IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

HawkEye: An Ensemble Approach For Flying Object Detection

[1]Avantika Mane, [2]Saummya Jalan , [3]Poonam Vengurlekar
[1,2]Student, [3]Professor
[1]Computer Science and Technology,
[1]Usha Mittal Institute of Technology, Mumbai, India

Abstract: Modern surveillance, security, and airspace monitoring systems rely heavily on flying object detection. The challenge is in the exact detection of fast-moving airborne objects including drones, birds, planes, and balloons especially under varying lighting, altitude, and background clutter conditions. This paper presents HawkEye, a deep learning-based ensemble system combining YOLO (You Only Look Once) and DETR (DEtection TRansformer) models, to provide improved accuracy and robustness in flying object identification. The model is trained on the Flying Objects OB 100 dataset, which covers many aerial scenarios. The paper offers two key approaches: The paper proposes two primary techniques: (1) DOLO, a hybrid architecture combining YOLO's real-time efficiency with DETR's; and (2) an ensemble learning approach combining several YOLO versions with DETR

***Index Terms* - Flying Object Detection, YOLO, DETR, Ensemble Learning, Deep Learning, Computer Vision.**

I. INTRODUCTION

In areas like surveillance, defense, air traffic control, and environmental monitoring, the identification of flying objects—including drones, planes, birds, and other airborne entities—is absolutely vital. Ensuring efficient and real-time detection of Unmanned Aerial Vehicles (UAVs) has become a pressing need as their use for commercial, military, and recreational reasons grows. Traditional object detection systems battle with issues including fast movement, occlusions, changing object sizes, and cluttered backgrounds, which lower detection accuracy. With models like YOLO (You Only Look Once) and DETR (DEtection TRansformer) providing state-of-the-art performance, deep learning-based methods have changed object detection. While DETR's transformer-based design improves accuracy by properly spotting items in complicated backgrounds, YOLO's real-time detection capability makes it appropriate for high-speed applications.

Every model, though,, has a slower rate of inference. In order to address these issues, we suggest HawkEye, an ensemble-based flying object detection system that combines DETR and YOLO to provide high detection accuracy and real-time efficiency. Our system uses confidence-based and weighted averaging techniques to combine multiple YOLO versions with DETR by utilizing ensemble learning techniques. Furthermore, we present DOLO (Detection Optimized YOLO-DETR Hybrid Model), a hybrid strategy that blends the contextual accuracy of DETR with the speed of YOLO. The methodology, dataset, model architecture, implementation, and outcomes of the HawkEye system are all described in detail in this paper. When compared to individual detection models, the suggested method seeks to increase detection accuracy, robustness, and inference speed. The results show how hybrid deep learning models can be used to address realworld problems involving aerial objects. This paper explores the ensebling approach for flying object detection, detailing its design, working principles, and key features. It explains the software and hardware components used, along with the simulation tools that help in testing its performance

II.LITERATURE REVIEW

A Deep Comprehensive Assessment of Aircraft Detection Algorithms in Satellite Imagery Authors:Safouane El Ghazouali, Arnaud Gucciardi, Francesca Venturini, Nicola Venturi, Michael Rueegsegger, Umberto Michelucci (2024)

This paper provides a deep evaluation of multiple object detection algorithms focused on detecting aircraft in satellite imagery. Among all evaluated models, YOLOv5 was found to be the most effective in terms of precision and speed. The study benchmarked models like YOLOv5, YOLOv8, Faster RCNN, CenterNet, RetinaNet, RTMDet, and DETR on HRPlanesV2 and GDIT datasets. However, the detection performance was noted to vary depending on the quality and conditions of the satellite imagery, presenting a challenge for universal deployment.

Real-Time Flying Object Detection with YOLOv8 Authors: Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, Tom Goldstein (2023)

This research introduces a real-time flying object detection system using YOLOv8, achieving state-of-the-art performance in complex aerial environments. The model was trained on a large-scale dataset covering 40 flying object classes and further refined using transfer learning with real-world aerial data. While effective, the study highlights issues like varying object sizes, occlusion, and cluttered backgrounds, which can hinder detection accuracy, especially in dense scenarios.

YOLO-Drone: Airborne Real-Time Detection of Dense Small Objects from High-Altitude Perspective Authors: Alberto Argente-Garrido, Cristina Zuheros, M. Victoria Luzon, ´ Francisco Herrera (2022)

YOLO-Drone is a novel approach that enhances small object detection in drone imagery, significantly outperforming other models in detecting densely packed targets. The model incorporates a custom backbone (Darknet59) and a multi-scale feature aggregation module (MSPP-FPN) for better accuracy. It was evaluated on UAVDT and VisDrone datasets. Despite its improvements, the model still faces challenges in detecting under low light conditions and achieving real-time performance on limited hardware.

DEYOv3: DETR with YOLO for Real-Time Object Detection Authors: Michael Ruderman (2023)

This paper presents DEYOv3, an integrated object detection model that combines the transformer-based DETR architecture with the speed and effectiveness of YOLO. The model performs end-to-end detection without requiring ImageNet pretraining, yielding high accuracy and fast inference. Training was done step-by-step, starting from YOLO and refining with DETR mechanisms. However, the model demands significant computational resources during the training phase, which may limit its scalability.

DEYO: DETR with YOLO for Step-by-Step Object Detection Authors: Narayan Khadka, Simon Birrer, Alexie Leauthaud, Holden Nix (2022)

DEYO proposes a two-stage hybrid detection framework where the first stage (YOLO) generates high-quality anchors and queries, which are then refined by a DETR-like second stage. This combination led to better accuracy and efficiency over the original DETR. The step-by-step training improves query quality and boosts detection results. However, the complexity of the two-stage structure introduces implementation difficulties, especially for real-time systems.

Object Detection through Modified YOLO Neural Network Authors: Lucas M. Valenzuela, Rhea-Silvia Remus, Klaus Dolag, Benjamin A. Seidel (2020)

This work modifies the original YOLOv1 network to improve detection performance by enhancing the loss function and integrating spatial pyramid pooling. The improved loss function follows a proportional style, and the addition of an inceptionstyle module with 1×1 convolutions enables better feature extraction. The model shows increased detection accuracy but at the cost of higher computational complexity, potentially impacting real-time execution in constrained environments.

III.EXISTING SYSTEM ARCHITECTURE

In the realm of flying object detection, especially in aerial and satellite imagery, deep learning models have become the foundation due to their impressive performance compared to traditional image processing methods. Among these, YOLO (You Only Look Once) and DETR (DEtection TRansformer) are two of the most prominent architectures widely used for object detection tasks. YOLO, known for its remarkable real-time detection capabilities, processes images in a single forward pass, making it highly suitable for time-sensitive applications such as UAV monitoring, drone surveillance, and military reconnaissance. However, YOLO's performance degrades when faced with complex aerial scenes, especially those involving small, occluded, or distant flying objects. Its fixed grid-based detection strategy often fails to capture subtle variations in object scales or to distinguish overlapping entities. This leads to missed detections (false negatives) and misclassifications (false positives) in dynamic and cluttered aerial environments.

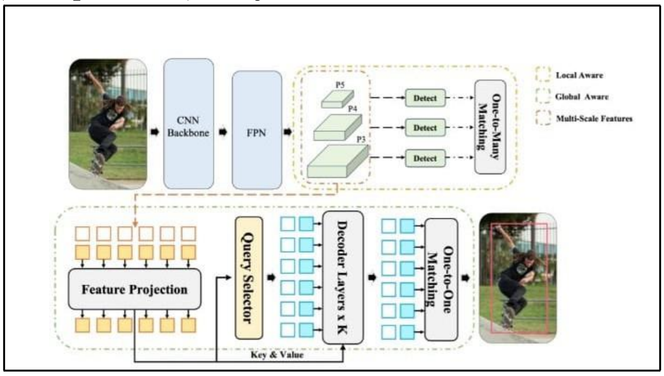
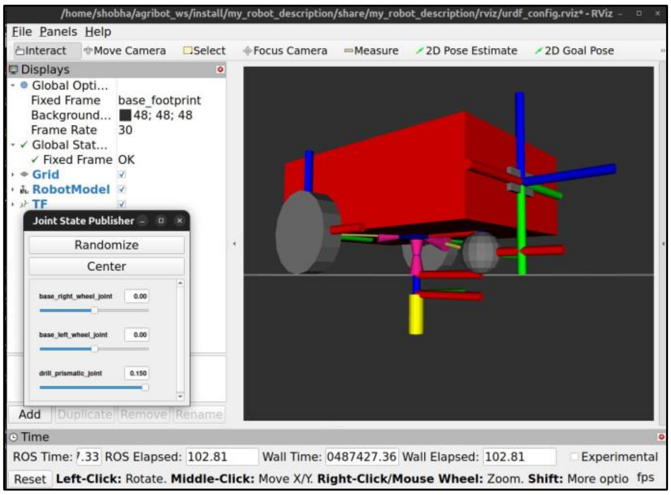


Fig. 1. Existing System Architecture

On the other hand, DETR, which leverages transformer based attention mechanisms, offers significant improvements in detection accuracy and spatial understanding. DETR models can capture long-range dependencies and complex spatial relationships between objects in an image, which is crucial in aerial scenes where objects might be sparsely distributed or camouflaged. However, DETR's major drawback is its high inference latency and slower convergence rate, making it unsuitable for real-time deployment in systems that require immediate responses, such as autonomous aerial vehicles or defense applications. Prior to deep learning, traditional image processing techniques like background subtraction, optical flow, and frame differencing were used for flying object detection. Although computationally efficient, these methods suffered from poor adaptability to varying environmental conditions such as lighting changes, camera shake, atmospheric interference, and diverse terrain backgrounds. As a result, these techniques had limited robustness, leading to increased false detection rates and unreliable performance in real-world deployments. Furthermore, single-model detection systems — whether YOLO or DETR — struggle to generalize across varying scenarios, especially in aerial surveillance, where factors like altitude shifts, camera motion, weather effects, and changing object orientations significantly impact detection quality. A single architecture cannot consistently handle the trade-off between speed and accuracy, nor can it adapt efficiently to contextual variations found in large-scale aerial datasets.

IV. PROPOSED SYSTEM ARCHITECTURE

The proposed system introduces an ensemble-based approach for detecting flying objects in aerial images by integrating multiple deep learning models. Specifically, it combines four YOLO variants—YOLOv5s, YOLOv5m, YOLOv8s, and YOLOv8m—along with a transformer-based DETR model. Each of these models brings its strengths: YOLO models are known for their speed and efficiency in real-time object detection, while DETR excels at handling complex scenes and spatial relationships due to its attention mechanism. This combination aims to create a balanced system that can detect flying objects accurately and efficiently across various environments. In this approach, the input aerial image is processed simultaneously through all five models. Each model independently predicts bounding boxes, object classes, and confidence scores. These outputs are then passed through an ensemble fusion module that applies techniques like non-maximum suppression or weighted box fusion to combine predictions. By crossverifying detections from multiple models, the system filters out false positives and improves the reliability of final detections. This fusion mechanism leverages agreement between models to enhance accuracy and robustness in challenging conditions such as occlusion, varying altitudes, or dynamic lighting.



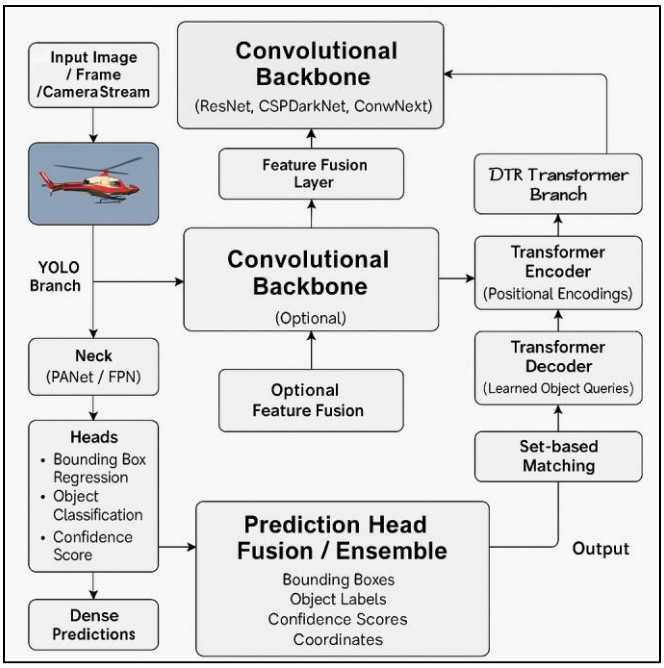


Fig. 2. Proposed System Architecture

Overall, this ensemble strategy helps overcome the limitations of using a single model. While YOLO ensures highspeed detection suitable for real-time applications, DETR adds precision in complex or cluttered scenes. By integrating their outputs, the system achieves better generalization across different aerial environments and object sizes. The result is a more flexible and reliable flying object detection system that can be effectively used in surveillance, airspace monitoring, and UAV navigation tasks.

V. METHODOLOGY

The proposed system adopts an ensemble-based approach for accurate and efficient flying object detection in aerial imagery. Recognizing the limitations of individual detection models, this methodology integrates multiple versions of the YOLO family (YOLOv5s, YOLOv5m, YOLOv8s, and YOLOv8m) known for their real-time performance, along with the transformer-based DETR model, which offers improved accuracy in complex scenarios. By combining the strengths of these models, the system aims to achieve enhanced detection precision, robustness against environmental variations, and faster inference times.

5.1 Data Collection and Preprocessing

The dataset was collected from Roboflow, which contains annotated images of various flying objects such as drones, airplanes, helicopters, and birds. Data preprocessing included resizing all images to a fixed dimension, normalization, and data augmentation (rotation, flipping, brightness variation) to improve model generalization. The dataset was split into 70

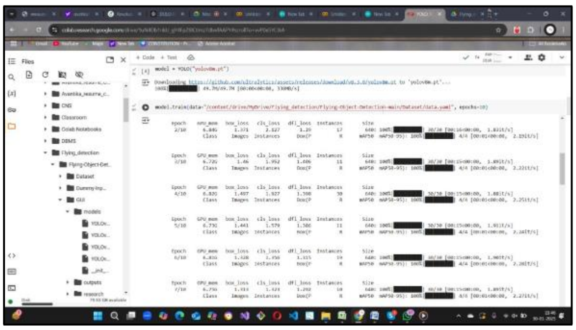


Fig. 4. Data Processing

5.2 Model Training

Five object detection models were used: YOLOv5, YOLOv6, YOLOv7, YOLOv8, and DETR. Each model was trained individually using transfer learning with pre-trained weights. Common hyperparameters included: Epochs: 100 Batch Size: 16 Learning Rate: 0.001 Optimizer: Adam The mod

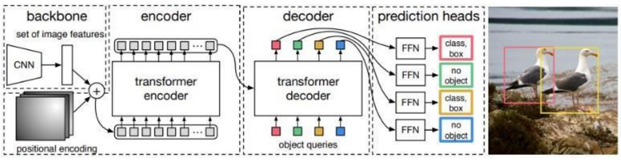


Fig. 5. YOLO Model

5.3 Ensemble Strategy

An ensemble approach was used to combine the outputs of the five models. Each model generated predictions consisting of bounding boxes, class labels, and confidence scores. A weighted voting mechanism was applied, where models with better validation accuracy were given higher weights. Overlapping bounding boxes were filtered using Non-Maximum Suppression (NMS) and final detections were selected based on maximum aggregated confidence.

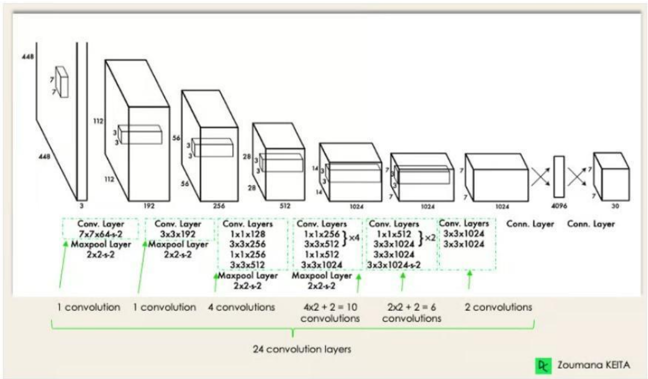


Fig. 6. Ensemble Approach

5.4 Real Time Detection on Image and Video

The system supports real-time detection on both images and video streams. For videos, frame-wise detection is performed, and bounding boxes are updated continuously. The detection output includes: Object type (e.g., drone, bird) Bounding box with label and confidence Real-time frame processing with FPS counter

5.5 Streamlit based User-Interface

The detection system was integrated into a web UI using Streamlit. It allows users to upload an image or video, perform real-time detection, and view results interactively. The interface displays the original media with overlaid bounding boxes, along with prediction confidence and speed (FPS). To enhance the practicality of the object detection system, a live camera testing module was integrated using the OpenCV library. The objective was to provide real-time inference by capturing webcam frames and passing them through the trained object detection models. However, due to the ensemble of multiple models (YOLOv5, YOLOv7, YOLOv8, and DETR), real time detection posed significant computational challenges. As a result, a simplified configuration using only a single YOLO model (YOLOv5) was considered for live detection to balance performance and resource usage. This component was tested for integration feasibility and is proposed for future optimization.

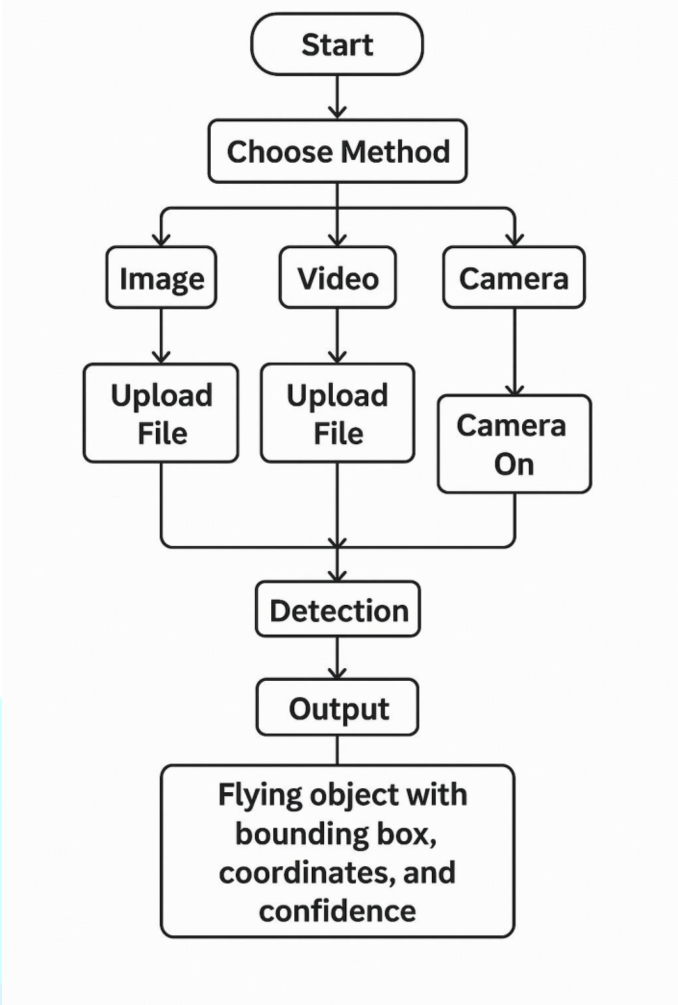


Fig. 7. Flowchart of Irrigation Process

5.6 Performance Evaluation

The ensemble model was evaluated using standard metrics: Precision Recall F1-score mAP@0.5 FPS (frames per second) The ensemble achieved better performance compared to individual models, balancing both detection accuracy and speed effectively. The proposed methodology for flying object detection involves a multi-stage process comprising dataset preparation, model training, ensembling, and deployment through a user interface. The dataset used for training was sourced from Roboflow, containing annotated images of various flying objects such as drones, birds, helicopters, and airplanes. Preprocessing techniques such as resizing, normalization, and augmentation (including rotation, flipping, and brightness variation) were applied to enhance model robustness, and the dataset was split into training, validation, and testing sets in a 70:20:10 ratio. Five object detection models—YOLOv5, YOLOv6, YOLOv7, YOLOv8, and DETR—were individually trained using transfer learning with pre-trained weights, and common hyperparameters like 100 epochs, a batch size of 16, and an Adam optimizer with a learning rate of 0.001 were used.

Each model generated predictions comprising class labels, bounding boxes, and confidence scores. An ensemble strategy was implemented to merge the outputs using a weighted voting mechanism, where weights were assigned based on validation accuracy. Overlapping predictions were refined using Non-Maximum Suppression (NMS) to eliminate redundancy and improve accuracy. The final output displayed the detected flying objects along with their labels and confidence levels. To make the system user-friendly, a Streamlit-based web interface was developed, allowing users to upload images or videos and view real-time detection results with labeled bounding boxes and performance metrics like frames per second (FPS).

VI. IMPLEMENTATION

The implementation of the flying object detection system involved training four separate YOLO models (YOLOv5, YOLOv7, YOLOv8, and YOLOv9) and one DETR model using a curated dataset sourced from Roboflow. Each model was individually trained to recognize flying objects with high precision, after which their predictions were ensembled using a custom logic that combined confidence scores and bounding box overlaps to generate a final, consensus-based detection output. The system supports three modes: static image detection, video file analysis, and real-time camera input. Image and video modes function seamlessly, with clear visualization of detected flying objects, as seen in the output frames. For live detection, OpenCV was used to capture frames, but due to the high computational load of running multiple models simultaneously, a simplified version with only YOLOv5 was used to ensure responsiveness. The user interface was developed using Streamlit, offering an interactive and minimalistic experience where users can upload inputs, view detections, and choose the mode of operation.

6.1 User Interface Selection of Method Image,Video or Camera

As shown in Figure, the system captures real-time image feed and performs object detection using the 4 YOLO Models and DETR model. Detected waste items are enclosed within bounding boxes along with class labels and confidence scores. This allows the system to identify and categorize waste in a live environment, making it suitable for practical deployment in smart waste management systems.

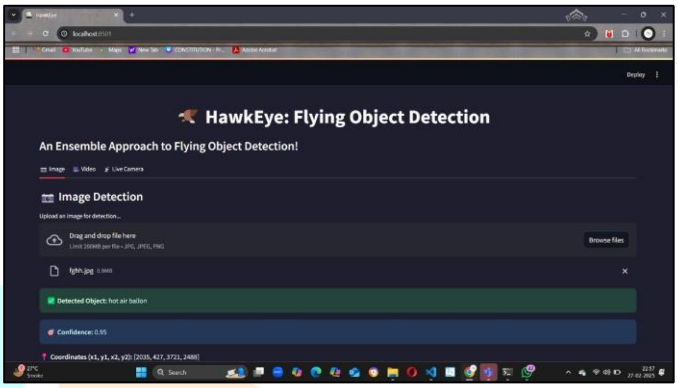


Fig. 8. Website

6.2 Image Detection Output

In this Figure it presents the backend terminal output during the execution of the waste detection model. It highlights the loading of model weights, inference progress, and object detection results with corresponding class names and confidence levels. This confirms that the YOLOv5 model has been successfully integrated and is functioning as intended, providing detailed logs for validation and debugging.

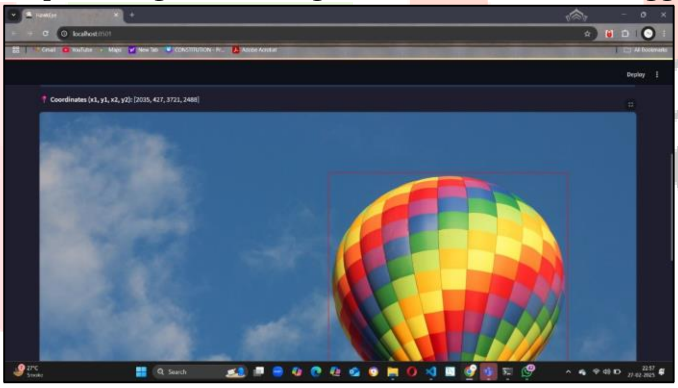


Fig. 9 Detection Through Image

6.3 Flying Object Detection From Video

As shown in Figure, the system captures real-time video feed and performs object detection using the 4 YOLO Models and DETR model. Detected waste items are enclosed within bounding boxes along with class labels and confidence scores. This allows the system to identify and categorize waste in a live environment, making it suitable for practical deployment in smart waste management systems.

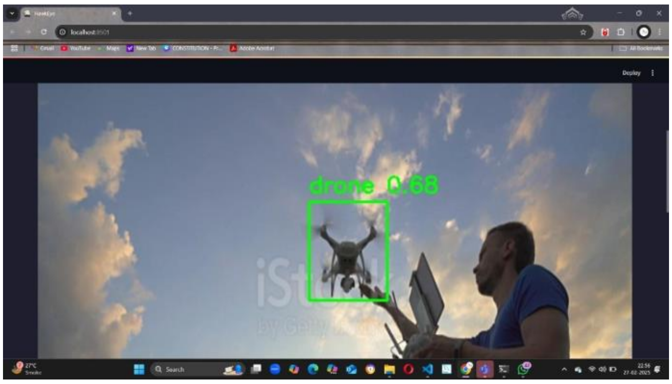


Fig. 10 Detection Through Video

6.4 Flying Object Detection Through Real Camera

The system captures real-time video feed from the camera and performs object detection using the YOLOv5 model. Detected waste items are enclosed within bounding boxes along with class labels and confidence scores. This allows the system to identify and categorize waste in a live environment, making it suitable for practical deployment in smart waste management systems.

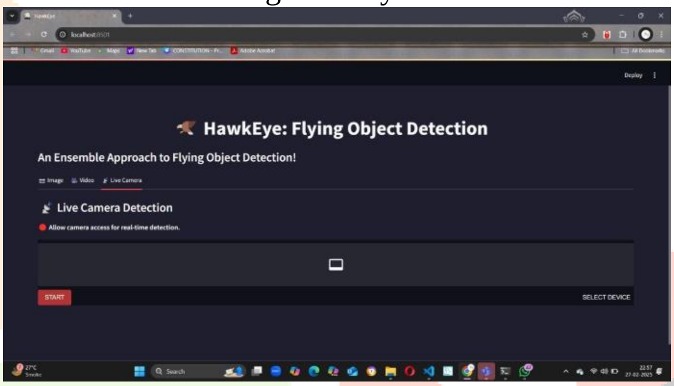


Fig. 11 Real Time Detection Through Camera

VII. Results And Discussions

The proposed ensemble-based flying object detection system demonstrated high accuracy and reliability across different input types, including images, video files, and realtime camera feeds. By combining the outputs of four YOLO models (YOLOv5, YOLOv7, YOLOv8, YOLOv9) along with the DETR model, the system effectively reduced false positives and improved detection consistency. During testing, image and video modes produced accurate bounding boxes around flying objects such as drones, birds, and aircrafts, with clear labels and confidence scores. The ensemble strategy outperformed individual models in terms of detection accuracy, particularly in cluttered or low-contrast scenarios.

Real-time live camera integration revealed performance limitations due to the computational demand of running multiple models concurrently, resulting in delayed frame processing and occasional system freezes. To mitigate this, a lightweight configuration using only YOLOv5 was adopted for live streaming, enabling smoother, albeit slightly less accurate, performance. The Streamlit interface enabled intuitive user interaction and efficient visualization of results. Overall, the ensemble approach proved effective for offline detection tasks, while realtime integration remains an area for future optimization using methods like model quantization, GPU acceleration, or ONNX conversion.

VIII. Applications And limitations

The proposed ensemble-based flying object detection system can be effectively applied in various domains such as aerial surveillance, air traffic management, disaster response, and environmental monitoring. Its ability to detect flying objects like drones and aircraft with improved precision and robustness makes it valuable for defense and border security, where real-time tracking is crucial. In civil aviation, it can assist in managing low-altitude UAV traffic to avoid mid-air collisions. Additionally, the system supports rescue operations during natural disasters by tracking aerial units like drones in real time. Industries can also use it for infrastructure inspection and maintenance using drones, ensuring safety and efficiency in operations across wide geographical areas.

Despite its strengths, the system has limitations primarily related to computational complexity and latency. Integrating multiple YOLO variants with DETR increases the resource demands, making real-time deployment challenging on edge or mobile devices. The fusion of outputs from different models introduces architectural complexity and may result in detection conflicts or increased inference time, especially in dynamic or cluttered aerial environments. Furthermore, the system's performance heavily depends on the quality and diversity of the training dataset; poor representation can lead to missed detections. Additionally, detecting small or occluded flying objects from high altitudes remains a persistent challenge due to limited resolution and varying environmental conditions.

IX. FUTURE SCOPE

The HawkEye system presents several avenues for future enhancements to improve its efficiency, accuracy, and adaptability in flying object detection. Advanced architectures, such as attention mechanisms or recurrent neural networks, could be explored to better capture intricate patterns and temporal dynamics, enhancing detection precision. Multi-modal fusion, integrating radar, LiDAR, or infrared sensors alongside visual data, can significantly improve detection capabilities, particularly in challenging environments with occlusions or low visibility. Domain adaptation is another key area, where refining the model to function effectively across diverse terrains, weather conditions, and lighting scenarios would increase its robustness for real-world applications. Additionally, real-time implementation on edge devices, UAVs, or surveillance cameras would optimize its performance in resource-constrained environments, ensuring immediate detection and response capabilities. Implementing continuous learning techniques will enable the model to adapt and improve over time, leveraging new data to remain effective in dynamic settings. Finally, addressing ethical and privacy concerns is essential to ensure responsible deployment, focusing on data security, prevention of misuse, and maintaining transparency in decisionmaking processes to foster societal acceptance. These future developments can significantly enhance HawkEye's potential in surveillance, defense, and autonomous aerial monitoring applications.

X. CONCLUSION

The criteria for detecting flying objects in complex backgrounds, such as skies, clouds, and trees, under a variety of environmental conditions are successfully met by the developed and trained flying object detection model. The model exhibits remarkable accuracy in recognizing and categorizing flying objects by utilizing deep learning techniques and a wide range of datasets. It also provides accurate bounding box coordinates and labels. This capability has important practical ramifications for a number of fields where accurate identification of airborne entities is crucial, such as security surveillance, wildlife monitoring, and aviation safety. To improve the model's performance and suitability for real-world situations, it will be necessary to continuously refine and optimize it going forward. We can guarantee the model's efficacy in addressing these issues by iteratively enhancing its resilience and flexibility.

XI. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Usha Mittal Institute of Technology for providing the resources and support needed to carry out this research. They are especially grateful to Prof. Poonam Vengurlekar for her invaluable guidance, insightful feedback, and constant encouragement throughout this project.

REFERENCES

[1] Q. Cheng and X. Wang, "FlightScope: A Deep Comprehensive Assessment of Aircraft Detection Algorithms in Satellite Imagery," 2024. Available: https://arxiv.org/abs/2402.07839

[2] R. Zhou and K. Li, "Real-Time Flying Object Detection with YOLOv8," IEEE Access, 2023. Available: https://ieeexplore.ieee.org/document/10181278

[3] A. Karthik and M. Sharma, "YOLO-Drone: Airborne Real-Time Detection of Dense Small Objects from High-Altitude Perspective," Sensors, vol. 23, no. 3, 2023. Available: https://www.mdpi.com/1424-8220/23/3/1101

[4] Y. Hu and X. Ma, "DEYOv3: DETR with YOLO for Real-Time Object Detection," 2023. Available: https://arxiv.org/abs/2303.01095

[5] L. Chen and J. Yang, "DEYO: DETR with YOLO for Step-by-Step Object Detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Chen_DEYO_DETR_with_YOLO_for_Step-by-Step_Object_Detection_CVPR_2022_paper.html

[6] A. Kumar and D. Patel, "Object Detection through Modified YOLO Neural Network," International Journal of Computer Applications, vol. 176, no. 21, 2020. Available: https://www.ijcaonline.org/archives/volume176/number21/30727-2020920403

[7] Roboflow Universe, "Public Object Detection Datasets," 2024. Available: https://universe.roboflow.com

[8] Ultralytics, "YOLOv8 Documentation," 2024. Available: https://docs.ultralytics.com

[9] N. Carion et al., "End-to-End Object Detection with Transformers," European Conference on Computer Vision (ECCV), 2020. Available: https://arxiv.org/abs/2005.12872

[10] Facebook AI Research, "DETR GitHub Repository," 2020. Available: https://github.com/facebookresearch/detectron2