Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3125

DESIGN OF A DYNAMIC MONITORING AND TIERED EARLY WARNING SYSTEM FOR CULVERT WATERLOGGING BASED ON A CLOUD-EDGE COLLABORATIVE ARCHITECTURE

YiTao Li
School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, Liaoning, China.

Abstract: In response to the frequent waterlogging of urban culverts induced by extreme rainfall, the vulnerability of traditional monitoring methods, and the lack of targeted early warnings, this paper designs and implements an Internet of Things (IoT)-based dynamic monitoring and tiered early warning system utilizing a cloud-edge collaborative architecture. At the perception layer, the system employs a non-contact computer vision approach. By deploying dual lightweight YOLOv8 object detection models on edge computing nodes (Raspberry Pi 4B), it synchronously extracts on-site water depth and the chassis height of passing vehicles in real time. At the decision layer, a tiered early warning logic based on dynamic "water level-vehicle type" matching is proposed. This approach overcomes the limitations of traditional uniform static thresholding strategies and enables sub-second local audio-visual traffic control directly at the edge. Furthermore, the edge nodes transmit lightweight structured data to the Huawei Cloud IoT platform for aggregation, ultimately delivering public services via a WeChat Mini Program. Users can leverage the embedded map component to monitor the real-time waterlogging status across various monitoring sites and receive vehicle-specific routing recommendations. Prototype testing results demonstrate that the system's visual perception error is maintained within a reasonable margin, and the overall response latency at the edge is less than 500 ms. Crucially, the system retains offline resilience and local warning capabilities even in extreme scenarios involving network interruptions. This research provides a low-cost, robust engineering solution for smart urban flood control and emergency traffic dispatching.
Keywords: Culvert waterlogging; Cloud-edge collaboration; Edge computing; YOLOv8; Tiered early warning; Internet of Things (IoT)

1 INTRODUCTION

Driven by global climate change and rapid urbanization, urban waterlogging disasters induced by short-term extreme rainfall have become increasingly frequent [1]. In low-lying topographies like underpass tunnels and railway culverts, rapid runoff accumulation frequently causes severe traffic paralysis and poses significant safety risks.

Current monitoring systems typically rely on two categories of water level sensors. Contact-based devices (e.g., submersible transmitters) suffer from frequent failures due to probe blockages and electrode corrosion in debris-filled wastewater [2]. Conversely, non-contact sensors (e.g., ultrasonic or radar meters) resist corrosion but offer strictly unidimensional functionality. They capture absolute water levels but cannot perceive traffic flow dynamics or identify approaching vehicle types.

This singular sensing capability limits existing warning mechanisms to a static thresholding approach (e.g., initiating full road closures at 20 cm of water). This coarse strategy ignores the varying wading capacities of different vehicles, often erroneously intercepting high-clearance emergency vehicles while inadvertently allowing low-clearance sedans to risk stalling. Furthermore, traditional architectures rely heavily on centralized cloud servers, rendering them highly vulnerable to paralysis during storm-induced network disruptions.

To address these vulnerabilities, this paper proposes a vision-based waterlogging monitoring system built upon a cloud-edge collaborative architecture. By replacing single-function sensors with lightweight machine vision algorithms, the system offloads core decision-making to the edge. This enables dynamic, localized early warnings based on precise "water level-vehicle type" matching. Concurrently, the system aggregates data via the Huawei Cloud IoT platform and integrates a WeChat Mini Program to provide drivers with vehicle-specific road condition queries and hazard-avoidance routing recommendations.

2 SYSTEM ARCHITECTURE DESIGN

To mitigate the latency associated with massive data transmission and enhance system robustness under extreme conditions, the proposed system employs a four-tier "Device-Edge-Cloud-Application" IoT architecture [3]. This design strategically restructures the distribution of computational resources. The system architecture is shown in Figure 1.

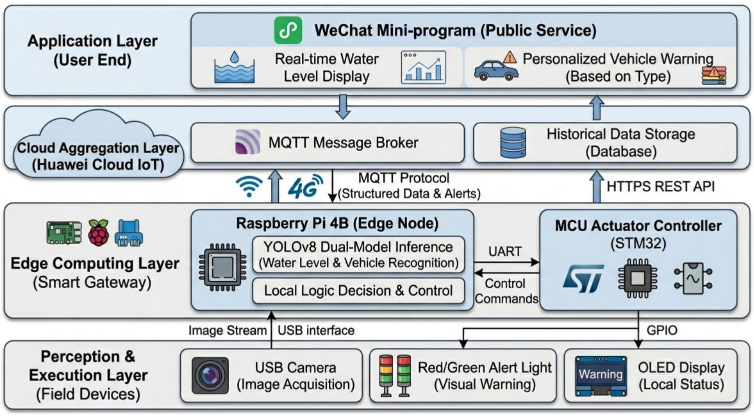

Figure 1 System Architecture Diagram

2.1 Perception and Execution Layer

At the perception layer, high-definition cameras are deployed at safe clearance heights within the culvert and at its access points to capture live video streams, ensuring complete physical isolation from the water body. The execution layer comprises high-brightness OLED displays, multi-color warning beacons, and boom barriers. These field devices are directly actuated by command signals issued via the serial port from the edge node, facilitating immediate physical traffic intervention and information dissemination [4].

2.2 Edge Computing Layer

Functioning as the core computational node, the edge layer utilizes a gateway based on a Raspberry Pi 4B. Rather than transmitting raw video streams to the cloud, the edge node locally deploys lightweight deep learning models to extract structured data (i.e., water depth and vehicle type). Furthermore, it processes the safety threshold comparison logic entirely within its local memory to directly drive the execution layer devices. This architecture substantially reduces network bandwidth consumption and guarantees the monitoring nodes' autonomous operation and control capabilities during network outages.

2.3 Cloud Scheduling Layer

Built upon the Huawei Cloud IoT platform, the cloud tier serves as the primary data aggregation and routing hub for large-scale distributed nodes. Edge devices transmit structured waterlogging data to the cloud via the MQTT protocol. The platform's rule engine then parses and sanitizes the real-time data, persisting long-term time-series records in the cloud database. Concurrently, it provides API interfaces to route real-time road conditions efficiently and stably to front-end public applications.

2.4 Application Service Layer

Addressing practical commuting needs, the application layer features a lightweight Mini Program developed via WeChat Cloud. Users can access the monitoring interface directly through WeChat without downloading a standalone application. Diverging from the dense data presentation of traditional dashboards, the system integrates intuitive map annotations and interactive features. By inputting their specific vehicle models, users receive customized, vehicle-specific routing recommendations.

3 KEY TECHNOLOGIES AND IMPLEMENTATION

3.1 Edge Dual-Perception Model Based on YOLOv8

To balance the limited computational capacity of edge devices with the requirement for high-precision recognition, the system adopts the YOLOv8n (Nano) object detection model. To accelerate inference, the model undergoes ONNX format conversion and quantization [5]. Within the edge nodes, two specific perception tasks run alternately:
(1) Non-contact water level recognition: A standard water gauge featuring an alternating black-and-yellow scale is deployed on the culvert sidewall. By identifying the relative pixel positions between the water surface intersection line and the gauge's reference point, the model translates image coordinates into physical depth [6]. To ensure system robustness, a dataset of 1,800 water level images under diverse environmental conditions was constructed. This dataset was partitioned into training, validation, and testing sets at a 7:2:1 ratio.
(2) Multi-scale vehicle classification: Before entering the monitoring blind spot, vehicles are classified in real time based on features including contour, wheel hub height, and body proportions. Categories include low-chassis (e.g., sedans, ~10 cm ground clearance), medium-chassis (e.g., SUVs, ~20 cm), and high-chassis vehicles (e.g., trucks/buses, ~30 cm) [7].

Through asynchronous data exchange within a local shared memory pool, the models achieve seamless state synchronization between the "environmental water level" and the "target vehicle".

3.2 Dynamic Tiered Control and Early Warning Logic

Based on the data acquired by the dual-perception model, the edge node incorporates a differentiated early warning decision tree. Assuming the current detected water depth is D, and the safe wading threshold for the detected vehicle category is $T_c(c\in\{Sedan,SUV,Truck\}$, the system's tiered control strategy is defined by the following discriminant function:

$$S(D,T_c)=\begin{cases} \&State\ 1\ (Green), & \&\&D<0.8\cdot\ T_c \\ \&State\ 2\ (Red), & \&\&D\geq T_c \quad \&\& \quad \&\& \end{cases} \tag{1}$$

The corresponding physical execution actions are described as follows:

When S=State1: Sufficient safety margin exists. The system actuates the execution layer to maintain a steady green light, displays the current water depth and "Caution, Flood Ahead, Slow to Pass" on the OLED screen, and keeps the boom barrier raised.

When S=State 2: The situation is deemed highly hazardous for the approaching vehicle. The boom barrier is lowered, the warning beacon transitions to a high-frequency flashing red mode, and the OLED screen displays "Warning, Flood Ahead, Please Stop" to intercept and divert the vehicle.

By implementing this vehicle-specific control policy, the logic maximizes the throughput of the urban road network during the nascent stages of waterlogging, effectively mitigating the misallocation of traffic resources [8]. The judgment flowchart is illustrated in Figure 2.

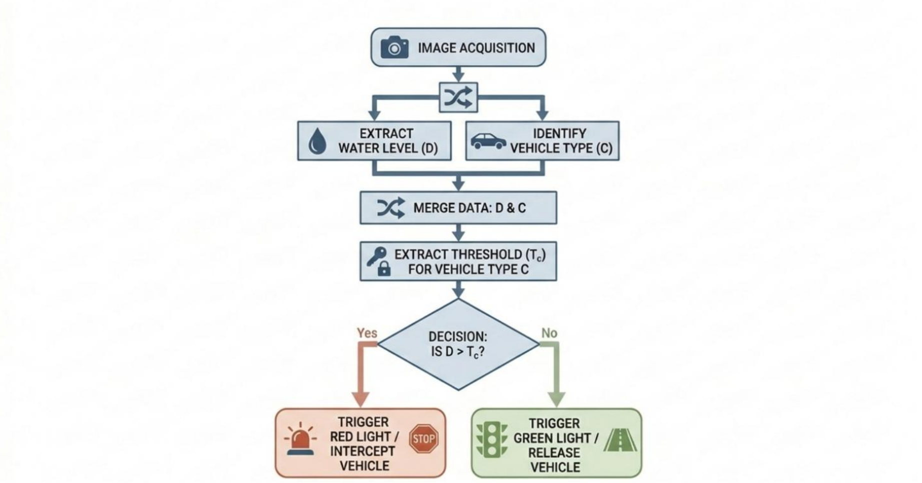

Figure 2 Flowchart of Dynamic Early Warning Logic

4 SOFTWARE AND HARDWARE SYSTEM IMPLEMENTATION

The edge node equipment must withstand all-weather, high-intensity operation in harsh outdoor environments. The edge node utilizes a 5W Raspberry Pi 4B (Cortex-A72, 4GB RAM) for low-power main control [9], paired with a 1080p camera for visual acquisition. Visual data is acquired via a 1080p high-definition camera. For control signal output, the Raspberry Pi communicates with an STM32 microcontroller via a UART serial port. The STM32 utilizes GPIO pins to drive on-site warning devices, including an OLED screen, warning beacons, and a servo-actuated boom barrier. The system simulation diagram is illustrated in Figure 3.

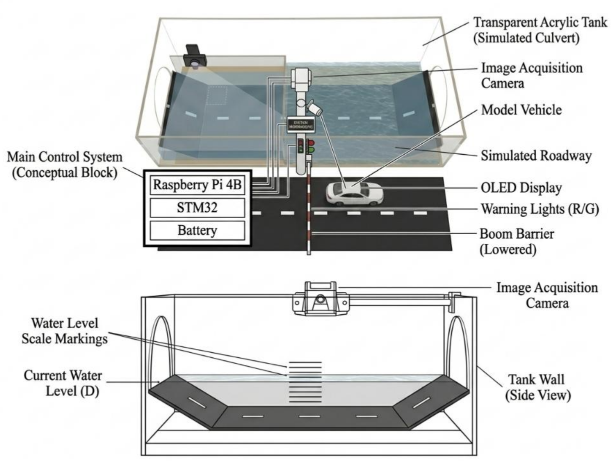

Figure 3 Hardware Setup of Edge Node

To guarantee uninterrupted operation, the entire hardware apparatus is encapsulated within an IP65-rated waterproof enclosure and integrates a backup lithium battery pack. During a mains power failure, the equipment autonomously transitions to battery power, sustaining essential local visual computation and audio-visual alarming functions.

4.1 Cloud Data Interaction and Collaboration

Addressing the information asymmetry regarding localized road conditions during rainfall, a native WeChat Mini Program was developed. Integrating the Tencent Map API for high-precision localization and the WebSocket protocol for sub-second data synchronization, the application provides real-time updates. Upon inputting their specific vehicle model, users receive intuitive passage safety recommendations for upcoming waterlogged segments, visualized via an elliptical dashboard (color-coded green or red).The user interface (UI) of the Mini Program is depicted in Figure 4.

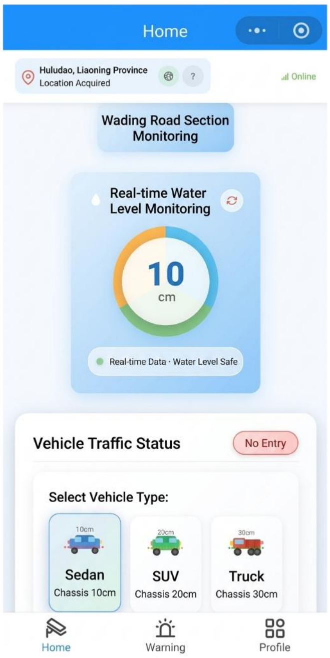

Figure 4 Dashboard and Mini-PROGRAM INTERFACES

5 SYSTEM PERFORMANCE EVALUATION AND ANALYSIS

To validate the proposed architecture and core performance metrics, a scaled-down culvert waterlogging simulation environment was constructed in a laboratory setting based on the physical prototype.

5.1 Accuracy Evaluation of Edge Perception Algorithms

This section evaluates the YOLOv8 model's inference accuracy under constrained edge computational resources. Dynamic testing was conducted by injecting water into a simulated tank and placing various vehicle models under diverse lighting conditions (e.g., standard indoor lighting and simulated low-illumination). Experimental results indicate that on the Raspberry Pi platform, the YOLOv8n model's recognition error for the water gauge scale remains stable within ±5 cm, satisfying engineering requirements. For vehicle classification, the implementation of a feature fusion strategy enables high robustness against reflective interference, yielding a comprehensive classification accuracy exceeding 92% for standard vehicle categories. Furthermore, the lightweight deployment ensures that the complete dual-model inference for a single image frame requires approximately 0.8 s at the edge, fulfilling real-time video processing constraints.

5.2 Offline Disaster Recovery and Fault Tolerance Testing

To simulate extreme scenarios where severe storms paralyze communication base stations, the Wi-Fi connection between the Raspberry Pi and the public network router was intentionally disconnected. Test logs confirm that while real-time updates to the Mini Program were interrupted, the edge node—relying entirely on the locally deployed YOLOv8 model—accurately identified the rising water level. Subsequently, it actuated the STM32 controller to trigger the high-frequency flashing red beacon and lower the boom barrier. Upon network restoration, alarm logs cached locally during the disconnection were automatically retransmitted to the cloud via MQTT. This validates the system's robust off-grid resilience and cluster fault tolerance. The robustness of this architecture in congested network environments is further substantiated by the theoretical bandwidth requirements in Table 1.

Table 1 Communication Overhead Comparison: Full-Cloud vs. Cloud-Edge Architecture

Metric	Full-Cloud Architecture	Proposed System	Reduction Rate
Transmitted Content	1080P Video Stream	Structured Detection Data	--
Average Bandwidth	~4 Mbps/node	~2 Kbps/node	> 99.9% ↓
Hourly Data Volume	Approx. 1.8 GB	Approx. 7.2 MB	> 99.5% ↓

6 CONCLUSION

This study demonstrates the efficacy of a cloud-edge collaborative architecture in resolving traditional urban waterlogging monitoring bottlenecks. The core contributions are:
(1) Advancement in Perception Modalities: By replacing traditional physical water level meters with non-contact visual recognition, this system resolves sensor corrosion and failure issues in harsh aquatic environments, extending equipment operational lifecycles [10].
(2) Refinement of Control Logic: Moving beyond uniform static thresholding, this study proposes a dynamic "water level-vehicle type" matching strategy powered by dual YOLOv8 models. This enables differentiated, millisecond-level intersection interventions tailored to specific vehicle wading capacities, bolstering the traffic resilience of the urban road network.
(3) High Architectural Reliability: The proposed architecture restructures computational workloads between the cloud and the edge. By leveraging Raspberry Pi nodes to execute ultra-low latency, offline early warnings, the system exhibits robust disaster recovery and fault tolerance capabilities.
Experiments confirm that the proposed system minimizes response latency, conserves network bandwidth, and optimizes emergency traffic management efficiency. Future work will expand the visual dataset to encompass complex meteorological conditions (e.g., dense fog, heavy snow cover) and investigate solar-powered, off-grid deployment schemes for remote areas lacking mains electricity, aiming for comprehensive smart urban water management coverage.

COMPETING INTEREST

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Tellman Beth, Sullivan Julia, Kuhn Collin, et al. Satellite imaging reveals increased proportion of population exposed to floods. Nature, 2021, 596(7870): 80-86. DOI: 10.1038/s41586-021-03695-w.
[2] Shah Syed, Seker Dursun, Hameed Shariq, et al. Waterlogging and flood management using IoT-based smart edge computing. Water, 2021, 13(8): 1083-1095. DOI: 10.3390/w13081083.
[3] Kaur Kuljeet, Garg Sahil, Kaddoum Georges, et al. Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay. IEEE Communications Magazine, 2021, 59(12): 52-58. DOI: 10.1109/MCOM.101.2100281.
[4] Huang Yufang, Peng Hongtao, So Massoud, et al. The city management based on smart information system using digital technologies in China. IET Smart Cities, 2022, 4(2): 1-15. DOI: 10.1049/smc2.12035.
[5] Terven Juan, Cordova-Esparza Diana. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning with Applications, 2023, 13(1): 100508. DOI: 10.1016/j.mlwa.2023.100508.
[6] Zhang Ming, Liu Jian, Wang Ying, et al. An Internet of Things based system for urban waterlogging monitoring and early warning. Sustainable Cities and Society, 2021, 74(1): 103233. DOI: 10.1016/j.scs.2021.103233.
[7] Wang Guojian, Chen Yu, An Ping, et al. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. Sensors, 2023, 23(16): 7190-7205.
[8] Lähde Hannu, Khadka A, Tahvonen O, et al. Ensuring Ecosystem Service Provision of Urban Water Nature-Based Solutions in Infill Areas: Comparing Green Factor for Districts and SWMM Modeling in Scenario Assessment. Environmental Processes, 2023, 10(4): 61-75. DOI: 10.1007/s40710-023-00678-x.
[9] Li Yuxuan, Ru Xiaohui, Chen Yun, et al. Real-time vehicle detection and tracking in edge computing architecture. Journal of Systems Architecture, 2022, 128(1): 102559. DOI: 10.1016/j.sysarc.2022.102559.
[10] Zhao Xin, Li Ming, Wang Hong, et al. Urban resilience assessment framework and spatiotemporal evaluation of driving forces: A case study of Hubei Province, China. Scientific Reports, 2024, 14(1): 30125.