detection and what that means for dependable real-world deployment by examining validation trends, visual detection instances, and failure situations.

2. Background: Deep Learning for Object Detection

2.1 Overview of Deep Learning in Computer Vision

Deep learning has transformed computer vision by replacing handcrafted feature pipelines (like SIFT and HOG + SVM) with end-to-end neural networks. Convolutional Neural Networks (CNNs) employ data to directly learn feature representations: lower layers detect edges and textures while deeper layers represent shapes, parts and whole objects. Following AlexNets 2012 success with ImageNet classification CNN-based architectures such as VGG, ResNet and EfficientNet have become standard, in vision tasks.

Object detection enhances image classification by forecasting the presence and positions of objects than assigning just one label per image. Modern detectors utilize a CNN backbone to generate feature maps, which detection heads then use to predict and classify bounding boxes. Supervised training generally relies on annotated datasets containing bounding box annotations, such, as COCO and Pascal VOC. Mean Average Precision (mAP) serves to evaluate performance across Intersection, over Union (IoU) thresholds; commonly used standards include mAP@0.5 and mAP@0.5:0.95.

These advancements are critical for drone detection since drones frequently appear as little objects in the picture, and effective detection necessitates both specialized detection heads that can localize small targets against complex backgrounds and robust feature extraction.

2.2 Comparing One-Stage and Two-Stage Object Detectors

Two-stage and one-stage methods are the two main categories of deep learning object detectors.

- ❖ Two-phase detectors like R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN initially generate a collection of region proposals that probably contain objects. In this phase these proposals are fine-tuned and categorized. Faster R-CNN significantly sped up R-CNN models by incorporating a Region Proposal Network (RPN) that utilizes shared features,

with the detection component. In scenarios where precise positioning's essential and speed is less important two-step algorithms are often utilized and generally deliver high accuracy.

- ❖ One-stage detectors like YOLO, SSD and RetinaNet eliminate the need for a proposal phase; they simultaneously predict bounding boxes and class probabilities in a single pass through the image. To enable real-time processing early versions of YOLO divided the image into a grid. Predicted bounding boxes, within each grid cell. To tackle class imbalance issues, SSD and RetinaNet applied loss and utilized multi-scale feature maps to enhance the precision of one-stage detection.

When latency and speed are crucial in real-time scenarios like drone detection, autonomous driving, and video surveillance, one-stage detectors like YOLO are frequently chosen. Although two-stage detectors are typically slower and heavier, they can provide higher accuracy on some benchmarks, which makes them less appropriate for deployment on edge devices or on systems that need to monitor many video streams at once.

In this study, we compare a contemporary transformer-based detector (RT-DETR), which can be viewed as an alternative paradigm that integrates detection with attention mechanisms instead of the traditional proposal vs. dense prediction dichotomy, with one-stage detectors (YOLOv8, YOLO11).

2.3 Evolution of YOLO Family (YOLOv1 → YOLOv8 → YOLO11)

Each version of the one-stage detector YOLO ("You Only Look Once") has aimed to find an improved compromise between accuracy and speed. By forecasting boxes and categories, in one pass, the initial YOLO (v1–v3) boosted detection speed. Subsequently improved backbones, anchor boxes and multi-scale predictions improved results in complex situations. Following versions like YOLOv4 and YOLOv5 came with model sizes ranging from micro to extra-large allowing you to select the most appropriate option based on your hardware and latency needs along, with various smart training and design enhancements (enhanced data augmentation, refined loss functions and superior feature fusion). YOLOv7 and other community variants greatly improved performance. Also introduced numerous modifications targeting "tiny object" detection and aerial imagery.

The architecture was revised in the Ultralytics releases: YOLOv8 adopted an anchor-free decoupled framework and evolved into a strong widely used standard for speed and precision extending its applications beyond just detection. The newest edition, YOLO11 targets a balance between accuracy and efficiency through updated training protocols and refined backbone/head choices beneficial especially for smaller hardware. YOLOv8n. Yolov8s function as lightweight CNN baselines, in our work whereas YOLO11n represents a more recent model. They enable us to examine how model capacity and generation impact drone detection effectiveness since they all embrace the same "single-shot real-time" concept but vary in scale and design details.

In this study, we employ the most recent YOLO model, YOLO11n, and representative CNN-based one-stage detectors, YOLOv8n and YOLOv8s. We can empirically investigate how model capacity and architecture generation impact drone detection performance because all three have the same fundamental design philosophy—single-stage prediction, many scale-specific variants—but differ in architectural details and parameter counts.

2.4 Transformer-Based Detectors (DETR, RT-DETR, etc.)

Transformers represent an alternative, to YOLO and various convolution-heavy detectors. They utilize self-attention mechanisms that enable examining the visual feature map simultaneously instead of relying mainly on convolutions. This approach aids in capturing long-distance context and relationships among objects, which's beneficial when objects are small or appear in crowded environments.

This started with DETR (Detection Transformer) which approached detection as a "set prediction" challenge. It eliminates the need for designed elements such as anchor boxes and generally skips NMS by producing a fixed collection of boxes and labels in an end-, to-end manner. The limitation of DETR models was their bulk and lengthy training duration. By utilizing multi-scale attention that focuses exclusively on the most relevant areas, Deformable DETR and its variants improved upon this boosting small-object detection and speeding up convergence though they may still be challenging for real-time applications. Incorporating efficiency strategies (backbones, multi-scale features, refined attention) to cut down latency RT-DETR is a more practical speed-oriented model that preserves the transformer-style global reasoning thus making it a superior choice, for real-time implementation.

Detectors based on transformers can theoretically leverage object relationships and overall context, which proves beneficial when objects interact or in crowded environments. Nonetheless they can be harder to train and fine-tune, on specialized datasets and generally demand greater computational resources. In this research we utilize the drone dataset to directly contrast RT-DETR-l, a typical transformer-based detector with YOLOv8 and YOLO11 models. This enables us to test empirically whether well-tuned CNN-based one-stage detectors are still better for single-class, small-object drone identification or whether the additional complexity of a transformer-based detector results in better performance.

3. Deep Learning for Drone Detection: Literature Review

3.1 Conventional Techniques for Drone/UAV Detection

In the stages of drone (UAV) detection studies, non-visual sensors and conventional signal processing dominated. Drones were. Classified by analyzing radar, RF (radiofrequency) signatures and acoustic arrays based on their electromagnetic signals, propeller sound or radar cross-section. To enhance reliability and reduce positives in crowded environments multi-sensor setups integrate

these methods, for example RF + radar + microphone arrays. Diverse sensor signals are often integrated into one detection outcome, through tracking and Bayesian inference.[1]

At the time conventional vision-based methods relied on optical flow, background subtraction and hand-crafted features such as SIFT or HOG with tracking performed via Kalman/particle filters or traditional classifiers (SVMs random forests). These approaches struggle when drones appear small in the frame move rapidly or are set against backgrounds, like clouds, trees or crowds. Based on investigations into anti-UAV technology pipelines relying on SIFT are challenging to adapt for different camera setups and are vulnerable to motion blur and variations, in scale. [2]

Comprehensive anti-drone surveys highlight the fact that conventional approaches by themselves are frequently inadequate for contemporary threat scenarios: they either have a high false alarm rate when faced with birds, aircraft, or background clutter, or they lack the spatial resolution to detect small UAVs at long range. Deep learning-based vision systems, which are frequently combined with other sensors in multi-modal frameworks, have become more popular as a result.[3]

3.2 CNN-Based Methods (SSD, Faster R-CNN variations, YOLO)

With the rise of learning CNN-based detectors emerged as the favored method for detecting drones in images. Surveys focusing on UAV vision and 2D detection reveal that the leading models are YOLO, SSD and Faster R-CNN. YOLO is generally selected for scenarios demanding real-time processing while two-stage detectors like Faster/Mask R-CNN are preferred when achieving the accuracy and more precise localization is more important. [4], [5]

While two-stage frameworks like Faster R-CNN can detect drones with precision they often run too slowly on embedded systems to allow smooth real-time application. More current research employs one-stage CNN detectors. Lightweight YOLO versions deliver real-time operation on edge hardware such as NVIDIA Jetson sustaining a detection accuracy as shown by Yurchuk et al.[6]

YOLOv3/v4/v5/v8 serves as a base for real-time drone detection in UAV images and surveillance videos. Tailored CNN-based models designed for drone images are introduced by Zhangs Drone-YOLO and similar studies demonstrate that properly optimized YOLO variants can deliver high mAP with minimal latency, on GPUs or edge hardware. [7] Zhang (2023). In order to reduce detections in video sequences and maintain consistent trajectories across frames some approaches merge YOLO-based detectors with simple tracking methods like cross-correlation or Kalman filtering Zhao et al. 2025 [8]

Drone detection is fundamentally a small-object identification problem in many cases, according to CNN-based literature. This is because drones occupy a very small portion of the image, frequently at a long distance, which calls for specific architectural and training changes on top of generic detectors.

3.3 Attention-Based and Transformer Methods (DETR, RT-DETR)

Transformer-driven detectors for object detection have been developed more recently. DETR (Detection Transformer) employs a transformer encoder–decoder to directly generate a fixed number of object queries by treating detection as a prediction challenge. Vanilla DETR features computational demand and slow convergence on typical datasets yet remains elegant and end-, to-end (without relying on hand-crafted anchors or non-maximum suppression). Although certain issues with -scale deformable attention are mitigated by variants such as Deformable DETR they remain more complicated compared to conventional CNN-based one-stage models.[9] 2021 Cazzato et al.

Contemporary architecture focused on achieving real-time operation is known as RT-DETR (Real-Time Detection Transformer). When properly set up RT-DETR can equal or surpass YOLO-type detectors regarding speed–accuracy balance, on COCO. It achieves this by integrating a backbone with a powerful transformer encoder and carefully selected object queries.[10] Zhao and colleagues 2024. To narrow the disparity with CNN detectors even more additional research (like RT-DETRv3) explores enhanced supervision density and architectural enhancements. [11] [Wang et al., 2024]

Transformer-based detectors remain less explored in the drone-detection domain than YOLO variants. When fitted with feature encoders and tailored adjustments for specific tasks, transformer detectors can effectively manage small low-contrast and shapeless targets evidenced by recent research that modifies RT-DETR for analogous tasks such as smoke detection or recognizing objects floating in rivers. [12]. Colleagues, 2025. The bulk of workable systems, according to surveys on vision-based drone detection in complicated environments, still rely on CNN-based YOLO-style models, with transformers emerging as a possible alternative. [13] Liu and colleagues, 2024

3.4 LMWP-YOLO, YOLO-DD, and other specialized drone-detection YOLO variants

- Many researchers modify YOLO especially for UAV/small-object identification because drones are frequently small and difficult to identify in large, cluttered settings.
- These enhancements usually boost -scale feature integration, heighten attention or decrease the models dimensions to preserve speed while more precisely detecting small objects.
- Drone-YOLO and YOLO-Drone: To improve detection of UAVs, in aerial images Zhangs Drone-YOLO and related frameworks modify the neck and multi-scale feature pyramid of YOLOv8.[14] Zhang (2023). Incorporating feature scales and enhanced fusion modules enables these models to surpass standard YOLOv8 on UAV datasets while preserving an efficient runtime.
- Additional lightweight versions of YOLOv8n include LightUAV-YOLO: This variant enhances YOLOv8n by redesigning the neck and incorporating components like

orthogonal feature augmentation to fuse multi-scale features for small objects all while keeping parameter counts extremely low for edge device deployment. [15] Nghiem and colleagues 2024. Similar studies compared to YOLOv8 baselines propose lightweight YOLOv8n refinements utilizing attention modules or GhostConv blocks tailored for UAV imagery typically demonstrating improvements, in both mAP and reduced FLOPs. [16]. Colleagues, 2024

- Zhou et al. Introduce LMWP-YOLO (Enhanced YOLO for Long-Range small Drone detection) a YOLO-derived detector aimed at identifying tiny drones at long distances. Beyond employing depth wise convolutions and pruning methods to reduce the models size it includes multi-dimensional collaborative attention alongside enhanced multi-scale fusion. Tests reveal gains in mAP compared to YOLO11n maintaining real-time efficiency and substantially decreasing parameters, especially for long-range scenarios. [17] Zhou et al. 2025
- YOLO-DD (UAV Identification in Complex Environments): YOLO-DD introduces a boundary- fusion approach and an improved feature extraction pipeline grounded in YOLOv11n to improve UAV detection in intricate cluttered settings. Compared to YOLO models on UAV datasets, this model delivers real-time performance, with enhanced precision while demonstrating resilience against occlusion and background clutter. [18]Zheng and associates 2025
- Extra anti-UAV Yolo variants for example DCR-Yolo and LRDS-Yolo: To effectively differentiate drones from birds and other small airborne objects DCR-YOLO integrates upgraded classification heads with refined feature fusion, for anti-UAV applications. [19] Ding and colleagues 2025. By addressing information degradation and inadequate cross-layer communication via enhanced feature fusion modules, LRDS-YOLO attains mAP with similar or reduced complexity compared to its YOLO baselines, for detecting small objects in UAV aerial imagery. [20]. Colleagues, 2025
- Together, these specialized YOLO derivatives show a distinct pattern: cutting-edge drone detection research focuses more on carefully customizing YOLO-like architecture with attention, multi-scale fusion, pruning, and lightweight design to the unique challenges of small UAV targets than it does on creating completely new paradigms.

3.5 An overview of the current state of research and its gaps

According to publications deep learning has become the accepted norm for recognizing drones through vision. CNN-based single-stage detectors, YOLO models prevail in real-world applications due to their balanced performance in precision, speed and straightforward deployment, on GPUs and edge platforms. [21] Yurchuk and colleagues 2025. By improving fusion features, attentiveness, and lightweight design, specialized variants including Drone-YOLO, LightUAV-YOLO, LMWP-YOLO, YOLO-DD, and DCR-YOLO offer gradual but significant improvements for tiny UAV targets. [22] Zhou and associates, 2025. Despite transformer-based detectors like RT-DETR being relatively recent in the drone field and often more challenging to

train and implement they offer an alternative that can theoretically leverage global context and allow for end-, to-end training. [23]. Colleagues, 2024

Although significant advances have been made quickly numerous research gaps persist:

- Tiny and far-reaching drones: When drones appear very indistinct or occupy just a few pixels, even expert YOLO variants might struggle. At present there is a lack of assessment at extreme distances, and, under real atmospheric conditions, the majority of research focuses on medium-range UAV images.
- Robustness in demanding scenarios and complex settings: Research indicates that effectiveness often declines in urban areas poor lighting, foggy conditions or strong backlighting. While some models (such as YOLO-DD and LRDS-YOLO) are designed to handle backgrounds, comprehensive evaluation, across diverse situations remain rare [24] (Liu et al., 2024).
- Data and domain variation: In contrast with broad datasets like COCO there exist fewer and smaller high-quality annotated datasets specific to drone detection. Further research is needed to bridge the domain gap between synthetic and actual drone images; nevertheless, investigations into generating synthetic data with simulation tools (like Unreal Engine + AirSim) demonstrate potential, for improving generalization and supplementing real-world datasets. [25] In 2023 Wang et al. While accessible aerial datasets such as VisDrone serve well for benchmarking purposes they are generally not created with security contexts involving "drone versus non-drone" situations in focus. [26] VisDrone materials, from Ultralytics
- Integration of sensors and holistic systems: Although most deep learning studies focus on vision numerous real-world anti-drone solutions employ multiple sensors (RF, radar, audio and vision). Extensive deep learning models that incorporate all sensor types concurrently remain largely uninvestigated despite research on multi-sensory Bayesian models, for UAV detection suggesting that merging deep visual detectors with probabilistic fusion across different modalities presents a promising approach. [27] In 2025 Saadaoui et al.
- Edge deployment and power efficiency: Research indicates that alongside accuracy factors like energy usage, delay and reliability, under computing power hold equal significance in real-world applications despite many lightweight YOLO models delivering impressive FPS on edge hardware. [28]Nghiem et al., 2024

To summarize the latest advancements, indicate that especially for class real-time security scenarios, Designed YOLO-based detectors represent the most feasible approach for drone detection at present. While transformer-based detectors and multi-sensor fusion methods are not yet as mature or widely applied in this domain, they offer promising opportunities for investigation.

The Analytical portion of this study, which compares strong YOLO baselines (YOLOv8n, YOLOv8s, and YOLO11n) against a representative transformer detector (RT-DETR-l) on a sizable drone detection dataset, is motivated by this background.

4. Applications for Deep Learning-Based Drone Detection in Industry

4.1 Security and Monitoring (Public Events, Borders, Critical Infrastructure)

The application of learning for drone detection has evolved from experimental setups to real-world security implementations protecting facilities such as refineries, power stations, correctional facilities, data hubs and extensive urban surveillance systems. To enable drone identification, tracking and immediate reaction vendors are progressively adopting modal sensing (RF, radar and cameras PTZ). As an example, Dedrone offers systems designed for short-term protection at important sites and large public gatherings.[29] Dedrone

Airports represent another element: suppliers recount multi-year implementations and operational insights, from sites including Heathrow, Gatwick and Venice while operators demand continuous surveillance owing to extended clear approach trajectories and strict safety standards. After the December 2018 Gatwick incident, which affected around 140,000 passengers and disrupted approximately 1,000 flights, it is easy to appreciate the urgency because it shows how a single rogue drone can inflict significant operational and financial harm. [30]([en.wikipedia.org])

4.2 Aviation and Airports (Runway Safety, Restricted Airspace)

Since even minor disruption can lead to safety hazards and substantial costs airports are among the first and most extensive users of AI-driven drone detection systems. After the 2018 Gatwick incident airports boosted their spending on counter-drone technologies integrating cameras, radar, RF sensors. The vision/AI component assists in visually verifying whether a detected object is truly a drone (than a bird or aircraft) and enables faster operational decision-making. [31](Aviation Week)

Drone detection has become an aspect of airfield security within the industry: OSL outlines extended deployments at Heathrow, Gatwick and Venice sharing valuable insights such as optimal sensor positioning, incorporating alerts into airport/ATC processes and educating personnel to understand detections. Providers such as Sentrycs deliver tailored airport solutions that transmit alerts to airport operations centers for response and utilize RF-based identification combined with decision-making logic (and sometimes mitigation when permitted). While visual deep-learning detection is still useful as an independent cross-check, particularly for non-compliant drones or weak RF reception, regulators also require identification through FAA Remote ID (“digital license plate” broadcasting) [32]([OSL Technology]).

4.3 Defense and Military Applications

Drones present a challenge, in defense; they serve as valuable tools while also posing considerable threats. The military now employs AI-powered detection to protect supply lines,

bases and convoys from leftover explosives and small commercial quadcopters repurposed as bombs. For instance, Droplis Eyes in Ukraine examines thermal and optical footage captured by drones surveying in front of convoys delivering real-time analysis at speeds of up to ~130 FPS to detect dangers prior to vehicles arriving there.[33] [Business Insider]

Protection counter-UAS systems at locations (like forward bases and depots) are like civilian configurations RF, radar, EO/IR cameras—yet they enforce rigorous standards regarding engagement protocols, response time and reliability. For example, Dedrone offers its DedroneTracker.AI C2 platform to military users to recognize, monitor classify and neutralize drone threats on-site. Deep learning helps cut through clutter (birds/false alarms), estimate what the drone is and how dangerous it might be, and prioritize targets for electronic or kinetic responses all while feeding into command-and-control systems.[34]

4.4 Commercial & Civilian Use Cases (Delivery Drones, Urban Air Mobility)

Deep learning vision allows commercial drone deliveries and urban air mobility by enabling drones to navigate around structures and power cables evade obstacles and identify landing spots. Companies such as Amazon Prime Air utilize sophisticated onboard perception on devices like the MK30 and Amazon has been actively expanding its pilot initiatives across the U.S. (Including Texas and Arizona) with rollouts such as San Antonio and continuous trials, as the program advances.[35]

Beyond delivery, AI-equipped drones are frequently employed for monitoring and inspection (turbines, pipelines, bridges, electrical lines) because they can automatically identify flaws like corrosion, cracks, or vegetation encroachment, which lowers risk and costs compared to manual inspection. Drones are also being used by food and logistics platforms concurrently: Uber Eats and Flytrex are planning test deployments in the United States (Flytrex has already recorded extensive suburban delivery experience), demonstrating how autonomy and vision are becoming commonplace in business operations.[36]

4.5 Practical Constraints: Real-Time Performance, Edge Deployment, Regulations

Real-world systems must be quick, deployable, and human-useable within practical limitations in addition to being accurate.

Initially hardware and processing speed play a role. To enable personnel to react to a drone entering a restricted zone, models need to function almost in real time over multiple streams with minimal delay since airports and security teams might oversee hundreds of camera inputs. Given that many systems installed on rooftops, vehicles or mobile units with limited power supply and cooling capabilities teams are driven to use efficient detectors (usually YOLO-based pruned/compressed models or fine-tuned transformer versions) deployed on accelerators such, as embedded GPUs or edge processors.[37]

Secondly its practical effectiveness is influenced by reliability, regulations and procedures. Due to circumstances (fog, rain, glare, darkness) and common false alerts (birds, balloons, debris) outdoors systems often integrate RF/radar/EO-IR to reduce false positives; however maintaining robustness continues to be a difficult and ongoing challenge. Additionally, privacy standards are required when monitoring public places, and civil deployments must adhere to rules like FAA Remote ID (the FAA terminated its discretionary enforcement policy on March 16, 2024, and disobedience can result in penalties and certificate action). Lastly, the system needs to be connected to actual SOPs, which include training so that operators, ATC, and law enforcement can respond promptly, clear alerts, and reasonable thresholds to prevent alarm fatigue.[38]

5. Problem Formulation and Dataset

5.1 Task Definition (Single-Class Drone Detection)

We approach drone detection as a single-class object detection task: the model predicts bounding boxes (b_i) and confidence scores (s_i) given an RGB picture (I). Each (b_i) represents the location of a drone while each ($s_i \in [0,1]$) indicates the models' confidence that the drone is contained within the box. Input is a single RGB picture taken from the ground or from above. Output is a set of bounding boxes of varying lengths with the class label "drone" and evaluation by using a held-out test set, standard detection metrics (mAP@0.5, mAP@0.5:0.95, precision, recall) were calculated. The main issues are resilience to cluttered backdrops, maintaining high recall without an unacceptable rise in false positives, and localization quality (tight boxes around little drones) because there is only one foreground class.

5.2 Dataset Description (Source: Roboflow Drones Dataset)

For our study we employ Roboflow Universes openly available drone datasets specifically the Drones project (version 4) designed for YOLO training. This dataset consists of drone images captured across multiple environments (rural, city, sky scenes near structures and foliage with differing illumination and angle perspectives). Each image is accompanied by bounding boxes for one category:

- Number of classes: 1
- Image modality: RGB images (JPEG).
- Label format: YOLO text files (one label file per image).

The dataset is relatively large for a single-class detection task:

- Training set: 20,256 images
- Validation set: 1,679 images
- Test set: 1,063 images

We confirm that there are no missing or mismatched labels in any of the three splits. Each split comprises a one-to-one connection between image files (in images/) and label files (in labels/). The dataset is ideal for testing contemporary detectors and examining how architecture and model size affect drone detection performance because of its size and diversity.

5.3 Train/Validation/Test Splits and Annotation Format (YOLO)

The dataset is divided into training, validation and test segments preserved precisely as originally provided. This approach prevents data leakage. Ensure that our results can be compared with those from other research using the identical dataset. Multiple bounding boxes (for instance drones in one picture) appear as several lines, within the same label file with all coordinates scaled to $[0,1]$. There was no need for conversion or relabeling because YOLOv8/YOLO11 and RT-DETR in the Ultralytics framework naturally accept this short annotation format.

5.4 Data Preprocessing and Augmentation

We utilize the Ultralytics versions of YOLOv8 YOLO11. Rt-DETRs data-loading and augmentation framework, with configurations chosen to balance processing efficiency and authenticity. The primary stages, in preprocessing and augmentation are as follows:

- Image padding and resizing: Every image is letterboxed and resized to a fixed input size of 640×640 pixels.
- Normalization: Pixel values are scaled from $[0, 255]$ to $[0, 1]$. The models' internal preprocessing implicitly handles standard mean/std normalization, and the channel order is RGB.
- Geometry improvements: To allow the model to generalize to drones coming from side angles we apply the standard Ultralytics augmentations designed for object detection, during training including o Random horizontal flips (mirroring).
- Random translation and scaling within acceptable bounds, enhancing resilience to changes in size and location. Depending on the precise Ultralytics version, random cropping and mosaic-like compositions can successfully replicate various zoom levels and scenes with numerous elements.
- Photometric enhancements: The model is exposed to changes in lighting and backdrop color through color jittering, which is characterized by slight shifts in brightness, contrast, saturation, and hue. The detector can handle photographs shot at different times of day and with varying camera settings thanks to sporadic fluctuations in exposure and gamma.
- Label-preserving uniformity: Each augmentation is carried out in a manner that maintains label consistency: the Ultralytics dataloader adjusts the bounding boxes when images are flipped, resized or shifted. We avoided augmentations that might compromise the small drone bounding boxes and did not make any manual changes to the annotations.

For this study, no additional datasets or synthetic data generation were used; all experiments were conducted only on the Roboflow drone dataset that was supplied, using the preprocessing and augmentation workflow mentioned above. This guarantees that architectural and capacity differences, rather than variations in data distribution, are the main causes of performance disparities between YOLOv8n, YOLOv8s, YOLO11n, and RT-DETR-l.

6. Methodology

This section details the models we assessed the training process (employing a primary configuration and hyperparameters across experiments) and the metrics utilized for outcome comparison. Our focus is, on two detector "families": transformer-based detectors and CNN-based single-stage YOLO models.

Featuring a CNN backbone multi-scale feature fusion (PAN/FPN-style neck) and a decoupled head, YOLOv8n functions as the lightweight baseline (anchor-free one-stage) for YOLO. It is optimized for edge speed utilizing COCO-pretrained weights (yolov8n.pt), on the Roboflow drone dataset. Using yolov8s.pt for a fair same-family comparison, we also train YOLOv8s, which maintains the same architecture but adds capacity (more layers/channels), typically enhancing detection on small/occluded drones at a low computation cost.

We employ RT-DETR-l (starting with rtdetr-l.pt) to depict transformers. Due to its size we adopt much more cautious training configurations to maintain stability. It incorporates a CNN backbone alongside a transformer encoder/decoder utilizing object queries to exploit context. To compare (1) v8n vs. v8s (capacity), (2) v8n vs. 11n (generation), and (3) YOLO vs. RT-DETR (CNN vs transformer), we train YOLO11n (yolo11n.pt), a newer-generation nano YOLO designed to enhance the accuracy–efficiency trade-off at a similar scale to v8n.

6.2 Training Setup

Hyperparameters

For the YOLO models, we tried to employ a constant training budget; for RT-DETR-l, we made small tweaks to maintain stability.

- **YOLOv8n**
 - Epochs: 20
 - Batch size: 16
 - Image size: 640×640
 - Learning rate: default Ultralytics setting (with cosine LR schedule)
 - Early stopping: patience = 5 epochs without improvement on validation metrics
- **YOLOv8s**
 - Same configuration as YOLOv8n:

- Epochs: 20
- Batch size: 16
- Image size: 640×640
- Cosine learning rate schedule, default base LR
- Early stopping: patience = 5
- **RT-DETR-l**

Unstable training (NaN weights) resulted from the first attempt using more aggressive hyperparameters. As a result, we chose a more cautious configuration:

 - Epochs: 10
 - Batch size: 8 (reduced to decrease memory and stabilize gradients)
 - Image size: 640×640
 - Initial learning rate: $lr_0 = 0.0005$ (smaller than default)
 - Early stopping: patience = 3
 - Cosine learning rate schedule
- **YOLO11n**
 - Epochs: 15
 - Batch size: 16
 - Image size: 640×640
 - Learning rate: default Ultralytics base LR with cosine schedule
 - Early stopping: patience = 5

Since we employed the same picture size, training/validation/test splits, and data augmentation settings for all models, performance variations may be mostly ascribed to model architecture and capacity rather than variations in training data or pipeline configuration.

Loss Functions and Optimization

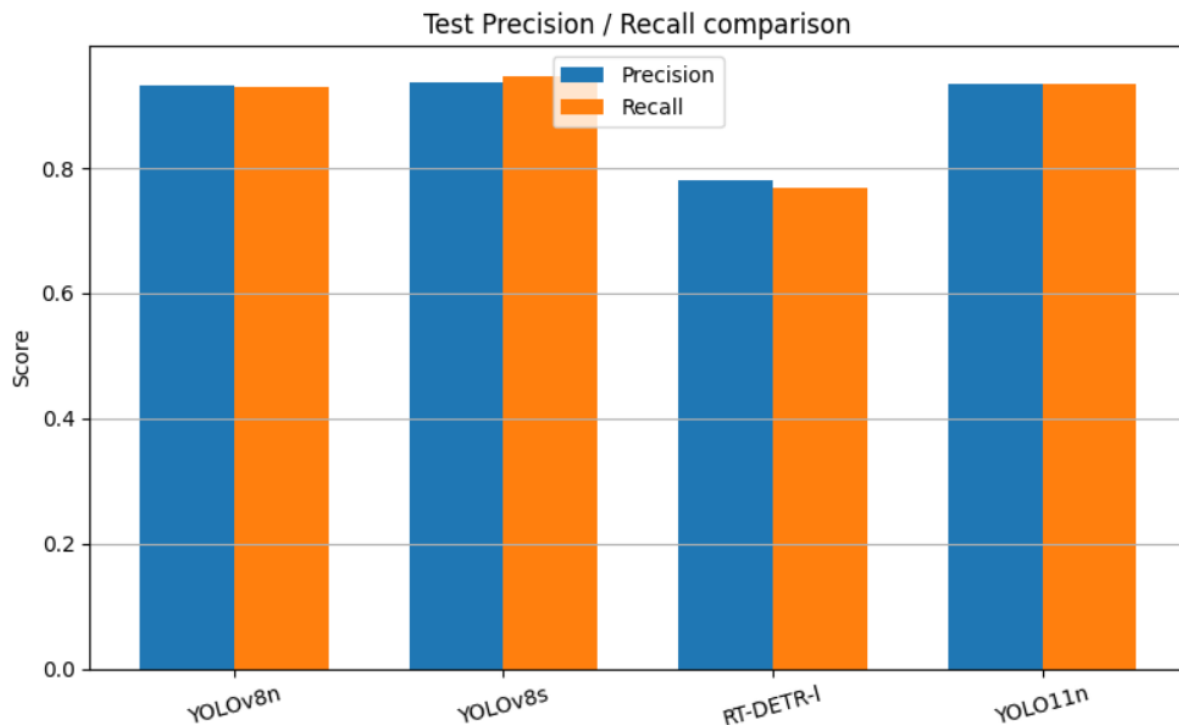
The YOLO models (YOLOv8n/YOLOv8s/YOLO11n) train by minimizing a sum of box regression loss, confidence/objectness loss and classification loss which are optimized with a typical deep-learning optimizer (commonly SGD or a variant similar, to AdamW based on default settings). These losses are handled internally by Ultralytics. Instead, RT-DETR-l employs a DETR-style objective, which allows for end-to-end prediction without anchors by first matching predictions to ground-truth boxes with Hungarian matching and then applying L1/IoU-based box losses plus a classification loss over object queries. Mini-batch gradient descent on GPU is used for training all models, and we rely on validation-monitored early stopping to minimize overfitting and prevent computation waste.

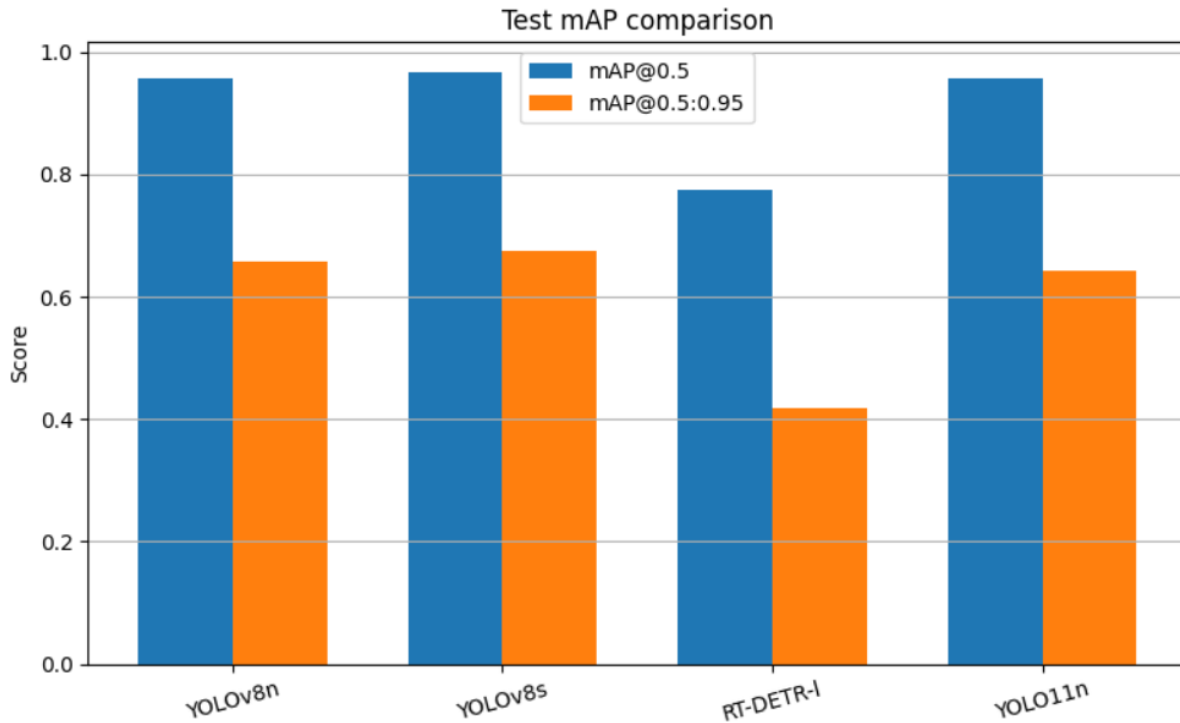
6.3 Evaluation Metrics

Employ the Ultralytics validation method along with detection metrics to assess every model on the reserved test dataset. Predictions are assessed through Intersection over Union (IoU), against ground-truth boxes (for instance deemed correct if $\text{IoU} \geq 0.5$) and the overall performance

is encapsulated by Average Precision (AP) and its mean counterpart mAP. We report $\text{mAP}@0.5:0.95$ the COCO-style metric averaged over IoU thresholds ranging from 0.5 to 0.95 offering a more precise measure of localization accuracy—especially important, for small drones—and $\text{mAP}@0.5$ (AP50)**, which evaluates detection performance at one IoU threshold. In addition, we provide a comprehensive picture of accuracy and robustness across architecture by reporting precision (P) and recall (R) to capture false-alarm vs. missed-detection behavior and by examining training/validation curves (loss and mAP vs. epoch) to comprehend convergence and check for overfitting.

7. Results





7.1 Quantitative Results on Validation and Test Sets

The YOLOv8n, YOLOv8s, RT-DETR-l, and YOLO11n models' respective performances. We note each model's optimal checkpoint on the validation set using mAP@0.5:0.95:

- **YOLOv8n** reaches its best validation performance at **epoch 19**, with
 - mAP@0.5 \approx **0.9607**, mAP@0.5:0.95 \approx **0.6769**
 - Precision \approx **0.9423**, Recall \approx **0.9367**
- **YOLOv8s** achieves the best overall validation results at **epoch 20**, with
 - mAP@0.5 \approx **0.9707**, mAP@0.5:0.95 \approx **0.6960**
 - Precision \approx **0.9468**, Recall \approx **0.9414**
- **YOLO11n** converges slightly earlier, with its best checkpoint at **epoch 15**, reaching
 - mAP@0.5 \approx **0.9578**, mAP@0.5:0.95 \approx **0.6655**
 - Precision \approx **0.9437**, Recall \approx **0.9330**
- **RT-DETR-l** behaves differently: its best validation performance occurs already at **epoch 1**, with
 - mAP@0.5 \approx **0.7840**, mAP@0.5:0.95 \approx **0.4356**

- Precision \approx **0.8026**, Recall \approx **0.7587**

These figures do not improve with subsequent epochs; in fact, performance deteriorates and the last epoch is unstable, which caused early stopping under the "safe" setting.

Using each model's optimal checkpoint on the held-out test set, we get the following metrics.:

- **YOLOv8n**
 - mAP@0.5 = **0.9575**
 - mAP@0.5:0.95 = **0.6564**
 - Precision = **0.9315**, Recall = **0.9297**
- **YOLOv8s**
 - mAP@0.5 = **0.9676**
 - mAP@0.5:0.95 = **0.6761**
 - Precision = **0.9367**, Recall = **0.9474**
- **RT-DETR-l**
 - mAP@0.5 = **0.7747**
 - mAP@0.5:0.95 = **0.4190**
 - Precision = **0.7810**, Recall = **0.7678**
- **YOLO11n**
 - mAP@0.5 = **0.9564**
 - mAP@0.5:0.95 = **0.6425**
 - Precision = **0.9340**, Recall = **0.9342**

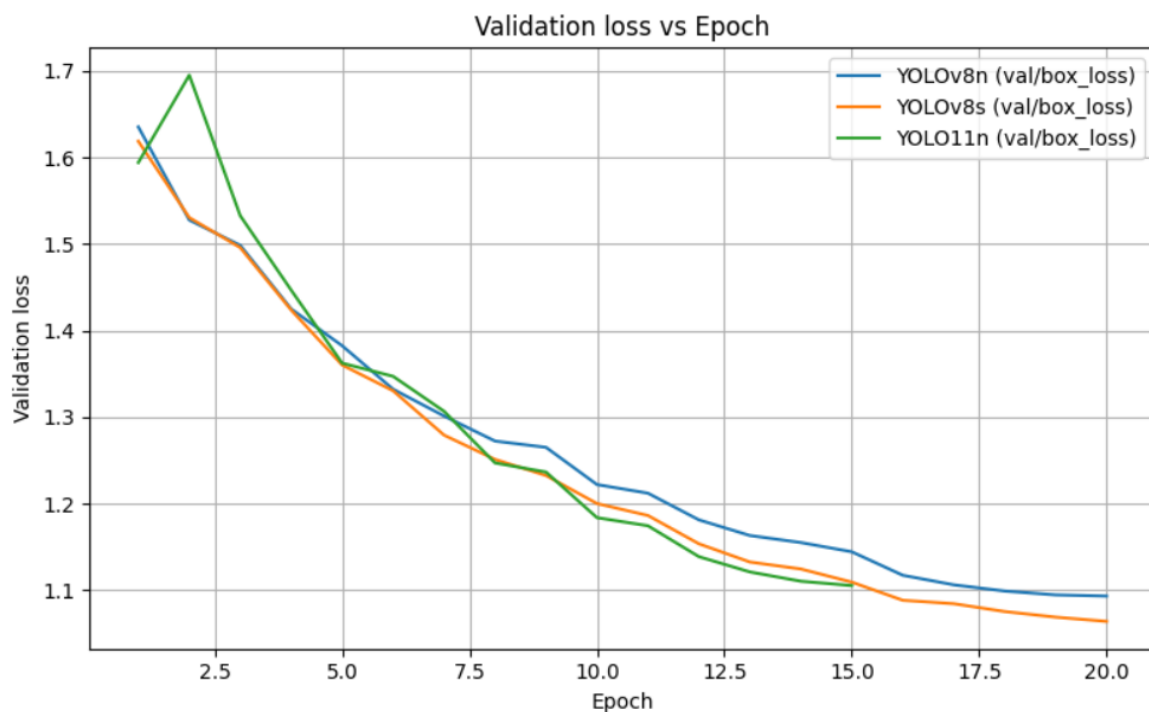
The strong correspondence between test and validation measures (YOLOv8s, for instance, obtains mAP@0.5:0.95 = 0.696 on validation and \approx 0.676 on test) indicates that all three YOLO-based models have good generalization and do not significantly overfit to the training/validation splits.

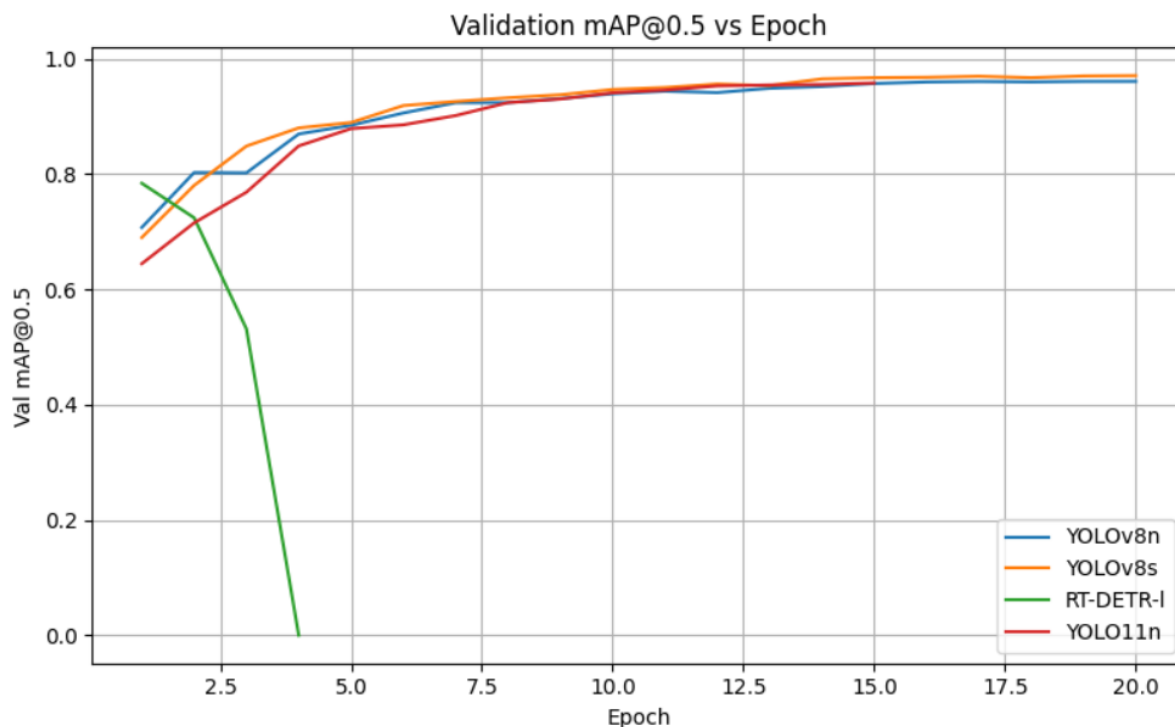
7.2 Comparison of Detectors (YOLOv8n vs YOLOv8s vs YOLO11n vs RT-DETR-l)

With the best mAP@0.5, mAP@0.5:0.95, and the highest test recall (0.9474)—meaning it misses the fewest drones, which is particularly crucial for security use-cases—YOLOv8s is unquestionably the best performance among the four detectors based on validation and test results. A very close second tier is occupied by YOLOv8n and YOLO11n: Both reach around 0.956–0.958

mAP@0.5 and 0.64–0.66 mAP@0.5:0.95 with comparable precision/recall around 0.93–0.94, demonstrating that architectural improvements allow YOLO11n to match (or marginally edge) YOLOv8n at a similar model scale. With mAP@0.5:0.95 = 0.419 and poorer precision/recall ($\approx 0.78/0.77$), RT-DETR-l lags well behind, indicating that its transformer advantages do not translate into improved practical detection for this single-class drone dataset—and within the selected training budget.

Overall, $\text{YOLOv8s} > \text{YOLO11n} \approx \text{YOLOv8n} \gg$ is the deployment-relevant lesson. RT-DETR-l is less desirable unless it is greatly adjusted or trained for a longer period with more data, whereas YOLO models provide good accuracy with manageable computation for real-time surveillance.





7.3 Training Dynamics

Validation mAP curves.

The YOLO models perform well during training. There are clear differences in speed and limits as shown by the validation mAP@0.5:0.95 curves. Continuing training past 20 epochs yields no improvements for YOLOv8n, which starts at approximately 0.37 at epoch 1 and climbs steadily to around 0.68 by epoch 19 before plateauing. YOLOv8s follows a pattern but stays above v8n most of the time growing from about 0.37 to roughly 0.70 by epoch 20 demonstrating that its greater capacity is truly advantageous, for difficult and tiny drone samples. YOLO11n gets slightly faster, increasing from roughly 0.35 to approximately 0.67 by the 15th epoch then it levels off with minimal progress.

RT-DETR-l has quite different behavior: it peaks early (~0.436 at epoch 1) and subsequently deteriorates, falling to ~0.29 by epoch 3. The final epoch becomes unstable (zero metrics owing to NaNs), indicating that this dataset is more sensitive to optimization and hyperparameter selections.

Validation loss curves.

Validation loss graphs reinforce the story. The box-regression loss for the YOLO models declines steadily and as expected: YOLOv8n reduces from 1.64 to about 1.09 by around epoch 20 YOLOv8s drops from roughly 1.62 to near 1.06 during the same period (aligning with marginally better localization) and YOLO11n falls from close, to 1.59 to approximately 1.10 by epoch 15 before leveling off. In line with its collapsing mAP curve and poorer test results, RT-DETR-l's

losses (e.g., GIoU and L1) start at reasonable values (val GIoU ~ 0.52 , val L1 ~ 0.38 at epoch 1), fail to improve consistently, and eventually become undefined due to NaNs. While RT-DETR-l finds it difficult to train steadily with the same training budget and settings, the training dynamics generally significantly favor the YOLO family, which converges consistently with stable gains and minimal indication of overfitting.

7.4 Qualitative Results

We visually examined predictions on a range of test images by superimposing bounding boxes and confidence ratings for the top YOLO models (YOLOv8s, YOLOv8n, YOLO11n) and RT-DETR-l to supplement the numerical results. These examples make it easier to grasp the performance differences in mAP, precision, and recall by highlighting the models' strengths and weaknesses in real-world scenarios.

Detection examples (correct detections).

All three YOLO models deliver performance in standard situations: a lone drone, against a plain sky is often identified with a precise bounding box and high certainty (commonly >0.9) while scenarios featuring multiple drones are generally accurately managed as separate entities especially by YOLOv8s and YOLO11n. Even though bounding boxes might be less precise when the object is smaller, or the background is more complex drones partially obscured by trees or buildings are still commonly detected. In generally difficult settings, YOLOv8s generates the most accurate and stable boxes, although YOLOv8n and YOLO11n resemble each other on many typical examples.

Failure cases (false positives / false negatives).

The challenging failures happen when drones are tiny and distant (just a handful of pixels) particularly if they merge with the sky. This often leads to confidence or missed detections, which also clarifies why mAP@0.5:0.95 (more stringent localization-sensitive) lags mAP@0.5 across all models. Missed or misaligned boxes may also occur due to severe lighting and low contrast (intense sunlight, heavy clouds, backlit shadows). False alarms sometimes occur due to birds and other small airborne items; this effect is more noticeable, in RT-DETR-l and YOLOv8n compared to YOLOv8s aligning with the precision of v8s.

In general RT-DETR-l proves to be the dependable; its poorer quantitative performance aligns with its habit of giving lower confidence scores and more often failing to detect drones that YOLOv8s/YOLO11n successfully recognize. Practically speaking this suggests that YOLOv8s serves as a main detector (with YOLO11n/YOLOv8n, as lightweight alternatives). Higher-resolution inputs, small-object-focused heads, or multi-sensor fusion (radar/thermal) could fill the remaining gaps, which include tiny long-range targets, harsh lighting, and bird confusion.

8. Discussion

8.1 Accuracy vs Model Size and Complexity

The findings reveal a balance between precision and computational effort: within this drone dataset the bigger model YOLOv8s surpasses the smaller YOLOv8n and YOLO11n regarding mAP@0.5, mAP@0.5:0.95 and recall. This aligns with the pattern that bigger YOLO versions (s/m/l/x) deliver higher accuracy due to their greater capacity especially for detecting small drones and, in dense environments. [39]Liu2024

Although more compact, YOLOv8n and YOLO11n exhibit performance comparable to YOLOv8s supporting results from lightweight UAV-centric models (like LightUAV-YOLO) that carefully crafted small networks can still deliver high mAP on edge devices. [40] Nghiem2024. In general, this supports a viewpoint: unless the task or dataset is significantly more challenging employing much larger CNN detectors yields only marginal improvements, in drone detection. Small CNN detectors often offer a trade-off between accuracy and efficiency.

Despite being larger and more complicated, RT-DETR-l performs significantly worse than all three of the YOLO models in this case, demonstrating that "more advanced" design does not always translate into better outcomes, particularly when tuning and time are limited. This is consistent with more general findings that DETR-style models frequently require longer training times and careful hyperparameter selection in order to compete; on our dataset, the more straightforward, well-optimized YOLO variants unquestionably prevail. [41] Zhao2024

8.2 CNN-Based YOLO vs Transformer-Based RT-DETR

An advantageous balance emerges when comparing CNN-driven YOLO models with the transformer-driven RT-DETR-l: YOLOv8/YOLO11 employ backbones alongside multi-scale feature pyramids and dense prediction heads making them typically simpler to train for consistent high performance and especially proficient at detecting small objects such as drones. [42] Liu2024 Conversely RT-DETR utilizes set-based prediction along with self-attention, which may successfully grasp greater context but often proves to be computationally more demanding and more sensitive to optimization choices. [43] Zhao2024. Although RT-DETR-l achieved its point immediately and then dropped with inconsistent subsequent epochs the YOLO models, in our trials using a moderate training schedule and a single-class dataset converged steadily with increasing validation mAP and decreasing loss.

This aligns with DETR studies which indicate that transformer-based detectors generally need adjustment (learning rates, warm-up extended schedules) and bigger/more varied datasets to perform optimally. [44] Wang2024. Most deployed drone-detection pipelines still prefer YOLO-like CNN detectors, and even specialized anti-UAV models (LMWP-YOLO, YOLO-DD, DCR-YOLO) typically build on YOLO backbones with additional attention/fusion rather than completely switching to full transformers, according to recent surveys and applied systems. [45]Yurchuk 2025

8.3 Suitability for Real-Time Drone Detection and Edge Deployment

The demanding criteria for drone detection at borders, airports and critical infrastructure involve minimal latency, substantial throughput (usually numerous video streams simultaneously) and a compact hardware design suitable for field use. Commercial counter-UAS systems from companies such as Dedrone, OSL and Sentrycs emphasize sensing and operational, near-real-time surveillance; these solutions are generally delivered through edge or remote operational environments as opposed to purely laboratory-based single-stream demonstrations.

In that regard, our findings are in line with previous lightweight UAV detector discoveries: YOLOv8n and YOLO11n provide accuracy comparable to YOLOv8s at significantly lower sizes, making them excellent candidates for Jetson-class/edge accelerators.

For server-side, multi-camera SOC setups, YOLOv8s is the most accurate option while still being sufficiently efficient (and aligning with the "fast vision module inside a multi-sensor pipeline" pattern seen in industry).

RT-DETR-l, on the other hand, is heavier and more difficult to train consistently, but it performs poorly here. As a result, unless it is further optimized and trained more carefully, it is currently a weaker fit for edge deployment. This is consistent with broader observations that transformer detectors can be more sensitive to tuning and cost for real-time workloads.

8.4 Lessons Learned from Hyperparameters and Training Behavior

Several clear and valuable insights emerged from the training experiments. To begin with lengthy training schedules are not required for the YOLO models (v8n, v8s and 11n) to deliver performance in this scenario. They reached accuracy within roughly 15–20 epochs displaying steady validation improvements and no evident overfitting—consistent, with many YOLO-focused UAV studies that use clean well-annotated datasets.[46] Zhou 2025. Secondly RT-DETR-l exhibits higher sensitivity: the initial attempt produced NaNs and training only became stable after lowering the learning rate (and using a more conservative "safe" configuration such as smaller batch size and patience). Nonetheless its performance remained inferior to YOLO suggesting that DETR-based models often need precise hyperparameter tuning and sometimes extended training to reach competitiveness. [47] Zhao2024

Given that this dataset contains one class and is extensive yet less varied compared to COCO we noticed that the model's capacity needs to align with the dataset's complexity. Under these circumstances YOLOv8s gains from increased capacity whereas a substantial transformer does not indicate diminishing returns (or even poorer results) when complexity outpaced the supervision signal and the amount of tuning applied. 8.1 Liu2024 Finally validation curves played a role in choosing checkpoints and justifying early stopping: the early rise and fall in RT-DETR-l indicated optimization issues and validated the cautious training choice whereas YOLOs steady mAP improvements boosted confidence, in its consistency. [48] Cazzato 2021. The most promising next steps—better small-object feature fusion/attention (as in LMWP-YOLO/LRDS-YOLO/DCR-YOLO) and, in deployments, multi-sensor fusion instead of just scaling up to a larger

architecture—are also indicated by the qualitative failure cases, which include tiny long-range drones, harsh lighting, and bird confusion.

9. Future Directions and Potential Developments

9.1 Improving Small-Drone and Long-Range Detection

Despite YOLO baselines, very small faraway drones (only a handful of pixels) continue to be the hardest to detect. Enhancements, in scale feature fusion, attention and even specialized small-object heads have demonstrated notable progress in recent studies.

Pruning and compression are also used by models such as LMWP-YOLO to increase accuracy while reducing model size for deployment. Tiny-target performance is routinely improved by YOLOv8/YOLOv11 UAV-focused enhancements (additional fusion blocks, scale-adaptive modules). The next steps for this project are to test structural pruning, add a small-target head to YOLOv8s/YOLO11n, and try those small-object modules.

9.2 Multi-Modal Fusion (Radar, Thermal, Audio + Vision)

Actual counter-UAS operations progressively depend on modal fusion, combining radar, RF, thermal/IR and sometimes audio alongside vision—since every sensor compensates for the limitations of the others (radar excels at extended distances RF supports identification and vision is ideal, for verifying and pinpointing the target). Nonetheless most of our tests utilize RGB images.

The idea of "strengths" is often highlighted in analyses of counter-UAV technologies and many investigations show that integrating tracking methods such as (extended) Kalman filters with radar and camera data helps to maintain stable tracks and reduces false positives compared to using vision exclusively; recent research on micro-drones also indicates that multi-sensor configurations are generally more reliable in noisy cluttered conditions, than relying on any individual sensor alone. Adding thermal/IR for night/backlighting, using radar/RF detections as priors that the vision model verifies and refines, and investigating a learned fusion layer (e.g., transformer- or graph-based) that combines features across sensors prior to detection and tracking are the obvious next steps for our pipeline.

9.3 Domain Adaptation and Adverse Weather / Night-Time Scenarios

Because actual deployments need to endure fog, rain, snow, haze and nighttime conditions—where effectiveness often declines significantly—most drone detectors, including ours, are mainly trained on daytime images. Consequently, recent research emphasizes domain adaptation and training that accounts for weather. For example, Foggy Drone Teacher enhances condition performance without the need for fully annotated fog datasets by employing a Mean Teacher approach to adjust a clear-weather drone detector, for fog using pseudo-labels.

According to models such, as DA-RAW, which integrate contrastive learning and attention-based alignment to develop weather-resistant features the clear, bad-weather gap can be separated into a "style gap" and a "weather gap." Night-oriented techniques such as SAM-DA modify daytime models for UAV images by employing segmentation-guided supervision to reduce the dependence on extensive night annotations. Similarly related approaches show that explicitly incorporating weather factors (weather-aware normalization in HyperFLAW/FLYAWARE) enhances aerial perception, under various conditions. This indicates two feasible next steps for our pipeline: investigate night/thermal-specific heads or normalization (shared backbone, domain-specific components) for reliable 24/7 operation, and train YOLO with strong synthetic weather/low-light augmentation plus unsupervised adaptation (Mean Teacher/DA-RAW-style).

9.4 Integration with Multi-Drone Tracking and Counter-UAV Systems

Genuine counter-UAV operations need to monitor drones continuously predict their paths, assess threat levels and support counteractions beyond simple frame-by-frame identification. To manage multi-drone environments with restricted computational resources current methods, merge powerful, per-frame detectors with multi-object tracking (MOT) and incorporate global-local transformer techniques. Additional studies show that integrating vision detections with radar/RF through EKF/particle-filter tracking can decrease trajectory inaccuracies and false positives relative to using vision. Current research highlights that detection, classification and tracking ought to be designed for continuous situational awareness.

Within operational and legal boundaries assessments from industry and counter-UAS studies also indicate "closed-loop" frameworks that connect sensing to decision-making processes and where legally permissible, mitigation options such as alerts, RF control, jamming/spoofing or kinetic actions. Our YOLOv8s/YOLO11n/YOLOv8n models are accurately viewed as front-end perception units, within this broader framework. The following actions involve integrating them into a MOT pipeline (, like DeepSORT or transformer-based MOT) to produce track IDs and speed estimates incorporate threat assessment (geofences, loitering approach-speed heuristics) and link the results to either simulated or real countermeasure systems. Ultimately the industry is progressing toward multi-UAV monitoring, were units exchange perception and control data.

Overall, the path is clear: our recent studies can provide a strong basis for end-to-end systems that unify small-drone-optimized models, multi-modal sensing, robustness to weather/night, and integrated tracking and response—rather than just better detectors.

10. Conclusion

This project explored drone detection through learning from both theoretical and applied viewpoints. The literature review covered how CNN-based detectors, YOLO-inspired one-stage architecture, have mostly supplanted traditional UAV detection techniques such as radar, RF, acoustic approaches and conventional computer vision. To address challenges, with objects and cluttered scenes we also reviewed recent adaptations of YOLO tailored for UAV scenarios

incorporating enhanced multi-scale fusion, attention mechanisms and streamlined designs. Although they are an intriguing substitute, transformer detectors such as RT-DETR are still less prevalent in developed, functional drone-detection stacks.

Utilizing the Roboflow drone’s dataset (over 20k training images) with YOLO-style annotations we developed a single-class drone detection system, for evaluation. Employing a training and testing framework we refined four modern detectors: YOLOv8n, YOLOv8s, YOLO11n and RT-DETR-l. YOLOv8s surpassed both YOLOv8n and YOLO11n on the reserved test dataset ($mAP@0.5 \approx 0.968$ $mAP@0.5:0.95 \approx 0.676$ precision ≈ 0.937 recall ≈ 0.947). Although RT-DETR-l was bigger its performance was notably poorer. The training graphs revealed the reason: RT-DETR-l reached its peak prematurely and then lost stability while YOLO models showed steady improvement and stabilized after approximately 15–20 epochs.

Together these results reinforce the conclusions of the literature: transformer-based detectors generally need more precise tuning, extended training periods and often a wider variety of data to achieve their full capability while well-optimized YOLO CNN detectors presently provide the best accuracy–efficiency trade-off for real-time single-class drone identification. Regarding deployment YOLOv8s is most appropriate for server-side or more advanced setups while YOLOv8n and YOLO11n serve as options for edge devices, with limited computing resources and power. Small-object-optimized modules, multi-sensor fusion (radar/RF/thermal), domain adaptation for fog/rain/night, and integration into full multi-drone tracking + threat-assessment pipelines are the obvious next steps, as the qualitative review revealed the same hard cases that others report—tiny distant drones, poor weather/lighting, and bird-like false positives.

11. References

- [1] Saadaoui, F.Z., Cheggaga, N. & Djabri, N.E.H. Multi-sensory system for UAVs detection using Bayesian inference. *Appl Intell* 53, 29818–29844 (2023)
- [2] 2025, Ding et al. <https://www.sciencedirect.com/science/article/pii/S1270963825013823>
- [3] Liu and colleagues, 2024”<https://www.mdpi.com/2504-446X/8/11/643>
- [4] Carrio et al. (2017)
“<https://arxiv.org/search/?query=Carri%C3%B3+2017+UAV+detection+tracking+survey&searchtype=all>
- [5] Cazzato et al (2021)
<https://www.sciencedirect.com/search?qs=Cazzato%202021%20UAV%20detection%20survey%20YOLO>
O
- [6] Yurchuk et al. <https://ceur-ws.org/Vol-4035/Paper7.pdf>
- [7] Zhang (2023): <https://www.mdpi.com/2504-446X/7/8/526>
- [8]Zhao et al. 2025 : <https://arxiv.org/abs/2502.05292>

- [9] 2021 Cazzato et al.: <https://arxiv.org/abs/2010.04159>
- [10] Zhao and colleagues 2024:
https://openaccess.thecvf.com/content/CVPR2024/papers/Zhao_DETRs_Beat_YOLOs_on_Real-time_Object_Detection_CVPR_2024_paper.pdf
- [11] Wang et al., 2024 — RT-DETRv3 (hierarchical dense positive supervision)
arXiv:<https://arxiv.org/abs/2409.08475>
- [12] “Colleagues, 2025” <https://www.mdpi.com/2571-6255/8/5/170>
- [13] Liu et al., 2024 — survey on vision-based drone detection in complex environments (MDPI Drones):
<https://www.mdpi.com/2504-446X/8/11/643>
- [14] Zhang (2023) — Drone-YOLO (YOLOv8-based UAV detection tweaks)
<https://www.mdpi.com/2504-446X/7/8/526>
- [15] Nghiem et al. (2024) — LightUAV-YOLO (YOLOv8n lightweight + orthogonal feature enhancement) <https://link.springer.com/article/10.1007/s11227-024-06611-x>
- [16] “Colleagues, 2024” — lightweight YOLOv8 refinements:
<https://www.sciencedirect.com/science/article/pii/S1051200424005888>
- [17] Zhou et al. (2025) — LMWP-YOLO (long-range small-drone detection; attention + fusion + pruning) <https://www.nature.com/articles/s41598-025-95580-z.pdf>
- [18] Zheng et al. (2025) — YOLO-DD (YOLOv11n-based, boundary aggregation/fusion)
<https://link.springer.com/content/pdf/10.1186/s13634-025-01253-4.pdf>
- [19] Ding et al. (2025) — DCR-YOLO (anti-UAV; better fusion + classification head)
<https://www.sciencedirect.com/science/article/pii/S1270963825013823>
- [20] “Colleagues, 2025” — LRDS-YOLO:
- [21] Yurchuk et al., 2025 (edge/Jetson-friendly vision-based UAV detection review/comparison)
<https://ceur-ws.org/Vol-4035/Paper7.pdf>
- [22] Zhou et al., 2025 (LMWP-YOLO long-range small-drone detection; attention + multi-scale fusion + pruning) <https://www.nature.com/articles/s41598-025-95580-z.pdf>
- [23] Colleagues, 2024 (transformer-based detectors like RT-DETR; general reference)
https://openaccess.thecvf.com/content/CVPR2024/papers/Zhao_DETRs_Beat_YOLOs_on_Real-time_
- [24] Liu et al., 2024 (survey: vision-based drone detection in complex environments)
<https://www.mdpi.com/2504-446X/8/11/643>
- [25] Wang et al., 2023 (synthetic drone data using Unreal Engine + AirSim for drone detection)
<https://pubs.aip.org/aip/acp/article/2939/1/030007/2929077/Generating-synthetic-data-for-deep>
- 26] VisDrone materials (Ultralytics) <https://docs.ultralytics.com/datasets/detect/visdrone/>

- [27] Saadaoui et al. (Bayesian multi-sensor UAV detection)
<https://link.springer.com/article/10.1007/s10489-023-05027-z>
- [28] Nghiem et al., 2024 (LightUAV-YOLO; edge efficiency emphasis)
<https://link.springer.com/article/10.1007/s11227-024-06611-x>
- [29] Dedrone: https://www.dedrone.com/solutions/dedrone-rapid-response?utm_source=chatgpt.com
"DedroneRapidResponse: Multi-Layered Mobile Drone Detection Unit"
- [30] https://en.wikipedia.org/wiki/Gatwick_Airport_drone_incident?utm_source=chatgpt.com "Gatwick Airport drone incident"
- [31] https://aviationweek.com/air-transport/gatwick-drone-scare-drives-countermeasures-deployments?utm_source=chatgpt.com "Gatwick Drone Scare Drives Countermeasures Deployments"
- [32] https://www.osltechnology.com/resources/drone-detection-for-airports-lessons-learned-from-years-of-experience-in-europe?utm_source=chatgpt.com "Lessons from Europe's Busiest Airports on Drone Detection | OSL Technology"
- [33] https://www.businessinsider.com/ukrainian-startup-dropla-blue-eyes-russian-ambush-drones-artificial-intelligence-2025-10?utm_source=chatgpt.com "Ukrainian Startup Using AI to Fight Russian Ambush Drones on Roads ..."
- [34] https://www.dedrone.com/press/dedrone-launches-dedronetracker-ai-the-ai-driven-command-and-control-platform-enabling-the-complete-counterdrone-kill-chain?utm_source=chatgpt.com "Dedrone Launches DedroneTracker.AI: AI-Driven Counterdrone C2 Platform"
- [35] https://www.aboutamazon.com/news/operations/mk30-drone-amazon-delivery-packages?utm_source=chatgpt.com "How Amazon delivers packages using it's new drone MK30"
- [36] https://apnews.com/article/0b50d5176d076ce60f050ac561f7c02b?utm_source=chatgpt.com "Uber Eats will soon launch US drone delivery in partnership with Flytrex"
- [37] https://www.sciencedirect.com/science/article/pii/S0952197625017774?utm_source=chatgpt.com
"Performance evaluation of low-power and lightweight object detectors ..."
- [38] https://www.faa.gov/newsroom/faa-ends-discretionary-enforcement-policy-drone-remote-identification?utm_source=chatgpt.com "FAA Ends Discretionary Enforcement Policy on Drone Remote"
- [39] Liu et al., 2024 (survey on vision-based drone detection; notes accuracy/efficiency patterns and challenges in complex scenes) <https://www.mdpi.com/2504-446X/8/11/643>
- [40] Nghiem et al., 2024 (LightUAV-YOLO; lightweight YOLOv8-based design for UAV/edge use)
<https://link.springer.com/article/10.1007/s11227-024-06611-x>
- [41] Zhao et al., 2024 (RT-DETR / "DETRs Beat YOLOs on Real-time Object Detection", CVPR 2024)
https://openaccess.thecvf.com/content/CVPR2024/papers/Zhao_DETRs_Beat_YOLOs_on_Real
- [42] Liu et al., 2024 (vision-based drone detection survey; YOLO strengths in complex scenes)
<https://www.mdpi.com/2504-446X/8/11/643>

[43] Zhao et al., 2024 (RT-DETR / “DETRs Beat YOLOs on Real-time Object Detection”, CVPR 2024)
PDF: https://openaccess.thecvf.com/content/CVPR2024/papers/Zhao_DETRs_Beat_YOLOs_on_Rea

[44] Wang et al., 2024 (RT-DETRv3; dense positive supervision + training improvements)
arXiv: <https://arxiv.org/abs/2409.08475>

[45] Yurchuk et al., 2025 (vision-based UAV detection for small edge devices; YOLO-heavy practical focus) <https://ceur-ws.org/Vol-4035/Paper7.pdf>

[46] Zhou et al., 2025 (LMWP-YOLO / long-range small-drone detection)
<https://www.nature.com/articles/s41598-025-95580-z.pdf>

[47] Zhao et al., 2024 (RT-DETR / “DETRs Beat YOLOs on Real-time Object Detection”, CVPR 2024)
https://openaccess.thecvf.com/content/CVPR2024/papers/Zhao_DETRs_Beat_YOLOs_on_Real-time_Obj

[48] Cazzato et al., 2021 (survey overview; if you share the exact title/DOI I can give the precise link)
<https://scholar.google.com/scholar?q=Cazzato+2021+UAV+detection+survey>

- Z. Liu, et al., “Vision-Based Drone Detection in Complex Environments: A Survey,” *Drones*, 2024.
- Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, “A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles,” 2017.
- D. Cazzato, et al., “A Survey of Computer Vision Methods for 2D Object Detection,” *Applied Sciences*, 2021.
- Saadaoui, et al., “Multi-sensory System for UAV Detection Using Bayesian Inference,” 2025.
- S. Ding, et al., “DCR-YOLO: An Enhanced Anti-UAV Detection Method,” 2025.
- N. Alshaer, et al., “Vision-Based UAV Detection and Tracking Using Deep Learning,” 2025.
- S. Zhou, et al., “LMWP-YOLO: Improved YOLO for Long Range Detection of Small Drones,” 2025.
- Zheng, et al., “YOLO-DD: A Lightweight Framework for UAV Detection in Complex Environments via Boundary-Aware Fusion,” 2025.
- Y. Han, et al., “LRDS-YOLO Enhances Small Object Detection in UAV Aerial Images with a Lightweight and Efficient Design,” *Scientific Reports*, 2025.
- Z. Wang, et al., “Generating Synthetic Data for Deep Learning-Based Drone Detection,” 2023.
- U.S. Federal Aviation Administration (FAA), “Remote Identification of Drones” and related guidance, 2023–2025.
- Amazon Prime Air, public regulatory documents and news reports on MK-30 drone delivery operations in the U.S. and Europe, 2022–2025.
- “Zipline and Amazon Battle It Out for Drone Delivery,” *Barron’s* and related coverage, 2024.
- Associated Press, “Uber Eats Will Launch U.S. Drone Delivery in Partnership with Flytrex,” 2025.
- AWS Machine Learning Blog, “Using AI and Drones to Simplify Infrastructure Inspections,” 2025.
- KritiKal Solutions, “AI-Powered Drone Security Systems: Smarter Surveillance,” 2025.
- Business Insider, “A Ukrainian Startup is Using AI to Outsmart Russian Ambush Drones,” describing Dropla’s Blue Eyes system, 2025.
- Systematic reviews and surveys on multi-sensor UAV detection and counter-UAS systems (e.g., multi-sensor fusion of radar, EO/IR, RF, and acoustics), 2023–2025.

- Foggy Drone Teacher and related works on domain adaptation for drone detection under foggy and adverse-weather conditions, 2023–2024.
- Recent works on multi-UAV detection, tracking, and intelligent multi-UAV systems for surveillance and disaster response (e.g., GL-DT, cooperative multi-UAV frameworks), 2023–2025.