


Article

Smart Drone Surveillance System Based on AI and on IoT Communication in Case of Intrusion and Fire Accident

Minh Long Hoang 

Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy; minhlong.hoang@unipr.it

Abstract: Research on developing a smart security system is based on Artificial Intelligence with an unmanned aerial vehicle (UAV) to detect and monitor alert situations, such as fire accidents and theft/intruders in the building or factory, which is based on the Internet of Things (IoT) network. The system includes a Passive Pyroelectric Infrared Detector for human detection and an analog flame sensor to sense the appearance of the concerned objects and then transmit the signal to the workstation via Wi-Fi based on the microcontroller Espressif32 (Esp32). The computer vision models YOLOv8 (You Only Look Once version 8) and Cascade Classifier are trained and implemented into the workstation, which is able to identify people, some potentially dangerous objects, and fire. The drone is also controlled by three algorithms—distance maintenance, automatic yaw rotation, and potentially dangerous object avoidance—with the support of a proportional–integral–derivative (PID) controller. The Smart Drone Surveillance System has good commands for automatic tracking and streaming of the video of these specific circumstances and then transferring the data to the involved parties such as security or staff.

Keywords: drone; AI; YOLO; Cascade Classifier; PID; flight algorithms; IoT; sensor security



Citation: Hoang, M.L. Smart Drone Surveillance System Based on AI and on IoT Communication in Case of Intrusion and Fire Accident. *Drones* **2023**, *7*, 694. <https://doi.org/10.3390/drones7120694>

Academic Editors: Liuguo Yin and Shu Fu

Received: 26 October 2023

Revised: 26 November 2023

Accepted: 29 November 2023

Published: 2 December 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Currently, unmanned aerial vehicles (UAVs) are utilized in a wide range of applications [1–5], especially in surveillance systems [6,7]. Drone surveillance involves visually monitoring an individual, a group, items, or a situation to prevent potential threats. The establishment of an efficient surveillance system with drone fleets necessitates the smooth integration of dependable hardware and sophisticated automation software. In buildings and factories, there is a high demand for smart security systems with drone applications. Drones perform significantly faster than patrol vehicles or security personnel, enabling them to promptly arrive at the location of an incident, thereby facilitating a swift remedial response.

Research [8] uses a drone attached to a USB camera interfaced with Raspberry Pi, which is capable of autonomous flight monitoring in a campus, office, and industrial areas. However, the system does not include an object identification function and mainly focuses on video streaming. Therefore, this system cannot track a specific intruder automatically. Another article [9] studies the human motion tracking algorithm with a drone based on the MediaPipe Framework [10,11]. A drone-based method for 3D human motion capture has been developed by researchers [12], in which a drone circles a human subject, records a video, and then reconstructs 3D full-body postures. These studies try to reconstruct the 3D posture sequence of a subject rather than concentrating on a methodology for autonomous human motion tracking.

Another paper discusses the deep convolutional neural network (CNN) model (EfficientNet-B3) [13] for plant disease identification, and images were captured using a drone with a convolution neural network (CNN). Nevertheless, EfficientNet’s performance can be reduced on hardware accelerator such as GPUs, which are designed for large computation models and where data movement is a relatively small component of the overall performance. In this case,

EfficientNet requires significantly less computation and more data movement than comparable networks.

In this work, a Smart Drone Surveillance System (SDSS) is developed for the case of intruders and fire accidents. Sensors are responsible for detecting the intrusion or the accident flame and then sending the signal to the workstation via Wi-Fi. The drone is already located in a hidden place at a proper distance inside the building. After receiving the sensor signal, the workstation controls the drone to reach the sensor location. On its trajectory, the drone can also track suspicious objects. The drone turns around the sensor place to identify a person or fire. Once it captures the concerned object, the drone starts tracking at a safe distance. The object identification is trained by YOLOv8 [14] and Cascade Classifier [15]. The drone-controlling software is developed based on Python [16], which sets the drone's trajectory to the sensor locations and transfers the video camera to the workstation. After obtaining the object video, the first 5 s or 10 s of the videos are sent to other PCs or mobile phones of the emergency contact, such as the security or police. Then, the drone keeps following the detected object and stores the video in the workstation. In this way, considerable circumstances are under security monitoring, and all the evidence is saved.

In addition, three algorithms are implemented into the flight control system: distance maintenance, automatic yaw rotation, and potentially dangerous object avoidance. Distance maintenance is regulated by a proportional–integral–derivative (PID) controller [17] based on the prediction box area of the Artificial Intelligence (AI) object identifier. The area box must be at a specific threshold to guarantee a safe distance between the object and the drone. The automatic yaw rotation is also adjusted by the PID controller, depending on the difference between the box center x-coordinate and the frame center. The drone rotates its yaw angle, following the right/left motion of the object to ensure that the object is at the center of the frame. The third algorithm helps the drone avoid dangerous objects that can be used to throw, such as knives, scissors, cups, and bottles. Once the drone detects these objects close to it (based on the object area size in the image), the drone moves to one side and rotates its heading to the opposite side, so the tracking process continues with the intruder.

The devices in SDSS communicate with each other under the IoT protocol [18,19]. The IoT refers to the interconnection of things over the Internet, enabling smart devices to engage in communication, data exchange, and interaction between electrical components, software systems, networks, and sensors, thus facilitating effective communication processes. Effective communication among smart devices plays a crucial role in the IoT, as it facilitates the gathering and sharing of data. This capability significantly contributes to the overall success of IoT products and projects.

This paper contributes the following points to scientific research:

- Both YOLOv8 and Cascade Classifier are successfully implemented together into the flight system to support each other in object detection, which accomplishes high accuracy and speed for surveillance purposes.
- The distance maintenance and yaw rotation algorithms based on the PID controller are described in detail, providing deep comprehension for the reader about the drone control field with the support of AI techniques.
- An algorithm for potentially dangerous object avoidance is proposed, which utilizes a straight strategy to dodge the approaching object based on the trained model.
- The strong point of this paper is to combine the computer vision models and the UAV algorithms into a smart system. There is a highly effective connection between the two sides of implementation. The drone-controlled algorithms are based on AI models. Thus, this paper not only describes the robust flight control methods in detail but also describes the automatic operation connected to the trained AI object identifier models.

The paper is organized as follows: the 1st part describes the utilized components, the computer vision models, and the control algorithms for the drone. The 2nd part contains

the experimental setup and result analysis. Finally, the conclusion and the future work are outlined.

2. Related Work

SDSS utilizes AI models in computer vision, such as YOLOv8 and Cascade Classifier, to identify the suspected objects. A human detection method based on YOLOv5 was proposed in [20], which successfully identified the human object. When comparing YOLOv8 against YOLOv5, YOLOv8 demonstrates faster speed and improved throughput with a similar number of parameters due to the implementation of hardware-efficient and architecturally reformed approaches [21]. On the other hand, Cascade Classifier, based on the AdaBoost algorithm, is speedy, robust, and accurate. Meanwhile, deep learning methods like convolutional neural networks (CNNs) [22] require significant data and processing power.

A system for autonomous human-following drones used 3D pose estimation to gather data on human direction for following a moving subject from a certain direction [23]. This system uses PID controls with input values such as the longitudinal movement speed, vertical movement speed, and rotational angular speed with the neck and middle hip as the reference points. In our paper, the PID controller works based on the output bounding box area from computer vision models, comparing it with the desired threshold. A larger box area corresponds to a closer distance and vice versa.

The same research [23] also proposes a system that calculates human direction from the 3D joint points of the left shoulder and right shoulders. From that, the left-right movement speed of the drone is changed to approach the target value. Unlike that control system, our manuscript utilizes the PID controller to regulate motor speed by calculating the difference between the prediction box center and the image center in pixels. This method can constantly track the person's direction change to adapt, aiming to set the human-object box around the frame center.

Typically, to avoid obstacles or approaching objects, the drone needs to be equipped with a light detection and ranging (LiDAR) sensor to enable the perception of obstacles within the surrounding environment [24] or ultrasonic sensors for distance estimation [25]. To minimize the extra cost and high power consumption, our drone system has implemented the algorithm to avoid the right/left by detecting the approaching object position respecting the image frame.

Another study uses a tracking camera with a path-planning algorithm for collision avoidance in horizontal space [26]. The tracking camera on the drone is used to track its position. The idea is to drive the drone to the target with three possible trajectories. If there is an obstacle, the drone moves to another planned path. However, this method requires the external sensor system, such as the motion tracker of Optitrack, to track obstacle positions [27]. Our developed system tracks the obstacles directly based on the object identifier YOLOv8. Hence, the SDSS has good time responses with self-detection and optimized cost construction for the drone.

The sensors are essential in the Internet of Things (IoT) [28], which acquire and transmit data from their surroundings to the center of wireless communication. In the proposed system, the PIR sensor [29] for human appearance detection and the flame detection sensor [30] are mounted in the monitoring places to transmit the alert to the workstation based on ESP32 [31]. The smart system receives this alert and then automatically activates the whole flight system, implementing the robust algorithm with AI models inside. In this way, the SDSS can be seen in Figure 1.

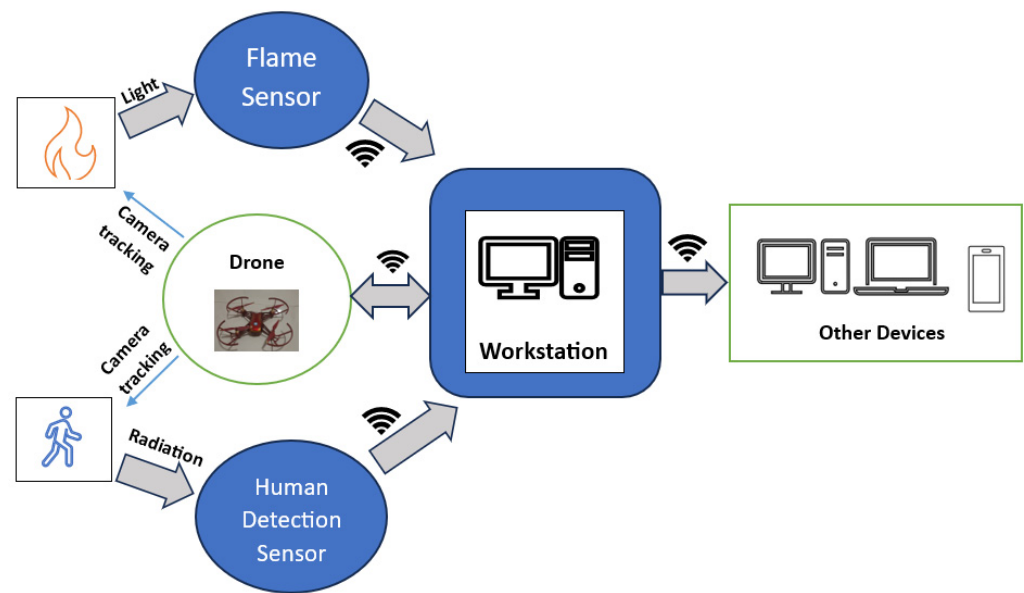


Figure 1. Smart system diagram based on IoT.

3. Materials and Method

3.1. Drone Components

The drone is a continuous low humming sound. As shown in Figure 2, a quadcopter drone has four propellers and motors, a power distribution board, a frame, an electric motor controller (ESC), a flight controller, a battery, a receiver, a camera, and sensors. Pressure sensors measure the altitude or distance between the ground and the drone. The inertial measurement unit (IMU) sensors [32–34] measure the accelerations and angles of the drone.



Figure 2. Tello drone.

3.2. Drone Working Principle

The drone can work in 4 degrees of freedom (DOF), which translates in 3 directions and rotates in 1. As illustrated in Figure 3, 2 propellers rotate clockwise, and the other 2 rotate anticlockwise, generating zero angular momentum and creating the lift to fly the drone. This characteristic keeps the drone stationary rather than rotating in one direction while hovering.

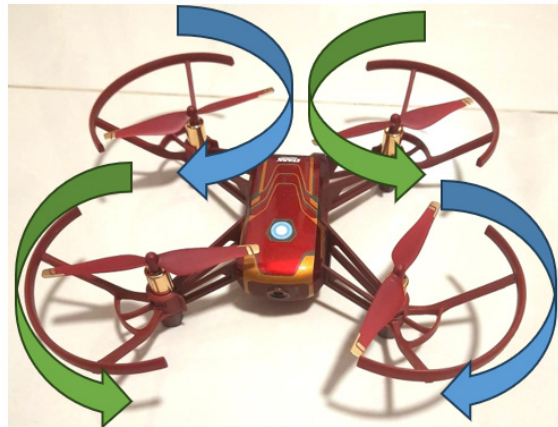


Figure 3. Rotation direction illustration of drone propeller.

The translations are up–down, left–right, and forward–backward. To rotate the drone yaw following the counterclockwise movement, the speed of the counterclockwise motor must be increased, and the speed of the anticlockwise motor must be reduced and vice versa.

The speed of drone motors is controlled in cm/s by a Python library: DJITelloPy [35].

- Left/right velocity: -100 to 100 cm/s;
- Forward/backward velocity: -100 to 100 cm/s;
- Up/down velocity: -100 to 100 cm/s;
- Yaw velocity: -100 to 100° /s.

The position translation:

- Move to left or move to right: 20 to 500 cm;
- Move forward or move backward: 20 to 500 cm;
- Rotate clockwise or anticlockwise: 1 – 360° .

3.3. Computer Vision Technologies

3.3.1. YOLOv8

YOLO is a widely adopted ensemble of object detection models utilized for real-time object identification and classification within computer vision. The primary characteristic of YOLO is its singular-stage detection methodology, which was specifically devised to identify objects rapidly and accurately in real time. YOLOv8 provides the most significant advantages in both accuracy and speed in detection among all YOLO versions [36]. As illustrated in Figure 4, the trained models output the prediction boxes with bounding heights and widths.

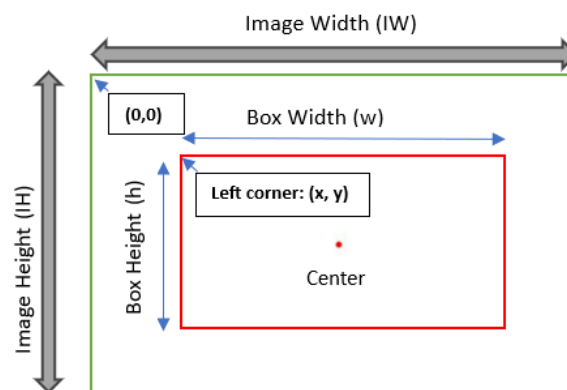


Figure 4. Image frame and prediction box in pixel coordinate.

YOLOv8 incorporates the C2f module, which effectively combines two parallel branches of gradient flow, enhancing the resilience and effectiveness of gradient information propagation. The integration of advanced characteristics with contextual information enhances the precision of detection. The detection module utilizes a combination of convolutional and linear layers to effectively transform the high-dimensional information into the desired output bounding boxes and item classifications. The system's backbone changed with the C2f, replacing C3 (composed of 3 convolutional layers) in YOLOv5 [37]. C2f has the outputs from the concatenated bottleneck (a combination of two 3×3 convolutional layers with residual connections), while C3 only utilizes the output from the last bottleneck. Every convolution filter is responsible for extracting a particular characteristic from the image. Figures 5 and 6 depict the bottleneck and C2F structures, respectively.

Note: CBS = Conv + BN + SiLU

where in the convolutional layer, BN (batch normalization) normalizes the previous layers' output using the current batch's mean and variance and SiLU (Sigmoid Linear Units) is an activation function for neural networks.

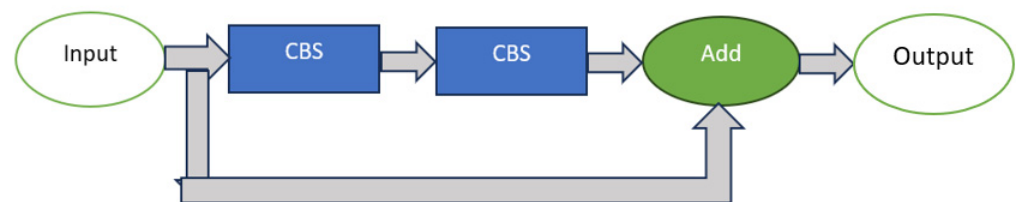


Figure 5. Bottleneck structure.

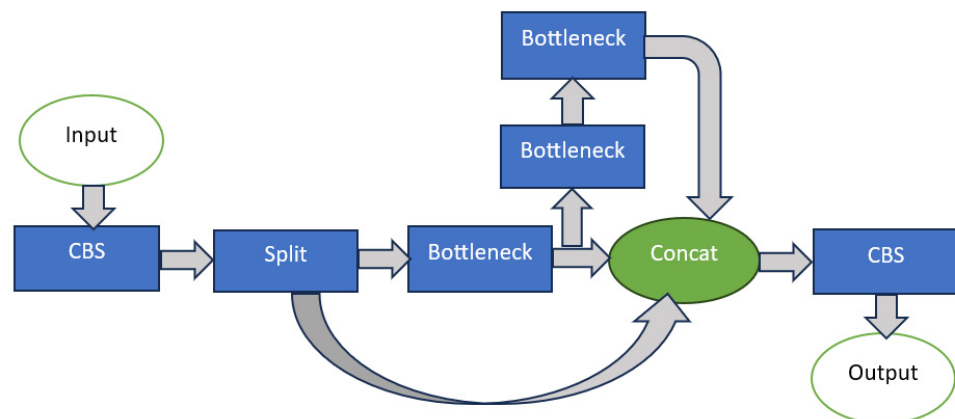


Figure 6. C2f structure.

Overall, the YOLOv8 structure mainly contains an input segment, a backbone, a neck, and an output segment with a detection head.

The input segment implements mosaic data augmentation, adaptive anchor computation, and adaptive grayscale padding on the input picture.

In the backbone network, the input picture undergoes processing by several convolutional (Conv) and C2f modules in order to extract feature maps at various scales. The feature maps produced as output undergo processing through the spatial pyramid pooling fast (SPPF) module. This module utilizes pooling with different kernel sizes to merge the feature maps. The combined outputs are subsequently transmitted to the neck layer. The utilization of Sequentially Connected Three Maximum Pooling Layers (SPPFs) results in a reduction in computing effort and a decrease in delay.

The neck layer of YOLOv8 utilizes the Feature Pyramid Network (FPN) [38] and Path Aggregation Network (PAN) [39] architecture to augment the model's capacity to fuse

features. This architectural design integrates high-level and low-level feature maps by utilizing upsampling and downsampling techniques, enabling the effective transmission of both semantic and localization cues. By employing this methodology, the network gains improved capability to integrate characteristics from items with diverse scales, hence augmenting its detection efficacy on things with variable scales.

The detection component of YOLOv8 adheres to the conventional approach of segregating the classification component from the detection component. The process involves the computation of loss and target detection box filtering. The computation of loss has two main components, namely classification and regression, with the exclusion of the object branch. The classification branch employs the Binary Cross-Entropy (BCE) loss function, whereas the regression branch utilizes the Distribution Focal Loss (DFL) [40] and CIoU loss functions [41]. The formation of prediction boxes in YOLOv8 involves the utilization of decoupled heads, which have the capability to predict categorization scores and regression coordinates concurrently. The representation of classification scores is accomplished by a two-dimensional matrix, which signifies the existence of an item within each individual pixel. The regression coordinates can be denoted by a four-dimensional matrix, which represents the displacement of the object's center with respect to each pixel. YOLOv8 utilizes a task-aligned assigner to calculate a task alignment measure by utilizing the classification scores and regression coordinates. The task alignment measure integrates the classification scores with the Intersection over Union (IoU) value, facilitating the joint optimization of classification and localization while mitigating the impact of low-quality prediction boxes. The IoU metric is extensively utilized in the field of object identification. It plays a crucial role in identifying positive and negative samples and estimating the distance between predicted boxes and ground truth. An item is commonly categorized as detected when the IoU surpasses a threshold of 0.5.

Although YOLOv8 is a robust model, it still has a limit in small object detection or objects with low contrast. Thus, another OpenCV [42] model is implemented for fire detection since it is essential to discover the flame from the ignition state when it just starts in a small shape.

3.3.2. Cascade Classifier

The Cascade Classifier technique works based on the Haar feature-based Cascade Classifier, which is an effective object detection method proposed in the article [43]. In this case, there are about 600 positive and 400 negative image samples to train classifiers for fire detection. Positive images contain fire, and negative images do not contain fire. As illustrated in Figure 7, the Haar Cascade Classifier consists of 4 main steps:

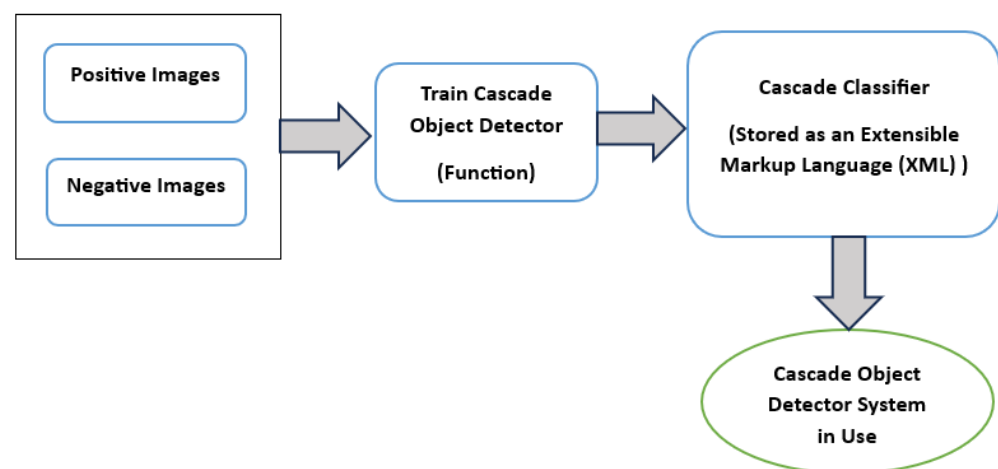


Figure 7. Cascade Classifier diagram.

- Step 1: Gathering the Haar Features. In a detection window, a Haar feature is effectively the result of computations on neighboring rectangular sections. The pixel intensities in each location must be summed together to determine the difference between the sums. Figure 8 shows the Haar feature types.



Figure 8. Haar feature types.

- Step 2: Creating Integral Images. In essence, the calculation of these Haar characteristics is sped up using integral pictures. It constructs sub-rectangles and array references for each of them rather than computing each pixel. The Haar features are then computed using them.
- Step 3: Adaboost Training. Adaboost selects the top features and trains the classifiers to utilize them. It combines weak classifiers to produce a robust classifier for the algorithm to find items. Weak learners are produced by sliding a window across the input image and calculating Haar characteristics for each area of the image. This distinction contrasts with a learned threshold distinguishing between non-objects and objects. These are weak classifiers, whereas a strong classifier requires a lot of Haar features to be accurate. The last phase merges these weak learners into strong ones using cascading classifiers.
- Step 4: Implementing Cascading Classifiers. The Cascade Classifier comprises several stages, each containing a group of weak learners. Boosting trains weak learners, resulting in a highly accurate classifier from the average prediction of all weak learners. Based on this prediction, the classifier decides to go on to the next region (negative) or report that an object was identified (positive). Due to the majority of the windows not containing anything of interest, stages are created to discard negative samples as quickly as possible.

3.3.3. Evaluation Metrics

Precision (P), recall (R), and average precision (AP) are the evaluation metrics to validate the object detection models. The AP is the average accuracy of the model [44].

The formula to calculate P and R is as follows:

$$P = \frac{TP}{TP + FP} \quad (1)$$

$$R = \frac{TP}{TP + FN} \quad (2)$$

$$AP = \int_0^1 p(r)dr \quad (3)$$

where TP is true positive; FP is false positive; and FN is false negative.

3.4. Human-Tracking Algorithms

3.4.1. Distance Maintenance

To track a person or fire, a drone needs to identify these objects using AI in computer vision. A rectangular boundary will cover the detected object, as shown in Figure 4.

Then, the safety distance must be adjusted in real time, which is carried out based on the area threshold in pixels. This area of the prediction box must be maintained in a specific range $[A, B]$ following the below algorithm:

- If the box area $< A$ \rightarrow Drone is too far away \rightarrow 2 front motor speed is increased \rightarrow Drone moves forward.
- If the box area $> B$ \rightarrow Drone is too close \rightarrow 2 back motor speed is increased \rightarrow Drone moves backward.
- If the box area $\in [A, B]$ \rightarrow Drone is at the proper distance from the object \rightarrow Drone maintains motor speed.

For instance, the image width and length are 640 and 480, respectively.

The threshold can be about [33,200, 33,800] for person detection, and the threshold for fire detection can be about [600, 1800].

PID controls the speed variation to maintain the proper distance between the drone and the target, as shown in Figure 9.

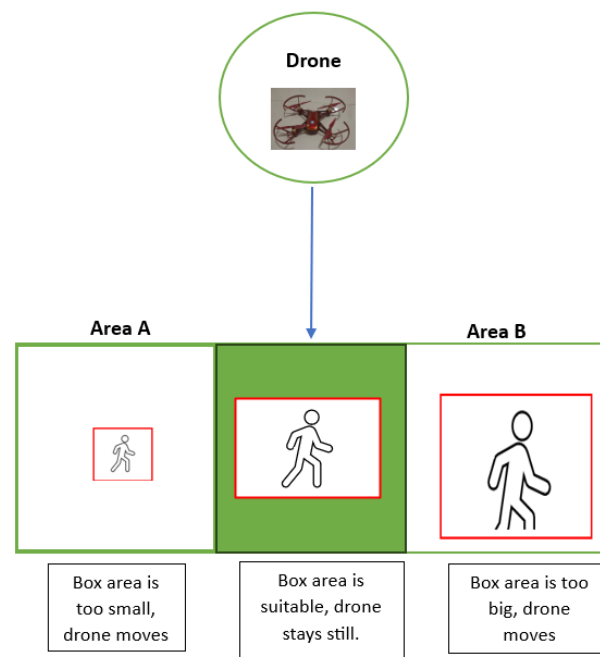


Figure 9. Drone tracks object at the proper distance.

- The P-controller is an essential element in the control systems. The system offers a direct control action proportional to the error between the target setpoint and the measured process variable. The drone's controller continuously modifies the motor speed depending on the difference between the desired and predicted box areas to ensure the drone maintains the appropriate distance from the item. The back motors are slowed to gently return the drone back if it is too near than intended and vice versa. The difference between the required and measured rectangle areas determines how much correction is made; higher differences yield more vital adjustments.
- The D-controller aids in system control by monitoring the rate of change. The focus is placed on the rate of change between the target value and the measured value. When the drone reaches the proper distance, the D-controller helps keep the drone steady by looking at how quickly the drone's speed is changing. If the drone is going backward or downward too fast, the D-controller will adjust to slow it down. This feature helps the drone stay at the desired distance smoothly, ensuring stability and precise speed control.
- The I-controller operates by continuously summing the error signal over a period of time and utilizing the resultant integrated value to provide suitable modifications to control inputs. If the drone deviates from its setpoint, the integral controller calculates the duration and magnitude of the accumulative error and applies corrective actions

proportionally. The P and D controllers can make quick adjustments but struggle to remove minor, persistent errors that occur over time, leading to steady-state errors.

As shown in Figure 10, the overall control function:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (4)$$

where

- $u(t)$: PID control variable.
- K_p , K_i , and K_d are the proportional, integral, and derivative coefficients, respectively.
- $e(t)$ is the error between the desired and current values.
- K_p should be great enough if the error is significant; the control output will be proportionately high. K_d should be set higher if the change is rapid. K_i should be suitable to eliminate the residual error due to the historic cumulative value of the error.

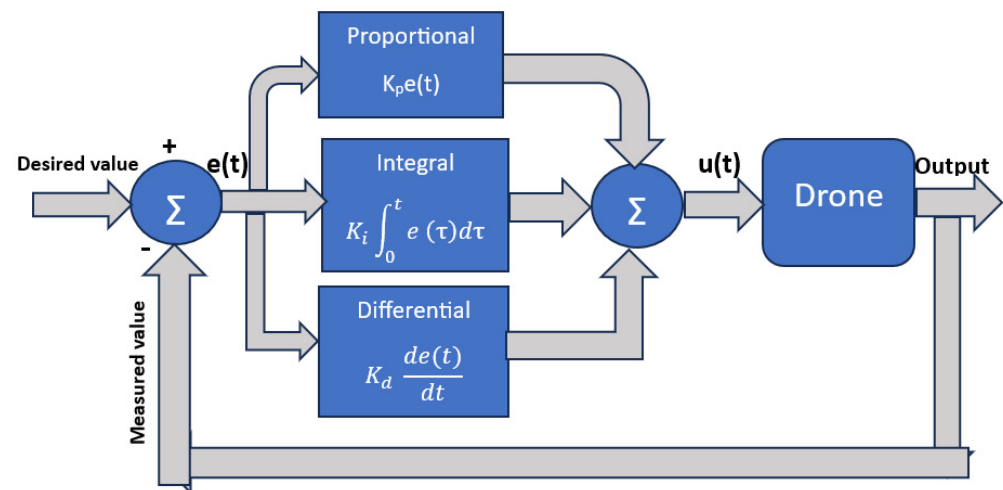


Figure 10. PID controller diagram.

3.4.2. Yaw Rotation for Object Position Adaption

When the object starts moving to the side, the drone has to rotate the yaw to capture the object in the image center and follow the target to the left or right motion. If the person moves to the left side, the drone has to rotate to the left to bring the object back to the frame center. Thus, the speed of the left motors should be adjusted to be higher than the right motors and vice versa, as demonstrated in Figure 11.

The PID is also applied to this case to avoid overshooting issues. When the drone rotates closely to the object, its speed must be decreased gradually. The PID will adjust the motor speed depending on how far the drone's yaw is from the object's actual point.

- If the x-coordinate of the rectangle center < x-coordinate of the image center → Target moves to the left → PID adjusts the drone yaw to increase the left motor speed.
- If the x-coordinate of the rectangle center > the x-coordinate of the image center → Target moves to the right → PID adjusts the drone yaw to increase the right motor speed.

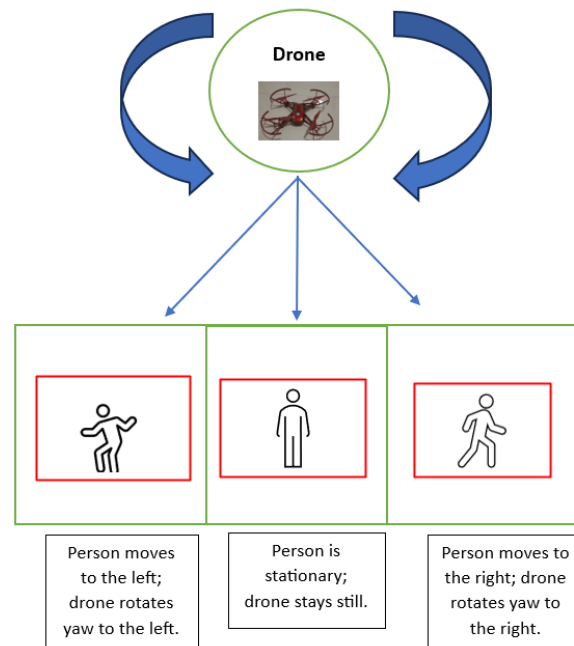


Figure 11. Yaw rotation for object position adaption.

3.4.3. Potentially Dangerous Object Avoidance

Some other objects are learned in model YOLOv8 to support drone safety, such as knife, bottle, cup, cell phone, and scissor. If the intruder throws these objects, the drone detects them at a specific range; it is programmed to move to one side, and the yaw rotates to the opposite side to keep tracking the person. For instance, if a knife is thrown at the drone, and the drone moves to the left and then rotates the yaw to the right. Another area threshold (pixel) is set to detect whether the objects approach close to the drone. When the drone identifies those objects, the area is greater than this area threshold on camera, and the drone will avoid them. Figure 12 describes the object at the frame center; Figure 13 shows the cases when the approaching object is at the left and right.

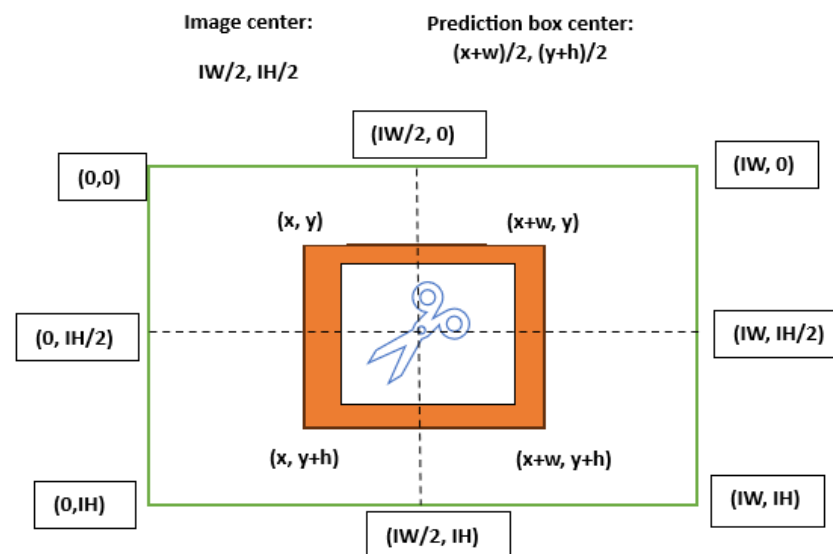


Figure 12. Object in the frame.

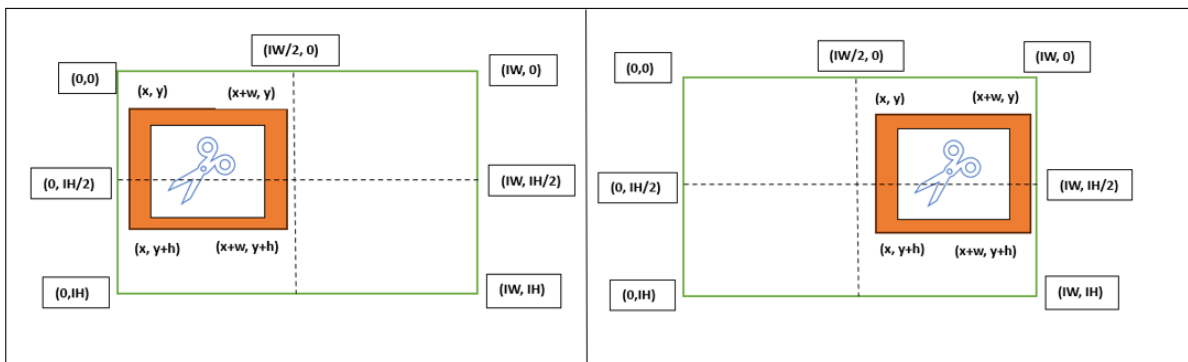


Figure 13. Approaching object at the left (1st case) and right (2nd case).

If approaching object > area threshold:

- If $(x + w)/2 \in [0, IW/2]$ and $(y + h)/2 \in [0, IH]$:
Object at the left → Drone moves to the right 20 cm and then turns yaw to the left 30° .
- If $(x + w)/2 \in [IW/2, IW]$ and $(y + h)/2 \in [0, IH]$:
Object at the right → Drone moves to the left 20 cm and then turns yaw to the right 30° .
- If $(x + w)/2 = IW/2$ and $(y + h)/2 = IH/2$:
Object at the center → Drone moves to the right 25 cm and then turns yaw to the left 32° .

3.5. Sensor Utilization

As shown in Figure 14, the “Passive Pyroelectric Infrared Detector” (PIR HAT) is an M5StickC compatible human body induction sensor that detects infrared radiation from the body. The sensor will output HIGH when infrared is detected, which will continue for two seconds until the next detection cycle.



Figure 14. PIR sensor.

The sensor detects infrared radiation from objects in its field of vision. It is referred to as “passive” since it does not produce any energy of its own and instead monitors changes in the quantities of infrared radiation around it. A pyroelectric substance, which produces an electric charge when subjected to temperature variations, is the main component of a PIR sensor. This substance typically has a crystalline structure and persistent electric polarization. When a person moves within the sensor’s range, it causes a rapid change in the infrared radiation levels falling on the pyroelectric material. The pyroelectric material produces an electric signal in response to this radiation change.

As shown in Figure 15, flame sensor DFR0076 [8] is an analog flame sensor consisting of an amplifier circuit, a lens, and a photodiode/phototransistor. When a flame is present, it emits light throughout a broad band of wavelengths, including ultraviolet and infrared. The sensor lens directs this light onto the photodiode/phototransistor, which generates an electrical current according to the light’s brightness. The amplifier circuit amplifies this

signal, and the resulting analog voltage output shows the flame's presence and intensity. The flame sensor can be used to sense the fire or other wavelengths at 760~1100 nm light. The flame sensor probe is positioned at an angle of 60 degrees, which allows for enhanced sensitivity to the unique spectral characteristics of flames. The operational temperature range of the flame sensor is from -25 to 85 °C.



Figure 15. Flame sensor.

4. Experiments and Results

4.1. Experimental Setup

In this research, the Tello drone [45], as shown in Figure 16, is used in the experiment. It can shoot up to 720p video at 30 frames per second.



Figure 16. Tello drone in air.

The system with the AI models were trained in the host machine with an NVIDIA Quadro P620 (combining a 512 CUDA core Pascal GPU), 2 GB GDDR5, Intel Core i7 vPro-10850H Processor (2.70 GHz), and RAM of 32 GB. The image annotation was carried out by using OpenCV [46]. YOLOv8 was trained based on Ultralytics [47], and Cascade Classifier was trained by OpenCV [48,49]. Python was used as an integrated development environment for the project. The image width and length for the AI model are 640 and 480, respectively.

The flame sensor is connected with the ESP32, as shown in Figure 17, and the PIR sensor with M5StickC is set up, as shown in Figure 18.

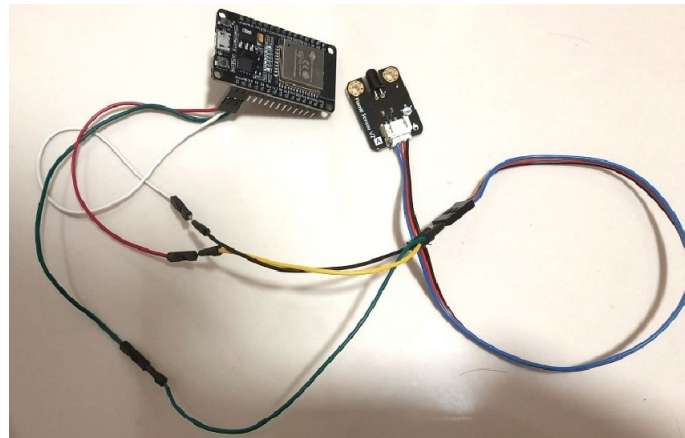


Figure 17. Flame sensor in connection with ESP32.

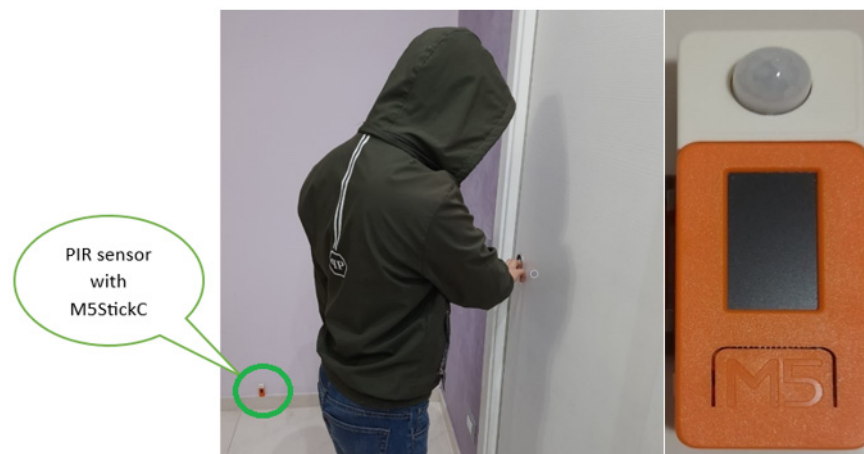


Figure 18. PIR sensor with M5StickC setup for person detection.

4.2. Results and Analysis

4.2.1. Computer Vision Test Performance

About 1000 images are utilized for each model, where 70% are used for testing and 30% are used for testing.

Table 1 reports the performance of two computer vision models from YOLOv8 and Cascade Classifier for person and flame detection, respectively. Although the flame detection model has slightly superior metrics, both models achieve high accuracy, guaranteeing good efficiency in object tracking.

Table 1. Performance result on test-set image.

Model	Precision (%)	Recall (%)	AP (%)
Person, knife, bottle, cup, cell phone, scissors detection YOLOv8	88.4	86.5	88.9
YOLOv7	85.1	84.8	85.3
YOLOv5	72.5	71.2	72.7
Flame detection (Cascade Classifier)	89.1	88.3	90.1

On the other hand, two other popular YOLO models, YOLOv5 [50] and YOLOv7 [51], are also trained to compare with YOLOv8 performance. The metric evaluation shows that YOLOv8 obtains the best execution in precision, recall, and AP.

4.2.2. Person Detection

The PIR sensor detects a human appearance; it sends the signal to the workstation via Wi-Fi thanks to the ESP32 microcontroller embedded inside M5StickC. The workstation controls the drone to fly to the sensor place to detect that person and keep following, as illustrated in Figure 19.



Figure 19. Person motion detection by drone.

4.2.3. Evaluation on the Distance Maintenance

The distance and direction evaluation techniques were carried out. The prediction box area is implemented to maintain a specific range [33,200, 33,500]. In order to evaluate the effectiveness of the proposed system in terms of distance, we conducted an experiment where the drone was tasked with tracking a participant who walked and periodically stopped along a linear path. The person stood for about 10 s, then walked fast for 1 step and stopped for another 10 s, then walked again and stopped. In this case, when the distance between the drone and the person is maintained, the bounding box area of person detection is supposed to be approximately the same.

The result in Figure 20 shows the drone's success in following the person. Since the person moves very fast, the distance between the drone and the object is extended at high speed, corresponding to the sudden drop of the bounding box area. However, the drone recovers the distance by moving forward immediately. Here, the person moves away only one short step, so the drone covers the distance quickly, and then the AI system continues to detect the person with a bounding box.

4.2.4. Evaluation of Direction Rotation

To evaluate the drone's rotation ability, the person moves to the left and then moves to the right side of the drone at the same distance with normal speed. The purpose is to observe whether the drone yaw can rotate according to the motion. Figure 21 demonstrates that the drone has good capability of adapting its yaw rotation, following the direction of the tracked object.

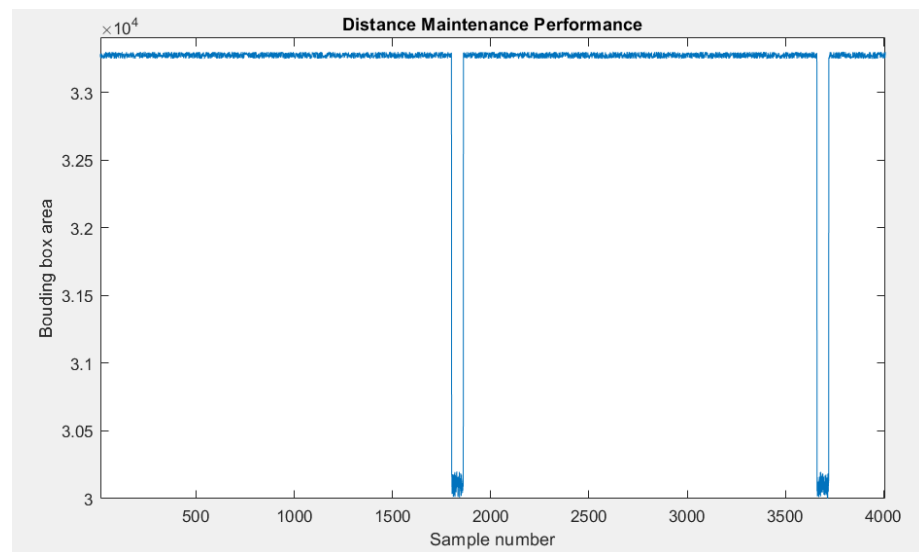


Figure 20. Distance maintenance evaluation.

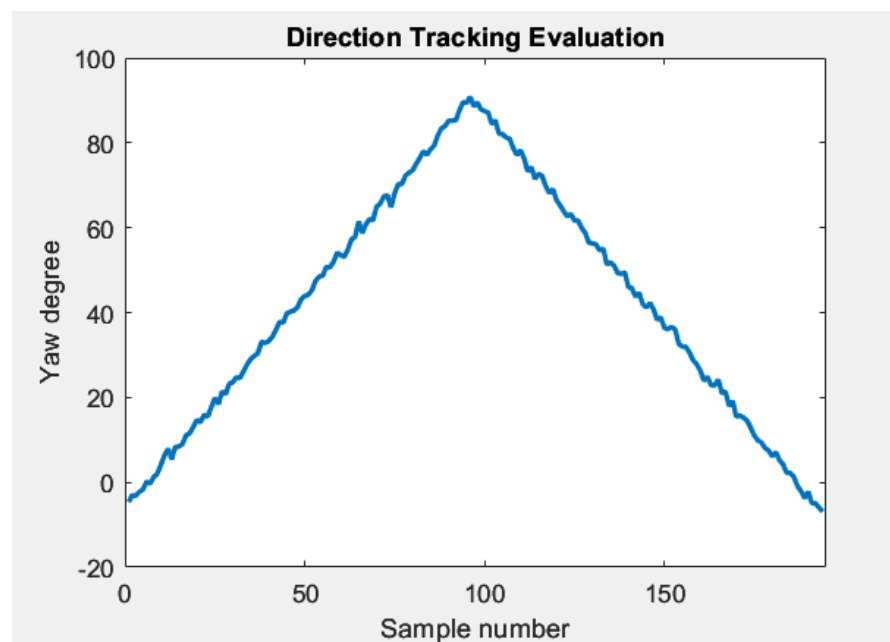


Figure 21. Direction tracking evaluation.

4.2.5. Potentially Dangerous Object Detection

The drone is able to detect the potentially dangerous object detection, as shown in Figure 22. Once these identified objects are thrown at the drone, it can avoid them by moving to one side and adjusting the yaw orientation to keep tracking the person due to algorithm of automatic yaw rotation. The maximum moving speed is about 100 cm/s. For other weapon types, the system needs further training to gain the ability to detect diverse weapons such as a gun.

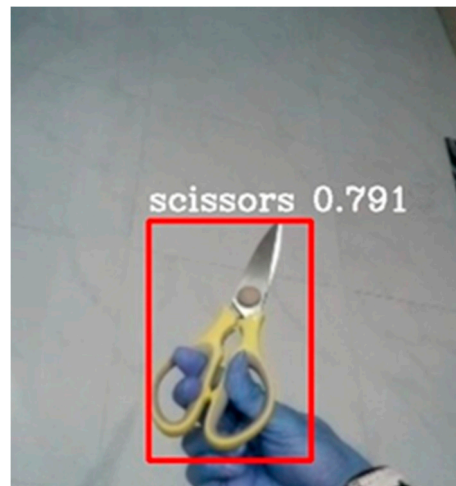


Figure 22. Potentially dangerous object detection.

4.2.6. Fire Detection

Similarly, when the flame sensor collects the firelight, it sends the signal to the workstation via Wi-Fi. Then, the drone begins its action and tracks the fire, as shown in Figure 23. After 5 or 10 s of recording, the video is delivered to the security, building administrator, or the house owner via mail or other telecommunication tools.

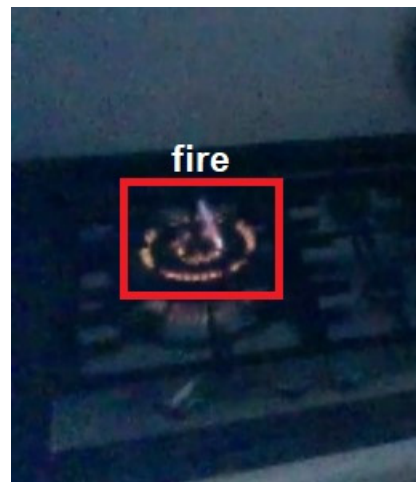


Figure 23. Fire detection.

The Cascade Classifier is quick, since each classifier in the cascade only requires processing a small portion of the data. This feature enables quicker object recognition by reducing data that have to be processed. In addition, objects may still be detected by the chain of classifiers despite noise and other distortions. Due to each classifier in the cascade being trained on a fraction of the data, the Cascade Classifier technique can provide high accuracy in object detection. In future work, more fire scenarios will be tested in diverse circumstances.

4.3. System Overview

Table 2 reports the main pros and cons of the developed system.

Table 2. Advantages and limits of SSDS.

Advantages	Limits
<ul style="list-style-type: none"> • Drone is automatically controlled to arrive and track the concerned object with wireless communication from sensors. • Drone is implemented the practical algorithms to enhance its security with the capability of maintaining a safe distance, following the object's direction, and realizing the dangerous object. • System has fast object tracking thanks to robust computer vision models. 	<ul style="list-style-type: none"> • The drone should stay at the proper distance from other sensor positions for easy activation. • SSDS is robust but complex, requiring a good working station to operate.

5. Conclusions

An SSDS was successfully developed to detect and track the objects concerned in case of intrusion and fire accidents with the support of AI models and IoT communication. Both of the computer vision models we used, YOLOv8 and Cascade Classifier, were trained and implemented in the workstation for object classification. Furthermore, three algorithms for drone control were implemented for the automation optimization of drone functions like target following and dangerous object avoidance. The entire system is capable of collecting the alerts from IoT sensors and manipulating the drone for acquiring the data, monitoring the stream to store the data, and transmitting the data to other responsible electronic devices via Wi-Fi. In the future, the developed system will be applied to more circumstances, such as particular factories, for further experiments to collect more information about the system's pros and cons. From this stage, more drone control algorithms and AI models can be employed in the smart flight system.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Zuo, Z.; Liu, C.; Han, Q.-L.; Song, J. Unmanned Aerial Vehicles: Control Methods and Future Challenges. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 601–614. [[CrossRef](#)]
2. Alsawy, A.; Hicks, A.; Moss, D.; Mckeever, S. An Image Processing Based Classifier to Support Safe Dropping for Delivery-by-Drone. In Proceedings of the 2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS), Genova, Italy, 5–7 December 2022.
3. Harrington, P.; Ng, W.P.; Binns, R. Autonomous Drone Control within a Wi-Fi Network. In Proceedings of the 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Porto, Portugal, 20–22 July 2020; IEEE: Piscataway, NJ, USA, 2020.
4. Chen, K.-W.; Xie, M.-R.; Chen, Y.-M.; Chu, T.-T.; Lin, Y.-B. DroneTalk: An Internet-of-Things-Based Drone System for Last-Mile Drone Delivery. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 15204–15217. [[CrossRef](#)]
5. *Drones in Smart-Cities: Security and Performance*; Al-Turjman, F. (Ed.) Elsevier Science Publishing: Philadelphia, PA, USA, 2020; ISBN 9780128199725.
6. Sun, Y.; Zhi, X.; Han, H.; Jiang, S.; Shi, T.; Gong, J.; Zhang, W. Enhancing UAV Detection in Surveillance Camera Videos through Spatiotemporal Information and Optical Flow. *Sensors* **2023**, *23*, 6037. [[CrossRef](#)] [[PubMed](#)]
7. Kim, B.; Min, H.; Heo, J.; Jung, J. Dynamic Computation Offloading Scheme for Drone-Based Surveillance Systems. *Sensors* **2018**, *18*, 2982. [[CrossRef](#)] [[PubMed](#)]
8. Zaheer, Z.; Usmani, A.; Khan, E.; Qadeer, M.A. Aerial Surveillance System Using UAV. In Proceedings of the 2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN), Hyderabad, India, 21–23 July 2016; IEEE: Piscataway, NJ, USA, 2016.
9. Boonsongsrikul, A.; Eamsaard, J. Real-Time Human Motion Tracking by Tello EDU Drone. *Sensors* **2023**, *23*, 897. [[CrossRef](#)] [[PubMed](#)]
10. MediaPose. Available online: <https://google.github.io/mediapipe/solutions/pose> (accessed on 11 September 2022).

11. Bazarevsky, V.; Grishchenko, I.; Raveendran, K.; Zhu, T.; Zhang, F.; Grundmann, M. BlazePose: On-Device Real-Time Body Pose Tracking. Available online: <https://arxiv.org/abs/2006.10204> (accessed on 11 September 2022).
12. Zhou, X.; Liu, S.; Pavlakos, G.; Kumar, V.; Daniilidis, K. Human Motion Capture Using a Drone. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; IEEE: Piscataway, NJ, USA, 2018.
13. Shah, S.A.; Lakho, G.M.; Keerio, H.A.; Sattar, M.N.; Hussain, G.; Mehdi, M.; Vistro, R.B.; Mahmoud, E.A.; Elansary, H.O. Application of Drone Surveillance for Advance Agriculture Monitoring by Android Application Using Convolution Neural Network. *Agronomy* **2023**, *13*, 1764. [[CrossRef](#)]
14. Jia, X.; Wang, Y.; Chen, T. Forest Fire Detection and Recognition Using YOLOv8 Algorithms from UAVs Images. In Proceedings of the 2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 26–28 July 2024; IEEE: Piscataway, NJ, USA, 2023.
15. Cuimei, L.; Zhiliang, Q.; Nan, J.; Jianhua, W. Human Face Detection Algorithm via Haar Cascade Classifier Combined with Three Additional Classifiers. In Proceedings of the 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, China, 20–22 October 2017; IEEE: Piscataway, NJ, USA, 2017.
16. Python. Available online: <https://www.python.org/> (accessed on 20 December 2022).
17. Peretz, Y. A Randomized Algorithm for Optimal PID Controllers. *Algorithms* **2018**, *11*, 81. [[CrossRef](#)]
18. Aivaliotis, V.; Tsantikidou, K.; Sklavos, N. IoT-Based Multi-Sensor Healthcare Architectures and a Lightweight-Based Privacy Scheme. *Sensors* **2022**, *22*, 4269. [[CrossRef](#)]
19. Kumar, M.; Kumar, S.; Kashyap, P.K.; Aggarwal, G.; Rathore, R.S.; Kaiwartya, O.; Lloret, J. Green Communication in Internet of Things: A Hybrid Bio-Inspired Intelligent Approach. *Sensors* **2022**, *22*, 3910. [[CrossRef](#)]
20. Wijesundara, D.; Gunawardena, L.; Premachandra, C. Human Recognition from High-Altitude UAV Camera Images by AI Based Body Region Detection. In Proceedings of the 2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS), Ise-Shima, Japan, 29 November–2 December 2022; IEEE: Piscataway, NJ, USA, 2022.
21. Hussain, M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines* **2023**, *11*, 677. [[CrossRef](#)]
22. Taye, M.M. Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation* **2023**, *11*, 52. [[CrossRef](#)]
23. Yamashita, H.; Morimoto, T.; Mitsugami, I. Autonomous Human-Following Drone for Monitoring a Pedestrian from Constant Distance and Direction. In Proceedings of the 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), Kyoto, Japan, 12–15 October 2021; IEEE: Piscataway, NJ, USA, 2021.
24. Liang, Q.; Wang, Z.; Yin, Y.; Xiong, W.; Zhang, J.; Yang, Z. Autonomous Aerial Obstacle Avoidance Using LiDAR Sensor Fusion. *PLoS ONE* **2023**, *18*, e0287177. [[CrossRef](#)] [[PubMed](#)]
25. Majchrzak, J.; Michalski, M.; Wiczynski, G. Distance Estimation with a Long-Range Ultrasonic Sensor System. *IEEE Sens. J.* **2009**, *9*, 767–773. [[CrossRef](#)]
26. Kawabata, S.; Lee, J.H.; Okamoto, S. Obstacle Avoidance Navigation Using Horizontal Movement for a Drone Flying in Indoor Environment. In Proceedings of the 2019 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), Majorca Island, Spain, 3–5 May 2019; IEEE: Piscataway, NJ, USA, 2019.
27. Motion Capture Systems. Available online: <https://optitrack.com/> (accessed on 24 November 2023).
28. Hercog, D.; Lerher, T.; Truntič, M.; Težak, O. Design and Implementation of ESP32-Based IoT Devices. *Sensors* **2023**, *23*, 6739. [[CrossRef](#)] [[PubMed](#)]
29. M5StickC P.I.R. Hat (AS312). Available online: <https://shop.m5stack.com/products/m5stickccompatible-hat-pir-sensor> (accessed on 5 October 2022).
30. Flame_sensor_SKU_DFR0076-DFRobot. Available online: https://wiki.dfrobot.com/Flame_sensor_SKU_DFR0076 (accessed on 17 October 2022).
31. ESP32. Available online: <https://www.espressif.com/en/products/socs/esp32> (accessed on 23 May 2023).
32. Hoang, M.L.; Carratu, M.; Paciello, V.; Pietrosanto, A. A New Orientation Method for Inclinometer Based on MEMS Accelerometer Used in Industry 4.0. In Proceedings of the 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, UK, 20–23 July 2020; IEEE: Piscataway, NJ, USA, 2020.
33. Hoang, M.L.; Pietrosanto, A. New Artificial Intelligence Approach to Inclination Measurement Based on MEMS Accelerometer. *IEEE Trans. Artif. Intell.* **2022**, *3*, 67–77. [[CrossRef](#)]
34. Hoang, M.L.; Carratu, M.; Ugwiri, M.A.; Paciello, V.; Pietrosanto, A. A New Technique for Optimization of Linear Displacement Measurement Based on MEMS Accelerometer. In Proceedings of the 2020 International Semiconductor Conference (CAS), Sinaia, Romania, 7–9 October 2020; IEEE: Piscataway, NJ, USA, 2020.
35. DJITelloPy: DJI Tello Drone Python Interface Using the Official Tello SDK. Available online: <https://djitellopy.readthedocs.io/en/latest>. (accessed on 10 July 2023).
36. Wang, X.; Gao, H.; Jia, Z.; Li, Z. BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8. *Sensors* **2023**, *23*, 8361. [[CrossRef](#)]

37. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting Tassels in RGB UAV Imagery with Improved YOLOv5 Based on Transfer Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8085–8094. [[CrossRef](#)]
38. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
39. Li, H.; Xiong, P.; An, J.; Wang, L. Pyramid attention network for semantic segmentation. *arXiv* **2018**, arXiv:1805.10180.
40. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
41. Gai, W.; Liu, Y.; Zhang, J.; Jing, G. An Improved Tiny YOLOv3 for Real-Time Object Detection. *Syst. Sci. Control Eng.* **2021**, *9*, 314–321. [[CrossRef](#)]
42. Culjak, I.; Abram, D.; Pribanic, T.; Dzapo, H.; Cifrek, M. A Brief Introduction to OpenCV. In Proceedings of the 35th International Convention MIPRO, Opatija, Croatia, 21–25 May 2012; pp. 1725–1730.
43. Viola, P.; Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 8–14 December 2001; IEEE Computer Society: Piscataway, NJ, USA, 2005.
44. Zhang, E.; Zhang, Y. Average Precision. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: Boston, MA, USA, 2009; pp. 192–193.
45. Tello Iron Man Edition. Available online: <https://www.ryzerobotics.com/ironman> (accessed on 10 October 2022).
46. Annotating Images Using OpenCV. Available online: <https://learnopencv.com/annotating-images-using-opencv/> (accessed on 2 January 2023).
47. Ultralytics Home. Available online: <https://docs.ultralytics.com/> (accessed on 4 January 2023).
48. OpenCV: Cascade Classifier Training. Available online: https://docs.opencv.org/4.x/dc/d88/tutorial_traincascade.html (accessed on 2 January 2023).
49. Hoang, M.L. Object Size Measurement and Camera Distance Evaluation for Electronic Components Using Fixed-Position Camera. *Comput. Vis. Stud.* **2023**. [[CrossRef](#)]
50. Karthi, M.; Muthulakshmi, V.; Priscilla, R.; Praveen, P.; Vanisri, K. Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace Validation of Dataset. In Proceedings of the 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSSES), Chennai, India, 24–25 September 2021; IEEE: Piscataway, NJ, USA.
51. Zeng, Y.; Zhang, T.; He, W.; Zhang, Z. YOLOv7-UAV: An Unmanned Aerial Vehicle Image Object Detection Algorithm Based on Improved YOLOv7. *Electronics* **2023**, *12*, 3141. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.