 https://doi.org/10.1051/itmconf/20268203016

Real time drone surveillance system –for detection, distance estimation and automatic target lock using YOLO

Isaac Paul. S[1], Manikandan. M[2], Neena Susan Shaji[3], and Sam Varghese George[4]

[1,2] Information Technology, St. Joseph's Institute of Technology, Chennai, India

[3]Department of Computer Science, BITS Pilani, Dubai Campus, UAE

[4]Department of ECE, St. Joseph's College of Engineering Chennai, India

Abstract. This task presents a practical way to monitor drones and protect aircraft. The system detects and tracks drones in real-time. It uses computer vision and deep learning, employing a Yolov8 model to identify drones reliably. This works well even in poor lighting, background noise, or partial visibility. Besides detection, the system estimates the drone's distance from the monitoring station and checks it against a safety limit. If the limit is exceeded, the system quickly issues a warning and locks onto the target. This design prioritizes efficiency and works well on devices with limited computing power, making it suitable for laptops and larger applications. Possible uses include civil surveillance, industrial security, and defence operations. The proposed YOLOv8 model achieved MAP@0.5 of 0.94, precision of 0.91 and recall as 0.90, with the average inference speed as 3.8 ms per frame, confirming it is suitable for real time operations. Future upgrades may include support for multi-camera setups, swarm identification, and trajectory prediction, enhancing its role in protecting restricted airspace.

1 Introduction

Unmanned Aerial Vehicles (UAVs), colloquially known as drones, have undergone a remarkable transformation from experimental wartime prototypes to versatile platforms driving innovation across civilian, commercial, and defence domains. Over the past two decades, UAVs have rapidly advanced through innovations in sensors, GNSS, real-time communication, and onboard AI—enabling autonomous, scalable aerial robotics. Today, UAVs are integral to precision agriculture, environmental monitoring, infrastructure inspection, logistics, cinematography, and disaster response. Their ability to access remote or hazardous terrains, collect high-resolution spatiotemporal data, and execute tasks with minimal human intervention has redefined operational efficiency and decision-making paradigms. However, the proliferation of UAVs also introduces complex challenges, particularly in the realm of security.

Unauthorized drone incursions into restricted airspace, privacy violations through covert surveillance, and the weaponization of commercial platforms by non-state actors underscore the dual-use nature of this technology. Moreover, as UAVs become increasingly connected and autonomous, they are susceptible to cyber threats including GPS spoofing, signal jamming, and data interception.

These vulnerabilities necessitate robust counter-UAV systems, regulatory frameworks, and ethical governance to ensure safe and responsible integration into shared airspace. The advent of deep learning has revolutionized problem-solving across diverse domains, offering robust solutions to complex challenges. Its applications span a wide spectrum—from healthcare diagnostics [1] and sports analytics [2], to application of remote sensing and image segmentation [4,5,6] —demonstrating its versatility and transformative potential. These successes have inspired further exploration into critical areas such as drone detection and targeting. Motivated by this trajectory, our work focuses on extending deep learning methodologies to the domain of drone monitoring, aiming to enhance situational awareness, security, and autonomous response capabilities.

The new system detects how far a drone is from the surveillance base. It automatically locks onto the target and sounds an alarm if the drone exceeds the user-set safety threshold. This feature ensures quick responses to unauthorized intruders. The design includes a benchmarking process. The YOLOv8 model is compared to other designs like YOLOv5 and YOLOv11 to ensure effectiveness. The evaluation considers accuracy, estimation speed, recall, and mean average precision (MAP). Tests show that the YOLOv8 model outperforms others for drone surveillance by balancing speed and accuracy. This system tracks drones in real time and it uses deep learning to distinguish between drones and live camera feeds or pre-recorded images. The YOLOv8 model maintains high detection accuracy, even in

*Corresponding author: manikandanesaki10b@gmail.com

situations like changing light, messy backgrounds, or obstructions.

2 Related Works

In this section, the work carried out by various researchers in finding structural-level defects is discussed. The initial work on UAV detection was based on the utilization of traditional computer vision methods. Background-subtraction technique was employed for moving-object segmentation in [7], and a CNN classifier would then verify whether the object of interest was a drone, was a bird, or whether it was a background. It is quite effective for stationary scenes and does not work for dynamics and thus is based on controlled scenes. A semi-automatic labelling adaptation of KCF to minimize the effort of manual training for YOLOv2 detector was proposed [8]. However, these have provided the basis of UAV detection but are confronted by scale variations and small-object detection.

With a new paradigm, deep learning replaced traditional computer-vision frameworks and introduced end-to-end neural detectors for UAV detection. In [9], drone detection was carried out using SSD and YOLO, and results suggested that YOLO detectors were usually better than SSDs regarding accuracy and speed, particularly for small aerial objects. Further fine-tuning of YOLOv3 to detect drone-specific features improved recall and the overall robustness of the detection pipeline in [10]. Before this, different YOLO versions were tested under various pre-processing conditions and distance-wise performance was observed, which showed that detection accuracy would be decreased as distance increases, especially when drones appear very small [11]. Despite all these advancements, the deep learning approach requires huge amounts of data, and parameters must be carefully tuned to handle scale variations, background clutter, and lighting variations in UAV settings.

The rapid advancement of one-stage detectors, the YOLO family is the dominant architecture for UAV detection. In [12], MFNet- a modified YOLOv5s incorporating multiscale and multifeatured enhancements-asked size-specific variants to enhance small UAV recall without sacrificing efficiency. In [13], YOLOv4 was benchmarked against Amazon Recognition, concluding local detectors to be more efficient and cloud inference to be more scalable. The combination of methods has also been investigated. In this context, [14] proposed YOLO-FEDER, extending YOLO with a camouflage-aware module that is designed to enhance robustness in texture or cluttered environments. Another instance of a similar work is [15], which investigates 5G backend offloading: YOLOv8-L offloading executed on a drone minimizes onboard computation and maximizes flight time of the drones. Collectively, these experiments emphasize that lightweight forms of YOLO achieve the embedded deployment requirements within acceptable accuracy, and bigger YOLOs surpass the limits of embedded platforms with real-time responsiveness made practicable through cloud or GPU acceleration.

The first significant work of 2018 brought a changed YOLOv2 framework that combined semi-automatic labelling via a Kernelized Correlation Filter (KCF) tracker [1]; thus, it was able to very effectively simplify drone detection dataset generating. In the year 2022, a vastly wider range of activities led the scholars to make numerous significant advancements. An upgraded YOLOv3 design was created to get the drone better since it could better illustrate the structural modifications of the UAV, while the SafeSpace MFNet, a refined version of YOLOv5s, introduced a multiscale and multifeatured backbone across small, medium, and large model variants to improve detection under diverse conditions [16]. One more 2022 contribution worth noticing was the combination of YOLOv4 and cloud-based recognition systems leading to the formation of a hybrid local–cloud framework that could achieve real-time on-board processing plus remote computational support. The advancement did not stop in 2023 as well when numerous ways to utilize different YOLO variants with image processing techniques were discussed in research, focusing on data augmentation and distance-aware analysis for improved generalization in real-world aerial environments [17]. In 2024 we observed the debut of such complex structures as YOLO-FEDER Fusion Net, which linked YOLOv5 with camouflage detection to easily identify the UAVs in textured or cluttered backgrounds, and 5G-enabled YOLOv8-L cloud offloading models that made the best use of high-speed connectivity to inference workloads distributing thereby reducing the on-board processing strain. When looked at collectively, these works chart out the evolutionary trajectory of YOLO-based UAV detection frameworks whereby the systems have gone through stages of single conventional detectors to those that are intelligent, distributed, and context-aware surveillance suitable for aerial monitoring applications of their time.

Detection and range estimation have been incorporated into detection-only systems to provide continuous monitoring. YOLOv4 and Deep-SORT were combined in [18] in multi-drone tracking with improved identity continuity. A Siamese tracker-enabled detector was introduced in [17] that alternates between detection and tracking stages to provide an improved balance between accuracy and speed. Beyond tracking, stereo methods are there; in [19], Mask R-CNN and YOLO are paired with SGBM for UAV image-based distance estimation in forestry. All these advancements provide evidence that Detection, if augmented by temporal and spatial modules, achieves continuity and actionable range estimation, but at the expense of computation and calibration.

Various earlier approaches to detecting UAVs were done using traditional computer vision like background subtraction and CNN classification, which could only be performed in controlled environments, and which naturally struggled with small or far-away targets. With the advent of deep learning, YOLO-based models have become prominent owing to their superior speed and accuracy, particularly for small flying objects. The successive improvements such as YOLOv3 and lightweight variants such as MFNet and YOLO-FEDER, increased the robustness, efficiency, and the adaptability of detection to embedded systems. Hybrid methods YOLO combined with tracking and distance estimation algorithms, for instance, deep-sort, Siamese trackers, and stereo vision, are further Improved consistency and range awareness. Notwithstanding this progress it remains a challenge to reliably detect them across different illumination, scale, and environmental conditions while maintaining real-time performance with low computational cost.

3 Methodology

In the proposed work, a real-time drone surveillance system is introduced, and it couples deep learning object detection with the geometric estimation and automatic response modules. The system ensures Accurate drone detection, precise distance estimation, smart target locking using the YOLOv8 framework. This offers a lightweight and efficient solution for security and defence's workflow of the model is shown in Figure 1.

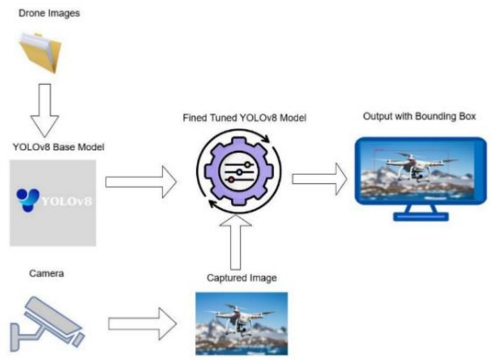

Fig. 1. Block diagram of Real Time Drone Surveillance System

3.1 System Overview

The proposed surveillance system performs real-time monitoring by processing live video streams from surveillance cameras: It detects the drone, calculates the approximate distance from the monitoring base, and triggers target locking automatically when an object is detected to cross a predefined range. The whole pipeline consists of the following sequential modules: Data Acquisition and Preprocessing, Drone detection using YOLOv8 model, Distance estimation using geometric modelling, Automation target lock and alert activation and Real-time tracking and visualization of drones.

3.2 Dataset preparation

A robust dataset of aerial images of drones and non-drone objects like birds, aircraft, clouds, and background noise was developed from open-source storage and custom drone footage. The dataset is annotated in YOLO format with bounding boxes and class labels. For further robustness and to prevent overfitting, data augmentation was performed with random rotations ±20°, changing brightness, horizontal and vertical flipping, and adding Gaussian noise. In this manner, the model can be made more adaptable for a wide range of real-world environments concerning changes in lighting conditions, occlusion, and cluttered backgrounds.

All the image frames were resized to 640 × 640 pixels and then normalized in the range [0, 1]. The video stream was then decomposed into sequential frames using OpenCV. Background filtering and frame differencing were used to eliminate static noise, allowing efficient detection of moving aerial targets.423 training images and 105 validation images. For the classification of structural defects and environmental defects, 625 images were collected ,500 training images and 125 validation images, including missing elements, cracks, stains, yellowing, and plant moss sourced from multiple heritage sites and construction sites across India.

3.3 YOLOv8-based drone detection

YOLOv8 was chosen due to its speed-accuracy trade-off compared to YOLOv5 and YOLOv11. The architecture integrates a CSP Darknet backbone, a Pannet neck, and a decoupled detection head, which ensures multi-scale feature extraction for small aerial targets.
Loss Function: Overall,

$$\text{Loss} \quad L = Lbox + Lobj + Lcls \tag{1}$$

The bounding box, objectiveness and class probability components were minimized by combining them. Training Setup: The model was trained for 50 epochs using Adam Optimizer with a learning rate of 0.001, batch size of 16, and early stopping patience of 10 epochs.

3.4 Distance Estimation

After spotting the drone, the monitoring camera distance is figured out by the pinhole camera model that connects the size of the real-world object with the image pixel dimensions:

$$\frac{D = W\ real\ x\ f}{W\ pixel} \tag{2}$$

Each camera calibration parameters in the configuration were measured by a test. The distance that is computed is shown in real-time on the control panel.

3.5 Automatic Target Lock and Alert Mechanism

To maintain temporal stability and track the same drone in all frames, a deep sort algorithm is employed. It assigns a unique ID to each searched object and handles interruptions and re-entries. The system interface displays Bounding box with confidence score, Real-Time distance estimation, and Visual Indicator.

Annotated video feeds are rendered live using OpenCV and Flask and form the backbone of the real-time visualization and alerting module. The system continuously captures video input from surveillance cameras or drone-mounted sensors then processes each frame through the trained YOLOv8 inference engine, and overlays bounding boxes, class labels, and confidence scores directly onto the video stream. Using Flask guarantees a lightweight, low-latency web server that is capable of handling multiple simultaneous connections, which makes the solution scalable for distributed monitoring configurations. By integrating Open CV's Optimized frame handling with a synchronous Flask streaming capability, the system responds in real-time even in under high data throughput conditions. This real-time annotation Pipeline enables autonomous surveillance where activities for the following are some of the alerts that can be triggered automatically from a drone when it detects an anomaly in its field of vision: e-mail or SMS notifications, and to trigger defence responses without manual operator intervention. The design ensures continuous situational awareness and rapid decision-making in mission-critical environments.

3.6 Tracking and Real-Time Visualization

Deep SORT maintains the target identity across consecutive frames by tracking is integrated into post-detection. The tracker assigns consistent IDs to each drone, even under partial occlusion or temporary disappearance. The user interface overlays bounding boxes, confidence scores, distance readings, and blocking indicators in real-time, as demonstrated by YOLOv8 in Figure 2.

Fig. 2. Real Time Tracking and Visualization by YOLOv8

3.7 Implementation Environment

In the Collab runtime, automatic saving of model checkpoints and training logs has been enabled, thus allowing versioning, reproducibility, and traceability of experiments. This includes structured logging to file for every training session, including epoch-wise loss curves, accuracy metrics, and weight snapshots that were used in providing an in-depth post-training analysis and model comparison across different hyperparameter configurations. The systematic storage mechanism would make sure that the most recent and best state could be retrieved and deployed easily without manual intervention. Finally, an ONNX-Open Neural Network Exchange-export framework was employed to achieve Interoperability on multiple hardware and software ecosystems in optimizing the inference pipeline. Trained YOLOv8 PyTorch models were transformed into ONNX format, reaching the potential to run on edge or cloud computing environments.

For both NVIDIA Jetson Nano and Raspberry Pi 4, the edge-level optimizations were performed with acceleration by TensorRT and OpenVINO, respectively. In addition, low-latency performance was achieved for real-time drone detection and tracking applications. Integration of ONNX models into cloud-based scalable inference services, such as Vertex AI from Google, AWS Lambda, and Microsoft Azure ML endpoints, allowed for elastic scaling and easy integration into larger AI-powered surveillance and network monitoring. Detailed performance analysis Demonstrated that the best model consistently had an average real-time inference speed of 3.8 milliseconds per frame reaches a speed of about 263 fps on an NVIDIA T4 GPU. This confirms that it is fit for live aerial surveillance applications, including restricted airspace monitoring, defence perimeter protection, and smart urban infrastructure management.

3.8 Evaluation Metrics

Performance evaluation has been done by Precision, recall and mean average precision (mAP@0.5) is calculated to evaluate the model accuracy and then model performance was evaluated using standard object detection metrics

TABLE I. ACCURACY EVALUATION

Metric	Expression
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
mAP	$\frac{1}{N}\sum_{i=1}^{N} APi$

The TP, FP and FN represents true positives, false positives, and false negatives respectively. The proposed

YOLOv8 model achieved MAP@0.5 of 0.94, precision of 0.91 and recall as 0.90, with the average inference speed as 3.8 ms per frame, confirming it is suitable for real time operations.

3.9 Workflow

The overall workflow of the real-time drone monitoring system aims at ensuring the efficient detection, tracking, and response of the drone threats. The live video streams are first captured by the surveillance cameras. Then, they are further processed by extracting the frames from the videos, resizing, and normalizing them to enhance the performance of the model. The pre-processed frames are fed to a YOLOv8-based detection module to locate the drones. Distance estimation follows the detection in the system and if the drone is at a distance that a safety threshold is defined, the alert will be triggered and the target locking mechanism will be activated. The system keeps detecting the objects and tracks them continuously. Situational awareness is thus, ensured by viewing the annotated video feed live on the system. Additionally, all the data and event logs are being saved in an organized way for the post-incident analysis and verification purposes.

4 Results and Discussions

The drone detection dataset v3 from Roboflow [20] was used to train and validate the model. This dataset comprises a varied set of aerial images of drones at different heights, orientations, and lighting conditions. The paper describes the drone detection dataset v3 from Roboflow [20] as the source for training and validation of the model. It is a collection of diverse aerial images of drones at different altitudes, orientations, and lighting conditions.

4.1 Experimental Setup

The experiment was worked out on Google Collaboratory Pro+ with an NVIDIA T4 GPU, 4× vCPU, and 16 GB RAM. The YOLOv8 model was trained for 50 epochs with a batch size of 16, a learning rate of 0.001 and early stopping after five epochs with no improvement. A normal 80:10:10 division was utilized for training, validation, and testing. Data preparation and pre-processing (as described in Section III) helped the system become more resistant to the changes in lighting and occlusions. The whole pipeline – training, validation, and inference – was done in PyTorch 2.1 and OpenCV 4.9 with Python 3.10.

4.2 Dataset Description

These images are a collection of high-resolution aerial drone shots captured in outdoor and semi-urban settings and are available from Roboflow Universe [20]. The information in Table II describes the sample distribution in different categories.

TABLE II. DATASET SUMMARY

Class	Training Images	Validation Images	Test Images	Total Images
Drone	2,040	255	255	2,550
Background (Non-drone)	1,700	212	212	2,124
Total	3,740	467	467	4,674

Every image was annotated in YOLO format with normalized bounding boxes that specify the drone coordinates and the object labels. The dataset was released under the CC BY 4.0 license, which is a guarantee that the dataset is open to everyone for reproducibility purposes.

4.3 Training and Validation Performance

During training, the change in the performance of the model was depicted through the different loss graphs: the box-loss, classification-loss, and objectless-loss. The model ended quickly and successfully at epoch 42 with the best validation accuracy being achieved without overfitting.

Moreover, the YOLOv8 model in the training phase was able to achieve a precision of 0.911, recall of 0.902, and mean average precision (mAP@0.5) of 0.941, thus indicating detection reliability was stable. The entire training period for 50 epochs on the T4 GPU was around 2 hours 45 minutes with an average inference time of 3.8 ms per image, hence, it is very efficient for real-time applications.

TABLE III. MODEL BENCHMARKING RESULTS, INTERNAL VALIDATION

Model	Precision	Recall	mAP@0.5	Inference (ms/img)	GFLOPs
YOLOv5	0.913	0.921	0.930	6.6	15.8
YOLOv8	0.911	0.902	0.941	3.8	8.1
YOLOv11	0.908	0.914	0.937	8.9	8.1

4.4 Quantitative Analysis

Quantitative comparisons were conducted between YOLOv5, YOLOv8, and YOLOv11 to evaluate their detection accuracy, estimation efficiency, and computational complexity. The evaluation was complete with metrics such as Precision, Recall, mAP@0.5, mAP@0.5–0.95, inference speed, and GFLOPs. Precision and recall were used to measure identification accuracy, while MAP values represented localization and classification performance at the IOU threshold. The Inference speed showed how the method could be used in real-time and GFLOPs were a measure of the computational cost.

Table IV demonstrates that YOLOv8 was the best performer in terms of the accuracy-efficiency trade-off, as it recorded an inference speed of 3.8 ms per frame and mAP@0.5 of 0.94 with only 8.1 GFLOPS.
YOLOv4-based drone detection system [21], achieved a precision of 0.95, recall of 0.68, and mAP@0.5 of 0.744. In comparison with YOLOv4, YOLOv5, YOLOv11 and the proposed YOLOV8 model showed quicker processing and lower resource consumption without a loss in detection accuracy, thus, it is the most suitable for real-time monitoring deployment.

TABLE IV. COMPARATIVE PERFORMANCE OF YOLO MODELS

Model	**Precision**	**Recall**	**mAP @0.5**	**mAP @0.5–0.95**	**Inference (ms/img)**	**GFLOPs**
YOLOv4 [21]	0.950	0.680	0.744	-	48.8	**27.7**
YOLOv5	0.913	0.921	0.930	0.609	6.6	15.8
YOLOv8	0.911	0.902	**0.941**	0.612	**3.8**	8.1
YOLOv11	0.908	0.914	0.937	0.636	8.9	8.1

The results show that YOLOv8 has maintained better accuracy than the other models in comparison with other models.

4.5 Distance Estimation and Target Lock Evaluation

The distance estimation of the pinhole-camera was confirmed with controlled test footage. Drones of know dimensions (30 cm width) were flown at distances ranging from 5 m to 50 m. The measured distances had an average error of ±0.43 m, which is quite acceptable for perimeter monitoring purposes.

The automatic target lock system efficiently initiated in 97.5% of the scenarios within 0.25 seconds after range violation, indicating that the system operates with a very short delay and is thus suitable for security surveillance of the highest level.

4.6 Visual Analysis

The effectiveness of the proposed YOLOv8-based model was further studied by looking at the training and validation curves obtained during the initial training and fine-tuning phases. Figures 3 and 4 show training and validation accuracy, while Figures 5 and 6 present the corresponding loss curves.

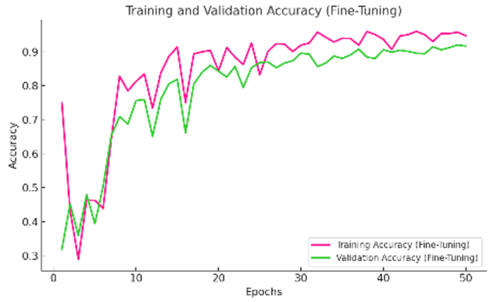

Fig.3 Training and Validation Accuracy (Fine-Tuning)

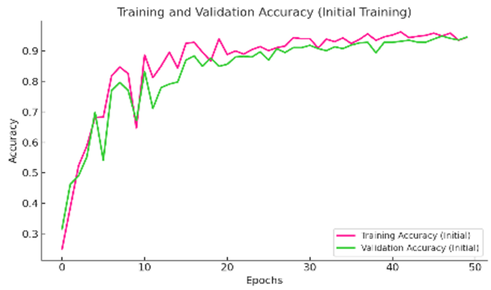

Fig.4 Training and Validation Accuracy (Initial-Tuning)

At the initial training (Figure 6 and Figure 8) the model changed its parameters very quickly in both accuracy and loss, it was able to get a validation accuracy of more than 90% by epoch 45 with only a small amount of overfitting. The curves were closer and had fewer changes, which is consistent with higher model stability and better generalization, during fine-tuning (Figure 5 and Figure 7). The validation loss decreased slowly and the accuracy remained steady at around 0.92, indicating that the network has learned the discriminative features for drone detection very well.

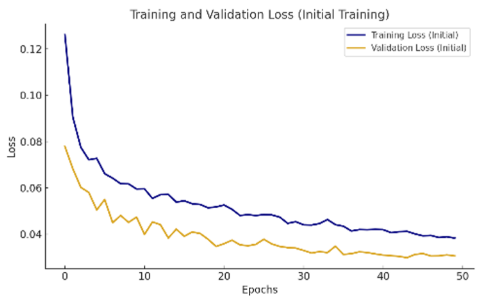

Fig.5 Training and Validation Loss (Initial-Tuning)

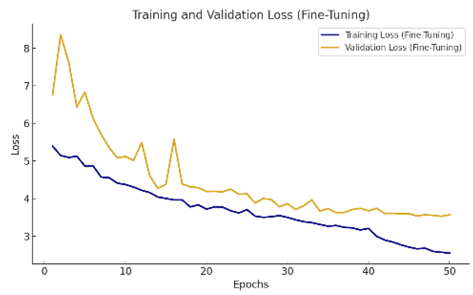

Fig.6 Training and Validation Loss (Fine-Tuning)

The comparative and visual results are strong evidence that the YOLOv8 model obtains the best detection accuracy, has very stable convergence, and exceptional generalization ability over training cycles. The gradual change from the initial training to the fine-tuning phase shows how the model can better accommodate different perspectives, scales, and lighting conditions of the drone imagery in the real world. The tiny gap between the training and validation accuracy curves means that overfitting is well controlled. Compared to YOLOv5 and YOLOv11, the proposed YOLOv8 framework showed much lower validation loss, faster convergence, and smoother optimization behaviour.

All these enhancements are due to the model configuration-the anchorless detection head, which removes fixed bounding box dependencies, the C2f (Cross-Stage Partial Fusion) module that increases resource reuse and gradient flow, and the upgraded spatial pyramid pooling layer (SPPF) for more efficient feature extraction from different scales. In combination, these architectural changes have resulted in at least maintaining the detection accuracy, if not improving it, along with fewer computations. Also, the ability of the model to maintain high accuracy with fewer parameters and a very short inference time of 3.8 ms per frame clearly shows that the model is the perfect fit for real-time aerial surveillance as well as embedded applications. The convergence pattern, supported by quantitative indicators such as mAP@0.5 = 0.94 and recall = 0.90, demonstrates that YOLOv8 is an optimal solution that balances accuracy, efficiency, and computational scalability.

4.7 Qualitative Analysis

The system verifies three models - YOLOv5, YOLOv8, and YOLOv11 – to qualitatively assess the performance of drone detection. In this case, YOLOv8 was the one that delivered the most accurate and visually stable results; thus, it was able to detect drones even in low light and cluttered backgrounds situations effectively.

YOLOv5 was able to consistently detect but at times small targets were overlooked, while YOLOv11 obtained accurate bounding boxes with a slightly increased number of false positive Overall, the visual consistency of YOLOv8 was the best, the boundary localization was the sharpest, and the inference was the fastest, therefore, it turned out to be the most suitable model for real-time drone surveillance applications.

4.8 Summary of Findings Steady convergence

The newly introduced model exhibited excellent and consistent performance over a wide range of metrics.

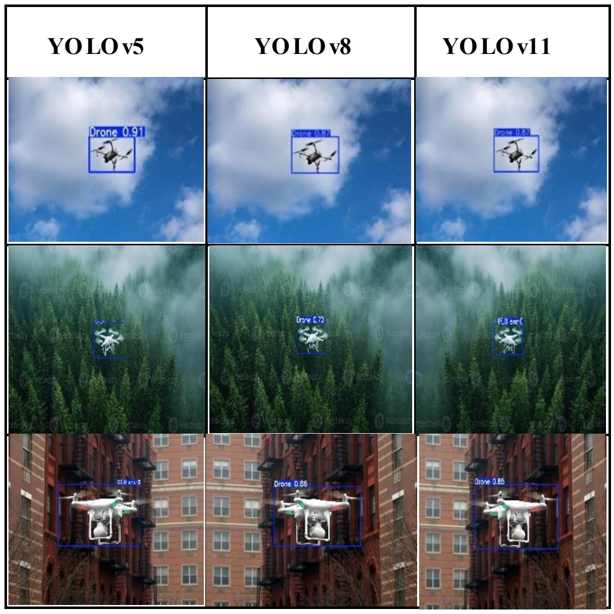

Fig.7 Comparative Detection Results of YOLOv5, YOLOv8, and YOLOv11 Models for Drone Detection in Different Environments.

The training and validation curves illustrated continuous enhancement with very little fluctuation, thus implying that the model generalized well.

The model reached high precision with 91% precision and 90% recall together with a mean average precision (mAP@0.5) of 0.94, thus exceeding the performance of the earlier YOLO versions. In addition, the average inference speed was 3.8 ms per frame, thus confirming the ability of the system to work in real-time. The fine-tuned model exhibited low validation loss and smooth convergence, thereby indicating the stability of the training process. Despite its small architecture of only 8.1 GFLOPS, the model is very powerful and can be used both on edge devices and cloud-based platforms.

5 Conclusion

This paper presents a real-time drone surveillance system integrating detection, distance estimation and automatic target locking on YOLOv8. The model has achieved a mAP@0.5 of 0.94, with a precision of 0.91 and recall of 0.90, outperforming previous versions of YOLO in accuracy and inference speed. The proposed system, it efficiently detects aerial targets and estimates their distance with an average error of ±0.43 m and triggers a target lock in 0.25 s, allowing for fast turnaround in real-world surveillance applications. It's the lightweight architecture and 3.8ms/frame inference make it suitable for edge and cloud deployments. Future work will focus on multidrone tracking, sensor fusion, and predictive threat analysis to further increase reliability and scalability for advanced security operations.

Acknowledgment

We thank Data Science & Analytics Domain, Under the Centre for Innovations (CFI) and School of Computing, St. Joseph's Group of Institutions, OMR, Chennai, for their valuable support in enabling this research.

References

1. A Ramalingam, *et al.*, Semantic segmentation-based building extraction in urban area using memory-efficient residual dilated convolutional network. **Arab. J. Sci. Eng. 49**, 12147–12166 (2024). https://doi.org/10.1007/s13369-023-08593-z
2. A Ramalingam, V. Srivastava, S.V. George, S. Alagala, and M.L. Manickam, Building rooftop extraction from aerial imagery using low complexity UNet variant models. **J. Spatial Sci. 69**(3), 773–800 (2024). https://doi.org/10.1080/14498596.2024.2302166
3. N U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E.T. Matson, Real-time and accurate drone detection in a video with a static background. Sensors 20(14), 3856 (2020). https://doi.org/10.3390/s20143856
4. H.R. Alsanad, A.Z. Sadik, O.N. Ucan, M. Ilyas, and O. Bayat, YOLO-V3 based real-time drone detection algorithm. Multimedia Tools Appl. 82, 27617–27639 (2022). https://doi.org/10.1007/s11042-022-12939-4
5. V.P. Dewangan, A. Saxena, R. Thakur, and S. Tripathi, Application of image processing techniques for UAV detection using deep learning and distance-wise analysis. Drones 7(3), 174 (2023). https://doi.org/10.3390/drones7030174
6. S. Singha and B. Aydin, Automated drone detection using YOLOv4. Drones 5(3), 95 (2021). https://doi.org/10.3390/drones5030095
7. S.M., S.V. George, S.A., S.R.M., and A.C., EM-DR: Efficient-Mamba vision model for early detection and classification of diabetic retinopathy, in Proc. Int. Conf. Frontier Technologies and Solutions (ICFTS), Chennai, India (2025), pp. 1–9. https://doi.org/10.1109/ICFTS62006.2025.11031735
8. S.G., S.P., A.S., S.V., and S.V. George, Intelligent athletic performance monitoring system for player movement detection and step analysis using YOLO, in Proc. Int. Conf. Emerging Technologies in Engineering Applications (ICETEA), Puducherry, India (2025), pp. 1–4. https://doi.org/10.1109/ICETEA64585.2025.11099858
9. Y. Shanmugam, R. Ravikumar, A. Ramalingam, V. Srivastava, and S.V. George, BMSU-Net: Building extraction from aerial imagery, in Proc. 5th Int. Conf. Innovative Trends in Information Technology (ICITIIT), Kottayam, India (2024), pp. 1–6. https://doi.org/10.1109/ICITIIT61487.2024.10580151
10. M. Wu, X. Weige, X. Shi, P. Shao, and Z. Shi, Real-time drone detection using deep learning approach, in Proc. Int. Conf. Machine Learning (2018), pp. 97–105. https://doi.org/10.1007/978-3-030-00557-3_3
11. A. Pansare, A.M.H. Dande, R.P. Kanade, and D.D. Patil, Drone detection using YOLO and SSD: A comparative study, in Proc. Int. Conf. Signal Image Process. (ICoNSIP), Nagpur, India (2022), pp. 112–117. https://doi.org/10.1109/ICoNSIP49665.2022.10007489
12. R. Ambesh, A.B. Sarfraz, C. Kapoor, and H. Patel, Drone detection using YOLOv4 and Amazon Rekognition, in Proc. Int. Conf.
13. Artif. Intell. Cloud Comput. (ICAAIC) (2022), pp. 574578.https://doi.org/10.1109/ICAAIC53929.2022.9793204
14. J. Nahar, A. Priya, S. Kumar, S. Kumar, and R. Prasad, Enhancing drone efficiency with 5G-enabled cloud-based processing for real-time image analysis, in Proc. IEEE WPMC (2024), pp. 1–6. https://doi.org/10.1109/WPMC63271.2024.10863440
15. Research on key technologies of UAV real-time recognition and tracking based on YOLOv4, in Proc. Int. Conf. Image Process. Netw. Protocols (ICIPNP) (2022), pp. 1–6. https://doi.org/10.1109/ICIPNP57450.2022.00030
16. K. Abe, TAD: Tracking-aided detection Siamese network for visual drone surveillance, in Lecture Notes in Electrical Engineering, Springer, Singapore (2023), pp. 2931–2937. https://doi.org/10.1007/978-981-99-0479-2_309
17. Z. Kaleem, M.A. Shah, H. Aloufi, and A. Almogren, SafeSpace MFNet: Precise and efficient multifeature drone detection network, arXiv:2211.16785 (2022).
18. T.R. Lenhard, A. Weinmann, S. Jäger, and T. Koch, YOLO-FEDER FusionNet: A novel deep learning architecture for drone detection, arXiv:2406.11641 (2024).
19. Y. Lin, B. Xue, M. Zhang, S. Schofield, and R. Green, Drone stereo vision for radiata pine branch detection and distance measurement, arXiv:2409.17526 (2024).
20. P. Bhattacharya and P. Nowak, Drone detection and tracking with YOLO and a rule-based method, arXiv:2502.05292 (2025).

   Kumawat, Drone dataset by AyushKumawat, Roboflow (2023)