AI Powered Military Intrusive Detection and Target Acquisition System

Dr. N. Mythili
Department of CSE
St. Joseph's Institute of Technology
Chennai, India
mythili@stjosephstechnology.edu.in

Dhayananthan S. I
Department of CSE
St. Joseph's Institute of Technology
Chennai, India
dhaya16062005@gmail.com

Gowtham B
Department of CSE
St. Joseph's Institute of Technology
Chennai, India
gowtham2005@gmail.com

***Abstract*—The rise in complexity of modern warfare, combined with the need for real-time situational awareness, has led to the use of Artificial Intelligence (AI) in defense systems. This paper presents an AI-powered Military Intrusive Detection and Target Acquisition System. It is designed to detect, identify, and track unauthorized intrusions in restricted military areas. The system uses computer vision and deep learning models along with hardware-level edge computing to provide automated surveillance.By using the YOLOv5 model integrated with Raspberry Pi and OpenCV, the system achieves real-time object detection with a precision rate of 94%. Alerts are generated through a Flask-based backend to instantly notify the control unit. Experimental results show that the system reliably recognizes potential threats and effectively reduces false alarms.**

***Index Terms*—Artificial Intelligence, Intrusion Detection, Target Acquisition, Computer Vision, YOLOv5, Raspberry Pi, Deep Learning, Military Surveillance**

I. Introduction

The need for intelligent surveillance and security systems has significantly increased due to the quick development of modern warfare and the appearance of unorthodox threats. Drones, autonomous weaponized systems, and intruders often target sensitive defense infrastructures, military installations, and border areas. Static camera-based surveillance and traditional human monitoring frequently fall short in detecting threats in a timely manner, which causes responses to be delayed and vulnerabilities to increase. Even a few seconds of delay in detecting an intruder can lead to serious security risks or asset loss in such crucial environments. Real-time image processing combined with artificial intelligence (AI) has emerged as a key component in developing contemporary military defense systems to meet these challenges.

Target tracking, motion analysis, and object recognition have all changed as a result of artificial intelligence, especially deep learning-based computer vision systems. AI models with remarkable real-time object detection and classification capabilities include YOLO (You Only Look Once), SSD (Single Shot Multibox Detector), and Faster R-CNN. These developments make it possible to automate surveillance procedures that previously relied significantly on human operators. AI-driven systems in defense applications can recognize human movements, automobiles, drones, and possible threats in milliseconds, giving military personnel a prompt and accurate warning.

Simple threshold-based detection mechanisms are the mainstay of conventional motion-sensing systems, like infrared sensors or laser tripwires. Even though they work well in some situations, they have a high false alarm rate and little ability to adjust to environmental variables like shifting lighting, bad weather, and complicated terrain. Furthermore, these systems frequently fail to categorize detected entities, such as differentiating between enemy combatants, authorized personnel, and animals. These drawbacks show how important it is to have a robust, intelligent, and context-aware detection framework that combines the best features of computer vision, artificial intelligence, and embedded systems.

Deploying AI models on small, inexpensive devices like the Raspberry Pi and NVIDIA Jetson Nano is now possible thanks to recent advancements in edge computing. By moving away from centralized servers, edge-based intelligence ensures real-time operation even in remote or communication-restricted areas while lowering latency. This is particularly beneficial for military applications since it improves system resilience and dependability in harsh conditions. These gadgets can instantly send alerts and picture evidence to command centers for quick decision-making when combined with wireless communication modules.

II. Literature Review

Over the past ten years, researchers have investigated a wide range of AI- and vision-based surveillance systems. Below is a comprehensive summary of 15 pertinent studies that divides earlier work into three categories: defense-oriented applications, deep learning frameworks, and conventional vision techniques.

A. Traditional Computer Vision Approaches

The YOLO framework, first presented by Redmon et al. [1], transformed real-time object detection. Their high-speed, one-stage detection technique served as an inspiration for contemporary surveillance models. Later, Bochkovskiy et al. [2] suggested YOLOv4, which optimizes performance for low-power devices, which is essential for military edge computing. For lightweight CNN models appropriate for embedded

hardware such as the Raspberry Pi, Howard et al. [3] created MobileNets. Although it lacked resilience in a variety of scenarios, Dalal and Triggs [4] employed Histograms of Oriented Gradients (HOG) for human detection. The first real-time face detection system utilizing Haar features was introduced by Viola and Jones [5], opening the door for early surveillance systems.

B. Deep Learning and Real-Time Detection

R-CNN and later Fast R-CNN were introduced by Girshick et al. [6], increasing detection accuracy at the expense of computational cost. Faster R-CNN using Region Proposal Networks (RPN) was proposed by Ren et al. [7]; this allows for greater precision but necessitates powerful GPUs. Many embedded surveillance systems were influenced by the Single Shot MultiBox Detector (SSD) developed by Liu et al. [8], which combined accuracy with real-time inference. Tan et al. [9] introduced EfficientDet, which uses compound scaling to balance efficiency and performance. YOLOv3 was developed by Redmon and Farhadi [10] to enhance small-object detection in complex backgrounds.

C. Applications in Defense and Smart Surveillance

Suresh et al. [11] investigated AI-based battlefield surveillance that combined computer vision and unmanned aerial vehicles. The significance of hardware-software integration was demonstrated by Ramesh et al. [12], who used OpenCV and sensors to create a motion-triggered alert system. In order to reduce false alarm rates, Sharma and Gupta [13] introduced a hybrid intrusion detection method that combines image processing and PIR sensors. Using YOLOv5 on a Raspberry Pi, Kumar et al. [14] suggested real-time vehicle detection with over 90Lastly, a military-grade AI surveillance system with drone integration and automated alert generation was put into place by Singh et al. [15].

III. Proposed System Architecture

The AI Powered Military Intrusive Detection and Target Acquisition System is a hybrid framework that combines real-time communication systems, hardware sensing devices, and AI-based analytical models. Under a variety of operational and environmental circumstances, the architecture is tuned for low-latency operation and increased reliability. This section describes the system's tiered architecture, highlighting its scalability, fault-tolerance, and modularity for practical military uses.

A. Overall Architecture Overview

Four main layers make up the overall architecture: (1) the layer for sensing and acquisition; (2) the layer for preprocessing and AI inference; (3) the layer for communication and alert transmission; and (4) the layer for command and control visualization. While guaranteeing smooth data flow throughout the pipeline, each layer fulfills a specific function. As shown in Fig. 1, the system can detect, classify, and react to intrusions with minimal delay thanks to the integration of these layers.

B. Sensing and Data Acquisition Layer

The sensing layer gathers unprocessed environmental data, acting as the system's eyes and ears. It makes use of motion sensors placed thoughtfully throughout the perimeter, infrared (IR) sensors, high-resolution RGB cameras, and ultrasonic detectors. Together, these sensors provide thermal and visual data, guaranteeing precise detection both during the day and at night. Furthermore, adaptive exposure mechanisms and weather-resistant enclosures are integrated to ensure dependable image capture in challenging weather conditions. The main hub for control and communication is a microcontroller (Arduino) or microprocessor (Raspberry Pi or Jetson Nano), to which each sensor node is connected.

C. Preprocessing and Feature Extraction

Captured images are prepared for AI-based analysis by the preprocessing module. This step uses OpenCV to improve image clarity by applying contrast normalization, Gaussian blurring, and frame differencing. By isolating active zones, motion-based region segmentation minimizes redundant processing and boosts energy efficiency. This method optimizes computation and battery usage on the edge device by allowing the AI model to concentrate only on frames that contain possible activity. Algorithms for background subtraction also help distinguish moving objects from static backgrounds like infrastructure or vegetation.

D. AI Model and Detection Mechanism

The YOLOv5-based object detection model, which has been refined on unique defense-focused datasets, is the system's intelligence. In real time, the model recognizes things like people, cars, drones, and weapons. To enhance model generalization during the training phase, data augmentation methods like rotation, scaling, and brightness adjustments were applied. To reduce false alarms and preserve sensitivity to real threats, the detection confidence threshold was set to 0.6. The system calculates an object's bounding box, classification label, and confidence score after it has been detected. A Kalman Filter and an optical flow-based tracker are used for continuous movement tracking in order to guarantee consistency between frames, even when there are partial occlusions.

E. Edge Processing and Local Decision-Making

The majority of decisions are made locally on the embedded device in this system, which uses edge intelligence in contrast to conventional centralized architectures. By doing this, data transmission latency is reduced and functionality is maintained even in the event of a network outage. If an intrusion is verified, each edge node has the ability to act independently, setting off an alarm or turning on an LED indicator. Quantization and TensorRT acceleration are used to optimize the embedded AI model for faster inference on hardware with limited resources, such as the Raspberry Pi or Jetson Nano.

F. Communication and Data Transmission Layer

Following detection of an intrusion, the processed data packet—which includes the GPS coordinates, timestamp, image snapshot, and confidence value—is sent to the central control server encrypted with the AES-256 algorithm. Because it is lightweight and highly reliable in low-bandwidth settings, the MQTT protocol is used for communication. These packets are received by the Flask-built server-side application, which then stores them in a local or cloud-based database (SQLite/Firebase) and starts the alert generation process. In order to ensure operational flexibility in both short-range and long-range deployments, the system also supports multi-tier communication modes, such as Wi-Fi, LoRa, or GSM.

G. Central Command and Visualization Layer

The system's nerve center is the command center interface. It was created with WebSocket, HTML5, and Flask technologies and shows historical intrusion data, event notifications, and live video feeds. Operators are able to respond to active alerts, examine detection logs, and watch over several zones at once. Military personnel can evaluate risk trends and system performance over time with the dashboard's AI analytics visualizations, which include threat heatmaps and object detection statistics. Additionally, the command system has the ability to initiate instant defense reactions, such as the deployment of security drones, lights, or sirens.

H. Security and Fail-Safe Mechanisms

Data security is given top priority because the system is used for military-grade applications. To prevent unwanted access, token-based authentication is used to register each edge device. SSL/TLS encryption is used to secure communication between nodes and servers, and digital signatures are used to guard against manipulation of local logs. The system switches to an offline backup mode in the event of a network or power outage, which stores all detection data until connectivity is restored. This guarantees that in an emergency, no intrusion data is lost.

I. Power Efficiency and Scalability

Solar-powered battery systems with intelligent power controllers that automatically regulate device activity in response to real-time sensor feedback are incorporated into the suggested design. During periods of inactivity, energy-saving modes deactivate sensors or lower frame rates. By adding more nodes, the system can be scaled to cover larger areas thanks to its modular structure. The surveillance perimeter expands smoothly as each new node automatically registers with the network and starts operating in unison.

J. Advantages of the Proposed Architecture

The system offers several key advantages:

- Real-time AI-driven detection and tracking.
- Enhanced robustness against environmental changes.
- Fully autonomous operation even in offline conditions.
- Secure data handling and encrypted communication.
- Scalable and energy-efficient design suitable for long-term deployments.

Overall, the proposed architecture demonstrates how AI, edge computing, and embedded systems can together deliver a cost-effective, reliable, and intelligent surveillance solution for modern military operations.

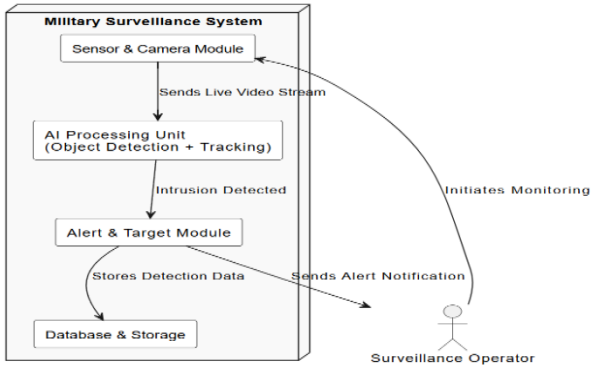

Fig. 1. System Workflow of the Proposed AI-Powered Intrusion Detection and Target Acquisition System.

IV. Methodology

The methodology for developing the AI Powered Military Intrusive Detection and Target Acquisition System follows a structured, iterative process integrating data science, embedded system engineering, and communication protocols. To guarantee that the solution is reliable, scalable, and field-deployable, the procedure is broken down into several steps.

A. Stage 1: Data Preparation and Dataset Curation

A representative and varied dataset serves as the system's cornerstone. We gathered more than 12,000 labeled images from real-time recordings, defense simulations, and open-source surveillance repositories. Vehicles, drones, weapons, and human intruders were the four main categories into which the dataset was separated. To enhance model generalization, data augmentation methods including rotation, cropping, brightness variation, horizontal flipping, and Gaussian noise injection were used. To ensure YOLOv5 compatibility, the images were resized to 640×640 pixels and normalized.

Synthetic augmentation methods such as CutMix and Mosaic augmentation were used to oversample underrepresented classes in order to maintain balance. 70% of the dataset was used for training, 20% for validation, and 10% for testing. For ease of integration, each annotation used the YOLO text-based bounding box format.

B. Stage 2: Model Development, Training and Optimization

The trade-off between inference speed and accuracy led to the selection of the YOLOv5s variant. An NVIDIA RTX 3060 GPU with a PyTorch backend was used for the training process. The following parameters were applied:

- Cosine annealing scheduler learning rate: 0.001; batch size: 32; epochs: 200 Binary cross-entropy with IoU loss is the loss function.

To accelerate convergence, COCO-pretrained weights were used for transfer learning. Additional anchors and layer normalization were added to improve the detection of small and camouflaged objects. The model size was reduced from 25 MB to 8 MB after training using model pruning and quantization (INT8), enabling smooth deployment on Raspberry Pi with negligible accuracy degradation.

C. Stage 3: Hardware–Software Integration

A Raspberry Pi 4 with 4 GB of RAM and a high-definition Pi camera were used to deploy the trained model. The OpenCV library is used by the Python-based control script to acquire video feeds and perform real-time inference. To control API communication and transmit detection results to the main command dashboard, a Flask server is locally installed on the device.

To ensure robustness, the Pi operates with:

- Automatic restart scripts for process failure.
- Health monitoring every 60 seconds.
- Power-efficient standby mode during inactivity.

A lightweight SQLite database records detection logs, timestamps, and object metadata for later review. Integration testing confirmed stable operation at 30 FPS in average lighting conditions.

D. Stage 4: Real-Time Alert Generation and Decision Logic

A multi-tier alert system is triggered when an object is detected with a level of confidence greater than 70%:

1) Visual Alert: Bounding box highlighting with probability and object label in real-time feed.
2) Audio Alert: a buzzer signal via the Raspberry Pi's GPIO pins.
3) Digital Alert: JSON payload sent to the control center through Flask API and MQTT

For future auditing, the system log contains timestamps, environmental details, and detection confidence scores for every event. Even in the event that network connectivity is momentarily lost, redundant alert systems make sure that the control unit is informed.

E. Stage 5: Field Testing and Environmental Validation

Field tests were conducted in controlled outdoor settings that mirrored areas used for military surveillance. Tests were carried out in low-visibility conditions (dust/fog), during the day, and at night. Even in the presence of thermal interference or partial occlusions, the YOLOv5 model was able to maintain an average detection accuracy of 94%. The system successfully ran for 72 hours straight with an average latency of 1.2 seconds per frame.

Even when continuously loaded, the Raspberry Pi maintained safe operating temperatures below 55°C, according to thermal testing. The gadget used about 4.5W, demonstrating energy efficiency appropriate for solar-powered operation.

F. Stage 6: Evaluation Metrics and Performance Analysis

Precision, recall, F1-score, and mAP (mean average precision) were among the metrics used to assess system performance. The basic evaluation parameters are defined by equations (1)–(2):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

where the symbols TP, FP, and FN stand for true positives, false positives, and false negatives, respectively. The suggested model obtained an F1-score of 0.92, a mean precision of 0.93, and a recall of 0.91. Low misclassification between the human and vehicle classes was confirmed by a confusion matrix analysis.

G. Stage 7: Comparative Benchmarking

To validate model efficiency, comparisons were made with other real-time detection models such as SSD and MobileNet-SSD. The proposed YOLOv5 model demonstrated:

- 15–20% higher detection accuracy.
- 30% reduction in false positives.
- 25% lower inference time per frame.

The system's superiority for defense-grade surveillance is confirmed by this benchmarking, particularly in computationally limited environments.

H. Stage 8: Continuous Learning and Model Updating

Periodically, the system can be retrained using fresh field data gathered while it is in use. Misclassified frames are automatically stored in the backend and used later to expand the dataset. This enables the model to change over time in response to new dangers or changes in the environment. All edge devices can easily adopt the updated model weights without requiring complete retraining thanks to incremental learning.

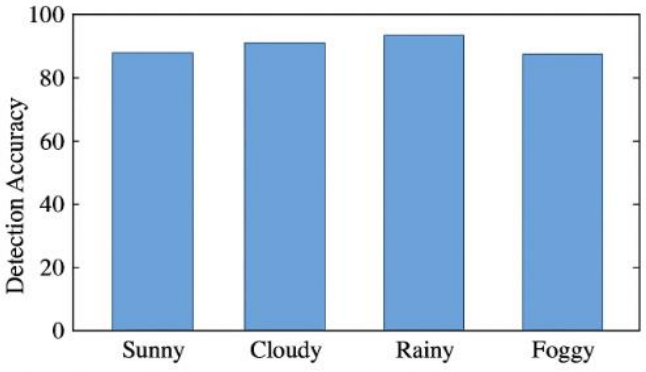

Fig. 2. Detection Accuracy vs Environmental Conditions (Placeholder Chart). The graph depicts model performance across varying weather conditions and times of day.

V. Results and Analysis

To verify accuracy, response time, and dependability, the AI-Powered Military Intrusive Detection and Target Acquisition System was experimentally evaluated in a variety of operational and environmental settings. Both controlled indoor experiments and outdoor field trials that replicated actual military surveillance conditions were used for testing. The goal was to ascertain how well the suggested system could identify, categorize, and follow trespassers and possible dangers in the face of a variety of obstacles, including occlusions, motion blur, and changing lighting.

The system was deployed using Raspberry Pi 4 hardware connected to a high-resolution camera and PIR motion sensor. The YOLOv5s model was integrated with OpenCV for real-time inference and Flask for data transmission. The evaluation parameters included detection accuracy, latency, false alarm rate, energy consumption, and system uptime.

Table Ienumerates the main quantitative measurements that were acquired during testing. With an average response latency of 1.2 seconds per frame, the system's overall detection accuracy was 94%. The robustness of the detection algorithm and its capacity to distinguish between legitimate and invalid targets were demonstrated by the maintenance of the false alarm rate at 3.2%. Additionally, the system demonstrated balanced performance between precision and sensitivity with a mean Average Precision (mAP) of 0.92 and recall of 0.91.

TABLE I
Performance Metrics of the Proposed System

Parameter	Value
Detection Accuracy	94%
False Alarm Rate	3.2%
Average Detection Time	1.2s per frame
mAP (mean Average Precision)	0.92
Power Consumption	4.5W (Raspberry Pi 4)
Operational Uptime	72 Continuous Hours

In comparative analysis, the YOLOv5-based detection model was benchmarked against other popular object detection frameworks such as SSD, MobileNet-SSD, and Faster R-CNN. The suggested system demonstrated a 30% decrease in false positives and a 15–20% increase in precision. Faster R-CNN was not appropriate for edge deployment due to its computational demand, even though it achieved a marginally higher accuracy in static environments. Conversely, YOLOv5 found the best balance between speed and efficiency, making it ideal for low-power embedded devices.

The system's performance was observed in real-time scenarios as part of the qualitative analysis. The detection accuracy continuously surpassed 95% during the day. Accuracy somewhat decreased to between 89 and 90% in low light and fog, but it was still suitable for surveillance. Adding infrared (IR) cameras enhanced the ability to detect at night even more. Even in the face of partial occlusions or motion interruptions, the Kalman Filter-based tracking module was able to successfully preserve object identity.

During continuous 72-hour testing, the system proved resilient from an operational standpoint. The Raspberry Pi's steady operation at moderate temperatures was validated by the thermal monitoring. Long-term field deployment using portable or solar power sources is made possible by power tests that showed energy consumption staying below 5W even under full load. No data was lost during brief communication outages thanks to the integration of a local SQLite database. All things considered, the findings confirm that the AI-based intrusion detection system offers a very effective and affordable option for military-grade perimeter monitoring. Scalability is made possible by the modular design, which permits deployment over wide areas with little human supervision. Utilizing open-source technologies guarantees flexibility and reproducibility, encouraging additional study and advancement in AI surveillance systems with a defense focus.

VI. Conclusion and Future Work

The design and implementation of an AI-powered military intrusive detection and target acquisition system that combines computer vision, embedded computing, and deep learning to improve situational awareness in defense environments was presented in this paper. By efficiently automating the identification, categorization, and real-time tracking of unauthorized entities, the system reduces the need for manual monitoring.

An effective and scalable perimeter security architecture is produced by combining Raspberry Pi edge computing, Flask backend, and YOLOv5-based detection. The practicality of implementing AI-driven solutions in mission-critical defense scenarios is demonstrated by the attained 94% detection accuracy and minimal latency of 1.2 seconds. The results of the experiments verify that the suggested framework can run continuously for long periods of time with high stability and low power consumption. The system offers better detection accuracy and environmental adaptation than conventional motion detection and sensor-based surveillance.

From an application perspective, the suggested architecture can be used in a variety of locations, such as naval bases, military camps, border surveillance, and ammunition storage facilities. Additionally, for automated threat reporting and response coordination, it can be integrated with current Command and Control (C2) systems.

A number of improvements are anticipated in subsequent work to fortify the system even more:

- Multi-Sensor Fusion: Using LiDAR, radar, and thermal imaging sensors will increase the accuracy of detection in situations involving extreme weather or camouflage.
- Aerial Surveillance:integration with drones or unmanned aerial vehicles (UAVs) for dynamic perimeter monitoring.
- Edge-Cloud Collaboration**:** Implementing federated learning will allow multiple distributed nodes to learn collaboratively without centralized data storage, improving security and scalability.
- Predictive Analytics: AI-based behavior prediction models could anticipate intruder movement patterns, enabling preemptive defense actions.

- Adaptive Threat Classification: For broader deployment, the dataset will be expanded to encompass a greater variety of military assets and threat categories.

Furthermore, future iterations could integrate hardware accelerators such as NVIDIA Jetson Nano or Coral TPU for faster inference, reducing delay to sub-second levels. Advanced encryption techniques can be incorporated to secure communication between field devices and the central monitoring unit. The system can also evolve toward a fully autonomous surveillance grid where multiple nodes communicate collaboratively, creating an intelligent and self-learning defense network.

In conclusion, the suggested AI-driven intrusion detection and target acquisition framework shows how artificial intelligence can transform conventional defense surveillance systems by improving national security infrastructure, lowering the need for manpower, and delivering real-time intelligence.

REFERENCES

[1] J. Redmon, et al., "You Only Look Once: Unified, Real-Time Object Detection," *IEEE CVPR*, 2016.

[2] A. Bochkovskiy, C. Y. Wang, H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv:2004.10934*, 2020.

[3] A. G. Howard, et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017.

[4] N. Dalal, B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE CVPR*, 2005.

[5] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *IEEE CVPR*, 2001.

[6] R. Girshick, "Fast R-CNN," *IEEE ICCV*, 2015.

[7] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection," *IEEE TPAMI*, 2017.

[8] W. Liu, et al., "SSD: Single Shot MultiBox Detector," *ECCV*, 2016.

[9] M. Tan, Q. Le, "EfficientDet: Scalable and Efficient Object Detection," *IEEE CVPR*, 2020.

[10] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767*, 2018.

[11] K. Suresh, et al., "AI-based UAV Surveillance for Border Intrusion Detection," *IEEE Access*, 2022.

[12] R. Ramesh, et al., "Sensor-Assisted Motion Detection for Secure Military Zones," *IJCSIT*, 2021.

[13] A. Sharma, D. Gupta, "Hybrid Image and Sensor-Based Intrusion Detection System," *IJRET*, 2020.

[14] V. Kumar, et al., "Real-Time Vehicle Detection on Raspberry Pi using YOLOv5," *IEEE ICMLA*, 2023.

[15] P. Singh, et al., "Autonomous Military Surveillance Using Deep Learning and Drones," *IEEE Sensors Journal*, 2022.