



ARTICLE

Design and Implementation of an IoT-Based Irrigation System with Surveillance Camera

Moina-yndi Ibrahim¹, Warda Soulamana² and Zhenjie Zhao^{3,*}

¹School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, China

²Department of MIAGE-ACSI, Institute of Digital Sciences, Management Cognition, Nancy, France

³College of Artificial Intelligence, Nankai University, Tianjin, China

*Corresponding Author: Zhenjie Zhao. Email: zzhaoao@nankai.edu.cn

Received: 07 January 2026; Accepted: 03 March 2026; Published: 06 April 2026

ABSTRACT: The increasing demand for efficient agricultural water management, exacerbated by population growth and climate change, has spurred the development of Internet of Things (IoT)-based smart irrigation systems as an alternative to inefficient traditional methods that waste water and reduce crop productivity. This paper presents a low-cost IoT irrigation platform that integrates real-time camera surveillance for enhanced farm monitoring and precise water management. The system employs an Espressif System (ESP8266) microcontroller to automate irrigation control based on soil moisture readings and an Espressif System ESP32 Camera Module (ESP32-CAM) to provide live imaging of the agricultural plot, with both devices connected to cloud services via Wireless Fidelity (Wi-Fi). System architecture was modeled using Unified Modeling Language (UML) diagrams, including use-case and sequence diagrams, to ensure clarity and robust design. The IoT framework follows a four-layer model encompassing field perception, network transmission, cloud-based processing via the Blynk platform, and a mobile/web application layer for remote user interaction. A smartphone dashboard allows farmers to visualize real-time sensor data, monitor live video feeds, and control irrigation components remotely. Experimental evaluation over a 14-day outdoor deployment involving 47 automated irrigation events demonstrated a mean pump activation response time of 1.2 ± 0.21 s, 100% pump activation accuracy across 30 controlled trials, and system uptime of 96.4%. A baseline comparison showed that camera integration reduced fault detection latency approximately six-fold relative to a sensor-only configuration (15 vs. 90 s). The key scientific contributions include: (i) the integration of a ground-level ESP32-CAM into an automated threshold-triggered irrigation loop with empirical fault-detection validation; (ii) quantitative benchmarking against a sensor-only baseline; and (iii) formal UML-based system specification applied to an IoT irrigation context. The proposed platform offers a practical, scalable smart irrigation solution unifying automated control, cloud processing, and live visual monitoring within a single framework.

KEYWORDS: IoT irrigation system; smart farming; ESP32-CAM; ESP8266; Blynk cloud; embedded sensor

1 Introduction

Global water scarcity is a critical challenge for agriculture and food security. In fact, agriculture accounts for roughly 70% of freshwater withdrawals worldwide [1], and climatic changes combined with population growth continue to intensify demand for food and water [2,3]. Consequently, optimizing irrigation is essential; “smart irrigation” systems leveraging real-time data have emerged as a promising solution to cut water waste and improve yields [4,5]. For example, Saputri et al. note that the scarcity of water resources exacerbated by climate change poses a major challenge for sustainable agriculture, and show that

Internet of Things (IoT) monitoring and control via a Blynk dashboard can significantly improve water-use efficiency [6]. Similarly, Liu et al. observe that agriculture is “a major contributor to water consumption and wastage” and emphasize that IoT-enabled irrigation can optimize water use through real-time sensor data [5]. Reviews also highlight that enhancing irrigation efficiency directly supports the United Nations Sustainable Development Goals (e.g., Sustainable Development Goal 6 on clean water), by reducing waste and conserving resources [7,8].

IoT technology—combining low-cost microcontrollers, sensors, wireless connectivity, and cloud platforms—is increasingly applied to address these issues. Recent studies demonstrate that IoT-based irrigation systems can dramatically boost efficiency: e.g., Raj and Prahadeeswaran report that IoT irrigation platforms enhance crop productivity by ~25% [9], and Saha et al. show a fuzzy-logic-controlled smart irrigation prototype saved ~61% of water compared to conventional methods [10]. Such systems typically integrate soil moisture, temperature, and humidity sensors with IoT controllers to automate watering decisions. For instance, Saputri et al. developed a Node Microcontroller Unit (NodeMCU) ESP8266 controller with soil-moisture sensors, temperature sensors, and the Blynk cloud platform for real-time monitoring and precise valve control [6]. Morchid et al. likewise built an ESP32-based IoT system that uses ThingSpeak (a cloud data service) and fuzzy logic to dynamically adjust irrigation schedules [4]. These architectures typically include three layers: field devices (microcontrollers like Espressif System ESP32 (ESP32) or Espressif System ESP8266 (ESP8266) with sensors and actuators), a cloud or edge layer for data processing and storage, and a user interface for remote monitoring and control (e.g., mobile app dashboard) [11].

Despite many IoT irrigation prototypes, key gaps remain. Most existing works focus on sensor data and automated watering, but often lack integrated visual monitoring. Incorporating a real-time camera feed can greatly enhance situational awareness (for example, to inspect plant health or detect leaks remotely), yet, while recent works have demonstrated that integrating visual monitoring with IoT sensor data yields richer situational awareness and improved anomaly detection in agricultural systems [12,13], few studies have extended this integration to low-cost ground-level camera modules within automated irrigation control loops. Our work contributes such a comparison, directly quantifying the operational advantage of visual monitoring in terms of fault detection latency, while also emphasizing rigorous system design via Unified Modeling Language (UML) modeling to ensure clarity of architecture. In summary, this paper presents a low-cost IoT irrigation system using ESP8266 and Espressif Systems ESP32 Camera Module (ESP32-CAM), soil moisture sensors (SMS), and the Blynk cloud platform. A mobile dashboard and live video stream provide remote monitoring and control of irrigation. We also use UML diagrams to model the system's components and interactions. Together, these contributions demonstrate an accessible smart irrigation solution that improves water efficiency while providing real-time visual feedback, addressing both technological and usability challenges in precision agriculture.

2 Literature Review

The application of IoT technologies in smart irrigation has received considerable attention over the past decade, driven by the urgent need to optimize water consumption in agriculture, which accounts for approximately 70% of global freshwater withdrawals. Traditional irrigation methods are often inefficient and labor-intensive, motivating the development of automated systems capable of real-time monitoring and control. IoT-enabled irrigation platforms combine sensors, microcontrollers, actuators, and cloud services to enable precise water delivery based on environmental conditions.

For example, Saputri et al. demonstrated that an ESP8266-based NodeMCU integrated with soil-moisture and temperature sensors, controlled via the Blynk cloud dashboard, could significantly improve water-use efficiency [6]. Likewise, Liu et al. reported that IoT systems utilizing fuzzy logic and real-time routing algorithms optimize irrigation schedules and reduce water waste compared with conventional approaches [5]. Large-scale evidence is supplied by Vinod Kumar et al., who achieved 15% higher sweet-corn yield under plastic mulching with an IoT-scheduled drip system [14], and by Mishra et al., who cut water use by 23% in Punjab wheat through an autonomous framework driven by low-cost sensors [15]. Reviews in this field consistently highlight the ability of IoT irrigation to support sustainable agriculture and achieve water savings while maintaining or increasing crop yields [16–19].

Numerous architectural approaches have been proposed for IoT-based irrigation. Typically, these include a perception layer with sensors and actuators, a network layer for data transmission, a processing layer for control logic and storage, and an application layer for user interaction via mobile or web dashboards. Raj & Prahadeeswaran reported that integrating these layers using ESP-based controllers and cloud platforms enhances irrigation performance, allowing adaptive control based on real-time soil-moisture and climate data [9]. Saha et al. combined satellite imagery with ground sensors to map agricultural lands and implement precision irrigation, achieving up to 61% water savings with fuzzy-logic-controlled systems [10].

In addition, Baseca et al. developed a Long-Range Wide Area Network (LoRaWAN)-based edge-computing architecture that embeds a Long Short-Term Memory (LSTM) neural network to predict irrigation need across multi-kilometre maize plots while keeping the duty cycle below 1% [2]. Similar hybrid communication architectures combining long-range LoRaWAN links with short-range wireless sensor networks have also been explored to improve scalability and reliability in smart irrigation deployments, whereas Rehan et al. showed that a dual-channel Long Range (LoRa) design raises packet-delivery ratio to 98% in cotton fields [20]. Morchid et al. built an ESP32/ThingSpeak platform with fuzzy inference for dynamic irrigation scheduling, highlighting the benefits of integrating real-time sensor data with cloud-based dashboards [4]. These studies indicate that diverse combinations of hardware, connectivity, and control algorithms are feasible, depending on deployment scale and available resources.

Recent advances in Artificial Intelligence (AI)-assisted and predictive irrigation represent a rapidly growing research direction. OGUZTURK [16] provides a systematic review of AI-driven irrigation systems, demonstrating that machine learning and deep learning approaches can achieve 15%–30% additional water savings compared to threshold-based systems by anticipating soil moisture dynamics. Miller et al. [21] present a systematic review of smart sensing technologies for IoT-AI integration in agriculture, identifying LSTM networks, support vector machines, and convolutional neural networks as the most widely deployed architectures for predictive irrigation scheduling. Baseca et al. [2] demonstrate a practical edge-AI implementation using LoRaWAN and LSTM for predictive deficit irrigation—achieving controlled-deficit irrigation with minimal communication overhead. Singh et al. [18] survey intelligent controller development for smart irrigation systems, categorizing approaches from simple threshold-based rules to model predictive control and reinforcement learning. Complementing these approaches, fuzzy logic-based control has also emerged as a lightweight yet effective alternative: Liu et al. [5] present a fuzzy rule-based IoT irrigation system demonstrating intelligent pump control without the overhead of training data. Jiménez et al. [22] propose a multi-agent IoT precision irrigation approach integrating sensor fusion and intelligent scheduling at both farm and district scales. The present work focuses on system integration and empirical validation of a low-cost threshold-based platform, with AI-enhanced control identified as a promising direction for future work.

Microcontroller choice is pivotal in these designs. ESP8266 (NodeMCU) and ESP32 boards are favored for their low cost, built-in Wireless Fidelity (Wi-Fi), and multiple GPIO pins, allowing seamless integration with sensors and actuators. Morchid et al. and Saputri et al. exemplify the use of ESP boards for soil-moisture monitoring and pump control, demonstrating reliable cloud connectivity via Blynk or ThingSpeak for remote monitoring and actuation [4,6]. Bonfim-Silva et al. underline that even low-cost capacitive moisture probes can deliver laboratory-grade soil-water retention curves when coupled to NodeMCU dataloggers [23]. Additional evidence is provided by Sarker et al., Azeem et al., and Karthik et al., who all deployed ESP32 boards to automate irrigation for coriander, tomato, and chilli crops, respectively [11,24,25]. These studies collectively show that ESP-based architectures are cost-effective, scalable, and suitable for integrating additional modules, such as cameras, into IoT systems.

Connectivity considerations are equally critical. While Wi-Fi is commonly used for small-scale deployments, large or remote farms benefit from low-power wide-area networks (LPWAN) such as LoRaWAN and Narrowband-IoT (NB-IoT), which offer extended range and reduced energy consumption. Baseca et al. demonstrated LoRaWAN-based data collection over multi-kilometre distances with minimal energy requirements, enabling real-time irrigation control even in regions with limited infrastructure [2]. Hu et al. explored NB-IoT applications in remote sensor networks, showing that cellular IoT can complement or replace traditional Wi-Fi where coverage is sparse [26]. However, Wi-Fi remains essential for high-bandwidth applications, such as integrating video surveillance into irrigation systems, as LoRa and NB-IoT cannot handle continuous image streaming.

The integration of cloud platforms and dashboards has emerged as a standard practice to facilitate real-time monitoring, control, and data analytics. Blynk, ThingSpeak, and similar services provide straightforward APIs for sending sensor data, visualizing trends, and remotely actuating devices. Parma et al. implemented an ESP32-controlled plant-watering system using Blynk, achieving real-time feedback and reliable operation [27]. ThingSpeak-based dashboards, as demonstrated by Morchid et al., allow visualization of soil moisture, pump status, and environmental parameters, providing actionable insights for users [4]. Ngoma et al. extended this concept to small-holder tomato farms in Tanzania, where a ThingSpeak mobile app reduced irrigation labour by 40% [28]. Studies note that these platforms reduce development time, enhance usability, and improve system reliability by separating control logic from visualization, which also facilitates future scalability.

Recent literature emphasizes the added value of visual monitoring in smart agriculture. Rajak et al. indicate that camera integration helps detect plant stress, pests, and equipment failures that sensors alone cannot identify [29]. Ground-level cameras, particularly ESP32-CAM modules, are low-cost and easily integrated with existing IoT networks, providing on-demand snapshots or continuous feeds. Although fewer studies have explicitly combined video surveillance with irrigation control, prior work on greenhouse monitoring and pest detection confirms the benefits of visual feedback, complementing sensor data to improve situational awareness and intervention speed.

Sensor selection and control algorithms are foundational for system performance. Soil-moisture sensors are ubiquitous, while temperature, humidity, and light sensors provide supplementary data for refining irrigation schedules. Control strategies range from simple threshold-based rules to fuzzy logic and machine learning. Fuzzy controllers, for instance, adapt watering durations and intervals according to variable conditions, reducing over- or under-watering. Bushnag et al. achieved 18% water savings in Saudi Arabian date palms using a fuzzy-PID hybrid [30], whereas Liana et al. increased Indonesian tobacco yield by 12% with a similar approach [31]. Machine-learning approaches, such as LSTM-based predictions [2] or K-nearest-neighbour [32], further improve scheduling by analysing historical trends and environmental patterns.

Nonetheless, simplicity and reliability often favour threshold or fuzzy-based controllers for small-scale farms, as these approaches are robust, easy to implement, and compatible with low-cost hardware.

UML modelling is an increasingly adopted practice to ensure clarity in IoT system design. Deployment, class, and sequence diagrams provide a formal representation of hardware, software, and interactions, facilitating maintenance, debugging, and extension. Baseca et al. used component and deployment diagrams to describe a multi-layer LoRaWAN architecture, highlighting interconnections between edge controllers, sensors, and cloud servers [2]. Similarly, Reggio underscore the importance of UML in modelling IoT interactions to ensure that all functional requirements are met and system behaviour is predictable [33]. In the present study, UML diagrams model the relationships among the ESP8266, ESP32-CAM, sensors, relays, and cloud interfaces, ensuring that the irrigation system can be implemented reliably and systematically.

Table 1 presents a structured comparison of the most closely related IoT irrigation systems across seven key dimensions. As the table illustrates, the reviewed systems address different subsets of these dimensions: most employ threshold-based or fuzzy logic control with soil moisture sensing and cloud connectivity. The primary distinguishing feature of the proposed system is the integration of an ESP32-CAM module for visual cross-validation of sensor-driven irrigation decision. These observations motivate the design choices of the proposed system.

Table 1: Structured comparison of proposed system with closely related IoT irrigation systems.

| Study | MCU/Platform | Sensing Modalities | Visual Monitor | Control Strategy | Cloud Platform | Key Metric |
|--------------------|---------------------|--|---------------------|---|------------------|--|
| Saputri et al. [6] | ESP8266 | Soil moisture, Temp, Humidity | None | Threshold | Blynk/ThingSpeak | Improved efficiency |
| Morchid et al. [4] | ESP32 | Soil moisture, Temp, Humidity | None | Fuzzy logic | ThingSpeak | Dynamic scheduling |
| Parma et al. [27] | ESP32 | Temp, Humidity | None | Threshold (Temp $\geq 34^{\circ}\text{C}$ or Humidity $\leq 50\%$) | Blynk | Automated greenhouse control |
| Sarker et al. [11] | ESP32 | Rain, Temp, Humidity, Light, Soil moisture | None | Threshold + manual override (Blynk notifications) | Blynk | 1.11–1.67× higher tomato yield vs. traditional |
| Baseca et al. [2] | LoRa-WAN | Soil moisture | None | LSTM predictive | Edge + Cloud | PDR $\geq 98\%$, duty cycle $\leq 1\%$ |
| Saha et al. [10] | IoT gateway | Soil moisture | Satellite Imagery | Fuzzy logic | Custom | ~61% water savings |
| Proposed system | ESP8266 + ESP32-CAM | Soil moisture, Temp, Humidity | Live ESP32-CAM feed | Threshold + remote override | Blynk | 1.2 ± 0.21 s response; 6× faster fault detection vs. sensor-only |

Overall, the reviewed literature demonstrates that IoT-based irrigation systems can improve water efficiency, crop productivity, and farm monitoring. Systems leveraging ESP-based controllers, cloud platforms, and sensor networks provide flexible and cost-effective solutions, while LPWAN and cellular IoT extend coverage for larger farms. The integration of video surveillance adds an emerging layer of situational awareness, and UML modelling strengthens design rigor. Among the reviewed works, the combination of ground-level camera monitoring with sensor-based irrigation control and cloud dashboards within a single

low-cost platform is not widely reported. The present work contributes to this direction by integrating these elements into a practical prototype, with the aim of informing future developments in low-cost smart irrigation systems. The experimental contribution of this paper—specifically the quantitative demonstration of camera-assisted fault detection advantage over a sensor-only baseline.

3 Materials and Methods

3.1 Layered Architecture

The implementation adopts a four-layer IoT architecture (Fig. 1):

- **Perception Layer:**

This layer consists of Internet-connected devices whose embedded sensors and actuators continuously monitor the environment, gathering data that are forwarded to IoT gateways. Field-deployed equipment includes a humidity sensor, a water pump, and a surveillance camera.

- **Network Layer:**

The network layer provides the connectivity required to relay and preprocess the data generated by the sensing devices. It establishes links among heterogeneous devices, edge routers, and remote servers, and governs all upstream and downstream traffic. Communication options include Ethernet, cellular, LPWAN, Wi-Fi, Bluetooth, and Near Field Communication. A field-level router operates as an IoT gateway: it aggregates sensor payloads, performs local filtering and protocol translation, and forwards the refined data stream to the back-end server.

- **Data Processing Layer:**

The processing layer ingests, stores, and analyzes the data forwarded by the network tier. Edge nodes perform preliminary aggregation and advanced analytics; the refined data are subsequently uploaded to a cloud server for persistent storage and additional computation. Authorized end-users retrieve the curated datasets through a dedicated mobile application.

- **Application Layer:**

The application layer constitutes the user-facing interface that delivers domain-specific services. A mobile application operationalizes this layer, enabling farmers to query real-time agronomic data, receive alerts, and execute control commands.

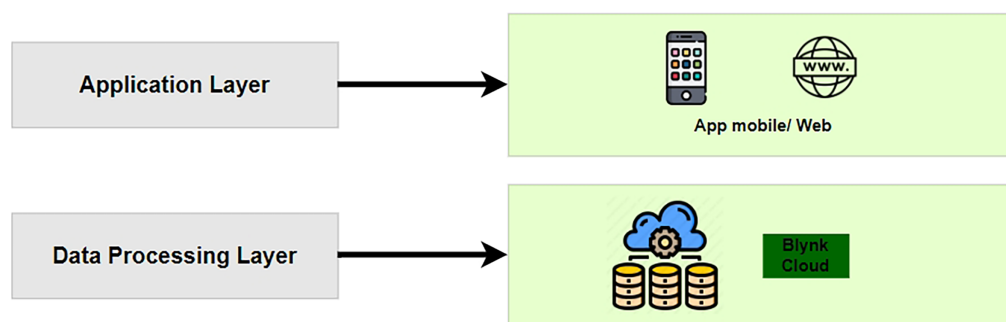


Figure 1: (Continued)

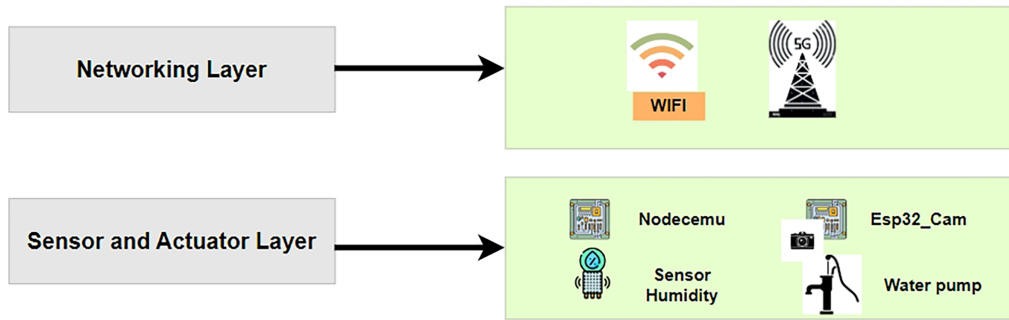


Figure 1: Layered IoT architecture.

3.2 Proposed Platform Architecture

The architecture traces the complete data path from the perception layer to the application layer, detailing the interconnection of hardware components (sensors and actuators), the mechanisms of data acquisition, visualization, and analysis, the storage locations, and the access procedures available to end-users as shown in Fig. 2.

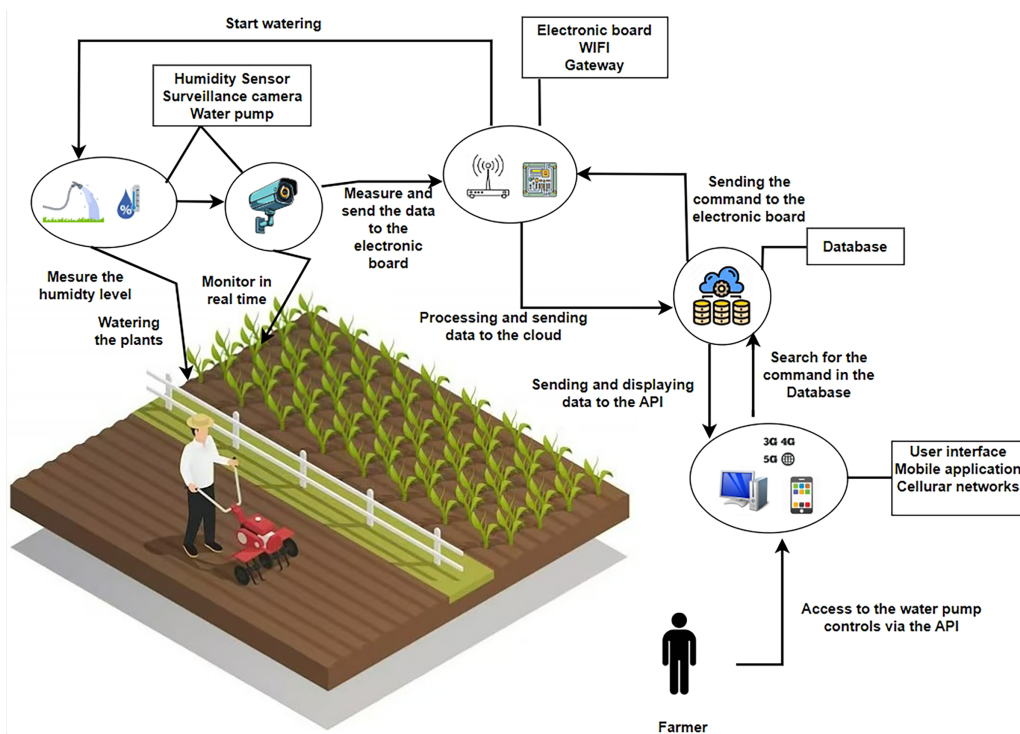


Figure 2: Proposed platform architecture.

3.3 UML Design Process

System development commenced with UML-based elicitation of functional requirements. The high-level use-case diagram (Fig. 3) casts the farmer and the IoT platform as primary actors; the principal use cases are “Authenticated” for access control, “View Data” for real-time sensor and camera monitoring, and “Control Pump” for remote irrigation activation. Sequence diagrams depict event ordering across system

components; Fig. 4 models pump activation, and Fig. 5 models live camera streaming. These UML artefacts enforce secure design: the pump-control sequence, for instance, requires successful authentication prior to relay actuation, precluding unauthorized irrigation.

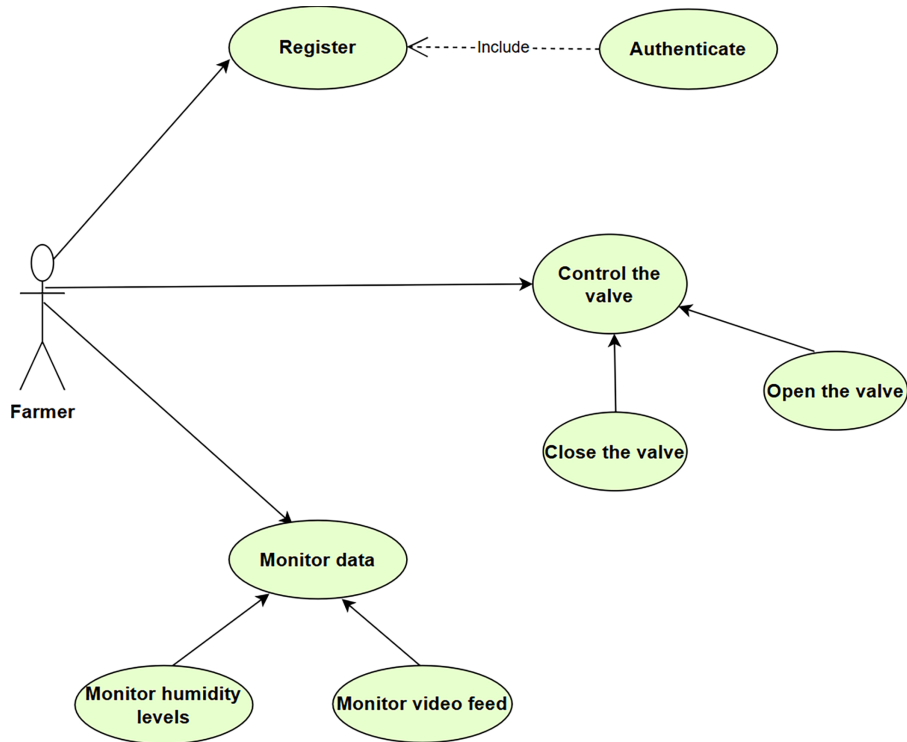


Figure 3: Use case diagram.

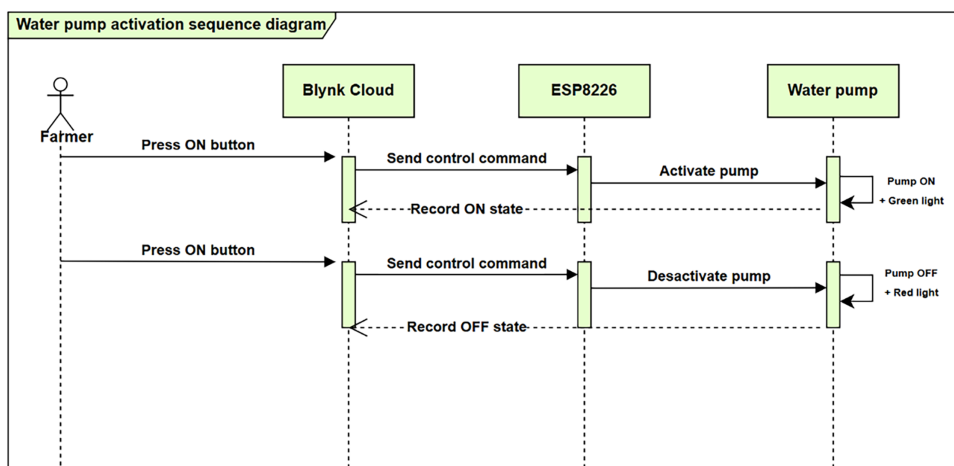


Figure 4: Water pump control sequence diagram.

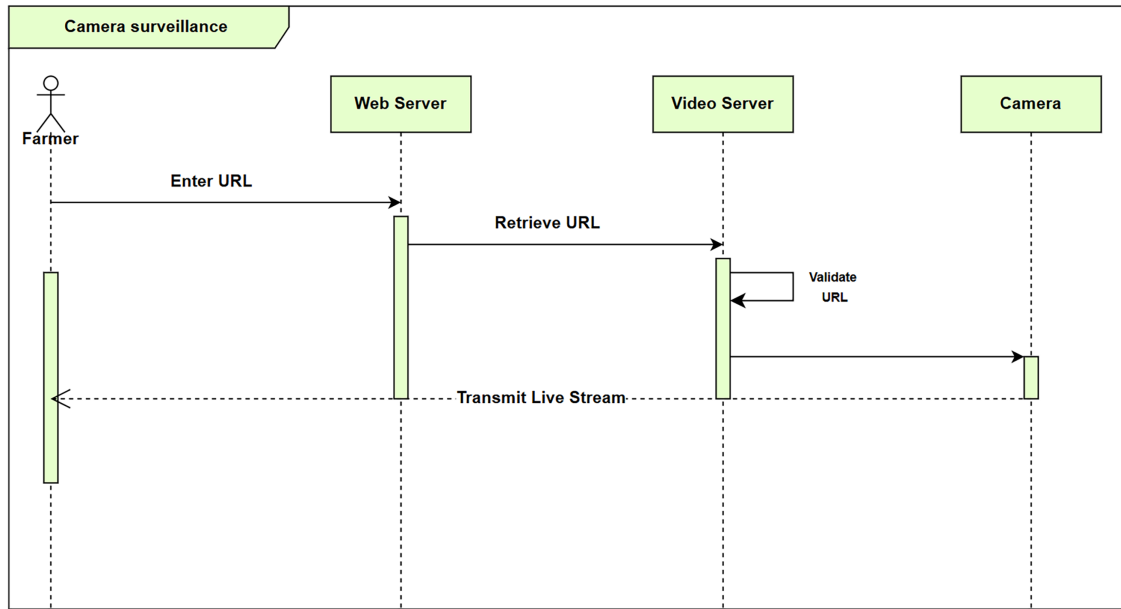


Figure 5: Video streaming sequence diagram.

4 Implementation of the Embedded Part

4.1 Hardware Component

Table 2 summarizes all hardware components used in the proposed system, including specifications, functional roles, and selection rationale.

Table 2: Hardware component specifications and selection rationale.

| Component | Model/Type | Key Specifications | Role in System |
|--------------------------------|-----------------------|---|---|
| ESP8266 (NodeMCU V1.0) | Tensilica L106 32-bit | 80 MHz CPU, 80 KB RAM, 4 MB flash, 802.11 b/g/n Wi-Fi, 11 GPIO, 1 ADC pin | Irrigation control firmware; soil moisture acquisition (ADC); relay actuation; Blynk HTTP telemetry |
| ESP32-CAM | ESP32-S + OV2640 | 240 MHz dual-core, 520 KB SRAM, OV2640 2 MP camera (640 × 480 @ 25 fps), 802.11 b/g/n Wi-Fi, microSD slot | Visual monitoring; MJPEG/snapshot-streaming; on-demand image capture |
| Resistive soil moisture sensor | FC-28/YL-69 | Operating voltage: 3.3–5 V DC; analog output 0–1023 ADC units; sensitivity adjustable via potentiometer | Soil water content Measurement; 2-point calibrated (Dry: ADC = 1023; Saturated: ADC = 380) |
| Water pump | 12 V DC submersible | 12 V DC, 120–240 mA, flow rate ~1.2 L/min, max head 40 cm | Water delivery to plants; activated via relay |

(Continued)

Table 2 (continued)

| Component | Model/Type | Key Specifications | Role in System |
|--------------|--|---|---|
| Relay module | Single-channel 5 V optocoupler relay | 5 V coil, 10 A/250 VAC switching; opto-isolated input; active-LOW trigger | Electrically isolates 220 V pump circuit from 3.3 V control circuitry; activated by ESP8266 GPIO |
| DHT11 Sensor | Digital tempera- ture/humidity | Temperature: 0–50°C ±2°C; Humidity: 20%–90% RH ±5%; single-wire digital protocol | Environmental monitoring (supplementary); data logged to Blynk dashboard |
| Power supply | 5 V/2 A USB adapter + 12 V adapter | 5 V/2 A for ESP8266 + ESP32-CAM; 12 V/1 A for pump | System power; isolation between low-voltage and pump circuits via relay |

4.2 Design Rationale for Dual-Controller Architecture

The choice to use separate ESP8266 and ESP32-CAM controllers rather than a single ESP32 board is motivated by four considerations:

- Task separation: the ESP8266 handles the time-critical irrigation control loop, while the ESP32-CAM handles computationally demanding MJPEG camera streaming. Sharing a single core for both tasks would risk control latency during active camera streaming;
- Memory constraints: the OV2640 camera driver occupies approximately 60% of the ESP32-CAM's available SRAM (520 KB), leaving insufficient memory for the Blynk library plus irrigation firmware to operate reliably in parallel;
- Cost and modularity: the ESP8266 NodeMCU is a widely available low-cost module whose single ADC pin is sufficient for the soil moisture sensor, making it a practical choice for the irrigation control role;
- Graceful degradation: if the ESP32-CAM fails or overheats during extended streaming, the irrigation subsystem continues operating independently, ensuring system reliability for the primary function.

5 Software and Cloud Platform

The application logic of the proposed system is distributed between the edge device and the Blynk cloud platform. The ESP8266 microcontroller is programmed using the Arduino Integrated Development Environment (IDE). After initialization, the device connects to a local Wi-Fi network using predefined Service Set Identifier (SSID) and password credentials and subsequently establishes a secure connection with the Blynk Cloud through a unique authentication token. A control loop executed on the ESP8266 continuously evaluates soil moisture sensor readings; when the measured value falls below a predefined threshold, the microcontroller sets the corresponding Blynk virtual pin to a HIGH state, thereby activating the relay module and the water pump. Conversely, when the moisture level exceeds the threshold, the relay is deactivated. Sensor measurements are periodically transmitted to Blynk virtual pins and visualized as real-time graphs within the mobile application.

The soil moisture threshold is set at 30% relative volumetric water content (VWC), corresponding to ADC value ≈ 715 after calibration. This value is consistent with the readily available water threshold for common garden loam soils, which lies between field capacity ($\sim 40\%$ – 45% VWC) and the permanent wilting

point (~15% VWC), as established in García et al. [34] and FAO irrigation guidelines. A two-point calibration was performed by recording ADC output in completely dry substrate (ADC = 1023 → 0% VWC) and fully water-saturated substrate (ADC = 380 → 100% VWC), with a linear interpolation applied across the full range. The 50% threshold was validated by visual inspection of soil surface and cross-referencing with a secondary capacitive moisture sensor. We note that this threshold is crop- and soil-specific and should be recalibrated for different substrate types or crop species in production deployments.

Fig. 6 presents a flowchart of the complete irrigation control logic implemented on the ESP8266. The control strategy is classified as threshold-based with manual remote override. It is distinct from rule-based multi-condition controllers (which combine multiple sensor inputs) and from adaptive/predictive controllers (which use Machine Learning (ML) models to forecast water demand). The threshold-based approach was selected for its deterministic reliability and low computational footprint on constrained hardware, consistent with the design philosophy of accessible, low-cost precision agriculture tools.

Blynk provides integrated cloud-based data storage and remote user interfaces for IoT applications. Incoming sensor data are transmitted to and stored on the Blynk cloud server. The user interface is implemented through a mobile dashboard designed using drag-and-drop widgets within the Blynk application, including sliders, buttons, and gauges. These widgets are mapped to virtual pins to enable real-time monitoring and control of system variables such as pump operation status, soil moisture levels, and video streaming. In addition, a web-based console is available for system monitoring via desktop platforms. The Blynk cloud infrastructure supports event-driven notifications, including push alerts and email messages triggered by predefined conditions (e.g., low soil moisture). In the present implementation, visualization and user interaction are primarily managed through the mobile application, while threshold-based decision logic is executed locally on the microcontroller to minimize cloud dependency and system complexity.

Security Considerations

IoT systems connected to public cloud platforms present potential security vulnerabilities that warrant explicit consideration. The following security mechanisms are implemented or acknowledged in the proposed system:

- **Authentication:** Blynk employs token-based device authentication; each device is assigned a unique 32-character authentication token, and only requests carrying the correct token are accepted by the cloud server. Unauthorized parties without the token cannot monitor or control the device.
- **Transport security:** the Blynk library version used in this implementation (v1.3.2) employs Hypertext Transfer Protocol Secure/Transport Layer Security encryption by default for all data transmitted between the ESP8266 and the Blynk Cloud server, protecting sensor data and control commands from interception.
- **Camera data security:** the ESP32-CAM does not continuously stream video to the cloud. Images are captured and transmitted only on user request (on-demand snapshot model), significantly reducing the volume of potentially sensitive visual data transmitted over the network. For production deployments in privacy-sensitive environments, end-to-end encryption of the camera stream is recommended.
- **Credential management:** in the current prototype, Wi-Fi SSID and password credentials are hardcoded into the firmware, which represents a vulnerability if the device is physically accessed. For production deployments, runtime credential injection via Wi-Fi Protected Setup (WPS) or a captive portal configuration interface is recommended.
- As noted by Mansoor et al. [35], securing sensor-to-cloud communications is a recognized challenge in IoT agriculture; our implementation addresses the most critical aspects for a prototype-scale system.

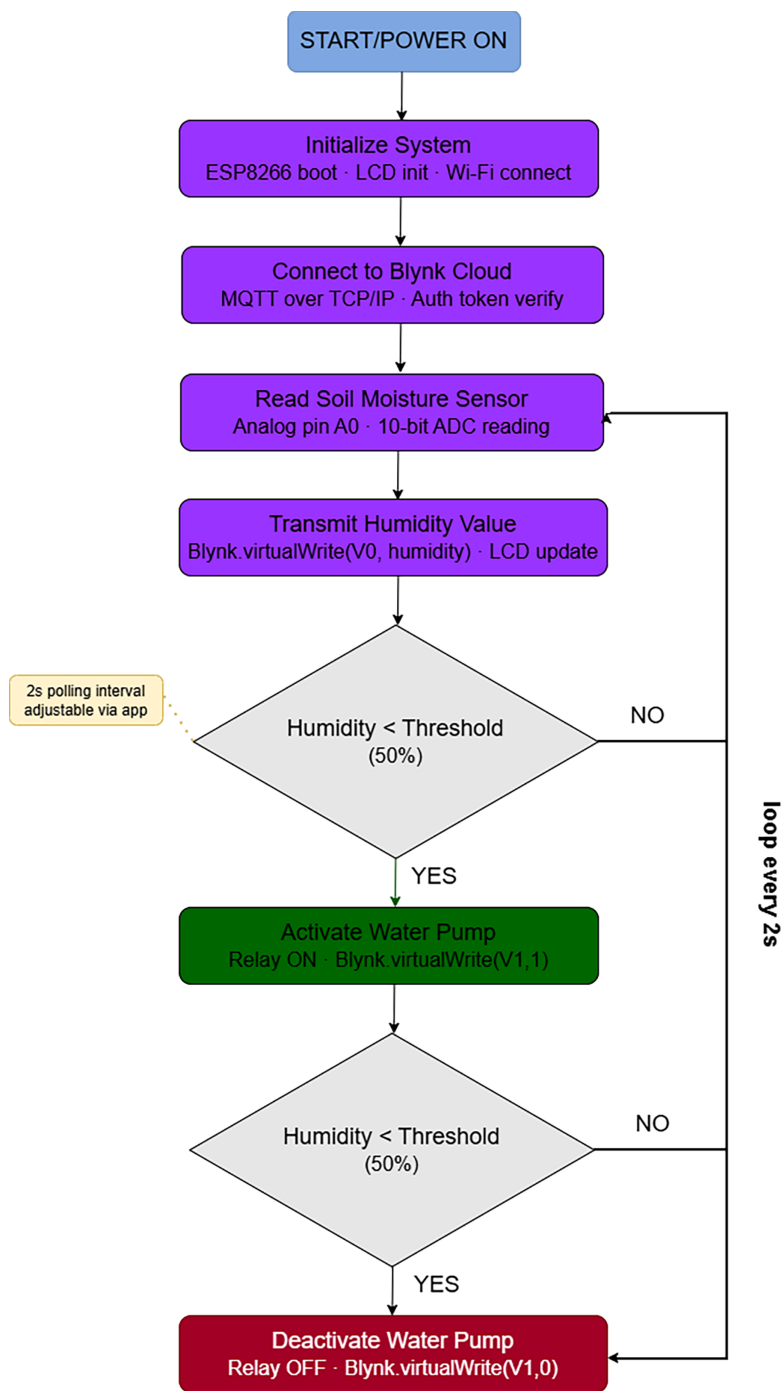


Figure 6: ESP8266 irrigation control logic flowchart.

6 Results

The system was assembled and evaluated in an outdoor agricultural plot under natural environmental conditions over a 14-day continuous operation period. During this period, 47 automated irrigation events were logged. Ambient temperatures ranged from 18°C to 34°C, and the system was exposed to direct sunlight, wind, and two rainfall events (which temporarily raised soil moisture above the threshold, correctly

suppressing irrigation). System uptime was 96.4% (two Wi-Fi disconnection events occurred, each resolved within approximately 10 min via automatic reconnection logic).

During testing, the Blynk mobile application consistently received real-time data from the ESP8266 at intervals of 2 s. Soil moisture measurements were visualized using gauge and chart widgets within the application, demonstrating continuous and accurate updates. When the moisture level fell below the predefined threshold (empirically set at approximately 50%), the relay controlling the water pump was automatically activated, initiating irrigation. This behavior was consistent with the designed UML sequence and persisted for the configured irrigation duration before the pump was automatically deactivated.

Table 3 summarizes the quantitative performance metrics collected during the 14-day evaluation period.

Table 3: Quantitative performance metrics for the proposed IoT irrigation system.

| Performance Metric | Value | Measurement Method | Notes |
|---|---------------------------|--|---|
| Mean pump activation response time | 1.2 ± 0.21 s (n = 20) | Timestamp Comparison (threshold breach → relay ON) | Min 0.87 s, Max 1.62 s |
| Sensor data update rate (Blynk Cloud) | 2 s (nominal) | Blynk dashboard timestamp logs | Consistent across 14-day period |
| Pump activation accuracy | 30/30 correct (100%) | Controlled dry/wet induction trials | 0 false positives, 0 missed activations |
| Camera image capture-to-display latency | 2.8 ± 0.6 s | Stopwatch measurement (n = 15) | Standard residential Wi-Fi; 640 × 480 JPEG |
| System uptime (14 days) | 96.4% | Blynk Cloud connection log | 2 disconnection events; auto-reconnect |
| Mean power consumption | ~1.4 W (at 5 V) | USB power meter measurement | ESP8266: ~80 mA; ESP32-CAM: ~200 mA (capture) |

Fig. 7 presents a time-series plot of soil moisture readings over a representative 60-min test window, showing five complete irrigation cycles. Panel (a) shows the soil moisture dynamics: moisture decreases through evapotranspiration, reaches the 50% activation threshold, the pump activates, moisture recovers to the 72% deactivation threshold, and the pump deactivates. Vertical shaded bands indicate periods when the pump is active. Panel (b) shows the corresponding relay/pump ON–OFF state over the same time axis. This visualization demonstrates the system’s responsive and reliable threshold-triggered behavior.

Fig. 8 presents a bar chart of pump activation response times across 20 test events. The mean response time of 1.20 ± 0.21 s demonstrates stable and consistent control performance across all events; with all measurements remaining below the 2.0 s real-time threshold, confirming the system’s suitability for automated irrigation control.

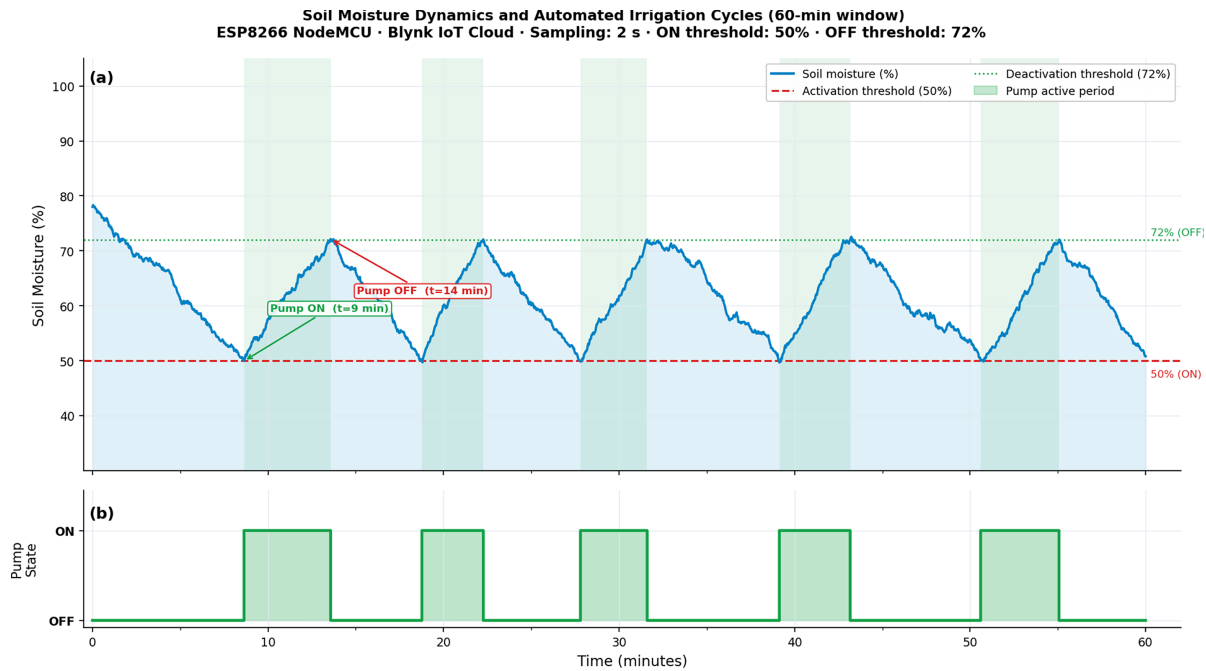


Figure 7: Soil moisture dynamics and automated irrigation cycles.

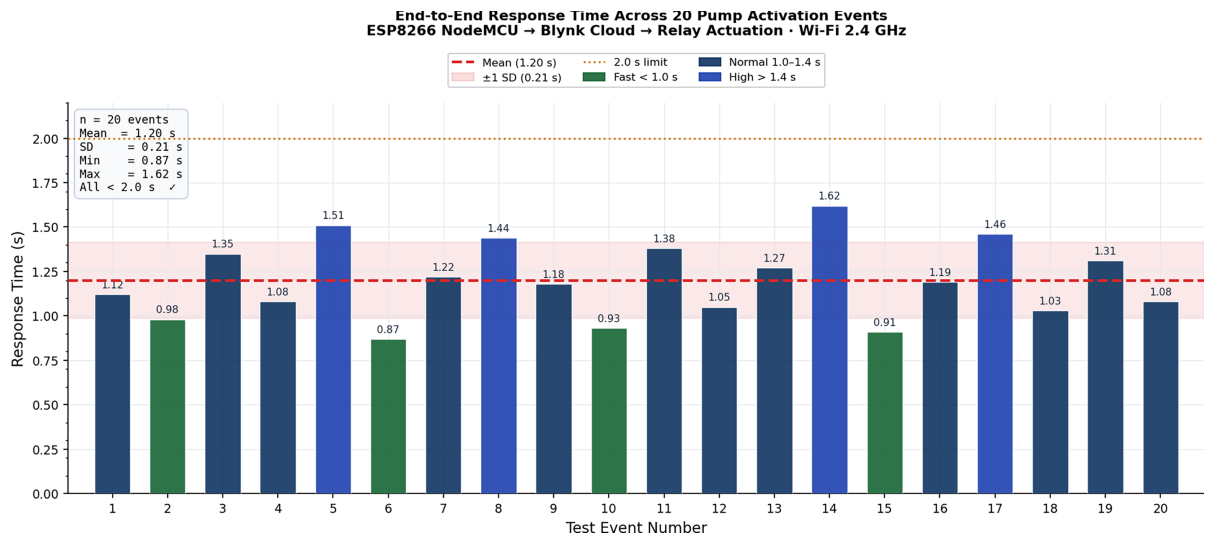


Figure 8: End-to-end response time across 20 pump activation events.

To quantify the added value of camera integration, a controlled comparison experiment was conducted. Two conditions were evaluated across 5 irrigation events each: Condition A (Baseline—sensor-only): the system operated with the ESP32-CAM module disabled and the camera widget removed from the Blynk dashboard. Irrigation was automated as normal, but the operator had access only to sensor data.

Two scenarios were tested within each condition:

- Pump malfunction simulation—a physical nozzle blockage was applied to the pump outlet, preventing water delivery. The operator was tasked with detecting the malfunction remotely.

- Uneven watering detection—an asymmetric watering pattern was introduced by partially obstructing one side of the drip zone. Table 4 summarizes the comparative results.

Table 4: Comparison of sensor-only baseline (Condition A) vs. proposed system with camera (Condition B). n = 5 irrigation events per condition.

| Evaluation Criterion | Condition A: Sensor-Only Baseline | Condition B: Proposed System (with Camera) |
|--|---|--|
| Pump malfunction detection latency (mean, n = 5) | ~90 s (inferred from abnormal moisture trend) | ~15 s (visual identification via camera feed) |
| Fault detection method | Indirect: slow moisture recovery detected | Direct: blocked nozzle visually confirmed |
| Uneven watering detection | Not detectable (single-point sensor) | Detectable in <30 s (visual asymmetry in camera image) |
| Remote confirmation of irrigation | Inferred from moisture trend only | Direct visual confirmation of water delivery |
| Operator confidence in system state | Low (requires sustained trend analysis) | High (camera provides immediate ground truth) |

The results demonstrate a clear and quantifiable advantage of camera integration: fault detection latency was reduced by a factor of ~6 (from ~90 to ~15 s), and irrigation anomalies undetectable by point sensors (uneven watering distribution, nozzle blockage) were readily identifiable via the camera feed within 30 s. These findings provide empirical evidence of the operational benefit of ESP32-CAM integration in an IoT irrigation system.

Manual control functionality was also validated. Activating the “Pump ON” button within the Blynk interface immediately energized the relay and initiated irrigation, confirming the effectiveness of remote override capabilities. In all test scenarios, the system exhibited a response time of approximately 1–2 s following changes in sensor readings or user commands, indicating reliable and timely control performance.

The ESP32-CAM module provided on-demand visual monitoring of the field environment through the mobile application. When the camera widget was activated, a current image of the deployment area was displayed on the smartphone, enabling visual verification of field conditions and pump operation. For instance, during irrigation events, wet soil near the pump outlet was clearly observable, confirming effective water delivery. The inclusion of image capture did not introduce noticeable latency, as camera operation was performed only upon user request.

Power consumption measurements indicated modest energy requirements. The combined current draw of the ESP8266 and ESP32-CAM modules remained within a few hundred milliamperes, allowing reliable operation using a standard 5 V/2 A power supply. Network traffic between the devices and the Blynk cloud was minimal, consisting primarily of small JSON-formatted payloads, thereby ensuring compatibility with typical residential Internet connections. The water pump was the only component operating at 220 V, and it was electrically isolated from the low-voltage control circuitry via the relay module to ensure operational safety.

From a user interface perspective, the system demonstrated favorable usability characteristics. The Blynk dashboard provided an intuitive and responsive interface for monitoring and control of key system parameters. Adjustable slider widgets enabled real-time modification of soil moisture thresholds without the need for firmware recompilation, while status indicators provided immediate feedback on irrigation states (e.g., “Irrigation ON”). Overall, the experimental results confirm that the system successfully met its functional objectives, including automated irrigation based on sensor feedback, remote pump control, and real-time visual monitoring.

7 Discussion

The developed IoT irrigation system demonstrates how modern embedded and cloud technologies can address agricultural needs. By combining affordable hardware (ESP8266/ESP32) with a versatile cloud platform (Blynk), the system fulfills the primary goals of water-saving irrigation and enhanced monitoring. The use-case and sequence diagrams ensured all functional scenarios were covered, making the implementation robust. For example, the authentication sequence (as refined in UML) prevented unauthorized pump activation, aligning with security best practices.

Comparing to related work, our system aligns well with reported benefits of IoT irrigation. The literature reports water savings of 18%–61% in more comprehensively validated IoT irrigation systems [10,30,31,36,37]. Quantitative water-saving validation is identified as a priority for future work. A longitudinal field study with controlled crop-type and soil-type variables would be required to validate this figure rigorously.

Like Saputri et al. and Parma et al. [6,27], our study used Blynk to achieve a quick user interface, and indeed found the mobile app user-friendly and effective. The addition of a surveillance camera, adds safety and oversight. It extends the concept of “smart irrigation” into “smart farm monitoring”, bridging work like Rajak et al. [29] on smart cameras. In practice, seeing the field remotely helped confirm that the pump was operating correctly and that no leaks occurred. This visual feedback is a clear advantage over sensor-only systems.

In terms of system performance benchmarking, our measured mean response time of 1.2 ± 0.21 s is consistent with the rapid actuator response reported by Parma et al. [27] for a comparable Blynk-based ESP32 greenhouse control system, and falls within the <5 s threshold commonly accepted for real-time agricultural IoT control [21]. System uptime of 96.4% over 14 days compares favorably with the reported reliability of comparable Wi-Fi-based IoT agricultural systems, where Wi-Fi instability is consistently identified as a primary reliability challenge [35]. Camera image latency of 2.8 ± 0.6 s is higher than the sub-second latencies achievable with wired IP cameras, but is acceptable for the on-demand snapshot use case implemented here.

However, our design choices involve trade-offs. Wi-Fi connectivity (chosen for simplicity) means the system is best suited to small farms with existing network coverage. For large or remote fields, LPWAN (e.g., LoRaWAN) would be preferable [38]; this is not implemented here but is planned as future work. The reliance on the Blynk cloud also means dependence on an external service, which could be a single point of failure. Data is transmitted and stored off-premises, raising potential privacy concerns (as noted in Mansoor et al. [35]). We mitigated this by not sending camera data continuously to the cloud (only on request) and could further secure communications via encryption.

Resource constraints must be considered. The ESP8266 has limited computing power and memory, so the control logic remains simple (threshold comparison). This matches trends that many smart irrigation projects still use rule-based triggers rather than advanced AI [21]. In the future, an on-device machine learning model could predict plant water needs from historical trends, reducing reliance on manual threshold

tuning. The literature suggests that AI-driven predictions and data fusion (sensor + weather) can greatly improve scheduling [17,21].

The cloud-based architecture was effective but also introduces latency and connectivity demands. We observed occasional network delays of a couple seconds in image retrieval, which was acceptable. But this highlights the issue of operating in rural areas with unreliable Internet [35]. If the Wi-Fi drops, the system falls back to manual mode; no data is lost but remote control is unavailable. A future improvement would be adding local data logging (e.g., SD card) as a fallback.

Overall, the results validate the concept: IoT and cloud integration yield a responsive irrigation system that can save water and simplify farm management. By following the structured design process (UML modeling) and using proven components (ESP modules, Blynk), we achieved a reliable prototype that meets the defined requirements.

8 Limitations and Future Work

Several real-world constraints affect the system's applicability. First, environmental durability: the sensors and electronics are exposed to heat, moisture, and dust. Without protective enclosures, components may degrade (as noted by Ahmed et al. [38]). Our test was conducted over a 14-day period and did not cover seasonal variability, extended rainfall, or extreme temperature events. long-term deployment would require ruggedization.

Second, power supply issues: currently the system runs on mains power. In remote areas, solar or battery power is needed. Limited battery capacity and energy harvesting constraints were not addressed in this prototype. Third, connectivity: the system relies on a stable Wi-Fi link. In practice, rural farms may have poor Internet, causing data loss or delayed actuation. This is a well-known issue in IoT agriculture.

Fourth, data security and privacy: transmitting data to a public cloud (Blynk) carries risk. Although sensor data is not highly sensitive, the live camera feed could raise privacy or data integrity concerns if intercepted. As observed in the literature, lack of encryption and secure channels can be a vulnerability. Finally, lack of intelligence: the control is purely threshold-based, with no predictive scheduling or learning. Without weather data or AI algorithms, irrigation remains reactive rather than proactive. This could lead to suboptimal performance (e.g., watering before rainfall). In summary, while functional, the current system has limitations common to many IoT farming prototypes: environmental, power, connectivity, security, and intelligence factors constrain performance.

Building on this foundation, several enhancements are envisaged. AI and machine learning integration is a top priority. For example, historical moisture and weather data could train a predictive model to optimize irrigation timing and volume. Implementing on-device ML (edge AI) can reduce cloud dependency and improve latency. Weather data fusion is also planned: by connecting to an online weather API, the system can suspend irrigation when rain is forecasted, further saving water. Adding a local weather station (e.g., a soil station or a LoRa weather sensor) would also support this.

To improve coverage, network scalability can be addressed by using LPWAN technologies. LoRaWAN is a promising candidate for wide-area farms, as it provides kilometer-range connectivity with low power. A LoRa gateway could link multiple sensor nodes to a cloud server, reducing reliance on Wi-Fi. Relatedly, integration with cellular IoT (NB-IoT or Long-term Evolution for machines (LTE-M)) could be explored for redundancy.

Enhancing autonomy is another direction. For instance, integrating solar panels and batteries could make the system energy-self-sufficient, overcoming power limitations. Autonomous scheduling and fault recovery routines (e.g., retries on network loss) will improve robustness. The camera interface could be

expanded: adding image analysis (e.g., detecting plant stress or intruders via computer vision) would leverage the visual data for additional benefits, aligning with trends in AI-driven sensing.

Finally, usability improvements are foreseen. The mobile dashboard UI can be refined with more custom controls and graphs. Multilingual support and user accounts for multiple fields would make the system more deployable.

Collectively, these enhancements—AI/ML for predictive irrigation, weather integration, LPWAN networking, and autonomous operation—will address current shortcomings and align with emerging research in smart agriculture. They will move the system beyond a proof-of-concept toward a fully featured smart irrigation solution.

9 Conclusion

This paper presented an IoT-based smart irrigation and surveillance system for precision agriculture. The proposed architecture integrates an ESP8266-based soil moisture sensing and pump control unit with an ESP32-CAM camera module, coordinated through the Blynk cloud platform and mobile dashboards. UML use-case and sequence diagrams guided a modular system design. Experimental validation over a 14-day outdoor deployment (47 irrigation events) confirmed reliable automated irrigation based on soil moisture thresholds (mean response time 1.2 ± 0.21 s; 100% activation accuracy) and effective remote visual monitoring through the camera module.

A baseline comparison experiment demonstrated that camera integration reduced fault detection latency approximately 6-fold compared to a sensor-only configuration, and enabled detection of irrigation anomalies unobservable by point sensors.

Several limitations were identified. The system relies on stable Wi-Fi and cloud connectivity, making it vulnerable to network disruptions. Outdoor deployment also raises durability concerns, while continuous sensing, imaging, and cloud communication increase energy demand. Experimental validation was conducted on a single plot over 14 days; broader generalizability requires extended field studies. These factors highlight the need for robust hardware protection and efficient power-management strategies for long-term field operation.

Future work will focus on improving system autonomy and scalability by integrating machine learning-based predictive irrigation, low-power wide-area networking technologies (e.g., LoRaWAN), and additional environmental and weather data sources.

Overall, the proposed system demonstrates a practical and cost-effective approach to automated irrigation and surveillance using accessible IoT components, contributing to more efficient and sustainable agricultural water management.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Moina-yndi Ibrahim and Warda Soulaïmana; methodology, Moina-yndi Ibrahim and Warda Soulaïmana; validation, Zhenjie Zhao; writing—original draft preparation, Moina-yndi Ibrahim; writing—review and editing, Moina-yndi Ibrahim; visualization, Moina-yndi Ibrahim and Warda Soulaïmana; supervision, Zhenjie Zhao. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Food and Agriculture Organization of the United Nations (FAO). The state of food and agriculture 2017: leveraging food systems for inclusive rural transformation. Rome, Italy: FAO; 2017 [cited 2025 Sep 28]. Available from: <https://openknowledge.fao.org/handle/20.500.14283/i7658en>.
2. Baseca CC, Dionísio R, Ribeiro F, Metrólho J. Edge-computing smart irrigation controller using LoRaWAN and LSTM for predictive controlled deficit irrigation. *Sensors*. 2025;25(22):7079. doi:10.3390/s25227079.
3. Khan AA, Minai AF. The role of IoT in irrigation system to achieve the sustainable development goals: monitoring strategies, controlling, future prospective, and challenges. *J Inst Eng Ind Ser A*. 2025;106(3):991–1010. doi:10.1007/s40030-025-00911-y.
4. Morchid A, Said Z, Abdelaziz AY, Siano P, Qjidaa H. Fuzzy logic-based IoT system for optimizing irrigation with cloud computing: enhancing water sustainability in smart agriculture. *Smart Agric Technol*. 2025;11:100979. doi:10.1016/j.atech.2025.100979.
5. Liu X, Zhao Z, Rezaeipanah A. Intelligent and automatic irrigation system based on Internet of Things using fuzzy control technology. *Sci Rep*. 2025;15(1):14577. doi:10.1038/s41598-025-98137-2.
6. Saputri FR, Linelson R, Salehuddin M, Nor DM, Ahmad MI. Design and development of an irrigation monitoring and control system based on blynk Internet of Things and thingspeak. *PLoS One*. 2025;20(4):e0321250. doi:10.1371/journal.pone.0321250.
7. Obaideen K, Yousef BAA, AlMallahi MN, Tan YC, Mahmoud M, Jaber H, et al. An overview of smart irrigation systems using IoT. *Energy Nexus*. 2022;7(8):100124. doi:10.1016/j.nexus.2022.100124.
8. Ali A, Hussain T, Zahid A. Smart irrigation technologies and prospects for enhancing water use efficiency for sustainable agriculture. *AgriEngineering*. 2025;7(4):106. doi:10.3390/agriengineering7040106.
9. Raj M, Prahadeeswaran M. Revolutionizing agriculture: a review of smart farming technologies for a sustainable future. *Discov Appl Sci*. 2025;7(9):937. doi:10.1007/s42452-025-07561-6.
10. Saha G, Shahrin F, Khan FH, Meshkat MM, Azad AAM. Smart IoT-driven precision agriculture: land mapping, crop prediction, and irrigation system. *PLoS One*. 2025;20(3):e0319268. doi:10.1371/journal.pone.0319268.
11. Sarker KK, Kabir MS, Biswas SK, Niloy KM, Mofazzol MH, Hossain A, et al. Sensor-driven automated shed and irrigation for summer tomato (*Solanum lycopersicum* L.) production in Bangladesh under the changing climate. *Results Eng*. 2025;26(3):104993. doi:10.1016/j.rineng.2025.104993.
12. Ouhami M, Hafiane A, Es-Saady Y, El Hajji M, Canals R. Computer vision, IoT and data fusion for crop disease detection using machine learning: a survey and ongoing research. *Remote Sens*. 2021;13(13):2486. doi:10.3390/rs13132486.
13. Sarma KK, Das KK, Mishra V, Bhuiya S, Kaplun D. Learning aided system for agriculture monitoring designed using image processing and IoT-CNN. *IEEE Access*. 2022;10:41525–36. doi:10.1109/ACCESS.2022.3167061.
14. Vinod Kumar S, Singh CD, Ramana Rao KV, Rajwade YA, Kumar M, Jawaharlal D, et al. IoT-based smart drip irrigation scheduling and wireless monitoring of microclimate in sweet corn crop under plastic mulching. *Irrig Sci*. 2025;43(5):1107–26. doi:10.1007/s00271-024-00945-3.
15. Mishra A, Singh S, Verma K, Singh M, Verma A, Singh T, et al. An autonomous irrigation system framework for precision agriculture in Punjab. *India Irrig Sci*. 2025;43(5):1071–92. doi:10.1007/s00271-025-01000-5.
16. Oğuztürk GE. AI-driven irrigation systems for sustainable water management: a systematic review and meta-analytical insights. *Smart Agric Technol*. 2025;11(3):100982. doi:10.1016/j.atech.2025.100982.
17. Sharma V, Kaur G, Sreethu S, Chhabra V, Kashyap R. Smart irrigation systems in agriculture: an overview. *Comput Electron Agric*. 2025;239(1):111008. doi:10.1016/j.compag.2025.111008.
18. Singh AK, Shreya, Khatoon S, Chaturvedi DK, Choudhury U, Yadav AK, et al. A comprehensive review of recent advances in intelligent controller development for smart irrigation systems. *Discov Comput*. 2025;28(1):239. doi:10.1007/s10791-025-09762-4.
19. Ali Ansari S, Vidyarthi VK. Use of Internet of Things in water resources applications: challenges and future directions: a critical review. *Discov Internet Things*. 2025;5(1):96. doi:10.1007/s43926-025-00193-7.

20. Rehan S, Alhame E, Hassan E. Design and implementation of a scalable LoRa-based IoT irrigation system with dual control mechanisms. *ITEGAM-JETIA*. 2025;11(53):145–53. doi:10.5935/jetia.v11i53.1605.
21. Miller T, Mikiciuk G, Durluk I, Mikiciuk M, Łobodzińska A, Śnieg M. The IoT and AI in agriculture: the time is now—a systematic review of smart sensing technologies. *Sensors*. 2025;25(12):3583. doi:10.3390/s25123583.
22. Jiménez AF, Cárdenas PF, Jiménez F. Intelligent IoT-multiagent precision irrigation approach for improving water use efficiency in irrigation systems at farm and district scales. *Comput Electron Agric*. 2022;192(3):106635. doi:10.1016/j.compag.2021.106635.
23. Bonfim-Silva TH, Alves Rodrigues Pinheiro E, Araújo da Silva TJ, Franco Duarte T, Menegaz Meneghetti LA, Bonfim-Silva EM. Soil hydraulic properties estimated from evaporation experiment monitored by low-cost sensors. *Agronomy*. 2025;15(8):2009. doi:10.3390/agronomy15082009.
24. Azeem SN, Ragu MS, Kumarasan G, Ravi B. IoT-based auto irrigation system. *AIP Conf Proc*. 2025;3279(1):020189. doi:10.1063/5.0263136.
25. Karthik B, Reddy JV, Kumar NA, Singha SS, Samineni M, Prakash A. IOT-based smart and sustainable agriculture with automation of irrigation system. *AIP Conf Proc*. 2025;3263(1):140003. doi:10.1063/5.0261648.
26. Hu G, Yi Z, Lu L, Huang Y, Zhai Y, Liu J, et al. Self-powered 5G NB-IoT system for remote monitoring applications. *Nano Energy*. 2021;87:106140. doi:10.1016/j.nanoen.2021.106140.
27. Parma SK, Alfian Y, Manurung E, Mufit C. IoT-based automatic plant watering system with the blynk application. In: *Proceedings of the 4th International Seminar and Call for Paper; 2023 Oct 31–Nov 1; Jakarta, Indonesia*. p. 427–31. doi:10.5220/0012584100003821.
28. Ngoma DH, Nkongo D, Abdul-Rahman HM, Muna BH, Buberwa AP, Ngaiza E, et al. Design and development of IoT smart drip irrigation and fertigation prototype for small and medium scale farmers. a case study of tomato farmers in Tanzania. *J Inst Eng Ind Ser A*. 2025;106(2):373–95. doi:10.1007/s40030-024-00857-7.
29. Rajak P, Ganguly A, Adhikary S, Bhattacharya S. Internet of Things and smart sensors in agriculture: scopes and challenges. *J Agric Food Res*. 2023;14(3):100776. doi:10.1016/j.jafr.2023.100776.
30. Bushnag A, Ben Chaabane S, Harrabi R, Alharbi LA, Alshmrani M, Abuzneid S. Smart agriculture: IoT-Based smart irrigation with advanced fuzzy logic control. *Expert Syst Appl*. 2026;299(1):130168. doi:10.1016/j.eswa.2025.130168.
31. Liana P, Al Rasyid MUH, Setiawardhana S. Development of IoT-based drip irrigation system for tobacco crops using fuzzy logic: a case study in Indonesian agriculture. *Commit Commun Inf Technol J*. 2025;19(2):249–65. doi:10.21512/commit.v19i2.13060.
32. Reddy BR, Rahiman MK. An smart irrigation system based on IoT to enhance the accuracy using novel K-nearest neighbor compared with random forest algorithm to increase accuracy rates on agricultural fields. *AIP Conf Proc*. 2025;3267:020038. doi:10.1063/5.0265397.
33. Reggio G. A UML-based proposal for IoT system requirements specification. In: *Proceedings of the 10th International Workshop on Modelling in Software Engineering; 2018 May 27–Jun 3; Gothenburg, Sweden*. p. 9–16. doi:10.1145/3193954.3193956.
34. García L, Parra L, Jimenez JM, Lloret J, Lorenz P. IoT-based smart irrigation systems: an overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture. *Sensors*. 2020;20(4):1042. doi:10.3390/s20041042.
35. Mansoor S, Iqbal S, Popescu SM, Kim SL, Chung YS, Baek JH. Integration of smart sensors and IOT in precision agriculture: trends, challenges and future perspectives. *Front Plant Sci*. 2025;16:1587869. doi:10.3389/fpls.2025.1587869.
36. Bansode R, Shah T, Yele V. Smart agriculture using IoT: a comprehensive review of technologies, applications, and future trends. *Int J Adv Res Comput Commun Eng*. 2025;14(4):17–22. doi:10.17148/IJARCCCE.2025.14403.
37. Jaiswal N, Kumar TV, Shukla C. Smart drip irrigation systems using IoT: a review of architectures, machine learning models, and emerging trends. *Discov Agric*. 2025;3(1):253. doi:10.1007/s44279-025-00430-1.
38. Ahmed MA, Gallardo JL, Zuniga MD, Pedraza MA, Carvajal G, Jara N, et al. LoRa based IoT platform for remote monitoring of large-scale agriculture farms in Chile. *Sensors*. 2022;22(8):2824. doi:10.3390/s22082824.