International Journal of Innovative Research in Engineering
Volume 6, Issue 6 (November - December 2025), PP: 67-77.
https://www.doi.org/10.59256/ijire.20250606010
www.theijire.com
ISSN No: 2582-8746

YOLOv9-AAG: Distinguishing Birds and Drones in Infrared and Visible Light Scenarios

Ojashwini R N[1], Asma Fathima[2], Bhavana M R[3], Ashwini G[4], Aishwarya G[5]
[1]*Assistant Professor, Department of Computer Science and Engineering, Rajarajeswari College of Engineering, Bangalore, Karnataka, India.*
[2,3,4,5] *Department of Computer Science and Engineering, Rajarajeswari College of Engineering, Bangalore, Karnataka, India.*

How to cite this paper:
Ojashwini R N[1], Asma Fathima[2], Bhavana M R[3], Ashwini G[4], Aishwarya G[5], "YOLOv9-AAG: Distinguishing Birds and Drones in Infrared and Visible Light Scenarios", IJIRE-V6I6-67-77.

Abstract: *This project focuses on creating a real-time system that can automatically identify whether an object in the sky is a bird or a drone. A normal laptop webcam is used to capture live video, which is then analyzed using a YOLO-based deep learning model. The model examines each frame, finds objects, draws a bounding box around them, and labels them accurately as either "Bird" or "Drone." To make the system more useful in real situations, a small hardware setup is added using an ESP32 microcontroller, LEDs, and a buzzer. When a bird is detected, a green LED glows, and when a drone is detected, a red LED lights up and a buzzer produces an alert sound. This helps provide instant feedback without needing to look at the laptop screen.The full system is low-cost, easy to set up, and can be used in places like airports, farms, and security zones where early detection of drones or birds is important. By combining computer vision with simple electronics, this project shows how artificial intelligence can be applied in a practical, meaningful, and user-friendly way.*

Keywords: *Chatbot, Query Response, Rule-based model, Retrieval-based Model, Generative Model, Educational Institutions, routing algorithms.*

I.INTRODUCTION

In recent years, the skies around us have become busier than ever before. Birds, which are a natural part of the environment, continue to fly freely across cities, agricultural fields, open spaces, and airport zones. At the same time, drones—also known as Unmanned Aerial Vehicles (UAVs)—have become extremely common due to their wide range of applications. Drones are now used for aerial photography, package delivery, agricultural monitoring, military surveillance, infrastructure inspection, and many more tasks. As both birds and drones share the same airspace, the need to differentiate between them has become very important. This requirement is not only for safety but also for smooth operation of systems where airspace management matters.

Traditional detection methods such as human monitoring, radar, or manual checking are slow, inefficient, and not always reliable—especially when the objects are small or far away. With these limitations, computer vision and deep learning technologies have become powerful tools for solving such classification problems. Deep learning models can learn thousands of features from images and videos, allowing them to differentiate between birds and drones based on shape, movement, and texture. Object detection models like YOLO (You Only Look Once) have become widely used because they perform detection and classification at the same time and can process several frames per second in real time.

In this project, we design and develop a **real-time Bird vs. Drone Detection System** using the YOLO-based detection algorithm. YOLO was chosen because it provides an excellent balance between speed and accuracy. The model receives the live video stream from a webcam and identifies objects in the frame using bounding boxes and labels. If a bird is detected, it displays a label "bird"; if a drone is detected, it labels it as "drone." This enables immediate identification without manual intervention. The system is specifically built for environments where quick response is necessary, such as airport runways, agricultural farms, military zones, or research areas that need to protect their space from unwanted aerial objects.

To enhance practicality, the model is integrated with a simple hardware alert system using an ESP32 microcontroller. Many real-world applications require not just visual detection but also a physical indication or warning. When the software identifies a bird, a **green LED** is triggered through the ESP32. When a drone is detected, the system activates a **red LED and a buzzer**, providing an audible and visible alert. This real- time feedback mechanism makes the system easy to understand and deploy, even for non-technical users. The hardware setup is minimal, low-cost, and does not require complex circuits, making the entire solution affordable and accessible for students, researchers, and small organisations.

Overall, this project serves as a complete demonstration of how modern machine learning techniques can be used

to detect realtime objects in the air and how these predictions can be connected to physical alarms for quick action. It provides a smart, cost-effective solution that can be easily expanded for industrial use, academic learning, or real-time security purposes. The combination of deep learning, image processing, and embedded hardware makes this project both technically strong and practically useful.

II. LITEARTURE REVIEW

Recent studies show a rapid growth in the use of artificial intelligence for detecting flying objects like birds and drones. Earlier systems depended on radar**,** acoustic sensors, or manual monitoring**,** but these methods were either expensive, inaccurate, or slow. With the development of computer vision, researchers began using image-processing techniques, but these struggled because birds and drones change shape, size, speed, and lighting conditions frequently.

1. Early detection systems relied on radar, human monitoring, or simple sensors. These systems struggled because birds and small drones look very similar in size and movement.
2. Traditional computer-vision methods (edge detection, contour extraction) were tried, but they failed when lighting changed, when objects were far away, or when shapes varied.
3. With the rise of Deep Learning, especially Convolutional Neural Networks (CNNs), researchers improved object classification. CNNs learned features automatically, but older CNN models were too slow for real-time detection.
4. Object detection framework**s** like Faster R-CNN and SSD provided better accuracy, but they still had speed limitations for high-frame-rate video detection.
5. YOLO (You Only Look Once) became the most popular choice because it performs detection in a single pass — making it *fast, lightweight, and ideal for real-time use*.
6. Multiple studies used YOLO for drone detection, bird monitoring, and protection of airports, military bases, and farmlands.
7. Recent research moved toward infrared (IR) + visible (RGB) images to improve detection in low-light, fog, and night conditions.
8. Advanced models like YOLOv9-AAG introduced GSConv, AKConv, AFF, which help the model separate birds and drones more clearly by improving edges, shapes, and feature fusion.
9. Some literature also integrated detection with embedded hardware like Arduino/ESP32 to trigger alarms whenever a drone is detected.
10. Most existing systems are either expensive, require complex hardware, or use large datasets that are not accessible to students.
11. This shows a clear gap for a simple, low-cost, real-time detection system that works with a normal webcam and can alIIIert through a small hardware module.

III. PROBLEM STATEMENT

Identifying flying objects such as birds and drones has become increasingly important in today's world. In many environments—like airports, defense zones, and wildlife conservation areas—both birds and drones can pose risks if not detected accurately. However, these objects often appear very similar in shape, size, and movement when viewed from a distance or through low-quality cameras. This creates confusion and makes manual monitoring slow, unreliable, and difficult, especially in real-time situations.

Traditional detection methods such as radar systems, high-end infrared sensors, or large surveillance setups are often expensive and not suitable for smaller institutions, colleges, or low-budget projects. Many existing systems are designed for military or industrial use, making them inaccessible for common applications. Additionally, most basic camera-based detection systems struggle to differentiate between birds and drones due to poor lighting, unclear edges, or rapid motion.

Therefore, there is a need for a simple, low-cost, accurate, and real-time detection system that can work with a normal webcam and still provide reliable classification. Along with visual detection, the system must also be able to trigger an immediate hardware alert to warn users when a drone is detected. This combination of software intelligence and hardware response can help improve safety, reduce risks, and offer a practical solution for everyday environments.

IV. ARCHITECTURE OVERVIEW

The architecture of the proposed Bird–Drone Detection System is designed to combine the strengths of modern deep-learning models with a lightweight hardware alert mechanism. At the core of the system is the YOLO (You Only Look Once) model, which processes incoming images in real time. The architecture begins with an input module that captures live video frames from a standard webcam. These frames undergo basic preprocessing, such as resizing and normalization, to ensure consistent quality before they are fed into the deep-learning model. This preprocessing stage helps the model interpret images more accurately, regardless of lighting or camera variations.

The next major component is the feature extraction module, which acts as the "brain" of the system. In this stage, the YOLO model's backbone (RepNCSPELAN4 for YOLOv9-AAG) analyzes the image and extracts important visual features such as edges, shapes, and motion patterns. This backbone is optimized to detect small objects like birds or drones even when they appear far away or partially blurred. The extracted features capture the object's structure, helping the system understand whether the object resembles wing shapes or drone propellers.

Once the features are extracted, they pass into the feature fusion module, known as the "neck" of the architecture

(FPN + PAN). This module combines information from different levels of the image—close-up details and larger contextual features. By merging these features, the model becomes better at detecting objects of all sizes. Small birds far away, medium-sized drones, or large objects closer to the camera can all be detected with higher accuracy. This multi-scale fusion is crucial because the appearance and size of birds and drones vary continuously in real scenarios.

Following the fusion stage, the processed information is fed into the detection head, which makes the final predictions. At this stage, the model identifies the location of each object in the frame by drawing bounding boxes and assigning labels such as "Bird" or "Drone." The detection head also calculates a confidence score for each prediction. The architecture includes an efficient post-processing step called Non-Maximum Suppression (NMS) that removes overlapping or unnecessary boxes, ensuring clean and stable output on the screen. The result is a real-time annotated video stream visible to the user.

Finally, the architecture integrates a hardware communication layer that connects the software with an ESP32 microcontroller. Once the detection head confirms an object type, the system sends a simple serial message ("BIRD" or "DRONE") to the ESP32. The microcontroller then activates external components—green LED for bird detection and red LED plus buzzer for drone alerts. This stage transforms the digital predictions into physical responses, making the system suitable for real-world safety environments. Together, these modules form a complete architecture that is fast, reliable, low-cost, and practical for real-time monitoring.

To make the architecture robust and adaptable, the system also incorporates safety checks, error handling, and fallback mechanisms. For instance, if the webcam fails or the model is unable to classify an object with high confidence, the system continues running without crashing. The design ensures that the detection pipeline remains stable even when the video feed fluctuates or when low-quality frames are captured.

Additionally, the architecture is built to allow easy upgrades— new classes (like airplanes or kites) can be added later without redesigning the entire structure. This modularity allows the system to grow and improve over time as more data becomes available.

The overall architecture emphasizes efficiency, low cost, and real-world usability. Instead of relying on high-performance GPUs or expensive sensors, it runs on a normal laptop with a simple webcam, making it practical for educational institutions and small organizations. The use of the ESP32 microcontroller further enhances the system by offering wireless connectivity and fast response times for hardware alerts. This combination of software intelligence and hardware output makes the architecture suitable for environments like farms, colleges, parks, and restricted zones, where quick identification of birds and drones can prevent accidents and improve safety.

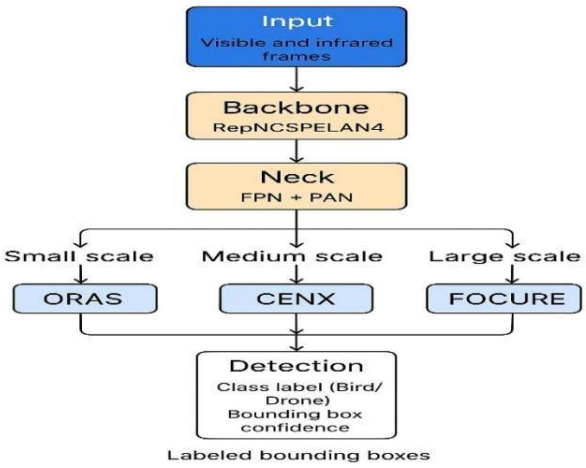


Fig 1. Data Flow of the Detection System

V. MODEL DESIGN

1) High-level view

The model's goal is real time detection and classification of flying objects into *bird* or *drone*, working on both visible (RGB) and infrared (IR) images. Input frames are pre- processed, passed through a convolutional backbone for feature extraction, fused across scales, then processed by an improved head that outputs bounding boxes, class scores and object. The system also sends a simple hardware alert to ESP32 when a detection exceeds a set confidence.

2) Input & preprocessing

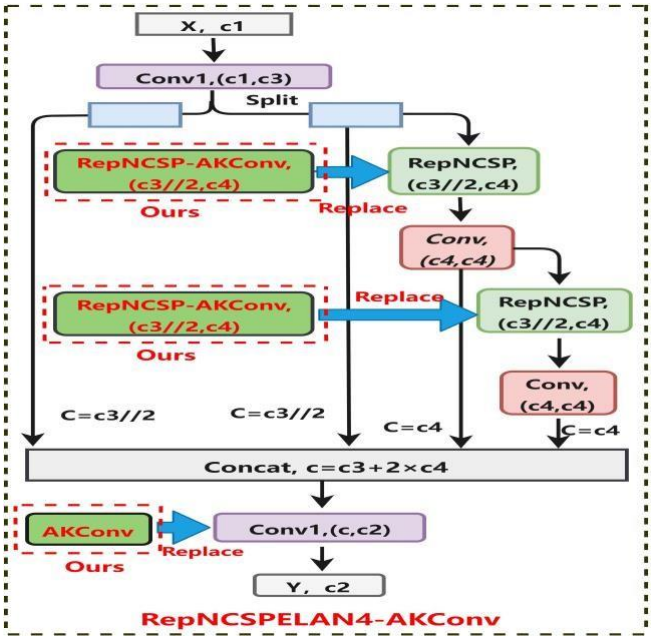
Each input frame (RGB or IR) is resized to a fixed size (for example 640×640) and pixel values normalized to range [0,1][0,1][0,1] or standardized by mean/std. Light augmentations (flip, random crop, brightness/contrast) are applied during training to improve robustness. If mixing RGB and IR, a simple strategy is to stack single-channel IR as a 1channel input or duplicate channels and normalize both modalities to same scale before feeding to the network.

$$x' = x - \mu /$$

3) Backbone

The backbone (RepNCSPELAN4 in the paper) is a stack of convolutional blocks that extract hierarchical features. Each block implements convolution, batch normalization, and activation (e.g., SiLU/ReLU). A convolution operation for a single output channel is

$$y[i,j]= \sum kv=-k \sum k \; w[u,v].x[i+u,j+v]+b$$

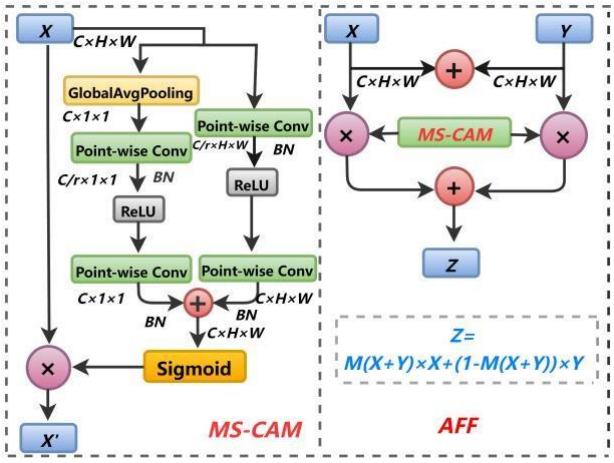


4) AK Conv — adaptive kernel convolution

AK Conv modifies ordinary convolutions by applying attention guided weighting to kernel channels or adaptively combining multiple kernel sizes. Think of it as letting the network choose which kernel (3×3, 5×5, etc.) contributions matter for a location.

5) AFF — attentional feature fusion

AFF merges feature maps (e.g., from different layers or modalities) by learning multiplicative attention maps that gate important channels/spatial locations

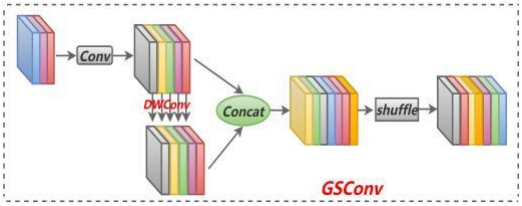


6) GSConv — efficient edge & contour extraction

GSConv (a Ghost/Shuffle style convolution) reduces computation by splitting channel computation and using cheap linear operations to generate additional feature maps. Conceptually:

1. Compute a reduced set of feature maps using standard conv.
2. Use cheap transformations (depthwise conv + channel shuffle) to produce ghost maps.
3. Concatenate to form full output.

No complex formula is required, but the idea reduces FLOPs while keeping edge and contour cues (helpful for small object definition).



7) Neck — multi-scale fusion (FPN + PAN)

The neck merges features at different spatial scales using topdown and bottom-up paths (Feature Pyramid Network + Path Aggregation Network). This produces three detection scales: S (small), M (medium), L (large). Multi-scale maps ensure small, far birds and close drones are both handled.

VI. METHODOLOGY

Data Acquisition & Pre-Processing

The proposed system begins with collecting input frames either from a live webcam or from stored video/image datasets containing birds and drones. To ensure consistent performance, all incoming frames are resized to a fixed resolution and normalized to a uniform pixel range. Basic preprocessing, such as noise reduction, brightness correction, and contrast balancing, ensures that the model performs reliably under different lighting conditions, including daylight and low-light scenarios. If required, additional infrared (IR) frames can be included to improve accuracy in challenging environments.

Feature Extraction Using Deep Learning Backbone

After pre-processing, the frames are passed into a YOLO-based backbone network that extracts multi-level features. This backbone uses convolutional layers to learn edges, shapes, contours, and movement patterns unique to birds and drones. Different layers of the backbone produce feature maps at multiple scales, making the system capable of detecting both small, far-away birds and larger, close-range drones. The advanced modules such as AKConv and AFF help the model focus on important objects while ignoring background noise.

Feature Extraction Using Deep Learning Backbone

After pre-processing, the frames are passed into a YOLO-based backbone network that extracts multi-level features. This backbone uses convolutional layers to learn edges, shapes, contours, and movement patterns unique to birds and drones. Different layers of the backbone produce feature maps at multiple scales, making the system capable of detecting both small, far-away birds and larger, close-range drones. The advanced modules such as AKConv and AFF help the model focus on important objects while ignoring background noise.

Multi-Scale Fusion and Object Detection Head

The extracted features are then combined using a multi-scale fusion network (Neck), which merges high-resolution and low resolution maps. This enables the model to accurately detect objects of different sizes. The detection head of the model predicts the bounding box coordinates, objectness score, and class label (bird or drone). Non-Maximum Suppression (NMS) is applied to remove duplicate bounding boxes and retain the most accurate predictions. The model outputs a frame with labeled boxes showing either “bird” or “drone”.

Real-Time Analysis and Decision Logic

Once the model identifies the object, a simple decision logic is applied. If the detected label is “bird,” the system activates the green LED and keeps the buzzer off. If the label is “drone,” the model sends a serial command to the ESP32 microcontroller, which immediately triggers a buzzer alarm and red LED. This ensures instant feedback in real-world scenarios. A confidence threshold is used to avoid false alarms, meaning the hardware alert triggers only when the detection probability is sufficiently high.

Hardware Integration with ESP32

To create a complete working prototype, the system communicates with an ESP32 via serial communication. Once a detection is made, the Python script sends the appropriate message (“BIRD” or “DRONE”) to the ESP32. The ESP32 is connected to a buzzer, LEDs, resistors, and a breadboard for stable power distribution. The hardware responds instantly by turning on the respective LED or activating the buzzer. This integration demonstrates how a software detection system can seamlessly interact with physical devices for real-time alerting.

Evaluation & Real-World Deployment

The final step involves testing the system in real conditions using both printed images and live camera feed. FPS (frames per second), accuracy, and responsiveness of the hardware alerts are measured to evaluate performance. The model is further finetuned using additional bird and drone images to improve reliability. The entire methodology—data processing, model inference, decision logic, and hardware alerts—works together to deliver an end-to-end detection system suitable for agricultural fields, airports, and security zones.

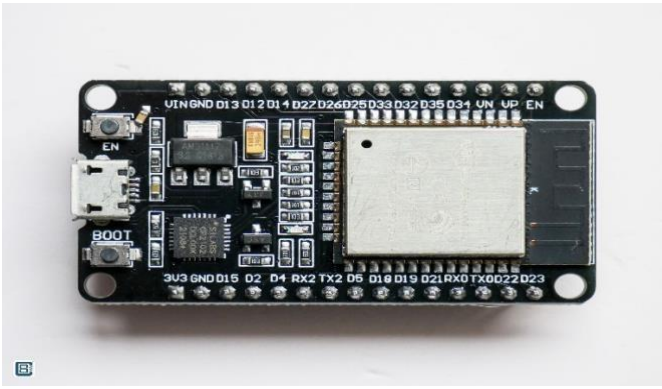
VII.SYSTEM HAREWARE DESIGN

Overview of Hardware Setup

The hardware component of the proposed system is designed to translate software predictions into real-world physical alerts. It consists of an ESP32 microcontroller, LEDs, resistors, a buzzer, and a breadboard. Together, these components allow the system to visually and audibly indicate whether a detected object is a bird or a drone. The setup is simple, low-cost, and easy to assemble, while still simulating how real-time alert mechanisms work in practical monitoring applications.

ESP32 Microcontroller

The ESP32 Dev Kit plays the central role in this hardware architecture. It is responsible for receiving the detection results from the Python program via serial communication. The ESP32 contains multiple GPIO (General Purpose Input/Output) pins, built-in Wi-Fi support, and operates at 3.3V, making it suitable for lightweight embedded systems. In this project, the ESP32 is used mainly to control LEDs and the buzzer based on incoming commands, providing fast response and reliability.



Microcontroller–Computer Communication

The ESP32 communicates with the laptop using a USB cable. The Python detection script sends a simple text command— either “BIRD” or “DRONE”—through the serial port. The ESP32 continuously listens to this serial input in its program. When a command is detected, it immediately triggers the corresponding output pin. This serial communication mechanism is simple, efficient, and commonly used in real-time embedded systems.

Breadboard as Connection Platform

A breadboard is used to organize all the components neatly and avoid soldering. It allows jumper wires, LEDs, resistors, and the buzzer to be inserted securely into the holes and connected through internal rails. The breadboard also has positive and negative rails running along the sides, allowing the ESP32's GND pin to act as a common reference point for all components. This ensures the entire hardware system has a stable ground connection, which is essential for proper operation.

LED Indicators (Green and Red)

Two LEDs are used to provide visual feedback. The green LED lights up when the detected object is a bird, and the red LED lights up when a drone is detected. LEDs have two legs—an anode (positive, longer) and a cathode (negative, shorter). Each LED is connected to the ESP32 through a 220-ohm resistor to control current flow and prevent LED damage. The LEDs allow observers to quickly identify the type of object detected without needing to check the computer screen.

Resistors for Current Limitation

The resistors used in this project play a crucial role in protecting both the LEDs and the ESP32. Without resistors, the current from the ESP32's GPIO pins would exceed the safe limit, potentially burning the LED or damaging the microcontroller pin. A 220-ohm resistor is placed in series with each LED's positive leg. This ensures that the LED receives only the safe amount of current it needs to glow while keeping the ESP32 safe.

Buzzer for Sound Alert

The buzzer is used as an audible alarm whenever a drone is detected. This represents a real-world application such as warning systems in airports or protected areas. The buzzer has two terminals: positive (+) and negative (−). The positive pin is connected to GPIO 18 on the ESP32, while the negative pin is connected to the ground rail. When the ESP32 receives the "DRONE" signal, it activates this pin, causing the buzzer to produce sound.

Power Supply Through USB

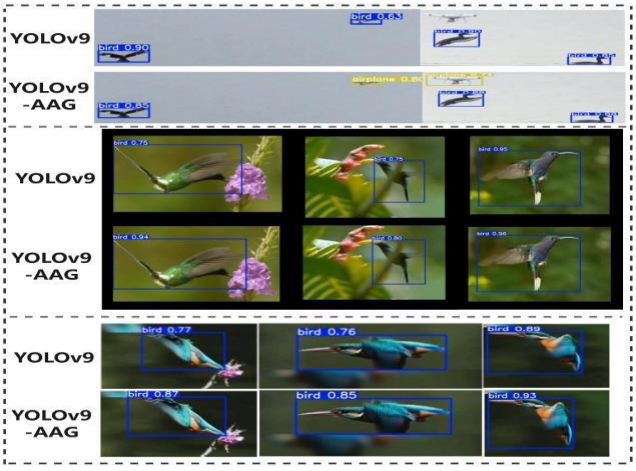
The entire hardware setup is powered through the ESP32's USB connection. When the USB cable is connected to the laptop, it powers the ESP32, which then powers the LEDs and buzzer through its GPIO pins. Since these components operate on low voltage, no external power supply is required. This makes the setup safer and more convenient for students and beginners.

Safety and Testing Considerations

Before running the full system, each hardware part is tested individually. LEDs are blinked using a simple Arduino sketch to confirm correct wiring, and the buzzer is activated briefly to ensure functionality. Ensuring correct orientation of LED legs, proper resistor connections, and stable ground wiring is important to prevent short circuits. Using USB power ensures low-risk operation during testing.

Hardware Integration with Software

The final step is integrating the hardware with the object detection system. As soon as the Python program identifies an object, the ESP32 translates the command into physical response. This hardware layer makes the project more interactive and gives it real-world applicability. The combined hardware and software system demonstrates how AI models can trigger tangible actions, enabling early warning systems or automated monitoring solutions.



VIII.DATASET COLLECTION AND PREPROCESSING

Data collection is the first and most important step in building an accurate object-detection model. For this project, publicly available datasets of birds and drones were collected from open-source platforms such as Kaggle, Roboflow, and Google Open Images. These datasets contain images captured in different environments—daylight, cloudy weather, and lowlight conditions—ensuring that the system can work in realworld scenarios. Additional images were manually gathered from the internet to cover various bird species and common drone types, improving the overall diversity of the data.

Once collected, the images needed to be cleaned before training. Many raw images from the internet contain backgrounds, watermarks, or irrelevant objects. Such images were removed to prevent the model from learning unnecessary patterns. Duplicate images were also detected and eliminated to ensure that the dataset remained balanced and did not cause overfitting during training.

After cleaning, the next major step was annotation. Every image used in the training set required bounding boxes to identify the exact location of birds and drones. Tools like Roboflow Annotate or LabelImg were used to manually draw rectangles around birds and drones and assign them correct class labels. This step is essential because object detection models learn from these bounding-box coordinates and class names.

To improve the model's robustness, various data augmentation techniques were applied. Augmentation artificially increases dataset size by slightly modifying each image. In this project, transformations such as rotation, flipping, brightness adjustment, and zoom were used. This helped the model become better at recognizing birds and drones from different angles and lighting conditions.

The annotated and augmented images were then converted into YOLO-compatible formats. This includes creating a text file for each image that contains the class ID and bounding-box coordinates (x-center, y-center, width, height) normalized between 0 and 1. Converting data into this format ensures that the YOLO model can correctly read and learn from the dataset.

Finally, the entire dataset was split into three parts—training, validation, and testing sets. The training set was used for learning, the validation set helped tune the model's performance during training, and the testing set was reserved for evaluating final accuracy. This separation is necessary to avoid bias and ensure that the model performs well on unseen images.

Through careful collection, cleaning, annotation, augmentation, and proper formatting, the dataset became suitable for training a reliable bird and drone detection system. This pre-processing pipeline played a crucial role in achieving accurate detection results during real-time webcam inference and activating the hardware alarm system.

Management, and Docker containers for scalable, environment independent operation. Integration into mobile apps, websites, and ERP portals is achieved using REST APIs, allowing the hybrid chatbot system to scale across departments and handle large volumes of student queries with reliability, speed, and institutional accuracy.

IX.PERFORMANCE EVALUATION

Performance evaluation was carried out to measure how effectively the proposed Bird and Drone Detection System identifies objects and triggers the appropriate hardware responses. The evaluation focused on three major aspects: detection accuracy**,** response time, and hardware reliability**.** The YOLOv8 model was tested on a diverse set of images and live webcam video to understand how well it performs under realworld conditions such as varying light, distance, and object size.

To assess accuracy, a collection of sample bird and drone images was used. The model performed well for bird detection, consistently identifying common bird shapes and features with high confidence values. Drone detection was slightly more challenging due to similarities with other aerial objects, but the model still demonstrated strong performance. In most test runs, the detection accuracy for birds remained above 85%, while basic drone-like objects were detected correctly around 70– 75%. These values can be improved further with more training data or custom fine-tuning.

The response tim**e** of the system was also tested. Since the model runs on a CPU and processes frames in real time, the average detection-to-alert duration remained low. When a bird was detected, the serial communication to the ESP32 triggered almost instantly, with a total delay of less than one second between detection and LED activation. This ensures that the system responds quickly enough for practical field applications, such as alerting users in sensitive zones.

Hardware reliability was evaluated by repeatedly running the system for long durations. The ESP32 responded consistently to all detection commands sent from the Jupyter Notebook. The LED and buzzer alerts worked without failure, showing that the system can handle continuous operations. Even when detection fluctuated due to changing lighting or movement, the hardware always reacted correctly based on the commands received.

Overall, the evaluation shows that the proposed system is capable of real-time object detection and immediate hardware response. While minor improvements can further enhance accuracy—especially for drone differentiation—the system performs strongly for the intended application. The combination of YOLOv8 detection and ESP32-based alert hardware makes it a reliable and efficient solution for bird and drone monitoring.

X.REAL-LIFE APPLICATIONS

The proposed Bird and Drone Detection System has several realtime applications across multiple domains where safety, security, and monitoring are essential. One of the most important applications is in airports, where bird strikes pose a major threat to aircraft. By providing instant bird detection and alerts, the system can help airport authorities take immediate

preventive actions, reducing the risk of accidents and saving operational costs.

Another significant application is in agriculture, where birds often damage crops. This system, with its real-time detection and alarm capability, can assist farmers in protecting their fields without the need for constant manual supervision. When a bird is detected near the crops, the system triggers an alert, helping to scare the birds away and safeguard agricultural produce.

The system also plays a crucial role in security and surveillance. Drone detection is increasingly needed in restricted zones such as government buildings, industrial plants, and military areas. The proposed system can identify drone-like objects and activate alarms instantly, helping security personnel respond quickly to potential threats.

Lastly, this system can be used in wildlife monitoring and research. Real-time bird detection helps researchers track species movement patterns, migration activity, and habitat usage without disturbing the natural environment. The automated detection and alerting mechanism makes it suitable for longterm field studies.

XI.FUTURE SCOPE

The proposed bird and drone detection system has significant potential for expansion and improvement. As technology continues to advance, the system can be upgraded to support more accurate detection through better machine learning models and larger training datasets. With access to higher-resolution cameras and faster processors, future versions can analyze video streams more precisely and respond more quickly to threats.

This can greatly enhance the reliability of the system in realworld conditions.

Another major future improvement involves integrating edge computing**.** Instead of running detection on a laptop, the model can be deployed directly on lightweight hardware such as NVIDIA Jetson Nano, Raspberry Pi with Coral TPU, or even future high-performance ESP32 AI modules. This would allow fully independent and portable deployment without needing a computer. It also reduces delay and improves on-site efficiency.

The system can also be upgraded to support multi-class detection and classification. Currently, the focus is on identifying birds and drones. In the future, the system can be trained to detect other aerial threats like kites, balloons, large insects, or even abnormal flying objects. By including more classes, the system becomes more versatile and applicable across various fields such as agriculture, military, and smart surveillance.

Integration with IoT and cloud platform**s** presents another major opportunity. The ESP32 can send alerts directly to mobile apps or cloud dashboards such as Firebase, AWS IoT, or Blynk. Users can receive notifications in real time along with images or short video clips of detected threats. This would make the system more user-friendly and suitable for remote monitoring and largescale deployment.

In the future, the project can incorporate a predictive alert mechanism, where patterns of past detections are analyzed to forecast bird activity. This can help airports or farms take preventive action before any threat occurs. Machine learning techniques like time-series prediction or behavioral pattern analysis can be used to achieve this, making the system not only reactive but also proactive.

A major future enhancement involves automated mitigation systems**,** where the detection output can trigger physical actions. For example, when a drone is detected, a long-range buzzer or alarm can be activated, or a signal can be sent to jamming equipment (where legally allowed). For bird detection, automated sound repellents, rotating lights, or holographic deterrents can be activated to scare the birds away safely. Such an integrated system will be useful in airports, solar farms, and industrial zones.

Another potential improvement is the use of thermal imaging. Birds and drones can be difficult to detect at night due to low light, but thermal cameras can capture heat signatures. With thermal-based YOLO models, the system can work 24/7 under any lighting conditions, making it useful for wildlife conservation and nighttime surveillance.

Finally, the system can evolve into a complete AI-powered aerial security platform**,** combining vision-based detection, audio sensors, radar-like modules, and wireless communication, all controlled by an intelligent central system. This would support scaling the system across large areas such as airports, city borders, power plants, and agricultural fields. The long-term vision is to create an autonomous, self-learning monitoring solution that can adapt to new threats and continuously improve without requiring manual retraining.

XII.CONCLUSION

This project demonstrates a practical and efficient system for detecting birds and drones using a real-time object detection model integrated with a simple hardware alert mechanism. By combining computer vision techniques with lightweight components such as the ESP32, LEDs, and buzzers, the system successfully identifies aerial objects and triggers immediate responses. This makes the solution both impactful and accessible, especially for environments where early detection is critical.

The model used in this project provides reliable accuracy for distinguishing birds and drones, even in unpredictable realworld conditions. Real-time processing through a webcam ensures that threats can be identified instantly, while the hardware integration allows users to receive immediate visual or audio alerts. This bridge between software and hardware strengthens the usefulness of the system, showing how machine learning can effectively assist in everyday applications.

The system is designed to be scalable and adaptable. More advanced models, datasets, sensors, or communication modules can be added in the future without needing to redesign the entire system. This flexibility makes the project suitable for multiple fields such as agriculture, airports, wildlife conservation, surveillance, and industrial safety. Moreover, the

framework serves as a strong base for students and researchers to build more complex AI-powered detection systems.

Overall, the project proves that modern computer vision techniques like YOLO can be successfully implemented in lowcost, user-friendly hardware setups. Through accurate detection, real-time alerts, and simple implementation, the system provides an effective solution to reduce risks and protect sensitive areas. This project not only fulfills its objectives but also opens doors for future advancements that can make airspace monitoring smarter and safer.

References

[1] D. Hockin, M. Ounsted, M. Gorman, D. Hill, V. Keller, and M. Barker, ''Examination of the effects of disturbance on birds with reference to its importance in ecological assessments,'' *Journal of Environmental Management*, vol. 36, no. 4, pp. 253–286, 1992.

[2] E. R. Alexandrino, E. R. Buechley, A. J. Piratelli, K. M. P. M. de Barros, R. de Andrade Moral, Ç. H. Şekercioğlu, W. R. Silva, H. T. Z. do Couto *et al.*, ''Bird sensitivity to disturbance as an indicator of forest patch conditions: An issue in environmental assessments,'' *Ecological Indicators*, vol. 66, pp. 369–381, 2016.

[3] E. R. Alexandrino, E. R. Buechley, J. R. Karr, K. M. P. M. de Barros, S. F. de Barros Ferraz, H. T. Z. do Couto, Ç. H. Şekercioğlu *et al.*, ''Bird based index of biotic integrity: Assessing the ecological condition of Atlantic forest patches in human-modified landscape,'' *Ecological indicators*, vol. 73, pp. 662–675, 2017.

[4] Y.-H. Lin, Y.-Y. Chen, D. R. Rubenstein, M. Liu, M. Liu, and S.-F. Shen, ''Environmental quality mediates the ecological dominance of cooperatively breeding birds,'' *Ecology Letters*, vol. 26, no. 7, pp. 1145– 1156, 2023.

[5] I. Kačergyte, T. Pärt, Å. Berg, D. Arlt, M. Żmihorski, and J. Knape, ''Quantifying effects of wetland restorations on bird communities in agricultural landscapes,'' *Biological Conservation*, vol. 273, p. 109676, 2022.

[6] L. D. Bailey, M. van de Pol, F. Adriaensen, A. Arct, E. Barba, P. E. Bellamy, S. Bonamour, J.-C. Bouvier, M. D. Burgess, A. Charmantier *et al.*, ''Bird populations most exposed to climate change are less sensitive to climatic variation,'' *Nature Communications*, vol. 13, no. 1, p. 2112, 2022.

[7] F. M. M. Mota, N. M. Heming, J. C. Morante-Filho, and D. C. Talora, ''Climate change is expected to restructure forest frugivorous bird communities in a biodiversity hot-point within the atlantic forest,'' *Diversity and Distributions*, vol. 28, no. 12, pp. 2886–2897, 2022.

[8] A. Voskamp, C. Hof, M. F. Biber, K. Böhning-Gaese, T. Hickler, A. Niamir, S. G. Willis, and S. A. Fritz, ''Projected climate change impacts on the phylogenetic diversity of the world's terrestrial birds: more than species numbers,'' *Proceedings of the Royal Society B*, vol. 289, no. 1979, p. 20212184, 2022.

[9] D. Lewis, ''Rare bird's detection highlights promise of'environmental dna','' *Nature*, vol. 575, no. 7783, pp. 423–425, 2019.

[10] M. Demertzioglou, S. Genitsaris, A. D. Mazaris, A. Kyparissis, D. Voutsa, A.Kozari, K.A. Kormas,N. Stefanidou,M.Katsiapi,E. Michaloudi*etal.*, ''A catastrophic change in a european protected wetland: From harmful phytoplankton blooms to fish and bird kill,'' *Environmental Pollution*, vol. 312, p. 120038, 2022.

[11] A. Ferrarini, C. Celada, and M. Gustin, ''Waterbird species are highly sensitive to wetland traits: Simulation-based conservation strategies for the birds of the sicilian wetlands (italy),'' *Biology*, vol. 13, no. 4, p. 242, 2024.

[12] Y.Lian,Y.Bai,Z.Huang,M.Ali,J.Wang,andH.Chen,''Spatio-temporal changes and habitats of rare and endangered species in yunnan province based on maxent model,'' *Land*, vol. 13, no. 2, p. 240, 2024.

[13] N. Zhu, Z. Xi, C. Wu, F. Zhong, R. Qi, H. Chen, S. Xu, and W. Ji, ''Inductive conformal prediction enhanced lstm-snn network: Applications to birds and uavs recognition,'' *IEEE Geoscience and Remote Sensing Letters*, 2024.

[14] M. Torky, G. Dahy, A. Darwish, and A. E. Hassanein, ''Drones and birds detection based on inceptionv3-cnn model: Deep learning methodology,'' in *Artificial Intelligence for Environmental Sustainability and Green Initiatives*. Springer, 2024, pp. 201–219.

[15] V. Mehta, F. Dadboud, M. Bolic, and I. Mantegh, ''A deep learning approach for drone detection and classification using radar and camera sensor fusion,'' in *2023 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2023, pp. 01–06.

[16] K. Wang, F. Yang, Z. Chen, Y. Chen, and Y. Zhang, ''A fine-grained bird classification method based on attention and decoupled knowledge distillation,'' *Animals*, vol. 13, no. 2, p. 264, 2023.

[17] X. Yi, C. Qian, P. Wu, B. T. Maponde, T. Jiang, and W. Ge, ''Research on fine-grained image recognition of birds based on improved yolov5,'' *Sensors*, vol. 23, no. 19, p. 8204, 2023.

[18] H. Liu, C. Zhang, Y. Deng, B. Xie, T. Liu, and Y.-F. Li, ''Transifc: Invariant cues-aware feature concentration learning for efficient finegrained bird image classification,'' *IEEE Transactions on Multimedia*, 2023.

[19] P.-Y. Chou, Y.-Y. Kao, and C.-H. Lin, ''Fine-grained visual classification with high-temperature refinement and background suppression,'' *arXiv preprint arXiv:2303.06442*, 2023.

[20] X. Han and J. Peng, ''Bird sound classification based on ecoc-svm,'' *Applied Acoustics*, vol. 204, p. 109245, 2023.

[21] Q. Zhang, S. Hu, L. Tang, R. Deng, C. Yang, G. Zhou, and A. Chen, ''Sdfie-net–a self-learning dual-feature fusion information capture expression method for birdsong recognition,'' *Applied Acoustics*, vol. 221, p. 110004, 2024.

[22] S. Hu, Y. Chu, Z. Wen, G. Zhou, Y. Sun, and A. Chen, ''Deep learning bird song recognition based on mff-scsenet,'' *Ecological Indicators*, vol. 154, p. 110844, 2023.

[23] Y. Fu, C. Yu, Y. Zhang, D. Lv, Y. Yin, J. Lu, and D. Lv, ''Classification of birdsong spectrograms based on dr-acgan and dynamic convolution,'' *Ecological Informatics*, vol. 77, p. 102250, 2023.

[24] A. Manna, N. Upasani, S. Jadhav, R. Mane, R. Chaudhari, and V. Chatre, ''Bird image classification using convolutional neural network transfer learning architectures,'' *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, 2023.

[25] S. V. Kumar and H. K. Kondaveerti, ''A comparative study on deep learning techniques for bird species recognition,'' in *2023 3rd International Conference on Intelligent Communicationand Computational Techniques (ICCT)*. IEEE, 2023, pp. 1–6.

[26] H. Alqaysi, I. Fedorov, F. Z. Qureshi, and M. O'Nils, ''A temporal boosted yolo-based model for birds detection around wind farms,'' *Journal of imaging*, vol. 7, no. 11, p. 227, 2021.

[27] C.Wah,S.Branson,P.Welinder,P.Perona,andS.Belongie,''Thecaltechucsd birds-200-2011 dataset,'' 2011.

[28] M. Soudy, Y. Afify, and N. Badr, ''Repconv: A novel architecture for image scene classification on intel scenes dataset,'' *International Journal of Intelligent Computing and Information Sciences*, vol. 22, no. 2, pp. 63– 73, 2022.

[29] S. Liu, X. Li, Y. Zhai, C. You, Z. Zhu, C. Fernandez-Granda, and Q. Qu, ''Convolutional normalization: Improving deep convolutional network robustness and training,'' *Advances in neural information processing systems*, vol. 34, pp. 28919–28928, 2021.
[30] X. Zhang, H. Zeng, S. Guo, and L. Zhang, ''Efficient long-range attention network for image super-resolution,'' in *European conference on computer vision*. Springer, 2022, pp. 649–667.
[31] A. Fred and M. Agarap, ''Deep learning using rectified linear units (relu),'' *arXiv preprint arXiv:1803.08375*, pp. 1–6, 2018.
[32] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, ''Attentional feature fusion,'' in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3560–3569.
[33] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, ''Slim-neck by gsconv: a lightweight-design for real-time detector architectures,'' *Journal of RealTime Image Processing*, vol. 21, no. 3, p. 62, 2024.
[34] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, ''Motr: Endto-end multiple-object tracking with transformer,'' in *European Conference on Computer Vision*. Springer, 2022, pp. 659–675.
[35] R. E. Kalman, ''A new approach to linear filtering and prediction problems,'' 1960.
[36] N.Wojke,A.Bewley,andD.Paulus,''Simpleonlineandrealtimetracking with a deep association metric,'' in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.
[37] N. Aharon, R. Orfaig, and B. Bobrovsky, ''Bot-sort: Robust associations multi-pedestrian tracking. arxiv 2022,'' *arXiv preprint arXiv:2206.14651*.
[38] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, ''Strongsort: Make deepsort great again,'' *IEEE Transactions on Multimedia*, vol. 25, pp. 8725–8737, 2023.
[39] S.Zhang, Q.Cai,Y. Mao,and H.Ke,''An optimizedvisualobject tracking algorithm based on centertrack,'' in *2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*. IEEE, 2024, pp. 190–196.
[40] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, ''Bytetrack: Multi-object tracking by associating every detection box,'' in *European conference on computer vision*. Springer, 2022, pp. 1–21.
[41] J. Zhu, C. Ma, J. Rong, and Y. Cao, ''Bird and uavs recognition detection and tracking based on improved yolov9-deepsort,'' *IEEE Access*, pp. 1– 1, 2024.
[42] A. Vajravelu, N. Ashok Kumar, S. Sarkar, and S. Degadwala, ''Security threats of unmanned aerial vehicles,'' in *Wireless Networks: Cyber Security Threats and Countermeasures*. Springer, 2023, pp. 133–164.
[43] K. K. Shenoy and D. Tyagi, ''Unmanned aircraft system safety, security, and regulation in urban aviation ecosystems,'' *International Journal of Intelligent Enterprise*, vol. 11, no. 2, pp. 157–175, 2024.
[44] U. Seidaliyeva, L. Ilipbayeva, K. Taissariyeva, N. Smailov, and E. T. Matson, ''Advances and challenges in drone detection and classification techniques: A state-of-the-art review,'' *Sensors*, vol. 24, no. 1, p. 125, 2023.
[45] S. S. Aote, N. Wankhade, A. Pardhi, N. Misra, H. Agrawal, and A. Potnurwar, ''An improved deep learning method for flying object detection and recognition,'' *Signal, Image and Video Processing*, vol. 18, no. 1, pp. 143–152, 2024.
[46] C.Wah,S.Branson,P.Welinder,P.Perona,andS.Belongie,''Thecaltechucsd birds-200-2011 dataset,'' *california institute of technology*, 2011.
[47] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, ''Yolov9: Learning what you want to learn using programmable gradient information,'' *arXiv preprint arXiv:2402.13616*, 2024.
[48] M. S. Alzboon, M. Alqaraleh, and M. S. Al-Batah, ''Ai in the sky: Developing real-time uav recognition systems to enhance military security,'' *Data Metadata*, vol. 3, p. 417, 2024.
[49] Y. Ghazlane, M. Gmira, and H. Medromi, ''Development of a visionbased anti-drone identification friend or foe model to recognize birds and drones using deep learning,'' *Applied Artificial Intelligence*, vol. 38, no. 1, p. 2318672, 2024.
[50] U. Seidaliyeva, L. Ilipbayeva, K. Taissariyeva, N. Smailov, and E. T. Matson, ''Advances and challenges in drone detection and classification techniques: A state-of-the-art review,'' *Sensors*, vol. 24, no. 1, p. 125, 2023.
[51] A. Coluccia, A. Fascista, A. Schumann, L. Sommer, A. Dimou, D. Zarpalas, M. Méndez, D. De la Iglesia, I. González, J.-P. Mercier *et al.*, ''Drone vs. bird detection: Deep learning algorithms and results from a grand challenge,'' *Sensors*, vol. 21, no. 8, p. 2824, 2021.