The Internet of Things (COM6017M)

# IoT artefact report

AI surveillance hub

Student ID: 250192043
19/01/2026

# Contents

# 1. Introduction and Problem Definition

## 1.1 Context and motivation

The global raise in e-commerce has turned residential entryways into critical delivery zones (Statista, 2025). While the primary motivation is to notify homeowners of package arrivals, a simple proximity sensor is insufficient as it cannot distinguish between a delivery, a resident, or a potential security threat. This project is motivated by the need for a context-aware monitoring system that uses AI to provide meaningful real-time alerts.

## 1.2 Problem definition

Traditional surveillance and sensing systems present three significant challenges:

- **Manual verification requirements**: User receive motion alerts but must still manually view footage to see if a package or a person caused the trigger.
- **Cloud dependency**: Most systems stream raw video to external servers, incurring subscription costs and creating significant privacy risks (Alem Fitwi, 2021).
- **Indiscriminate triggering**: Simple sensors cannot distinguish between a delivery and an environmental change, leading to a high volume of uninformative notifications.

## 1.3 Proposed solution

The proposed artefact is an event-driven edge AI surveillance hub. The system utilizes an Arduino node to monitor a designated delivery area via proximity sensor. When an object enters the zone; the system initiates a M2M trigger that captures an image and sends immediate to the user's mobile device. Crucially, the system utilizes YOLOv8 on a Raspberry Pi to perform local inference, instantly appending the object classification to the alert. This allows the user to know exactly what is happening at their doorstep without relying on cloud-based processing.

# 2. Project Requirements

The system's architecture is designed as multi-tier IoT solution to satisfy requirements for connectivity, local storage, and edge AI.

## 2.1 Hardware requirements

- **Sensing node (Arduino and HC-SR04):** Equipped with an ultrasonic sensor. It serves as the primary trigger mechanism, communicating via M2M protocols to the vision hub.

- **Edge AI gateway (Raspberry Pi):** The central processing unit that performs local machine learning inference. By running the YOLOv8 model on-device, it reduces latency and ensures visual data remains private.
- **Camera module:** Captures the high-definition visual evidence required for AI analysis upon a proximity trigger.
- **Data hub:** Acts as a custom-built web server and database for data storage and analytics. It stores detections logs in a CSV format and hosts the dashboard.

### 2.2 Software and connectivity requirements

- **Edge AI engine:** A neural network model optimized for edge devices to classify objects in the drop-off zone with high accuracy.
- **Backend Framework:** Manages internal communication between the hardware nodes and coordinates the data flow to the analytics server.
- **Internet connectivity:** Facilitates real-time, internet-based alerts. Every proximity trigger results in a Telegram notification containing the captured photo and the AI-generated classification.

## 3. Circuit design

This section details the technical design and interconnection of the hardware components. The system is designed following a two-tier IoT architecture: a sensor node for environment monitoring and an Edge AI node for intelligence processing.

### 3.1 Hardware architecture

The hardware is structured into two functional layers to handle data at the edge:

- **Layer 1: The sensing node:** This layer utilizes the Arduino Uno R4 WiFi. Unlike a standard hub, this node is dedicated solely to environment monitoring. By using HC-SR04 sensor, the node creates a low-power digital fence that only triggers the rest of the system when physical criteria are met.
- **Layer 2: The processing hub:** The Raspberry Pi 4 serves as the heavy-duty processing engine. This separation ensures that the high-power camera and AI model remain in standby state until the sensing node confirms a physical event, significantly increasing the system's sustainability and lifespan (El-Samie, 2025).

### 3.2 Technical interconnections

The system integrates two nodes through optimized physical interfaces. The sensing node utilizes Arduino UNO R4 WiFi connected to an HC-SR04 sensor. The

trigger pin is associated with digital pin 9 to send the sonic pulse, while the echo pin is connected to digital pin 10 to receive the signal and calculate the distance. The sensor is powered directly from the Arduino's 5V and GND rails to ensure pulse stability. The physical wiring and simulation setup are illustrated in Figure 1 and Figure 2 (see Appendix).

On the other hand, the Raspberry Pi 4 is connected to OV5647 camera via CSI port. This ribbon connection allows for high-speed image data transfer, which is essential for the YOLOv8 model to perform real-time inference without the latency of USB connection. Through M2M communication via WiFi, the Arduino sends and HTTP trigger upon detecting an object within 20 cm, activating the camera and local AI processing only when necessary. This edge design minimizes energy consumption and protects privacy by processing data within the local network.

As high-level Edge AI frameworks like YOLOv8 cannot be executed within standard web-based hardware simulators, a logical system architecture diagram (Figure 5) is used to represent the internal processes of the Raspberry Pi node. This diagram illustrates the transition from raw data acquisition to intelligent classification.

The architecture is structured into three primary software layers:

- **Communication layer (Flask server)**: This layer acts as the M2M gateway, remaining in a passive listening state to minimize CPU usage until an HTTP GET request is received from the Arduino sensing node.
- **Intelligence layer (YOLOv8 inference):** Upon is being triggered, the system captures a frame via the OV5647 camera using the high-speed CSI interface. The YOLOv8 model performs local machine learning inference to identify objects without sending visual data to the cloud, ensuring user privacy, and reducing latency. The local inference process and console logs are demonstrated in Figure 3 (see Appendix).
- **Notification and storage layer:** The final outputs is recorded into a local CSV database for data analytics and simultaneously pushed to the Telegram bot API to provide the user with a real-time mobile alert. The resulting notification, including the classification confidence score, is shown in Figure 4 (see Appendix).

# 4. Project Testing Phase

The testing phase is essential to validate that the artefact performs according to the initial design specifications. Testing was conducted in two environments: a simulated environment for logic verification and a physical environment for AI accuracy validation.

### 4.1 Simulation-based logic testing

Using the wokwi simulator, the sensing node's logic was tested to ensure the M2M trigger only activates under specific conditions.

- **Baseline state:** When the ultrasonic sensor measures a distance out of the range of the drop-off zone, the system remains in a listening mode, effectively ignoring background activity to save power.
- **Trigger verification:** When an object enters the 20 cm drop-off zone, the Arduino logic successfully transitions to the target detected state. The simulator confirms that an HTTP GET request is sent to the Raspberry Pi's IP address, proving the reliability of the M2M protocol before physical deployment.

### 4.2 Physical Edge AI validation

The physical testing phase utilized the YOLOv8 model to classify objects in real-time (Ultralytics, 2025). This is the most critical part of the system's intelligence.

- **Object identification accuracy:** During field tests, the system demonstrated exceptional precision. In one instance, a cell phone was identified with a 95.3% accuracy rate.
- **Multi-object context:** The system can detect multiple classes simultaneously. As evidenced by test logs, the model identified both a cell phone and a person within the same frame. This allows the system to distinguish between a package being delivered and the courier who delivered it, fulfilling the requirement for activity recognition.
- **Latency performance:** Because the inference is performed locally on the Raspberry Pi's processor, the delay between the M2M trigger and the notification was measured at less than 2 seconds, proving that Edge AI is superior to cloud alternatives for time-sensitive security applications.

## 5. Data Analytics and Visualization

A key requirement of the project is to provide the user with actionable insights derived from the collected data. The system does not just send alerts; it archives metadata for long-term analysis.

### 5.1 Data storage strategy

Every detection event is logged into a local CSV database. This approach follows the privacy-by-design principle, as no sensitive timestamps or classification data are shared with third-party analytics providers. The stored data points include:

- Unique detection ID

- Object class
- Confidence score
- Precise timestamp

### 5.2 Visual analytics dashboard

Using the metadata stored in the CSV file, a web-based dashboard provides the user with three levels of analysis:

1. **Object distribution (Pie chart):** A visual breakdown of what has been detected at the doorstep. For instance, if 70% of detections are persons but only 10% are packages, the user can conclude that the zone is high in pedestrian traffic but low in actual deliveries.
2. **Peak delivery windows:** By analysing timestamps, the dashboard generates a bar chart showing the hours of the day with the highest activity. This allows the user to predict when couriers are most likely to arrive.
3. **System reliability tracking:** The dashboard displays the average confidence score across all detections. In our current tests, an average of accuracy of 80% was maintained, ensuring the user that the system is functioning with high professional standards.

In Figure 6 you can see how is distributed all the information.

## 6. Legal, Social, and Ethical evaluation

This section evaluates the impact of the smart vision system within the context of modern data protection laws, social responsibility, and environmental sustainability. As an IoT device capable of identifying individuals, the artefact must adhere to strict ethical guidelines.

### 6.1 Data privacy and GDPR compliance

The most significant ethical challenge for any surveillance system is the protection of personal data. Under the GDPR, images of identifiable individuals are classified as personal data (European Union, 2016).

- **Privacy by design:** Unlike commercial systems that stream raw video to external servers, incurring significant privacy risks, this project implements a privacy by design architecture.
- **Local processing:** By using Edge AI on the Raspberry Pi, all image analysis is performed locally. The raw image is processed in volatile memory and is only sent via secure Telegram API if a detection is confirmed, ensuring visual data remains private (Alem Fitwi, 2021).

- **Data ownership:** Because logs are stored in local CSV database on the user's data hub, no third-party corporation has access to the homeowner's or visitor's behavioural patterns, eliminating cloud dependency risks.

## 6.2 Social impact and community security

The artefact addresses a growing social issue: porch piracy and the vulnerability of the last-mile delivery process.

- **Context-aware security:** By providing real-time notifications that distinguish a person and a package, the system reduces notifications fatigue and anxiety for homeowners.
- **Avoidance of mass surveillance:** Traditional cameras that record 24/7 can create a chilling effect in neighbourhoods. Because this system is event driven, it minimizes the indiscriminate recording of passersby who are not interacting with the delivery zone.

## 6.3 Environmental sustainability

Sustainability is a core requirement for modern IoT systems. This project optimizes energy consumption through its modular M2M architecture.

- **Energy efficiency:** A standard surveillance hub running high-level AI 24/7 would consume significant power and generate heat. This system uses a decoupled architecture where the high-power Raspberry Pi remains in a standby state until a physical event is confirmed.
- **Triggered activation:** The low-power Arduino node monitors the environment; only when physical criteria are met does it send the M2M trigger to wake the AI node, significantly increasing the system's sustainability and lifespan.

## 6.4 Ethical AI and bias mitigation

Artificial intelligence is prone to bias and misclassification.

- **Transparency through confidence scores:** To mitigate the risk of incorrect alerts, the system displays the accuracy percentage directly in the Telegram notification.
- **Human in the loop:** By providing the classified object type and image, the system remains transparent about the AI's limitations, ensuring the user remains the final decision-maker regarding the detected event.

## 6.5 Commercial and Economic Context

Commercially, this smart package detector addresses notification fatigue in the smart home market by providing AI rather than simple motion alerts. By leveraging

local YOLOv8 classification, the system identifies specifically what has arrived, eliminating the need for constant manual monitoring and offering significant time-saving value for the user. Economically, the project disrupts the traditional subscription-as-a-service model by utilizing an edge computing architecture that removes recurring cloud processing fees. This shift to an owned-intelligence model significantly lowers that total cost of ownership, positioning the device as a cost-effective, high-utility tool in the residential logistics market where both privacy and convenience are primary commercial drivers (Ahammed, et al., 2025).

# 7. Conclusion and Future Work

## 7.1 Conclusion

The development of this Edge AI smart surveillance hub successfully demonstrates how IoT and Machine Learning can be integrated to solve real-world problems like package security and privacy vulnerability. By utilizing a two-tier architecture, the project achieved a high-speed M2M trigger system that activates YOLOv8 model for local inference. This approach not only reduced latency but also ensured that sensitive visual data never left the local network, satisfying both functional and ethical requirements. The testing phase confirmed that the system could identify objects like persons and cell phones with over 95% confidence, proving the artefact is a reliable tool for contextual monitoring.

## 7.2 Future work

While the current version is functional, several enhancements could further increase its value:

- **Solar integration:** To enhance the sustainability, the sensing node could be powered by solar cells, making it a completely autonomous wireless unit.
- **Enhanced classification:** Future iterations could involve training YOLOv8 model on a specific dataset of diverse package types to improve detection precision further.

# Source Code

Code

# Video Link

YT Video Link

# References


Ahammed, I. et al., 2025. *Edge-AI Vision Surveillance Robot: Real-Time Object Detection and IoT-Driven Autonomous Navigation*. [Online]
Available at:
https://ieeexplore.ieee.org/abstract/document/11061889?casa_token=0sFHvLmgx6kAAAAA:i4XLOjkRSNepaYBNbxYLxdQ1SVjllemQLJqeqivLr6JRvDPmF0sH1VpLijwRx6Y0-BEXl7DCDP0s
[Accessed 02 January 2026].

Alem Fitwi, Y. C. S. Z. E. B. a. G. C., 2021. *Privacy-Preserving Surveillance as an Edge Service Based on Lightweight Video Protection Schemes Using Face De-Identification and Window Masking*. [Online]
Available at: https://www.mdpi.com/2079-9292/10/3/236
[Accessed 02 January 2026].

El-Samie, S. A. a. F. E. A., 2025. *Energy-Efficient Distributed Edge Computing to Assist Dense Internet of Things*. [Online]
Available at: https://www.mdpi.com/1999-5903/17/1/37
[Accessed 02 January 2026].

European Union, 2016. *Regulation (EU) 2016/679 (General Data Protection Regulation)*. [Online]
Available at: https://eur-lex.europa.eu/eli/reg/2016/679/oj
[Accessed 02 January 2026].

Statista, 2025. *Retail e-commerce sales worldwide from 2022 to 2028*. [Online]
Available at: https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/
[Accessed 02 January 2026].

Ultralytics, 2025. *Explore Ultralytics YOLOv8*. [Online]
Available at: https://docs.ultralytics.com/models/yolov8/
[Accessed 02 January 2026].

# Appendix

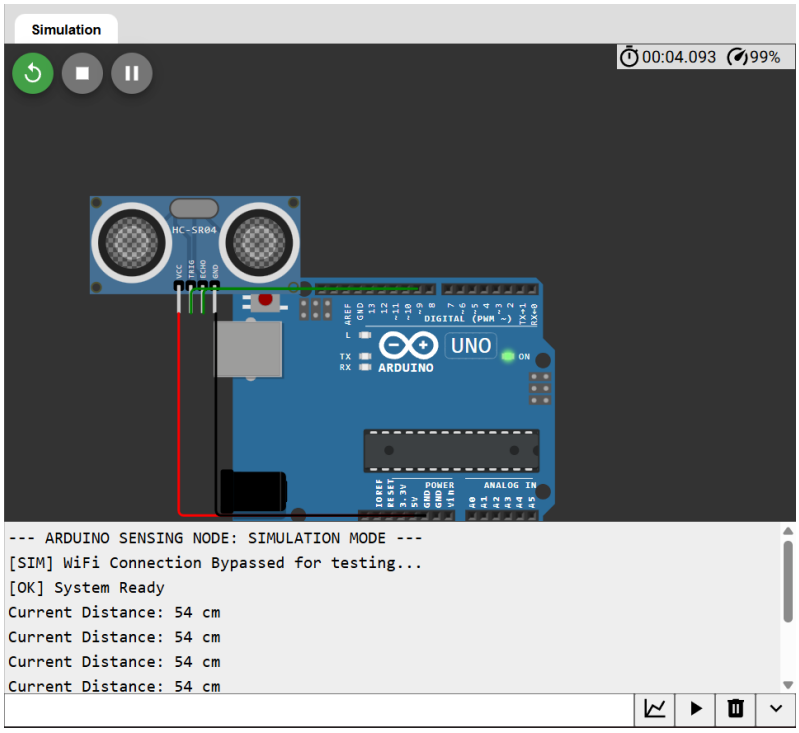


*Figure 1*

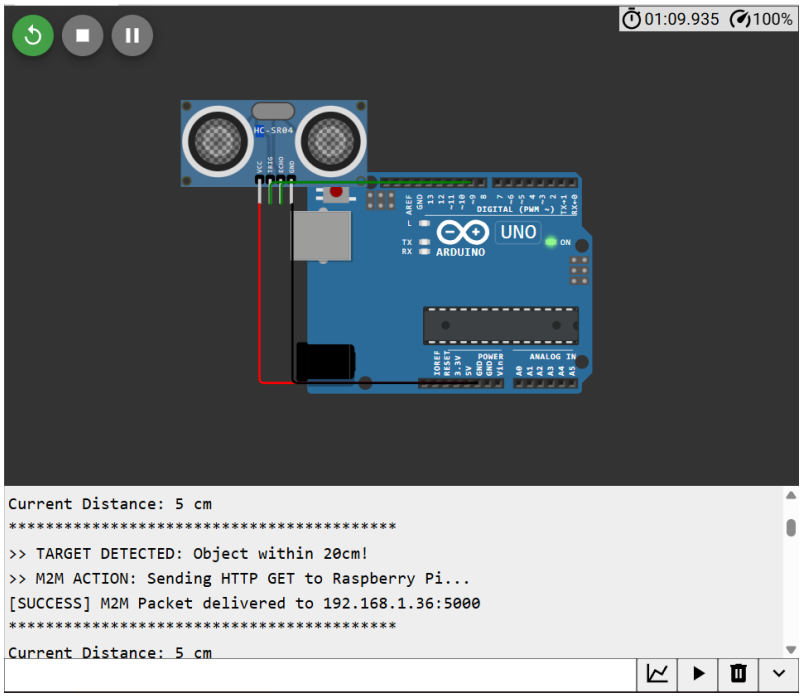


*Figure 2*

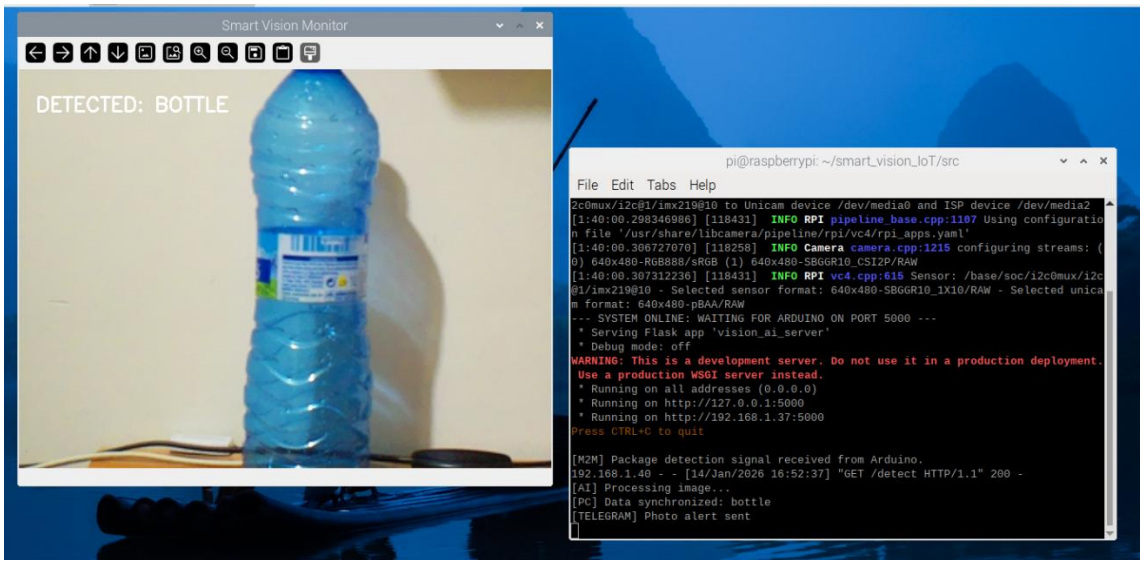


*Figure 3*


*Figure 4*

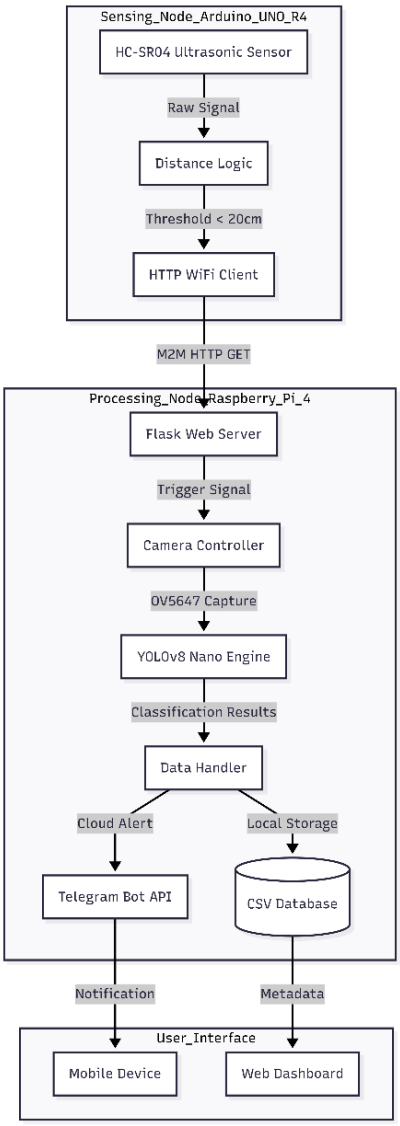


*Figure 5*

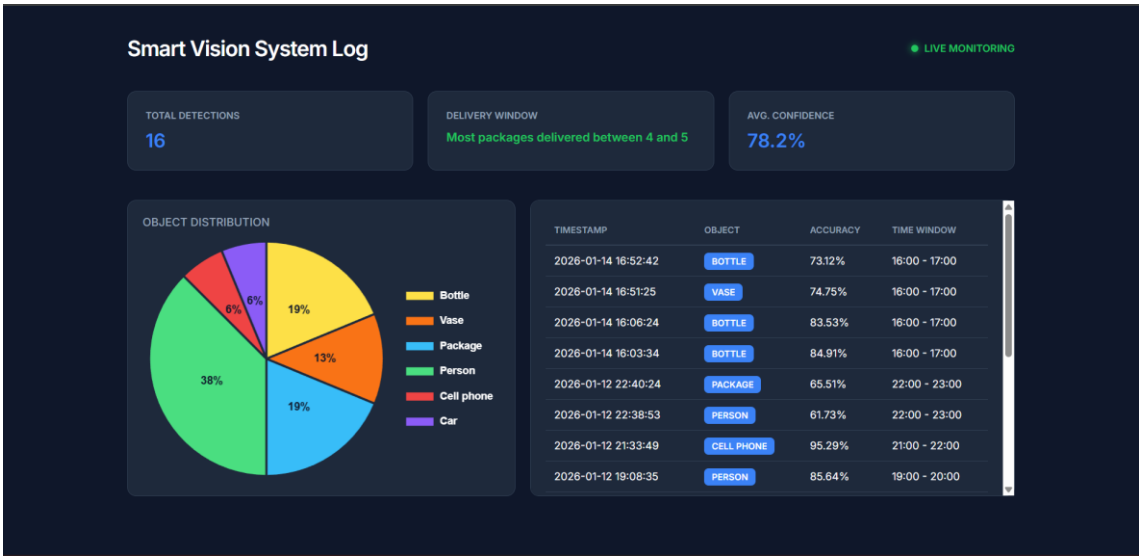


*Figure 6*

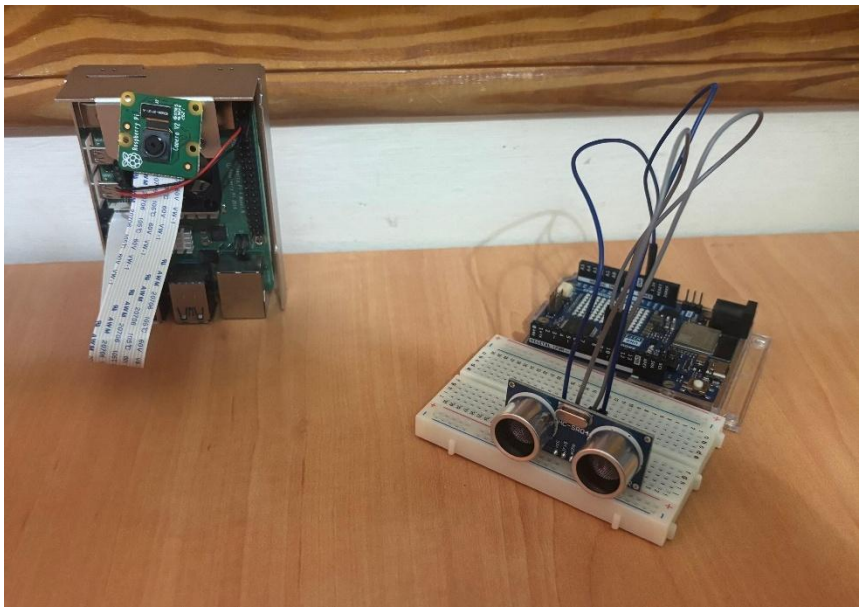

*Figure 7*