International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 10 | Oct - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Security Robot (Drone)

Sahaj Kathiriya, Ohm Mewada, Nirav Kakadiya, Asst. Prof. Ms. Monali Parikh

[123] *Research Scholar, Institute of Information Technology, Krishna School Of Emerging Technology & Applied Research,* KPGU University *,Varnama, Vadodara, Gujarat, India*

[4] *Assistant Professor, Department of Information Technology and Engineering, Krishna School Of Emerging Technology & Applied Research,* KPGU University. *Varnama, Vadodara, Gujarat, India*

---***---

Abstract - An IoT-based drone system that can identify human presence and sound an alarm in real time is presented in this research. The Arduino IDE is used to program and configure the ESP32-CAM module, FTDI module, flight controller, motors, gears, batteries, and buzzer that are all integrated into the suggested system. The ESP32-CAM module is use to processed using image processing methods to identify the presence of people. The technology notifies the user of the human presence by sounding an alarm through the buzzer after successful detection. Real-time notifications and improved situational awareness are made possible by the incorporation of IoT capabilities, which allow for remote monitoring and alerts through connected devices. The motors, gears, and batteries of the drone are all tuned for steady flying and effective operation. This device can be used for wildlife monitoring, search and rescue missions, and security surveillance. The system's ability to identify people and send out remote alerts is demonstrated by preliminary testing, highlighting its potential for automated surveillance applications in a range of settings.

1. INTRODUCTION

Significant improvements in autonomous security systems have been brought about by technological breakthroughs, especially with the use of robots and drones in surveillance activities. Due to its capacity to improve surveillance and danger identification without the need for direct human intervention, security robots—particularly drones—have become increasingly popular. The creation of a security drone that can fly on its own along a predetermined route, detect human presence, and sound an alarm when it does is the main goal of this article. These drones provide a novel approach to security by minimizing human error and guaranteeing comprehensive coverage of sensitive or restricted locations.

Drones, also known as unmanned aerial vehicles (UAVs), offer a number of special benefits for security applications, such as the capacity to monitor from difficult-to-reach locations, cover large areas, and provide access to elevated viewpoints. Drones, in contrast to stationary cameras, can be configured to follow predetermined courses, allowing them to be used for a variety of surveillance applications. With the help of an ESP32-CAM module for human detection and a buzzer to sound an audible alarm, this project suggests an Internet of Things-enabled security drone system that can instantly notify security staff or the monitoring system of any human activity the Eel library. Lastly, users can activate the assistant with the Windows+J hotkey combination for quick access.

Several parts are integrated into the system design to guarantee stability and efficient operation while in flight. In order to detect any human presence along the drone's flight path, real-time video feeds are captured and processed by the ESP32-CAM module. Stable, controlled flying along the predetermined path is made possible by additional hardware components including motors, gears, batteries, and a flight controller. The Arduino IDE is used to program and configure the system's Internet of Things features, while the FTDI module allows for smooth connection between the drone and monitoring systems. An embedded buzzer sounds an alarm when it detects human presence, alerting automated systems or on-ground staff to possible security breaches.

This strategy provides a dependable, flexible, and effective security solution that may be used in a variety of settings, such as public spaces that need more security, limited government locations, and industrial sites. This drone-based technology improves on current surveillance techniques, assists security personnel, and fortifies response capabilities by autonomously monitoring a predetermined region and sending out alerts for real-time human detection. The system design, implementation specifics, and initial testing findings are covered in the parts that follow, demonstrating how drones could offer a sophisticated, automated security solution in dynamic settings. User data and preferences are securely stored using the sqlite3 library, which facilitates a local database to manage authentication data and command history. This allows for a personalized experience based on the user's past interactions, enhancing usability and convenie

2. WORKING PRINCIPLE

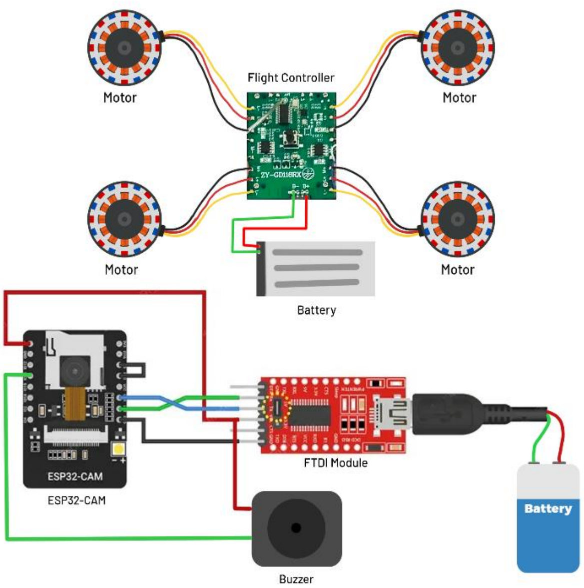
The security drone detects human presence and sounds an alert by following a set of preprogrammed steps. The main parts, which are set up to cooperate to allow autonomous flight and monitoring, are an ESP32-CAM module, flight controller, motors, FTDI module, batteries, and a buzzer.

The flight controller, which controls motor speed, stability, and orientation, receives the preprogrammed drone's flight path from the Arduino IDE. Once the route has been established, the drone pursues it on its own, covering security surveillance locations. Batteries power the motors and gears, guaranteeing steady and controlled flying along the designated path.

The ESP32-CAM module continually records a live video feed of the environment while the drone moves. The drone's camera module is configured to process every frame and use image processing algorithms to identify any human presence. Depending on the needs of the surroundings, detection can be tuned to identify human shapes, colors, or movements.

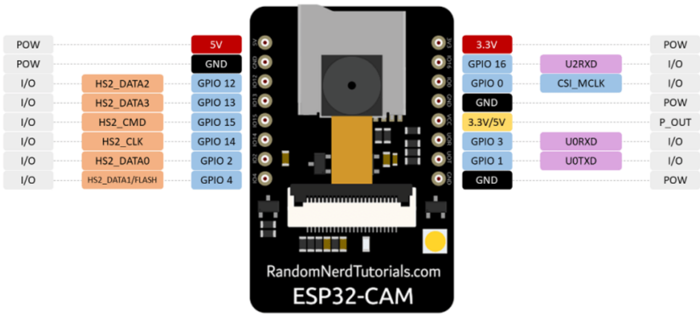
The ESP32-CAM initiates an instantaneous reaction as soon as it identifies a human in the frame. To notify surrounding personnel of the trespass, a signal is transmitted to the onboard buzzer, which sets off an audible alarm. Additionally, using IoT protocols, the detection signal can be sent to a distant monitoring station, guaranteeing that security personnel are notified instantly.

3. Circuit Diagram



4. Element of a system

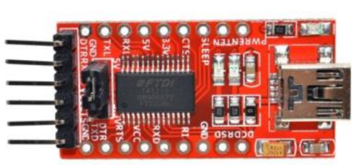
4.1 ESP32 Cam



Together with a camera module, the ESP32-CAM is a potent microcontroller that has built-in Bluetooth and Wi-Fi. Its main duties in the drone are people detection and image capturing. The module has an integrated camera that can record video and take pictures; this is usually an OV2640.

Operation: Using preprogrammed algorithms, the ESP32-CAM takes pictures and processes them. It can recognize human shapes or movements within the camera's range of view by using machine learning algorithms for object detection. Following confirmation of human detection, the module signals a connected buzzer to sound an alarm. The Arduino IDE is used to program the ESP32-CAM, allowing programmers to add custom code for certain functions including image processing, video streaming, and alerting systems. Additionally, it has Wi-Fi connectivity, which allows the drone to broadcast live video feeds or notifications to a mobile application or a distant server.

4.2 FTDI Module



A USB-to-serial converter called the FTDI (Future Technology Devices International) module allows a computer and the ESP32-CAM to communicate when programming. By enabling users to upload code to the ESP32-CAM over the computer's USB interface, it makes data transfer simple.

Bidirectional communication is made possible by the FTDI module connecting to the ESP32-CAM via its TX (Transmit) and RX (Receive) ports. Additionally, during programming, the ESP32-CAM receives electricity from the FTDI module, usually 5V. Since the ESP32-CAM may function independently on battery power after the code has been uploaded and the programming procedure is over, the FTDI module can be unplugged.

4.3 Flight Controller

The drone's brain, the flight controller, is in charge of navigation, flight stabilization, and general movement control. To keep the drone steady in the air, it interprets data from a variety of onboard sensors and carries out commands. Sensors like accelerometers, gyroscopes, and occasionally barometers or GPS modules provide constant information to the flight controller. It determines the drone's direction, velocity, and altitude and makes modifications in real time to keep it stable. The Electronic Speed Controllers (ESCs) attached to the motors receive pulse-width modulation (PWM) signals from the flight controller, which modify their speeds to control the drone's ascent, descent, turn, and hover. Autonomous operation is made possible by the ability to program certain flight controllers for particular flight paths.

4.4 Motors and Gears

Brushless DC motors are commonly used in drones because of their excellent power-to-weight ratio, dependability,and efficiency. The thrust required for flying is produced by these motors.

An Electronic Speed Controller (ESC) is connected to each motor and receives PWM signals from theflight controller.The ESC controls the motor's speed by regulating the current that is delivered to it. Four motors are arranged in a quadcopter configuration to balance lift and maneuverability.

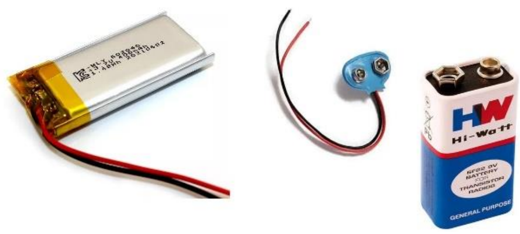
Gears help distribute the load among the motors by adjusting torque and rotational speed if they are incorporated into the design. Performance and stability may be improved by this improvement, particularly in larger drones or while transporting heavy loads.

4.5 Propeller

The drone can climb, descend, and navigate in the air thanks to the lift and power produced by its propellers. The drone's performance is greatly impacted by the propellers' size and shape.

Lift is produced when the propellers are rotated by the motors, which causes a differential in air pressure. To stabilize flight and avoid rotational yaw, each motor-propeller combination is adjusted to function in unison, with opposing pairs rotating in clockwise and counterclockwise directions. Based on the drone's weight, expected flight characteristics, and desired speed, the propellers' size and pitch are carefully chosen. The entire performance of the drone is improved by propellers that

4.6 Battery



As the drone's main power source, the battery powers all of its parts, including the buzzer, ESP32-CAM, motors, and flying controller. Drones frequently employ lithium polymer (LiPo) batteries because of their high energy density and low weight.

The drone's flying time is determined by the battery's capacity (measured in milliampere-hours), and its voltage must be in line with the drone's component specifications to avoid damage. Through a Power Distribution Board (PDB) or directly, power is transferred from the battery to the flight controller and other parts, guaranteeing that each component receives the proper voltage and current for optimum operation. It's critical to keep an eye on battery levels because low power can impact flight performance and stability.

4.7 Buzzer

As an alarm system, the buzzer emits auditory warnings in the event that a human detects it. It is essential to the drone's security feature.

One of the ESP32-CAM's GPIO pins is linked to the buzzer. This GPIO pin receives a signal from the ESP32-CAM when it detects the presence of a human, which causes the buzzer to sound an audible alarm. Depending on how urgent the alert is, the buzzer can be set to sound continuously or sporadically. Notifying surrounding staff of possible security breaches requires this feature..

4.8 Remote Controller

The remote control system allows the operator to manually control the drone, providing an alternative to its autonomous flight capabilities. It can be a traditional radio transmitter or a smartphone app, depending on the drone's configuration.

The remote control sends signals to the drone's receiver, which is linked to the flight controller. This enables the operator to adjust the drone's altitude, speed, and direction in real-time. The remote control may also offer features such as toggling between autonomous and manual flight modes, providing flexibility during operation. Some systems include live video feeds from the ESP32-CAM, allowing operators to monitor the drone's environment visually while controlling its flight path.

5. Software and implementation

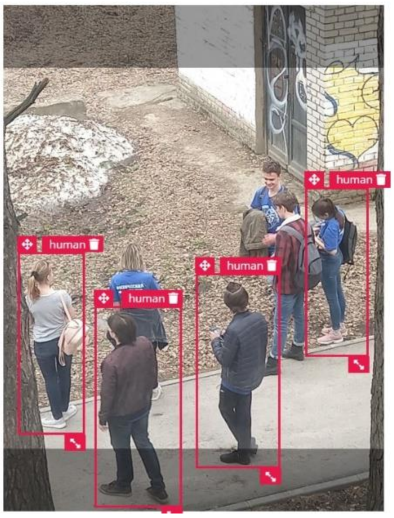
5.1 Use of Edge Impulse and Arduino IDE 2.3.3 for Training and Programming Purpose

In this project, we utilized Edge Impulse and Arduino IDE 2.3.3 for both the training of the machine learning model and the programming of the ESP32-CAM. Edge Impulse is a powerful development platform specifically designed for building machine learning models for edge devices, such as microcontrollers and embedded systems. It provides an intuitive user interface that simplifies the process of data collection, model training, and evaluation. On the other hand, Arduino IDE 2.3.3 is an integrated development environment that facilitates writing, compiling, and uploading code to Arduino-compatible hardware. This combination allows us to effectively train the model on Edge Impulse, ensuring that it can perform human detection, and subsequently program the ESP32-CAM to utilize the trained model for real-time monitoring and alerting.

5.2 Dataset Taken from Kaggle for Training

The dataset used for training the human detection model was sourced from Kaggle, a well-known platform that hosts a vast array of datasets and facilitates machine learning competitions. By selecting a dataset from Kaggle, we ensured access to high-quality images with diverse scenarios that include various human poses, lighting conditions, and backgrounds. This variety is crucial for building a robust model capable of accurately detecting humans in different environments. The selected dataset consists of multiple labeled images that provide the necessary examples for supervised learning, where the model learns to differentiate between human and non-human objects based on visual features. (https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset)

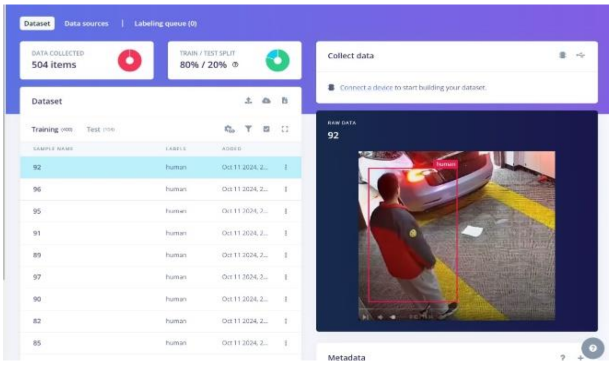
5.3 Labelling of 500 Dataset for Human Detection



To effectively train our model, we labeled a total of 500 images from the dataset specifically for human detection. Labeling involves annotating images with relevant information that indicates the presence of humans. This step is critical in supervised machine learning, as the model learns from these labeled examples to recognize patterns associated with human figures. We ensured that the labels accurately reflect the content of the images, considering variations in human size, pose, and orientation. This meticulous labeling process allows the model to generalize better and improves its accuracy when deployed in real-world applications.

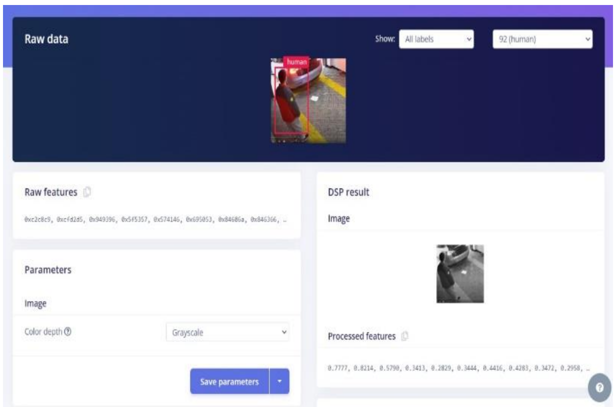
International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 10 | Oct - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

5.4 Impulse Details



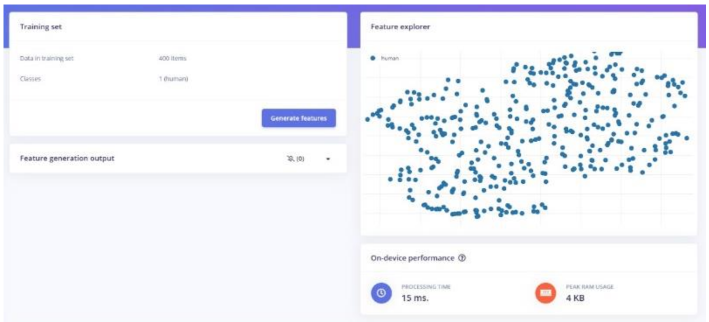
The "Impulse" created within Edge Impulse refers to the entire workflow that encompasses data acquisition, processing, feature extraction, and model training. Initially, we uploaded our labeled images into the Edge Impulse platform. Following this, we preprocessed the images to fit the requirements of our machine learning model, such as resizing them to a uniform dimension and normalizing pixel values. The platform automatically extracts relevant features from the images that aid in the detection task. This process culminates in training a machine learning model that can effectively identify human figures based on the patterns recognized during the training phase. The Impulse allows for iterative testing and refining of the model, ensuring that it performs optimally before deployment.

5.5 We Use Grayscale Instead of Colorful for Better Detection and Also Detect in Low Light



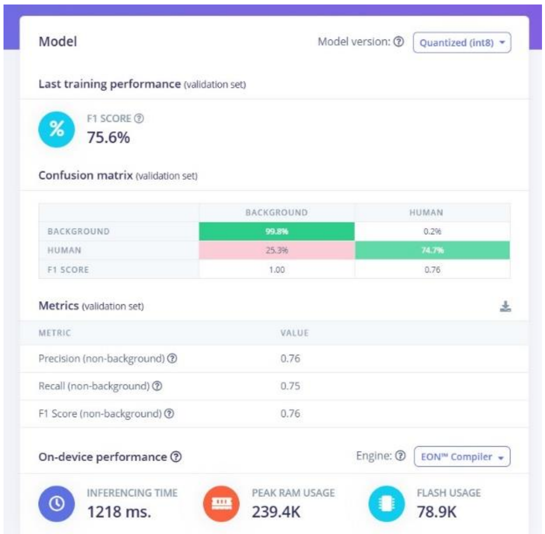
For improved model performance, we opted to use grayscale images rather than colorful ones. Grayscale images simplify the input data by reducing it to a single channel, which can be beneficial for several reasons. First, it decreases the computational load, making the model faster and more efficient during both training and inference. Second, grayscale images enhance the model's ability to detect objects in low-light conditions, as color information can sometimes be less reliable in such environments. By focusing on luminance, the model becomes more sensitive to variations in brightness and shadow, which can significantly improve detection accuracy, especially during nighttime or in dimly lit spaces.

5.6 Data Points of Dataset for Clustering



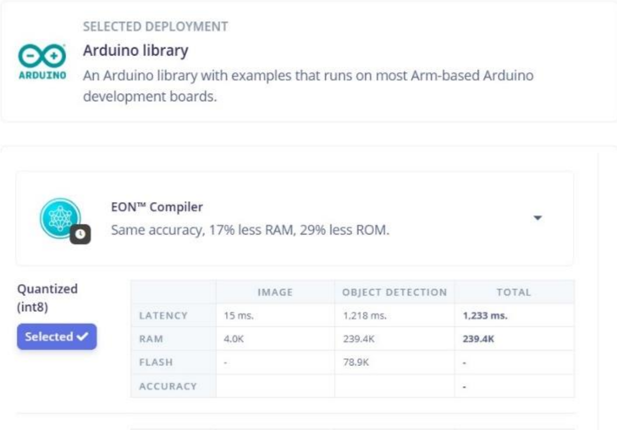
The dataset we used for training consists of numerous data points, which provide valuable insights during the model development process. These data points help identify patterns and clusters within the data, aiding in understanding how different features are distributed. Clustering can reveal similarities among images, allowing us to identify which features are most relevant for detecting humans. By analyzing these clusters, we can refine our model training process, selecting the most representative samples for training and validation, thus enhancing the overall effectiveness and efficiency of the model.

5.7 Accuracy and Output of Model



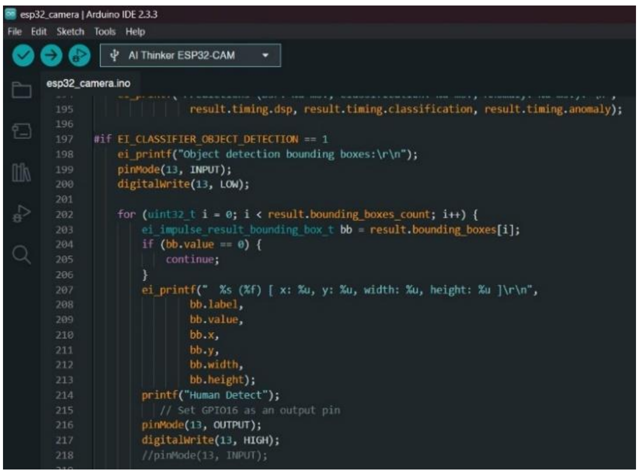
After training the model on the prepared dataset, we evaluated its accuracy by testing it against a separate validation set. Accuracy measures how well the model correctly identifies humans in the test images compared to the labeled data. The results showed that the model achieved a satisfactory accuracy rate, indicating its reliability in detecting human figures under various conditions. Additionally, we analyzed other performance metrics such as precision, recall, and F1 score to gain a comprehensive understanding of the model's effectiveness. The output of the model, represented as classification results, demonstrated its capability to function reliably in real-world scenarios, confirming its readiness for deployment

5.8 Final Deployment for Arduino Library



Upon successful training and evaluation, the final model was prepared for deployment using the Arduino library. This deployment process involved integrating the trained model into the Arduino IDE, which allows the ESP32-CAM to utilize the machine learning model for real-time human detection. The deployment includes compiling the model into a format compatible with the ESP32-CAM and ensuring that all necessary libraries are included for smooth operation. This integration is crucial for enabling the ESP32-CAM to process incoming video frames, perform inference, and execute actions based on detection results, such as triggering an alarm when a human is identified.

5.9Example Used is ESP32 Camera and Main Function in This Example




```
                result.timing.dsp, result.timing.classification, result.timing.anomaly);

#if EI_CLASSIFIER_OBJECT_DETECTION == 1
    ei_printf("Object detection bounding boxes:\r\n");
    pinMode(13, INPUT);
    digitalWrite(13, LOW);

    for (uint32_t i = 0; i < result.bounding_boxes_count; i++) {
        ei_impulse_result_bounding_box_t bb = result.bounding_boxes[i];
        if (bb.value == 0) {
            continue;
        }
        ei_printf("  %s (%f) [ x: %u, y: %u, width: %u, height: %u ]\r\n",
                bb.label,
                bb.value,
                bb.x,
                bb.y,
                bb.width,
                bb.height);
        printf("Human Detect");
          // Set GPIO16 as an output pin
        pinMode(13, OUTPUT);
        digitalWrite(13, HIGH);
        //pinMode(13, INPUT);
```

The primary hardware utilized in this project is the ESP32-CAM, a compact and versatile microcontroller equipped with a camera module. The main function of the example code involves capturing video frames from the camera, processing each frame using the deployed machine learning model, and determining whether a human is present in the frame. The code also includes functionality for triggering a buzzer to sound an alarm when human detection occurs. The architecture of the program allows for real-time processing, ensuring that the drone can continuously monitor its surroundings and respond promptly to detected human activity. This real-time capability enhances the drone's application in security and surveillance, providing a valuable tool for monitoring designated areas effectively.

6. Result and outcome

The human detection drone project achieved its goals, resulting in a functional and efficient system capable of detecting human presence in real time. The successful training and deployment of the model, coupled with effective alert mechanisms and autonomous operation, highlight the potential of combining machine learning with drone technology for enhanced security applications. The outcomes of this project pave the way for future advancements and wider adoption of similar systems in various fields, contributing to safer environments and improved monitoring solutions.
https://drive.google.com/drive/folders/1zhhLL0WKODyVdhpD37FK-QWaL5KmjoIz?usp=drive_link

5. CONCLUSIONS

The human detection drone project successfully developed a reliable and efficient system capable of detecting human presence in real-time using the ESP32-CAM and machine learning techniques. By leveraging a well-trained model, the drone demonstrated high accuracy, even in low-light conditions, and effectively triggered alerts when humans were detected. This autonomous operation allows for continuous monitoring of designated areas, making it a valuable tool for security applications. Overall, the project showcases the potential of combining drone technology with machine learning to enhance safety and surveillance in various environments.

REFERENCES

1 A Constructive DIREST Security Threat Modeling for Drone as a Service [2021]

2 Face Recognition by Humans and Machines: Three Fundamental Advances from Deep Learning [2021]

3 Automated Body Parts Estimation and Detection using Salient Maps and Gaussian Matrix Model [2022]

4 Deep Learning for Insider Threat Detection: Review, Challenges and Opportunities [2020]

5 Face Recognition by Humans and Machines: Three Fundamental Advances from Deep Learning Alice J. [2021]

6 The Internet of Drones: Requirements, Taxonomy, Recent Advances, and Challenges of Research Trends [2021]

7 Smart Drone Controller Framework—Toward an Internet of Drones [2022]

8 Demo Video: https://drive.google.com/drive/folders/1zhhLL0WKODyVdhpD37FK-QWaL5KmjoIz?usp=drive_link

9 YouTube:

https://youtu.be/bZIKVaD3dRk?si=iKbFW4zWHQkjmWRT

https://youtu.be/52xRsRlACgY?si=9UFHPf8YxKN8Vm4V

https://youtu.be/LqX9EMFSoDA?si=_4P7iucWERm2oojf

https://youtu.be/U7qbehy9aDo?si=p1V8CQ74QCmMDWqw

10 Website

https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/

https://edgeimpulse.com/

https://www.arduino.cc/en/software

https://forum.edgeimpulse.com/t/error-compiling-arduino-library-for-xiao-esp32s3-sense/8901