

Deep Neural Network Architectures for Real-Time Drone Detection in Aerial Surveillance Systems: A Comparative Evaluation of YOLOv7, YOLOv8, and YOLOv11

Nazerke Abylay¹, Elvira Kadylbekkyzy², Bakhytzhan Kulambayev³,
Bayan Moldakalykova⁴, Zharkyn Bimoldina⁵, Saken Mambetov^{6*}, Kuanysh Dossanbek⁷

Department of Telecommunication Engineering, Energo University, Almaty, Kazakhstan^{1,2}

Higher School of Telecommunications, Turan University, Almaty, Kazakhstan^{1,3}

Higher School of Information Technologies, Turan University, Almaty, Kazakhstan^{4,5,6}

Department of Computer Engineering, International IT University, Almaty, Kazakhstan⁷

Abstract—The proliferation of consumer drones has raised security concerns for critical infrastructure, airports, and urban surveillance, creating a need for reliable real-time detection. This study presents a controlled comparative evaluation of selected YOLO generations — YOLOv7x, YOLOv8x, and YOLOv11x — for single-class drone detection under matched high-level settings: 640 by 640 input, 50 epochs, the AdamW optimizer at an initial learning rate of 0.001, and an effective optimizer batch size of 64. Experiments used the YOLO Drone Detection Dataset (1,012 training and 347 validation images, 369 validation instances). All models were scored with one evaluation implementation applied to raw predictions, removing differences between framework-specific validators. Across five independent training runs, YOLOv8x achieved the highest mean average precision at an IoU threshold of 0.5, 0.912 with a standard deviation of 0.007, ahead of YOLOv11x at 0.892 (0.004) and YOLOv7x at 0.708 (0.008); a Welch t-test gives a p-value of 0.0012 for YOLOv8x against YOLOv11x. A provenance analysis found that 32 of the 347 validation images are near-duplicates of training images; re-scoring on the 315 remaining images lowers every result but preserves the ordering. Zero-shot evaluation on the independently collected DUT Anti-UAV test set, whose targets are predominantly small, reproduces the same ordering while absolute accuracy falls to 0.346 for the best model. Detection accuracy therefore does not increase monotonically with architectural generation, and accuracy measured on this dataset does not transfer to imagery dominated by small targets. YOLOv11x was the smallest model (56.8 million parameters) with the lowest peak inference memory, indicating a trade-off between accuracy and deployment cost.

Keywords—UAV detection; drone detection; YOLO; deep learning; real-time object detection; comparative evaluation

I. INTRODUCTION

The rapid increase in the number of unmanned aerial vehicles (UAVs) used in civil, commercial, and security applications has raised growing concerns regarding safety, privacy, and airspace management [1]. Drones can be weaponized, used for unauthorized surveillance, or inadvertently interfere with manned aviation, making real-time autonomous detection a critical research and operational

requirement [2]. The global drone market is projected to exceed USD 55 billion by 2028, intensifying the need for robust counter-UAV systems capable of operating under strict computational and latency constraints [3].

Traditional UAV detection approaches based on radar, acoustic sensors, and RF signal analysis face inherent limitations, including small target radar cross-sections, high false-positive rates from non-drone aerial objects, and vulnerability to complex electromagnetic environments [4]. Time-frequency CNN-based RF detection has demonstrated promising performance, although its effectiveness decreases at longer operating ranges [5]. Combining surveillance radar with YOLO-based classification can improve detection accuracy, but at the expense of increased system complexity associated with multi-sensor architectures [6]. Recent advances in deep learning, particularly convolutional neural networks (CNNs), have facilitated a shift toward visual and multimodal detection approaches with substantially improved discriminative performance [7]. Machine learning approaches have also been investigated for distinguishing UAVs from visually similar aerial objects such as birds, highlighting the importance of reducing false alarms in automated airspace surveillance systems [8].

This study contributes a controlled benchmarking protocol for comparing generations of YOLO detectors under matched high-level settings; a comparison covering detection accuracy, operating-point behaviour, robustness under image degradation, latency and throughput, with all models scored by one evaluation implementation; an analysis of training behaviour and computational efficiency under a fixed epoch budget; and practical recommendations for detector selection in real-time drone surveillance.

The evolution of object detection architectures from two-stage detectors such as R-CNN [9] and Faster R-CNN [10] to single-stage frameworks including SSD [11] and the YOLO family reflects the field's pursuit of higher accuracy at lower computational cost. For UAV detection, where detection latency

*Corresponding author

directly impacts response effectiveness and onboard power is constrained, this efficiency imperative is particularly acute [12].

Comprehensive reviews of deep learning methods for UAV detection indicate that YOLO-based architectures provide an effective balance between detection accuracy and computational efficiency [13]. The evolution of YOLO architectures from YOLOv1 to YOLOv8 demonstrates a continuous progression in feature extraction, optimization strategies, and detection performance [14]. A review of YOLO-based UAV detection technologies identified multi-scale feature fusion and attention mechanisms as the most influential architectural enhancements for improving detection accuracy [15]. A lightweight UAV detection framework based on GhostConv modules achieved a 40–60% reduction in FLOPs while maintaining competitive detection performance [16]. An efficient lightweight detection network demonstrated a processing speed of 682 FPS while reducing model parameters by 88.5% for UAV imagery analysis [17].

A lightweight object detection network optimized for UAV vision platforms was developed to improve deployment efficiency on resource-constrained devices [18]. CNN-based target detection has also been investigated for UAV reconnaissance imagery, where Faster R-CNN combined with optimized region proposals and image enhancement was used to improve detection under complex backgrounds and varying illumination [19]. An edge-oriented UAV detection framework achieved a favorable balance between accuracy and computational efficiency through an optimized backbone architecture [20]. A lightweight detection network achieved state-of-the-art efficiency for UAV object detection while maintaining competitive accuracy [21].

An enhanced YOLOv8-based architecture reduced the number of parameters by approximately 50% while maintaining an mAP@0.5 value of 98.4% [22]. A lightweight DRBD-YOLOv8 framework achieved higher detection accuracy with approximately half the parameters of YOLOv8n [23]. SEB-YOLOv8s enhanced unauthorized UAV detection through the incorporation of spatial attention mechanisms [24]. Small target-YOLOv5 improved drone aerial imagery analysis using specialized feature enhancement modules [25]. An optimized YOLOv7 framework for small-target detection in UAV aerial imagery combined anchor-box clustering, enhanced multi-scale feature fusion, and a refined detection head to improve detection efficiency on the VisDrone2019 dataset [26]. An improved YOLOv8n-based framework demonstrated competitive performance for lightweight tiny-object detection on the VisDrone2019 dataset [27]. A dual-branch YOLOv8 architecture was further proposed for small-object detection in UAV imagery, integrating enhanced feature extraction and multi-scale feature fusion to improve detection performance on the VisDrone benchmark [28].

Attention mechanisms have emerged as an effective strategy for improving UAV detection performance. EDGS-YOLOv8 combined EMA attention with Ghost convolutions and achieved 97.1% AP with a model size of only 4.23 MB [29]. A modified YOLOv8 architecture demonstrated effective multi-class detection of uncooperative aerial vehicles through attention-enhanced feature extraction [30]. Light-YOLOv5 integrated

transformer components and PEAM attention to improve performance on resource-constrained camera platforms [31]. YOLO-LE introduced adaptive feature fusion modules to enhance multi-scale feature sensitivity [32].

Research on hardware-oriented deployment has also progressed considerably. FPGA-based implementations of YOLOv3 and YOLOv8 enabled low-power onboard drone classification [33]. Dedicated FPGA accelerators have been developed to support real-time UAV object detection [34]. Lightweight small-target detection algorithms have been optimized for UAV platforms operating under strict power constraints [35]. LW-YOLOv8 combined CSP-CNN-Transformer fusion with SiLU loss to improve edge-device deployment efficiency [36].

Recent developments have increasingly focused on transformer-hybrid architectures. A Vision Transformer framework incorporating hierarchical attention and dynamic token pruning achieved linear-complexity attention for UAV edge computing applications [37]. LMWP-YOLO improved long-range small-drone detection through multidimensional collaborative attention and network pruning [38]. YOLO-DD employed boundary-aware feature fusion based on YOLOv11 to improve UAV detection in complex environments [39]. FFG-YOLO achieved lightweight UAV detection with only 1.77 million parameters and 5.7 GFLOPs [40]. EDNet introduced scalable model variants and reported a 5.6% improvement in mAP on mobile platforms [41]. A lightweight anti-drone YOLOv8 framework reduced the parameter count by 9.6 million while maintaining competitive detection performance [42]. Evaluation of YOLOv8 on distorted surveillance videos reported mAP values of 92.3% for visible-spectrum imagery and 97.9% for infrared imagery, demonstrating the effectiveness of deep learning-based detectors in multimodal surveillance environments [43]. Recent advances in UAV-based systems increasingly combine edge intelligence, distributed sensing, and machine learning to support autonomous monitoring and real-time data classification [44]. Beyond object detection, recent research has investigated multi-UAV surveillance systems that integrate aerial image stitching to enhance large-scale monitoring and situational awareness capabilities [45].

Despite significant progress in UAV detection research, most studies evaluate a single architecture or use different datasets and experimental configurations. As a result, it remains difficult to compare the performance of different detector generations objectively. A controlled cross-generational comparison under matched training settings and a single evaluation implementation is still lacking in the literature.

Single-stage detectors have been applied to drone detection across successive YOLO generations: Singha and Aydin demonstrated automated drone detection with YOLOv4 [46], Aydin and Singha extended this to YOLOv5 [47], and Hakani and Rawat reported real-time YOLOv9 detection deployed on an NVIDIA Jetson Nano edge device [48], underscoring the practical importance of on-device throughput. The three variants evaluated in the present study span the architectural transition from anchor-based detection (YOLOv7x) to anchor-free decoupled-head detection (YOLOv8x and YOLOv11x) and correspond to three stable, officially maintained extra-large

releases; such stable releases are a prerequisite for applying one training specification across all three. YOLOv9 and YOLOv10 are recent and promising architectures; their evaluation under the same conditions is a direct extension of this benchmark.

To the best of our knowledge, no prior study evaluates YOLOv7x, YOLOv8x and YOLOv11x under one set of matched high-level settings — identical dataset partition, input resolution, optimizer, schedule and hardware — with all three scored by a single evaluation implementation, while jointly reporting detection accuracy, operating-point behaviour, robustness under image degradation and computational cost. The present study is therefore positioned as a controlled cross-generational benchmark rather than a new detector. Its contribution is a like-for-like comparison that holds the training and data factors fixed as far as separately maintained implementations allow, and reports what remains unmatched (Section II-F), yielding a reproducible basis for model selection in real-time UAV surveillance.

II. MATERIALS AND METHODS

A. Dataset

All experiments used the YOLO Drone Detection Dataset [49], a public benchmark from Kaggle, with its predefined partition, which we did not re-partition: 1,012 training images containing 1,117 annotated drone instances and 347 validation images containing 369 instances. Annotations follow the standard normalized YOLO format for a single class (“drone”). Image resolution varies widely: 150–5,760 px wide by 160–3,840 px high in the training split and 200–6,000 by 159–4,000 px in the validation split. Instance size, computed from each annotation's absolute box area (the normalized label multiplied by that image's own pixel dimensions) and binned by the COCO convention, is strongly skewed towards large objects: in the validation split 13 instances (3.5 %) are small ($< 32^2$ px), 30 (8.1 %) medium (32^2 – 96^2 px) and 326 (88.3 %) large ($> 96^2$ px); the training split contains 1 small, 48 medium and 1,068 large instances. These figures are computed by the released `prepare_data.py` script rather than estimated. Images capture diverse UAV types under varying conditions, including daylight, overcast, different altitudes, and urban and rural backgrounds.

The dataset is publicly available on Kaggle [49]. At the time of access, no explicit license statement was provided on the dataset page. Therefore, the authors do not redistribute the dataset or reproduce any original dataset images in this article.

B. Model Architectures

Three extra-large (x) variants from consecutive YOLO generations were evaluated. Table I summarizes their architectural characteristics.

The three variants were selected as widely deployed milestone releases that span the transition from anchor-based to anchor-free detection, each having a stable, officially maintained extra-large variant at the time the experiments were run. The study is therefore a comparison of selected YOLO generations rather than an exhaustive one: YOLOv9 and YOLOv10 were not evaluated. A further consequence of using released checkpoints is that each architecture is initialized from

weights produced by its own upstream training pipeline, so initialization is confounded with architecture; this is discussed in the Limitations.

TABLE I. ARCHITECTURAL CHARACTERISTICS OF EVALUATED MODELS

Model	Backbone	Neck	Detection Head	Params (M)	GFLOPs
YOLOv7x	E-ELAN	PAFPN	Anchor-based (IDetect)	70.8	188.0
YOLOv8x	CSPDarknet53+C2f	PAN-FPN	Anchor-free (Decoupled)	68.1	257.4
YOLOv11x	C3k2	SPPF+PAN	Anchor-free (Decoupled)	56.8	194.8

C. Training Configuration

Table II details the training configuration, which is identical across all three models at the level of these settings. Section II-F documents the implementation-level differences that this level of matching cannot remove. Where an effective optimizer batch size of 64 is reported, it refers to the batch seen by the optimizer update only: gradients are accumulated from physical batches of 8, so batch-normalization statistics are computed over 8 samples and augmentation is sampled per physical batch. The two are not equivalent, and no claim of equivalence is made.

TABLE II. TRAINING HYPERPARAMETERS (IDENTICAL ACROSS ALL MODELS)

Parameter	Value
Input resolution	640 × 640 (letterbox)
Batch size	8 per step, accumulated to a nominal batch of 64
Epochs	50
Optimizer	AdamW (lr = 0.001)
Augmentation	Mosaic, HSV jitter, scale ± 0.5 , flip 0.5
Pre-trained weights	COCO (respective variant)
Confidence threshold	0.25
IoU threshold (NMS)	0.5

D. Experimental Setup

All experiments were conducted in a single standardized hardware and software environment so that performance differences are attributable to the models rather than to the experimental conditions. The computational platform was an NVIDIA GeForce RTX 5070 Ti Laptop GPU with 12 GB of VRAM, running Windows 10, Python 3.14.2, PyTorch 2.11.0, and CUDA 12.8. All models were trained and evaluated with the same data split, optimization settings, learning-rate schedule, and stopping criterion. Because the extra-large variants exceed 12 GB of VRAM at a physical batch of 16, training used a physical batch of 8 with gradient accumulation to an effective optimizer batch size of 64. A sensitivity study over accumulation factors was not performed, since a physical batch of 64 does not fit on this device at this resolution; this is recorded in the Limitations.

E. Reproducibility

YOLOv8x and YOLOv11x were trained with the Ultralytics framework (v8.4.117); YOLOv7x used the official implementation (github.com/WongKinYiu/yolov7, commit a207844b1ce82d204ab36d87d496728d3d2348e7). The software environment was Python 3.14.2, PyTorch 2.11.0+cu128, CUDA 12.8, NumPy 2.4.4, and OpenCV 5.0.0.93; every dependency is pinned in requirements.txt. Training started from the official COCO-pretrained weights: yolov7x.pt, yolov8x.pt and yolo11x.pt. The optimizer was AdamW with an initial learning rate of 1e-3, linear warm-up over the first three epochs, and cosine decay to 1e-5 over 50 epochs. The warm-up bias learning rate is set to 0.0: Ultralytics zeroes this parameter only when the optimizer is selected automatically, and leaving it at its default of 0.1 drives the validation loss to NaN within the first epoch under AdamW, so the protocol is not runnable without specifying it. The reported weights are the best validation checkpoint of each run, not the final epoch. The primary run of Table III uses seed 0, which fixes Python, NumPy, and PyTorch; the stability analysis of

Table V uses five independent retrains per model at seeds 0, 1, 2, 3, and 4, and the individual results of every run are reported there. The exact training and validation commands, configuration files, environment specification, per-run result files, and trained weights are available at <https://github.com/sakenski-afk/drone-detection-benchmark> and archived at Zenodo (<https://doi.org/10.5281/zenodo.21955344>), with the trained weights provided as release assets in the repository and SHA-256 checksums linking each checkpoint to the results computed from it.

The 50-epoch schedule is a fixed training budget applied identically to all three models rather than a demonstrated convergence point. As Fig. 1 shows, validation mAP@0.5 was still rising at the end of the schedule for every model, by +0.028 (YOLOv7x), +0.023 (YOLOv8x) and +0.021 (YOLOv11x) over the final ten epochs, with best checkpoints at epochs 35, 44 and 39 respectively. The comparison is therefore valid as a like-for-like comparison at a fixed budget, while absolute values would likely improve under a longer schedule.

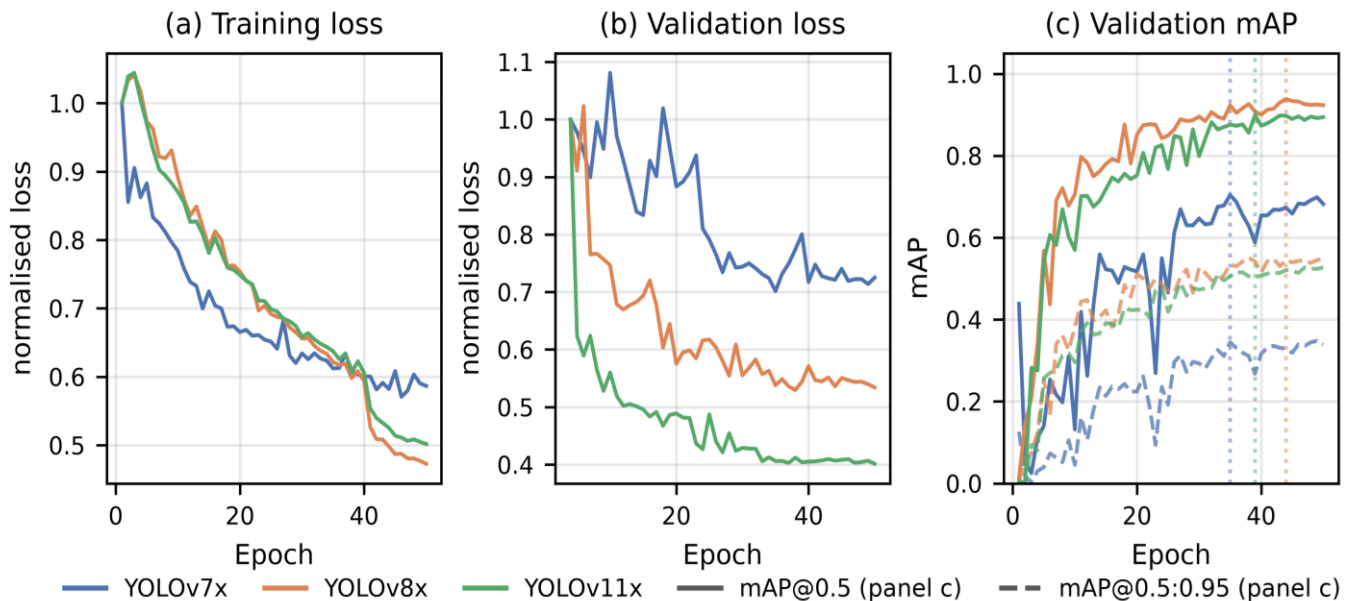


Fig. 1. Training loss, validation loss and validation mAP per epoch for the three architectures (seed 0). The two frameworks report different loss terms, so losses are normalized and only their shapes are comparable: training loss to its epoch-1 value, validation loss to its epoch-4 value, the first three epochs being warm-up. Dotted vertical lines in panel (c) mark the best validation checkpoint of each model. Validation mAP@0.5 is still rising at epoch 50 for all three architectures.

F. Implementation-Level Differences Between the Frameworks

Matching a protocol across separately maintained implementations is possible at the level of settings but not at the level of pipelines, and the residual differences are not hypothetical. The YOLOv7 repository provides no AdamW option: its --adam flag constructs torch.optim.Adam, so obtaining the optimizer of Table II required patching that call site, and the patch is recorded in the released repository. The loss functions differ in kind: YOLOv7x uses OTA assignment with objectness, box, and classification terms, while YOLOv8x and YOLOv11x use a decoupled anchor-free head with distribution focal loss, so loss values are not comparable across frameworks. Augmentation, exponential moving average, and pre- and post-

processing defaults also differ where the nominal hyperparameter names coincide. The one axis on which the comparison is genuinely identical is evaluation: all three models are scored by a single implementation applied to raw predictions, verified against the Ultralytics average-precision computation in the released test suite. We therefore describe this study as a comparison under matched high-level settings rather than under an identical training pipeline.

G. Evaluation Metrics

Performance was evaluated using precision, recall, F1-score and mean average precision at IoU = 0.5 (mAP@0.5) and IoU = 0.5:0.95 (mAP@0.5:0.95). All metrics for all three models are computed by a single implementation applied to raw predictions, so that the comparison does not inherit differences between the

frameworks' own validators; the average-precision computation is verified against the Ultralytics implementation to floating-point equality in the released test suite. Predictions are matched to ground truth greedily in descending confidence order, a match requiring $\text{IoU} \geq 0.5$. Precision, recall, F1, and the confusion matrices are all reported at the operating confidence threshold of 0.25: true positives are matched predictions, false negatives are ground-truth instances left unmatched, so that $\text{TP} + \text{FN}$ equals the 369 annotated validation instances exactly, and false positives are predictions matched to no instance. The background-background cell of the confusion matrix is undefined and is reported as such: an image contains an unbounded number of candidate windows, so object detection provides no finite set of negatives. For the same reason, ROC-AUC is not reported, since a false-positive rate would require a denominator the data does not determine; free-response ROC (FROC) is reported instead, plotting sensitivity against false positives per image, a rate whose denominator is the number of evaluation images. Robustness was assessed over five independent applications of each degradation to the trained model: additive Gaussian noise with zero mean and $\sigma = 0.05$ in normalized intensity, linear motion blur with a 15 px kernel at a random orientation, and low-light simulation scaling pixel intensities by 0.4 with $\gamma = 1.5$. This resamples noise realizations only and involves no retraining; it is distinct from the retraining variability reported as the standard deviation over five seeds.

III. RESULTS

A. Overall Detection Performance

Table III presents the detection performance of the evaluated models on the validation partition. YOLOv8x achieved the highest detection performance, with a precision of 0.895, recall of 0.902, F1-score of 0.899, $\text{mAP}@0.5$ of 0.921, and $\text{mAP}@0.5:0.95$ of 0.540, ahead of YOLOv11x (0.874, 0.846, 0.860, 0.891, and 0.507). YOLOv7x under the shared protocol reached $\text{mAP}@0.5 = 0.716$; retrained with its own repository's SGD recipe, it reached 0.853, and both are reported because the difference is a property of the matched protocol rather than of the architecture (Section II-F). Precision and recall are taken at the operating threshold of 0.25, consistent with the confusion matrices; Table VII shows that the reported ordering is unchanged when each model is instead scored at its own best-F1 threshold. Because the last-epoch checkpoint involves no selection, the mean of the per-seed differences between it and the reported best-validation checkpoint bounds the optimism introduced by selecting on this partition. Averaged over the five seeds that difference is +0.0056 for YOLOv7x, +0.0058 for YOLOv8x, and -0.0011 for YOLOv11x $\text{mAP}@0.5$; each figure is the mean of the per-seed differences computed before rounding, and the individual runs are reported in the released result files.

TABLE III. COMPARATIVE DETECTION PERFORMANCE ON THE VALIDATION SET (347 IMAGES, 369 INSTANCES)

Model	Precision	Recall	F1-score	$\text{mAP}@0.5$	$\text{mAP}@0.5:0.95$	Latency (ms)	FPS
YOLOv7x	0.805	0.672	0.733	0.716	0.357	16.12	62.1
YOLOv8x	0.895	0.902	0.899	0.921	0.540	17.78	56.2
YOLOv11x	0.874	0.846	0.860	0.891	0.507	16.39	61.0
YOLOv7x (native SGD)	0.774	0.799	0.787	0.853	0.450	16.12	62.1

All values are on the validation partition (347 images, 369 instances). Precision, recall, and F1 at the operating confidence threshold of 0.25 with $\text{IoU} \geq 0.5$ matching. Lat. = inference latency, measured on the GPU of Section II-D and not comparable with other hardware. The final row reports YOLOv7x trained with its own repository's SGD recipe instead of the shared AdamW protocol (Section II-F).

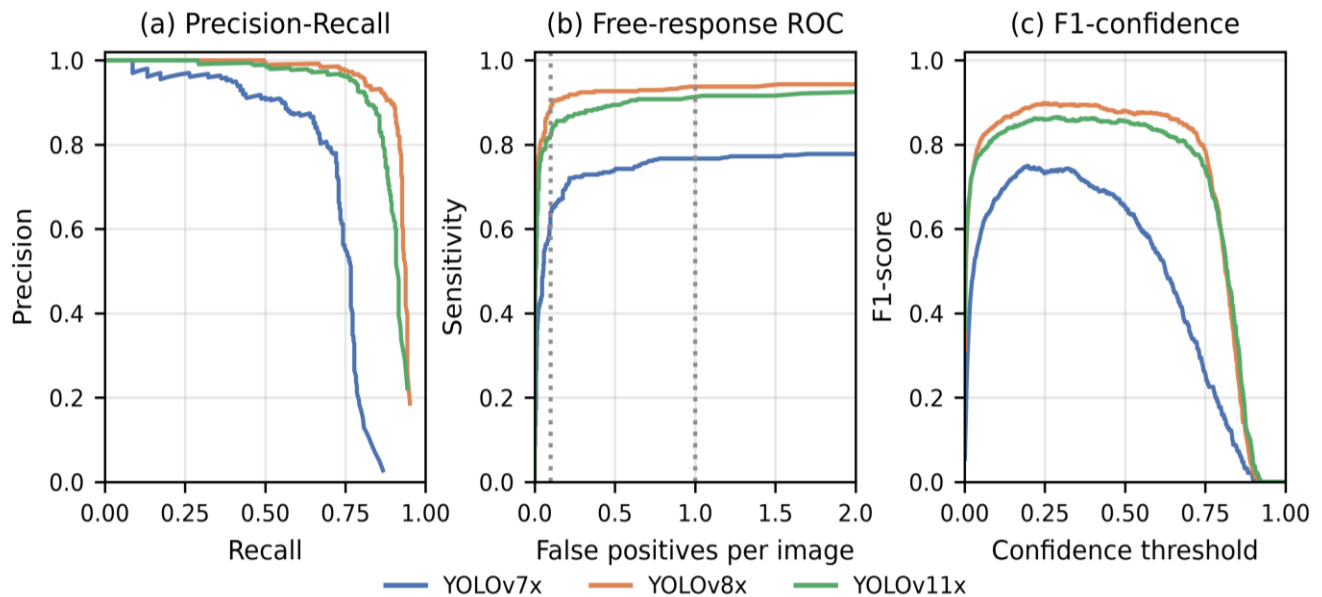


Fig. 2. Precision-recall curves, free-response ROC (FROC) curves, and F1-confidence curves.

B. Precision–Recall, FROC and F1 Analysis

Fig. 2 shows the precision–recall curves, the free-response ROC curves, and the F1-confidence curves for the three architectures. YOLOv8x attains higher precision across most recall levels and the highest sensitivity at every false-positive-per-image operating point examined. At 0.1, 0.25, 0.5, 1.0 and 2.0 false positives per image its sensitivity is 88.9 %, 91.9 %, 92.7 %, 93.8 % and 94.3 %, against 82.1 %, 87.3 %, 89.4 %, 91.3 % and 92.4 % for YOLOv11x and 64.0 %, 72.4 %, 74.0 %, 76.7 % and 77.8 % for YOLOv7x. The F1-confidence curves peak at similar confidence values for the two anchor-free models, indicating comparable confidence calibration.

C. Confusion Matrix Analysis

Fig. 3 presents the confusion matrices at the operating threshold of 0.25. YOLOv8x achieved the highest number of true positives (333 of 369 instances) with 39 false positives, against 312 true positives and 45 false positives for YOLOv11x and 248 true positives and 60 false positives for YOLOv7x. In every matrix, the true positives and false negatives sum to the 369 annotated validation instances; the background–background cell is undefined for object detection and is marked as such rather than filled with a count.

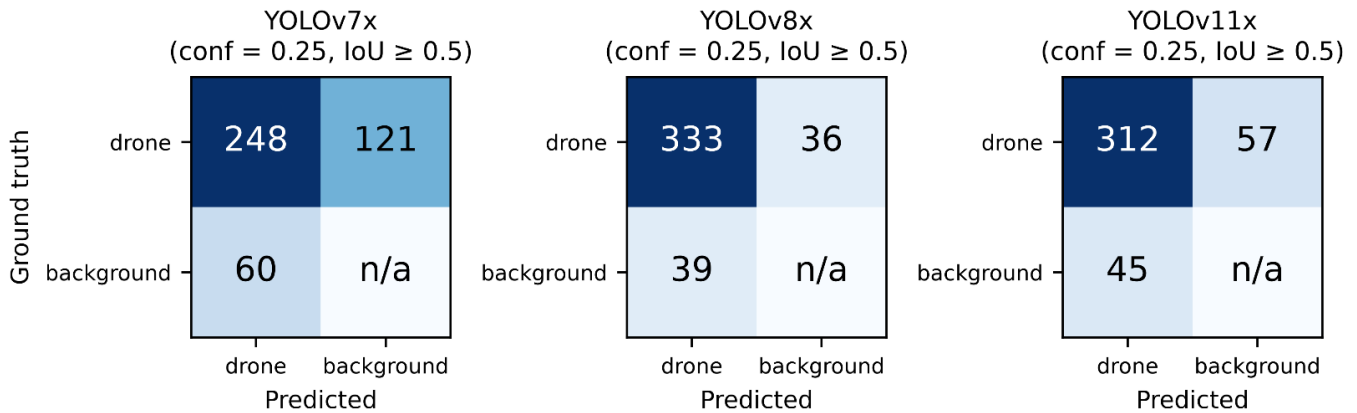


Fig. 3. Confusion matrices for each model on the validation partition, at the operating confidence threshold of 0.25 with $\text{IoU} \geq 0.5$ matching. True positives and false negatives sum to the 369 annotated instances in every matrix. The background – background cell is undefined for object detection and is marked as such.

D. Accuracy–Speed Trade-off

The trade-off between detection accuracy and inference speed is illustrated in Fig. 4. YOLOv8x attains the highest $\text{mAP}@0.5$ but the lowest throughput of the three, consistent with its higher computational cost, while YOLOv11x sits between the two on accuracy with the smallest parameter count. No single model dominates the accuracy–efficiency frontier on this dataset, and the ordering by accuracy does not follow the ordering by architectural generation.

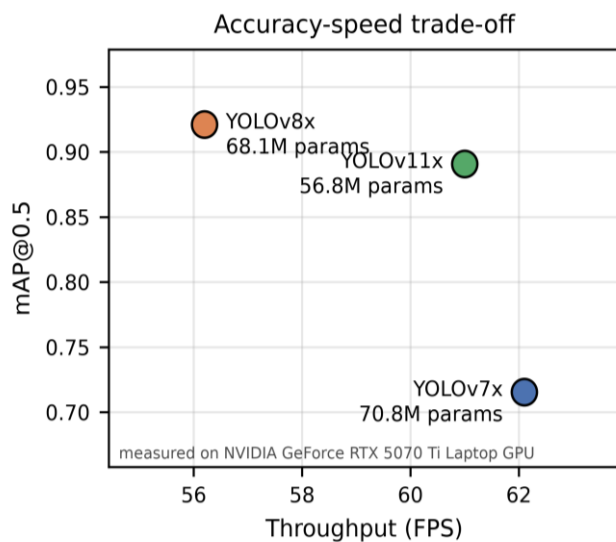


Fig. 4. Accuracy–speed trade-off ($\text{mAP}@0.5$ vs FPS).

E. Computational Efficiency

Table IV summarizes the computational complexity of the evaluated models, measured on the single-class ($nc = 1$) fused models. YOLOv11x required 56.8 million parameters, a 19.8 % reduction relative to YOLOv7x, and is the smallest of the three, while YOLOv8x carries the highest computational cost at 257.4 GFLOPs. Latency and throughput were measured on the GPU described in Section II-D as the mean of 100 forward passes at batch size 1 after 10 warm-up iterations, and track the GFLOPs ordering. These measurements are hardware-bound and are not comparable with figures obtained on other devices.

TABLE IV. COMPUTATIONAL EFFICIENCY COMPARISON

Model	Params(M)	GFLOPs	Layers	Lat.(ms)	FPS
YOLOv7x	70.8	188.0	362	16.12	62.1
YOLOv8x	68.1	257.4	113	17.78	56.2
YOLOv11x	56.8	194.8	191	16.39	61.0

F. Stability and Robustness Analysis

Reliability was assessed in two ways that do not carry the same evidential weight, and are therefore reported separately. Table V gives the mean and standard deviation of $\text{mAP}@0.5$ over five independent training runs, which differ only in random seed. Table VI gives the mean over five independent applications of each image degradation to the single seed-0 model; that procedure resamples noise realizations and blur orientations without retraining, and so says nothing about training variability. Across the five seeds, the ranges of

YOLOv8x (0.9035–0.9210) and YOLOv11x (0.8861–0.8968) do not overlap, so the ordering between them is not an artefact of a single seed.

TABLE V. STABILITY ACROSS FIVE INDEPENDENT TRAINING RUNS

Model	Baseline mAP@0.5 (seed 0)	Mean $\pm$ SD over 5 independent training runs
YOLOv7x	0.716	0.708 $\pm$ 0.008
YOLOv8x	0.921	0.912 $\pm$ 0.007
YOLOv11x	0.891	0.892 $\pm$ 0.004
<i>Stability of the evaluated models. Each value is mAP@0.5 on the validation partition from an independent training run; the five runs differ only in random seed.</i>		

TABLE VI. ROBUSTNESS UNDER IMAGE DEGRADATION (SINGLE MODEL, FIVE DEGRADATION REALIZATIONS)

Model	Baseline mAP@0.5	Gaussian noise	Motion blur	Low light
YOLOv7x	0.716	0.598	0.645	0.604
YOLOv8x	0.921	0.906	0.769	0.853
YOLOv11x	0.891	0.828	0.783	0.858
<i>Each value is the mean over five applications of the degradation to the seed-0 model. No retraining is involved, so these columns carry no information about training variability.</i>				

The robustness results are consistent with the detection performance reported above: the anchor-free models retain a higher mAP@0.5 than YOLOv7x under all three degradations, and at the operating threshold they produce fewer false positives (39 for YOLOv8x and 45 for YOLOv11x against 60 for YOLOv7x).

G. Sensitivity to the Operating Threshold

Precision, recall, and F1 in Table III are reported at a single confidence threshold of 0.25 shared by all models, which need not coincide with any one model's best operating point. Table VII therefore scores each model both at the shared threshold and at its own best-F1 threshold. The shared threshold happens to sit almost exactly at YOLOv8x's optimum and costs YOLOv7x 0.017 and YOLOv11x 0.006 F1, but the ordering is identical under both, and under mAP@0.5 and mAP@0.5:0.95, which integrate over the confidence sweep and cannot depend on the choice of operating point at all.

TABLE VII. DETECTION PERFORMANCE AT THE SHARED AND AT EACH MODEL'S OWN BEST-F1 THRESHOLD

Model	F1 at 0.25	Best F1	At threshold	Cost of sharing	mAP@0.5
YOLOv7x	0.733	0.749	0.199	-0.017	0.716
YOLOv8x	0.899	0.899	0.251	-0.000	0.921
YOLOv11x	0.860	0.865	0.289	-0.006	0.891
<i>The ordering YOLOv8x > YOLOv11x > YOLOv7x holds under all four criteria.</i>					

H. Independence of the Validation Partition

The train and validation partitions are those distributed with the dataset and were not created by us, so their independence cannot be assumed. Three signals were used to test it: an exact hash of the file bytes, a 64-bit perceptual hash, and cosine

similarity between ImageNet-pretrained ResNet-50 features. An ImageNet model is used deliberately, since features from a model trained on this dataset would beg the question. Table VIII reports the result.

The partitions are not independent. 15 validation images are byte-identical to training images, and several further pairs have a perceptual-hash distance of zero with a cosine similarity of 1.000. Taking the union of the three signals, 32 of 347 validation images (9.2 %) cannot be treated as independent evidence.

Every model was therefore re-scored on the 315 remaining images (Table IX). All results fall by 0.011 to 0.027 mAP@0.5, so the figures obtained on the full partition are optimistic. The ordering is unaffected: the per-seed ranges remain disjoint, YOLOv8x spanning 0.8923–0.9105 against 0.8729–0.8857 for YOLOv11x, and the difference remains significant (Welch t-test, $p = 0.00117$). The contamination inflated the measurements; it did not produce the result. The list of affected images is released with the code.

TABLE VIII. SIGNALS OF CONTAMINATION BETWEEN THE TRAINING AND VALIDATION PARTITIONS

Signal	Criterion	Validation images
Exact duplicate	identical SHA-256 of the file bytes	15
Perceptual hash	64-bit DCT hash within 5 bits	32
Embedding similarity	ResNet-50 cosine at least 0.95	20
Union of all three	—	32 of 347
<i>Each row counts validation images matching at least one training image under that criterion.</i>		

TABLE IX. DETECTION PERFORMANCE RE-SCORED ON THE LEAKAGE-FREE SUBSET

Model	Full partition (347 images)	Leakage-free (315 images)	Difference
YOLOv7x	0.7079 $\pm$ 0.0082	0.6813 $\pm$ 0.0089	+0.0266
YOLOv8x	0.9115 $\pm$ 0.0067	0.9002 $\pm$ 0.0075	+0.0113
YOLOv11x	0.8924 $\pm$ 0.0042	0.8790 $\pm$ 0.0052	+0.0134
<i>Mean $\pm$ standard deviation of mAP@0.5 over five independent training runs.</i>			

I. Cross-Dataset Generalization

Whether the ordering established above is a property of the architectures or of this dataset can only be decided on data the models have not seen. Every trained checkpoint — three architectures, five seeds each — was therefore evaluated, without retraining or fine-tuning, on the detection test split of the DUT Anti-UAV dataset [50]. That set was chosen because its target-scale distribution is close to the inverse of ours (Table X), which makes it a test of exactly the regime this dataset cannot represent. A perceptual-hash check confirms it shares no near-duplicate imagery with our training partition.

The ordering transfers unchanged (Table XI): YOLOv8x leads YOLOv11x, and YOLOv7x remains far behind, with the difference between the two leading models significant over the

five runs (Welch t-test, $p = 0.00206$, Cohen's $d = 2.83$). Unlike in our own data, the per-seed ranges overlap slightly here, so the lead is significant on means rather than disjoint run by run.

TABLE X. INSTANCE-SIZE DISTRIBUTION OF THE EVALUATION SETS

Instance size (COCO area convention)	This study (validation)	DUT Anti-UAV (test)
Small, below 32^2 px	13 (3.5 %)	848 (37.8 %)
Medium, 32^2 – 96^2 px	30 (8.1 %)	836 (37.2 %)
Large, above 96^2 px	326 (88.3 %)	561 (25.0 %)
Images/instances	347 / 369	2200 / 2245
<i>Sizes follow the COCO area convention, computed from each annotation against its own image dimensions.</i>		

Absolute accuracy does not transfer. The best model retains 38 % of its in-dataset $mAP@0.5$ and YOLOv7x only 19 %, on a set where 37.8 % of instances are small against 3.5 % here. Fig. 5 places the three evaluation sets side by side: removing the

contaminated images lowers every model slightly, while the change of domain lowers them by more than half, and the ordering survives both. Accuracy measured on this dataset therefore predicts the ranking of these architectures on other data, but not their accuracy on it, and in particular says little about performance on small and distant drones.

TABLE XI. ZERO-SHOT CROSS-DATASET PERFORMANCE, $mAP@0.5$

Model	This dataset	DUT Anti-UAV (zero-shot)	Retained
YOLOv7x	0.708 ± 0.008	0.132 ± 0.020	19 %
YOLOv8x	0.912 ± 0.007	0.346 ± 0.007	38 %
YOLOv11x	0.892 ± 0.004	0.327 ± 0.007	37 %
<i>Five independent training runs per model in both columns. Zero-shot means no retraining or fine-tuning on the external set.</i>			

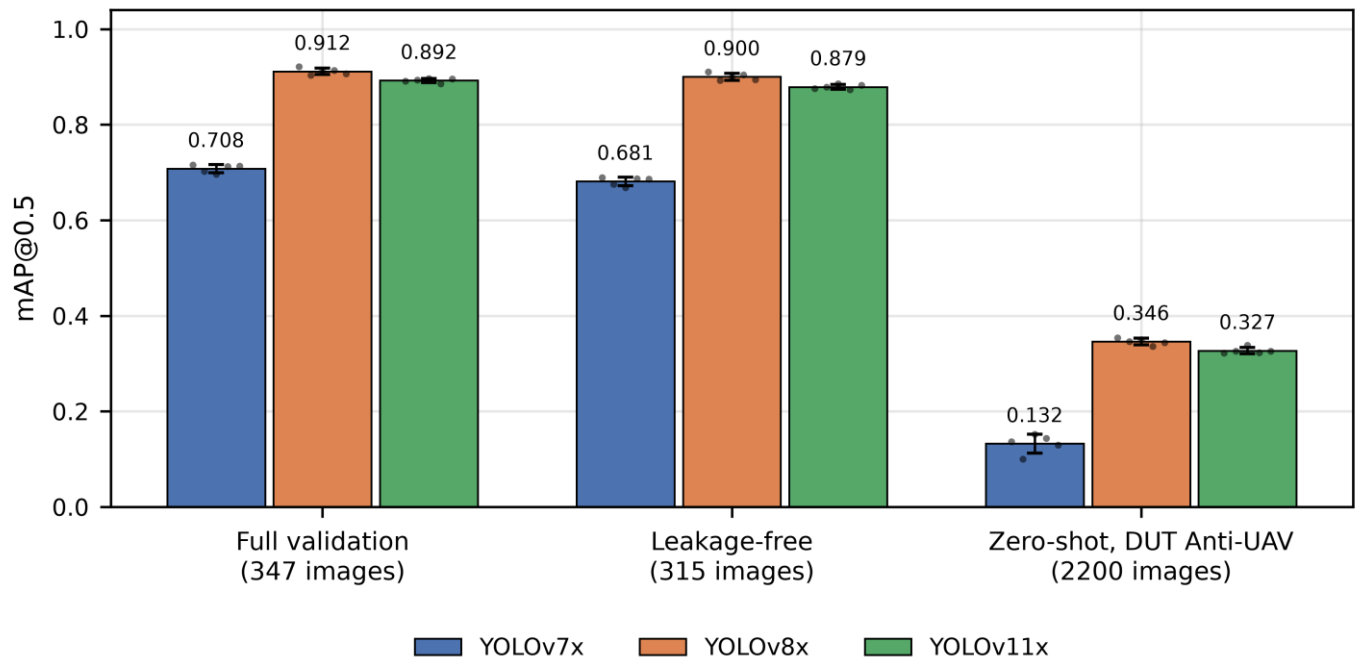


Fig. 5. Detection accuracy across the three evaluation sets. Bars are the mean over five independent training runs, error bars are one standard deviation, and points are the individual runs. Removing the contaminated validation images lowers every model; evaluating zero-shot on an independently collected dataset lowers them much further while preserving the ordering.

IV. DISCUSSION

A. Generational Improvements

Detection accuracy does not increase monotonically with architectural generation on this dataset. YOLOv8x exceeds YOLOv11x by 1.9 pp $mAP@0.5$ as a mean over five independent training runs (0.912 ± 0.007 against 0.892 ± 0.004), the per-seed ranges do not overlap, and a Welch unequal-variance t-test over the five runs per model gives $t = 5.351$, $p = 0.0012$ with Cohen's $d = 3.38$. Taking per-image $AP@0.5$ as the experimental unit instead, paired across the 347 validation images, gives a mean difference of -0.0349 in favour of YOLOv8x (paired t-test $p = 0.00065$; Wilcoxon signed-rank $p = 0.00064$; Cohen's $d_z = -0.185$; percentile bootstrap 95 %

confidence interval $[-0.0553, -0.0154]$ over 10,000 resamples). Both units of analysis agree in sign and exclude zero. YOLOv11x nevertheless uses 16.6 % fewer parameters than YOLOv8x, so the practical choice between them is a trade-off between accuracy and model size rather than a dominance in either direction. We do not attribute the differences to specific architectural components: isolating such causes would require an ablation study that we did not perform.

B. Operating-Point Behaviour

Because object detection provides no finite set of background samples, we characterize threshold behaviour with free-response ROC rather than ROC-AUC. The FROC curves show that YOLOv8x sustains the highest sensitivity at low false-

positive-per-image rates, which is the regime that matters for surveillance deployment, where each false alarm carries an operational cost. Reducing false positives at a fixed sensitivity remains the main practical difference between the models; identifying which architectural changes produce it is beyond what this benchmark can establish.

C. Edge Deployment Implications

Parameter count alone predicts neither the memory a model needs at inference nor the energy it costs per frame, so both were measured directly (Table XII). Peak inference memory does follow parameter count here: YOLOv11x has the smallest

footprint at 369 MiB against 535 MiB for YOLOv7x. Energy per frame does not. Board power differs by under 2 W across the three models, so energy per frame is set by throughput, and YOLOv7x — the largest model — is the cheapest at 1.53 J while YOLOv8x is the most expensive at 1.74 J. The claim that YOLOv11x is the most plausible edge candidate among those evaluated is therefore supported on memory and contradicted on energy. These remain desktop-GPU measurements: they do not establish on-device behaviour on an embedded accelerator, where memory bandwidth and kernel support differ, and on-device deployment is left to future work.

TABLE XII. MEASURED DEPLOYMENT COST AT BATCH SIZE ONE

Model	Parameters (M)	Weights	Activations	Peak memory	Sustained FPS	Energy per frame
YOLOv7x	70.8	288 MiB	247 MiB	535 MiB	53.5	1.532 J
YOLOv8x	68.1	276 MiB	110 MiB	386 MiB	46.3	1.738 J
YOLOv11x	56.8	237 MiB	132 MiB	369 MiB	51.0	1.587 J

Peak memory is measured from a single synchronized forward pass at 640×640. Energy is integrated from board power sampled during sustained inference, net of a 18.1 W idle draw, on the GPU of Section II-D. The throughput column is the sustained rate observed during that energy measurement, with power sampling running concurrently; it is therefore lower than the peak throughput of Table IV, which is the best of repeated 100-pass timings with nothing else running. Energy per frame is computed from the sustained rate, since that is the rate at which the measured power was drawn.

D. Limitations

Several limitations bound these findings. The dataset contains 1,012 training and 347 validation images, and its instances are strongly skewed towards large objects, 88 % of validation instances exceeding 96² px; the cross-dataset evaluation quantifies what this costs, with the best model retaining 38 % of its accuracy on a set dominated by small targets. The predefined partitions are not fully independent: 32 of 347 validation images are near-duplicates of training images, and while the leakage-free re-scoring preserves the ordering, all full-partition figures are optimistic by 0.011 to 0.027 mAP@0.5. The dataset provides no independent test partition, so the same data were used for checkpoint and threshold selection; the resulting optimism was measured against the unselected last-epoch checkpoint and is +0.006 mAP@0.5 or smaller. The 50-epoch budget was not run to convergence. Each architecture is initialized from a COCO-pretrained checkpoint produced by its own upstream pipeline, so initialization is confounded with architecture, and the differences cannot be attributed to architectural evolution alone; controlling this would require training all three from scratch, which a dataset of this size cannot support, or checkpoints pretrained under one common recipe, which do not exist for these architectures. Only selected generations were compared: YOLOv9 and YOLOv10 were not evaluated. Only extra-large variants were compared, and results may not generalize to multi-class detection, to adverse conditions such as night or fog, or to nano variants. Gradient accumulation was used to reach an effective optimizer batch size of 64, which does not reproduce a true batch of 64 for batch-normalization statistics or augmentation sampling, and no sensitivity study over accumulation factors was performed because a physical batch of 64 does not fit on the available device at this resolution. Edge measurements are desktop-GPU measurements, and on-device performance was not measured. Finally, matching a training protocol across separately

maintained implementations is achievable only at the level of settings, not of pipelines, as Section II-F documents.

V. CONCLUSION

This study presented a controlled comparative evaluation of three generations of YOLO architectures—YOLOv7x, YOLOv8x and YOLOv11x—for real-time drone detection on a publicly available dataset under matched high-level training settings, with all models scored by a single evaluation implementation. On this dataset, YOLOv8x delivered the best detection performance, reaching mAP@0.5 = 0.912 ± 0.007 over five independent training runs against 0.892 ± 0.004 for YOLOv11x and 0.708 ± 0.008 for YOLOv7x, with the YOLOv8x and YOLOv11x per-seed ranges not overlapping (Welch t-test, p = 0.0012). YOLOv11x is the smallest model among those evaluated, at 56.8 million parameters.

The results do not support a monotone improvement across architectural generations. The transition from anchor-based detection in YOLOv7x to anchor-free detection in YOLOv8x and YOLOv11x is associated with a substantial gain under the shared protocol, but the most recent generation did not outperform its predecessor here. The same ordering was obtained zero-shot on the independently collected DUT Anti-UAV test set, so it is not an artefact of this dataset, while absolute accuracy fell to 0.346 for the best model and does not transfer. We further found that a single training protocol does not treat separately maintained implementations equally: YOLOv7x reaches mAP@0.5 = 0.853 under its own repository's SGD recipe against 0.716 under the shared AdamW protocol, so a benchmark that imposes one optimizer on all architectures systematically understates it. We do not attribute the measured differences to individual architectural components, which would require an ablation study we did not perform, and initialization from architecture-specific pretrained checkpoints remains a confounding factor.

From a practical perspective, YOLOv8x is the most accurate of the evaluated models on this dataset, while YOLOv11x offers the smallest parameter count and is therefore the more plausible candidate where memory is the binding constraint; the choice between them is a trade-off rather than a dominance. YOLOv7x remains a viable option for legacy deployments integrated with the established WongKinYiu repository, provided it is trained with that repository's own recipe. Future work should extend this methodology to lightweight variants suitable for edge deployment, to multi-class aerial object detection, to validation under nighttime, adverse weather, and heavy occlusion, and to multi-modal sensor fusion. All experimental code, trained model weights, per-run results, and dataset configurations are publicly available to facilitate reproducibility.

In practical terms, the principal contribution is a reproducible, like-for-like selection guideline rather than a new detector, together with the finding that generational ordering cannot be assumed. Its conclusions are bounded by a modest single-class dataset skewed towards large instances, by the absence of an independent test partition, by a fixed training budget that was not run to convergence, and by a single-GPU evaluation with no on-device measurement. The main open challenges remain the reliable detection of small and distant drones, robustness under adverse weather and nighttime conditions, and verified real-time operation on embedded hardware.

ACKNOWLEDGMENT

This research received no grant from any funding agency.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could influence the results of the work presented in this study

DECLARATION ON GENERATIVE AI

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) to assist with language refinement, text editing, grammar improvement, and manuscript organization. The AI tool was not used for data collection, data analysis, model training, experimental evaluation, generation of results, or interpretation of findings. All scientific content, experimental design, analyses, figures, tables, conclusions, and final decisions were developed and verified by the authors. After using the AI tool, the authors carefully reviewed, edited, and validated all generated content and take full responsibility for the content of this publication.

REFERENCES

- [1] Yasmine, G.; Ahmed, E.; Medromi, H. Real-time lightweight drone detection model: Fine-grained Identification of four types of drones based on an improved Yolov7 model. *Neurocomputing* 2024, 596, 127941. <https://doi.org/10.1016/j.neucom.2024.127941>
- [2] Mun, J.; Sohn, C. Technology trends in deep learning for real-time object detection in drone environment. *J. Adv. Mil. Stud.* 2023, 6, 181–196. <https://doi.org/10.37944/jams.v6i2.220>
- [3] Lykou, G.; Moustakas, D.; Gritsalis, D. Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies. *Sensors* 2020, 20, 3537. <https://doi.org/10.3390/s20123537>
- [4] H. Haifawi, F. Fioranelli, A. Yarovoy and R. van der Meer, "Drone Detection & Classification with Surveillance 'Radar On-The-Move' and YOLO," 2023 IEEE Radar Conference (RadarConf23), San Antonio, TX, USA, 2023, pp. 1–6, <https://doi.org/10.1109/RadarConf2351548.2023.10149588>
- [5] S. Mandal and U. Satija, "Time-Frequency Multiscale Convolutional Neural Network for RF-Based Drone Detection and Identification," in *IEEE Sensors Letters*, vol. 7, no. 7, pp. 1–4, July 2023, Art no. 7003304, <https://doi.org/10.1109/LSENS.2023.3289145>
- [6] K. P. Solanki, P. Verma and S. Dara, "Evaluating YOLO V8 Models for Efficient Drone Detection in Anti-UAV Systems," 2025 6th International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, 2025, pp. 1–6, <https://doi.org/10.1109/RAIT65068.2025.11089241>
- [7] Averina, M.D., Levanova, O.A., Grushevskaya, D.V. et al. UAV Detection Using Neural Networks. *Aut. Control Comp. Sci.* 59, 910–919 (2025). <https://doi.org/10.3103/S0146411625700270>
- [8] Alqaraleh, M., Alzboon, M. S., & Al-Batah, M. S. (2024). Skywatch: Advanced Machine Learning Techniques for Distinguishing UAVs from Birds in Airspace Security. *International Journal of Advanced Computer Science and Applications*, 15(11). <https://doi.org/10.14569/IJACSA.2024.01511104>
- [9] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587. https://openaccess.thecvf.com/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf
- [10] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017, <https://doi.org/10.1109/TPAMI.2016.2577031>
- [11] Liu, W. et al. (2016). SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) *Computer Vision – ECCV 2016*. ECCV 2016. Lecture Notes in Computer Science(), vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2
- [12] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao; YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7475. <https://doi.org/10.48550/arXiv.2207.02696>
- [13] Al-Iqubaydhi, N., Alenezi, A., Alanazi, T., Senyor, A., Alanezi, N., Alotaibi, B., ... & Hariri, S. (2024). Deep learning for unmanned aerial vehicles detection: A review. *Computer Science Review*, 51, 100614. <https://doi.org/10.1016/j.cosrev.2023.100614>
- [14] M. Hussain, "YOLOv1 to v8: Unveiling Each Variant—A Comprehensive Review of YOLO," in *IEEE Access*, vol. 12, pp. 42816–42833, 2024, <https://doi.org/10.1109/ACCESS.2024.3378568>
- [15] Chen, C., Zheng, Z., Xu, T., Guo, S., Feng, S., Yao, W., & Lan, Y. (2023). YOLO-Based UAV Technology: A Review of the Research and Its Applications. *Drones*, 7(3), 190. <https://doi.org/10.3390/drones7030190>
- [16] Cao, J., Bao, W., Shang, H., Yuan, M., & Cheng, Q. (2023). GCL-YOLO: A GhostConv-Based Lightweight YOLO Network for UAV Small Object Detection. *Remote Sensing*, 15(20), 4932. <https://doi.org/10.3390/rs15204932>
- [17] Li, H., Ma, J., & Zhang, J. (2025). ELNet: An Efficient and Lightweight Network for Small Object Detection in UAV Imagery. *Remote Sensing*, 17(12), 2096. <https://doi.org/10.3390/rs17122096>
- [18] Chen, N., Li, Y., Yang, Z. et al. LODNU: lightweight object detection network in UAV vision. *J Supercomput* 79, 10117–10138 (2023). <https://doi.org/10.1007/s11227-023-05065-x>
- [19] Na, L. (2024). CNN-Based Salient Target Detection Method of UAV Video Reconnaissance Image. *International Journal of Advanced Computer Science and Applications*, 15(9). <https://doi.org/10.14569/IJACSA.2024.0150909>
- [20] X. Min, W. Zhou, R. Hu, Y. Wu, Y. Pang and J. Yi, "LWUAVDet: A Lightweight UAV Object Detection Network on Edge Devices," in *IEEE Internet of Things Journal*, vol. 11, no. 13, pp. 24013–24023, 1 July 2024, <https://doi.org/10.1109/JIOT.2024.3388045>

- [21] Hua, W., Chen, Q. & Chen, W. A new lightweight network for efficient UAV object detection. *Sci Rep* 14, 13288 (2024). <https://doi.org/10.1038/s41598-024-64232-z>
- [22] Zhao, Y., Wang, L., Lei, G., Guo, C., & Ma, Q. (2024). Lightweight UAV Small Target Detection and Perception Based on Improved YOLOv8-E. *Drones*, 8(11), 681. <https://doi.org/10.3390/drones8110681>
- [23] Jiang, P., Yang, X., Wan, Y., Zeng, T., Nie, M., & Liu, Z. (2024). DRBD-YOLOv8: A Lightweight and Efficient Anti-UAV Detection Model. *Sensors*, 24(22), 7148. <https://doi.org/10.3390/s24227148>.
- [24] Fang, A., Feng, S., Liang, B., & Jiang, J. (2024). Real-Time Detection of Unauthorized Unmanned Aerial Vehicles Using SEB-YOLOv8s. *Sensors*, 24(12), 3915. <https://doi.org/10.3390/s24123915>
- [25] Zhou, J., Su, T., Li, K., & Dai, J. (2024). Small Target-YOLOv5: Enhancing the Algorithm for Small Object Detection in Drone Aerial Imagery Based on YOLOv5. *Sensors*, 24(1), 134. <https://doi.org/10.3390/s24010134>
- [26] Liu, Y., Chen, S., & Luo, L. (2023). Optimized YOLOv7 for Small Target Detection in Aerial Images Captured by Drone. *International Journal of Advanced Computer Science and Applications*, 14(9). <https://doi.org/10.14569/IJACSA.2023.0140909>
- [27] Yue, M., Zhang, L., Huang, J., & Zhang, H. (2024). Lightweight and Efficient Tiny-Object Detection Based on Improved YOLOv8n for UAV Aerial Images. *Drones*, 8(7), 276. <https://doi.org/10.3390/drones8070276>
- [28] Tan, Y., Xu, B., Sun, J., Xu, C., Pan, W., Dai, S., & Liu, H. (2025). DBYOLOv8: Dual-Branch YOLOv8 Network for Small Object Detection on Drone Image. *International Journal of Advanced Computer Science and Applications*, 16(1). <https://doi.org/10.14569/IJACSA.2025.01601124>
- [29] Huang, M., Mi, W., & Wang, Y. (2024). EDGS-YOLOv8: An Improved YOLOv8 Lightweight UAV Detection Model. *Drones*, 8(7), 337. <https://doi.org/10.3390/drones8070337>
- [30] P. Varma, K. Singh, A. Das, D. K. Giri and P. N. Dwivedi, "Modified YOLOv8 for Real-Time Multi-Class Uncooperative Aerial Vehicle Detection and Classification," 2025 IEEE 21st International Conference on Automation Science and Engineering (CASE), Los Angeles, CA, USA, 2025, pp. 3010-3015, <https://doi.org/10.1109/CASE58245.2025.11163927>
- [31] J. Han, R. Cao, A. Brighente and M. Conti, "Light-YOLOv5: A Lightweight Drone Detector for Resource-Constrained Cameras," in *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 11046-11057, 15 March 15, 2024, <https://doi.org/10.1109/IJOT.2023.3329221>
- [32] Chen, Z., Zhang, Y., & Xing, S. (2025). YOLO-LE: A Lightweight and Efficient UAV Aerial Image Target Detection Model. *Computers, Materials & Continua*, 84(1). <http://dx.doi.org/10.32604/cmc.2025.065238>
- [33] Di Cecio, G., Manco, A., & Gigante, G. (2024, March). On-board drone classification with Deep Learning and System-on-Chip implementation. In *Journal of Physics: Conference Series* (Vol. 2716, No. 1, p. 012059). IOP Publishing. <https://doi.org/10.1088/1742-6596/2716/1/012059>
- [34] G. Li, J. Zhang, M. Zhang and H. Corporaal, "An Efficient FPGA Implementation for Real-Time and Low-Power UAV Object Detection," 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 1387-1391, <https://doi.org/10.1109/ISCAS48785.2022.9937449>
- [35] Lv, Y., Tian, B., Guo, Q., & Zhang, D. (2025). A Lightweight Small Target Detection Algorithm for UAV Platforms. *Applied Sciences*, 15(1), 12. <https://doi.org/10.3390/app15010012>
- [36] Ma, F., Zhang, R., Zhu, B. et al. A lightweight UAV target detection algorithm based on improved YOLOv8s model. *Sci Rep* 15, 15352 (2025). <https://doi.org/10.1038/s41598-025-00341-7>
- [37] Zhu, W., Chen, K. Real-time object detection for unmanned aerial vehicles based on vision transformer and edge computing. *Sci Rep* 16, 6814 (2026). <https://doi.org/10.1038/s41598-026-37938-5>
- [38] Zhou, S., Yang, L., Liu, H. et al. Improved YOLO for long range detection of small drones. *Sci Rep* 15, 12280 (2025). <https://doi.org/10.1038/s41598-025-95580-z>
- [39] Zheng, C., Liu, L., Fu, Q. et al. YOLO-DD: a lightweight framework for UAV detection in complex environments via boundary-aware fusion. *EURASIP J. Adv. Signal Process.* 2025, 44 (2025). <https://doi.org/10.1186/s13634-025-01253-4>
- [40] Wang, T., Yang, S., Wan, M., & Liu, Y. (2025). FFG-YOLO: Improved YOLOv8 for Target Detection of Lightweight Unmanned Aerial Vehicles. *Applied System Innovation*, 8(4), 109. <https://doi.org/10.3390/asi8040109>
- [41] Z. Song, Y. Zhang and A. Ebayyeh, "EDNet: Edge-Optimized Small Target Detection in UAV Imagery - Faster Context Attention, Better Feature Fusion, and Hardware Acceleration," 2024 IEEE Smart World Congress (SWC), Nadi, Fiji, 2024, pp. 829-838, <https://doi.org/10.1109/SWC62898.2024.00141>
- [42] Y. Benchen and D. Xuan, "Improving the lightweight anti drone target detection algorithm for YOLOv8," 2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Hangzhou, China, 2024, pp. 402-408, <https://doi.org/10.1109/ICCASIT62299.2024.10827908>
- [43] Zainudin, A. Z. B., Abdul Karim, H. B., & Aldahoul, N. (2024). Drone detection and tracking in distorted surveillance video. 8th International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia. IEEE. <https://doi.org/10.1109/ICSIPA62061.2024.10686164>
- [44] Mambetov, S.; Nurpeissova, D.; Taissariyeva, K.; Tleuberdiyeva, G.; Mukanova, Z.; Kulambayev, B.; Moshkalov, A.; Skakova, A. Integrated IoT-UAV Architecture for Three-Dimensional Electromagnetic Radiation Monitoring and Intelligent Source Classification. *Electronics* 2026, 15, 1941. <https://doi.org/10.3390/electronics15091941>
- [45] A. El-almi, Y. Nadir, and K. Mansouri, "An Efficient Geometric Transformation-Based Approach for Multi-UAV Image Stitching", *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 4, pp. 25507–25513, Aug. 2025.
- [46] Singha, S.; Aydin, B. Automated Drone Detection Using YOLOv4. *Drones* 2021, 5(3), 95. <https://doi.org/10.3390/drones5030095>
- [47] Aydin, B.; Singha, S. Drone Detection Using YOLOv5. *Eng* 2023, 4(1), 416–433. <https://doi.org/10.3390/eng4010025>
- [48] Hakani, R.; Rawat, A. Edge Computing-Driven Real-Time Drone Detection Using YOLOv9 and NVIDIA Jetson Nano. *Drones* 2024, 8(11), 680. <https://doi.org/10.3390/drones8110680>
- [49] YOLO Drone Detection Dataset. Kaggle. <https://www.kaggle.com/datasets/muki2003/yolo-drone-detection-dataset> (accessed 13 April 2026).
- [50] Zhao, J., Zhang, J., Li, D., & Wang, D. (2022). Vision-Based Anti-UAV Detection and Tracking. *IEEE Transactions on Intelligent Transportation Systems*, 23(12), 25323–25334. <https://doi.org/10.1109/TITS.2022.3177627>