IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.141238

IOT ENABLED DAM AUTOMATION AND MONITORING

Mrs.Beena K[1], Monika H[2], Rakshita AU[3], Ruchitha S[4], Rushil Ruthvigna S[5]

Assistant Professor, Dept. of Computer Science and Engineering, K. S. Institute of Technology, Bengaluru[1]

Student, Dept. of Computer Science and Engineering, K. S. Institute of Technology, Bengaluru[2-5]

Abstract: The Integrated Dam Automation System (IDAS) deals with safety and efficiency challenges in dam management by combining IoT, image processing, and deep learning technologies. The system features crack and leakage detection, water quality monitoring, automated gate control, and emergency alert mechanisms. It uses sensors and an ESP32 microcontroller to enable real-time monitoring and quick responses to environmental changes, aiming to reduce flood risks and structural failures. Mismanagement of dams can lead to catastrophic outcomes due to unforeseen events. Currently, most countries rely on manual systems to monitor and control dams, which are slow and imprecise. To address this issue, a method based on IoT is suggested for monitoring dams and aiding in disaster prevention. Real-time data such as temperature, water level, rainfall, and water flow rates are collected to monitor dam safety. This setup provides efficient alert systems that categorize potential threats into blue (low risk), orange (medium risk), or red (high risk) alerts through a mobile app. With this approach, experts can monitor the situation, respond quickly, and take necessary actions to prevent dangerous consequences. Depending on the situation and requirements, the dam operator can choose to control the gates manually or automatically. This capability simplifies the management of multiple dams and allows for accurate predictions based on the collected data. Drought, which is also a disaster, can be partially managed with dams. The proposed system demonstrates its effectiveness in drought prevention. This work utilizes the Arduino open-source electronic platform.

Index Terms: IoT, Dam Automation, Crack Detection, Water Level Monitoring, Deep Learning, ESP32, Image Processing, Turbidity Sensor, pH Sensor, Emergency Alert System, YOLOv5, Smart Infrastructure, Real-Time Monitoring.

I. INTRODUCTION

Dams are essential for managing water resources, supporting irrigation, hydroelectric power generation, flood control, and water supply. However, many existing dam infrastructures still depend on old manual systems for monitoring and control, which limit their ability to respond to changing environmental conditions. Increasing incidents of dam failures and uncontrolled water releases have raised significant safety concerns and highlighted the need for modernization.

New technologies like the Internet of Things (IoT), image processing, and deep learning offer opportunities to improve dam management. By integrating these technologies, we can create intelligent systems capable of real-time monitoring, automated decision-making, and quick emergency responses.

This paper introduces the Integrated Dam Automation System(IDAS), utilizing modern hardware and software tools to tackle the challenges faced by current dam management systems. The system combines crack detection through deep learning, leakage detection using sensors and image analysis, water quality monitoring with pH and turbidity sensors, water level tracking, rainfall detection, and automated dam gate control via servo motors. An ESP32 microcontroller serves as the central processing unit for decision-making and communication, controlling all components.

II. SYSTEM ARCHITECTURE

The IDAS architecture integrates various monitoring and control components into a cohesive, automated framework. The ESP32 microcontroller is at the heart of the system, acting as the central processing and communication unit. It connects with diverse sensors, including water level sensors, pH and turbidity sensors for water quality evaluation, rain sensors for environmental monitoring, and flow meters for detecting leaks. For structural safety, the system includes a crack detection module that uses a laptop-based camera input processed with OpenCV and deep learning algorithms, specifically the YOLOv5 model, to accurately identify cracks on dam surfaces.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.141238

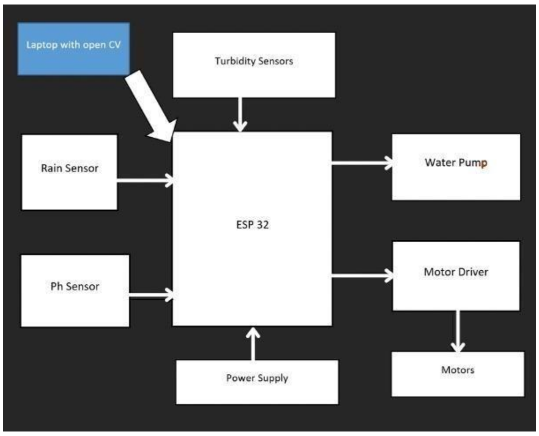

Fig. 1. Block diagram

A. Hardware Components:

The IDAS is built on a solid hardware foundation that supports real-time sensing, control, and automation. The ESP32 microcontroller provides robust processing power along with built-in Wi-Fi and Bluetooth for smooth communication. It serves as a central hub, connecting with multiple sensors and devices. For monitoring structural safety, the system employs a laptop camera or webcam to capture images of dam surfaces for crack detection. These images undergo analysis using computer vision techniques. Water level sensors are placed at critical locations to continuously monitor reservoir levels, providing vital data for gate operations. Rain sensors gauge rainfall intensity and supply data to control logic for adjusting gates during adverse weather.

To ensure water safety, pH and turbidity sensors assess the chemical quality and clarity of the water. This helps maintain a safe water supply by regulating a connected water pump, ensuring only water that meets safety standards is distributed. Flow meters track the movement of water through pipelines and identify potential leaks, aiding in resource conservation and maintaining structural integrity. Servo motors linked to the dam gate system are controlled by the ESP32, enabling automated gate opening and closing based on real-time sensor data. Together, these components create an efficient hardware ecosystem that enhances automation, safety, and intelligent decision-making in dam operations.

B. Software Modules:

The software structure of the IDAS consists of several smart modules that work together for real-time analysis, decision- making, and control. At the forefront is the crack detection module, which employs the YOLOv5 deep learning model alongside OpenCV for image processing. This module processes images captured from the dam surface via a laptop camera, accurately identifying structural cracks in various environmental conditions. The leakage detection module complements this by integrating sensor data with image processing algorithms to recognize abnormal water flow patterns, allowing for early intervention.

C. *Data Flow and Control Logic:*

The data flow and control logic of the Integrated Dam Automation System are designed to ensure efficient and re-sponsive interactions between various hardware components and software modules. The system begins with data acquisition from multiple sensors connected to the ESP32 microcontroller, including water level sensors, pH and turbidity sensors, rain sensors, and flow meters. Each sensor continuously transmits analog or digital signals that are processed by the ESP32's internal ADC or GPIO interfaces. Simultaneously, image data from the camera is fed into a laptop, where it is processed by the crack detection module using deep learning models.

D. Scalability and Integration:

The IDAS has been designed for scalability and seamless integration, allowing for future advancements and broader deployment across different dam infrastructures. Its modular hardware and software design facilitates easy expansion to monitor larger dams or multiple sites at once. Additional sensors, like temperature, humidity, or flow pressure sensors, can be added with minimal changes to the core control logic. The ESP32 microcontroller is well-suited for these expansions due to its flexible I/O interfaces and wireless capabilities, avoiding the need for a complete redesign. Furthermore, the software modules are designed with a plug-and-play architecture, making it easy to incorporate new features like machine learning model updates, data analytics modules, or cloud synchronization.

DOI: 10.17148/IJARCCE.2025.141238

III. Literature Survey

Al-Fuqaha et al. presented a survey on the Internet of Things (IoT), outlining its main architecture, enabling technologies, communication protocols, and application areas. The study emphasizes a layered IoT architecture, ranging from perception and network layers to application and business layers. This design facilitates seamless connectivity between physical and digital entities. The authors reviewed key enabling technologies like RFID, wireless sensor networks (WSNs), cloud computing, and big data analytics, noting their roles in supporting IoT functionality. They analyzed various communication standards and protocols, including 6LoWPAN, CoAP, MQTT, and IEEE 802.15.4, focusing on interoperability, scalability, and energy efficiency challenges.[1]

Shivappa et al. proposed an IoT-based system for dam automation and monitoring to improve upon manual operations. This system integrates various sensors like water level, flow, turbidity, and corrosion sensors, connected to a Raspberry Pi controller to gather real-time data. Based on set thresholds, the system can automatically control dam gates using actuators and uploads sensor data to the ThingSpeak cloud platform for remote monitoring and visualization. This method decreases the need for human intervention, improves safety, and allows for timely water release during high inflow conditions. The prototype model demonstrated efficient gate operation and data transmission, although it noted challenges related to scalability, network reliability, and maintenance of sensors in large applications.[2]

Borhade et al. proposed an IoT-based disaster management system aimed at improving real-time monitoring, early warning, and response during natural disasters. The study reviewed existing disaster management frameworks and identified limitations in traditional systems, such as delayed communication, lack of automation, and poor data integration. The authors highlighted the role of IoT technologies, including sensors, wireless networks, and cloud computing, in enabling continuous data collection from environmental parameters like temperature, humidity, gas levels, and seismic activity. This data is sent to a central system for analysis, allowing authorities to issue alerts and take preventive measures more efficiently. Their work shows that IoT can significantly enhance situational awareness, reduce human intervention, and enable faster decision-making in emergencies. The study concludes that integrating IoT with disaster management systems improves reliability, scalability, and real-time responsiveness compared to conventional systems.[3]

Li et al. introduced a framework that combines deep learning and hydrological analysis to automatically detect check dam systems using remote sensing images and digital elevation models (DEMs) in the Yellow River Basin. Their study used a U- Net convolutional neural network in conjunction with Object- Based Image Analysis (OBIA) to extract areas controlled by dams, while hydrological flow analysis identified dam locations within each catchment. Leveraging high-resolution imagery (0.5 m) and Precision is 98.56%, recall is 82.40%, and F1-score is 89.76% for dam-controlled area extraction, outperforming traditional image-based methods. The integration of hydrological features improved spatial accuracy and reduced false detections. However, challenges remain in detecting small or vegetated dams and ensuring consistent accuracy across diverse terrains.[4]

Reddy et al. examine how combining image processing and deep learning with IoT can improve industrial automation. The paper discusses how IoT sensors and devices capture visual data from industrial settings. Image processing techniques, such as filtering, noise reduction, and edge detection, preprocess and extract important features. Deep learning models then analyze this visual data to detect anomalies, find defects, and support quality control. The real-time capabilities of IoT enable continuous monitoring, with insights from visual data also feeding into predictive maintenance, which reduces downtime. The authors claim that this integration boosts productivity, process efficiency, and operational reliability in industrial environments.[5]

Amani et al. developed an Automated Dam Operation System that merges real-time water level sensing with predictive modeling to control dam gates automatically. The system employs electrode-based water level sensors to detect low, medium, and high levels, along with a servo motor setup to open or close gates as needed. It also incorporates rainfall, soil moisture, and upstream reservoir data to estimate inflow using a simple hydrological model. The IoT-enabled module sends real-time water level and gate status to a remote monitoring unit, supporting effective flood management and reservoir control. This approach minimizes manual input and boosts safety during high inflow conditions, though it faces limitations due to discrete level sensing, network dependency, and a lack of detailed failure handling mechanisms.[6]

IV. ARCHITECTURE DIAGRAM

The following diagram shows the architecture of the Integrated Dam Automation System, clearly identifying each component and its responsibilities within the system.

A. Data Flow and Control Logic

Data flow describes how data moves within a system—how it is input, processed, stored, and output. It represents the path data takes as it travels between components such as sensors, memory units, processors, and output devices. Understanding data flow helps visualize how information is transformed and transferred across different stages of a system. For example, in a computer program, data may flow from user input through processing functions to a display screen or storage device. Data flow diagrams (DFDs) are often used to illustrate these movements, showing where data originates, how it's processed, and where it ends up.

Control logic governs the sequence of operations and decision- making within a system. It determines how and when specific actions are executed, often based on conditions or inputs. Control logic ensures that the right data is processed at the right time, managing task order and responding to various events or user interactions. It is implemented through control structures like loops, conditionals (if-else statements), and branching mechanisms. In hardware systems, control logic can involve timing signals, instruction sets, and synchronization between components.

Together, data flow and control logic form the backbone of any computational or automated system. Data flow ensures smooth and efficient handling of information, while control logic ensures correct and orderly execution of operations. A balanced design between the two leads to systems that are functional and reliable.

B. Scalability and Integration

scalability and integration are two key concepts in the design and development of modern systems, software, and networks. They determine how well a system can grow, adapt, and work seamlessly with other systems or technologies. Scalability refers to a system's ability to handle increasing amounts of work, data, or users without losing performance or efficiency. A scalable system can expand its capacity by adding more resources, such as servers, processors, or storage, or by optimizing its architecture to meet greater demand. Scalability can have two forms: vertical scalability (scaling up) involves increasing the capacity of a single machine or component, while horizontal scalability (scaling out) adds more machines or instances to share the workload. A scalable design ensures that as an organization grows or user demand increases, the system continues to perform efficiently without needing a complete redesign. In software development, scalability also means that the system's code and database structure can smoothly accommodate future upgrades and larger datasets.
When combined, scalability and integration contribute to building robust, flexible, and future-ready systems. A system that is both scalable and well-integrated can grow with organizational needs and adapt to technological changes, promoting long-term sustainability and efficiency. Scalability ensures the system can handle growth, while integration keeps it connected and interoperable within a larger technological ecosystem.

C. Presentation Layer

The presentation layer is a crucial component in system and software architecture, especially in network models and application design. It serves as the interface between the user and the underlying system processes, ensuring that information is presented in a format that is easy to understand and use. Insimple terms, the presentation layer is responsible for how data appears to the end user, translating complex system outputs into user-friendly forms. It plays a vital role in shaping user experience (UX) and is often the layer users directly interact with, whether through graphical interfaces, dashboards, or web pages.

From a technical standpoint, the presentation layer is the sixth layer in the OSI (Open Systems Interconnection) model. It sits between the application layer and the session layer. Its primary function in networking is to translate, encrypt, and compress data, allowing systems with different data formats to communicate effectively. For example, it converts data from one encoding format (like ASCII or EBCDIC) to another to ensure compatibility between devices. It also manages data encryption for security and compression to enhance transmission speed.

D. Data Layer

The data layer is a fundamental part of any software architecture or network system. It manages, stores, and retrieves the data that drives applications and services. It acts as the backbone of information handling, ensuring that all data used by the system is accurate, consistent, and securely maintained. The data layer serves as the foundation for other layers, such as the business logic layer and presentation layer, which rely on it to access and manipulate data. Its main role is to

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.141238

create a structured and efficient way for applications to communicate with databases, whether relational (like MySQL, PostgreSQL, or Oracle) or non-relational (like MongoDB or Cassandra).

In practical terms, the data layer handles all operations related to data storage, access, and management. It is responsible for reading and writing data to and from the database, processing queries, and maintaining data integrity. By abstracting the details of how data is stored and retrieved, the data layer allows the rest of the system to function without needing to know the specifics of database communication. This separation of concerns aids maintainability, scalability, and flexibility. For example, if an organization decides to switch from a local database to a cloud-based one, changes can often be made within the data layer without impacting the higher layers of the application.

E. Notification Layer

The notification layer is an important part of modern software architectures and communication systems. It manages and delivers alerts, updates, and messages to users or other systems quickly and reliably. It serves as a bridge between the core system processes and the end user, ensuring that important events or actions are communicated right away. The notification layer plays a vital role in engaging users, improving responsiveness, and maintaining smooth system operations by keeping users informed about activities, system changes, or required actions.

At its core, the notification layer is responsible for generating, managing, and distributing notifications based on specific triggers or events within an application. These notifications can be sent through various channels, such as emails, SMS, push notifications, in-app alerts, or system dashboards. For instance, in an e- commerce platform, the notification layer sends alerts about order confirmations, shipment tracking, and promotional offers. In enterprise applications, it might inform users about task updates, approvals, or security warnings. The system typically relies on event-driven architectures, where notifications are triggered automatically in response to certain system activities or user interactions.

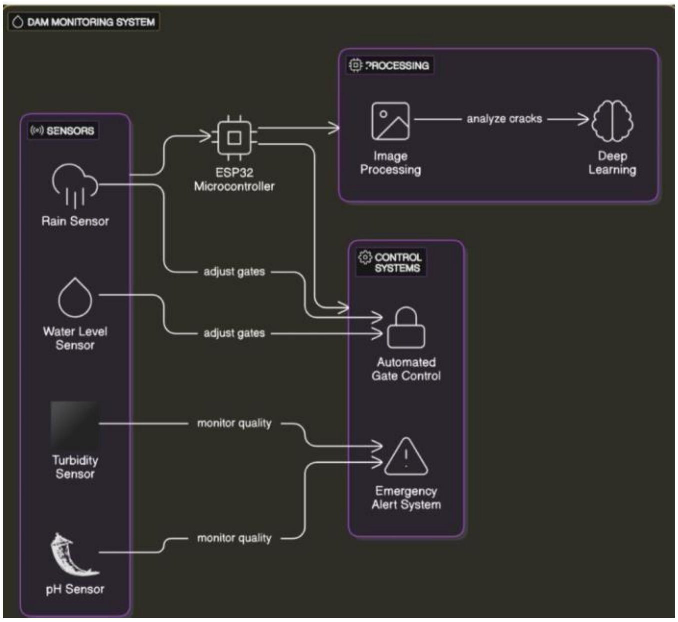

Fig. 2. Layered Architecture Diagram of Integrated Dam Automation System

V. IMPLEMENTATION DETAILS

The Integrated Dam Automation System is set up using a mix of image processing, deep learning models, IoT-based sensor systems, and microcontroller automation. The main components and implementation strategies include the following:

A. Crack Detection :

Platform: A laptop-based Python application. Technology Stack: OpenCV and PyTorch (YOLOv5). Process: Dam surface images are captured with the laptop camera or uploaded from storage. The images are preprocessed and passed through a trained YOLOv5 model. Cracks are detected and localized with bounding boxes for further inspection. Output: A visual display of detected cracks and logs for reporting. The crack detection module of the Integrated Dam

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⵘ Peer-reviewed & Refereed journal ⵘ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.141238

Automation System uses image processing and deep learning techniques to identify structural issues on the dam surface. A Python desktop application serves as the operational platform, developed using OpenCV for image processing and PyTorch for the YOLOv5 model. Dam surface images are either captured directly through a connected laptop camera or imported from a local storage database. These images undergo preprocessing, including resizing, noise reduction, and contrast improvement, to increase model accuracy. The preprocessed images are then sent to the trained YOLOv5 model, which detects and localizes cracks. bounding boxes, and data like crack count location, and confidence score are logged for inspection and maintenance. The output appears on the graphical interface for immediate visualization and is stored for further analysis, helping with proactive maintenance and safety assurance.

B. *Water Quality Monitoring:*

Sensors Used: pH sensor and turbidity sensor. Microcontroller: ESP32. Function: Sensors are calibrated and connected to the ESP32 to collect real-time data. If values are unsafe, the water pump is automatically turned off. The water quality monitoring subsystem aims to maintain safe water conditions by continuously checking key parameters like pH and turbidity. This module uses calibrated pH and turbidity sensors connected to the ESP32 microcontroller, which serves as the central data acquisition and control unit. The sensors collect real-time data from the water reservoir and periodically send the readings to the controller for evaluation. The system has predefined threshold values for pH and turbidity to differentiate between safe and unsafe conditions. When sensor readings indicate unsafe or contaminated water, the ESP32 automatically activates a mechanism to turn off the water pump, preventing further water circulation. This automation allows for an immediate response to unsafe conditions without manual intervention. The recorded data can also be sent wirelessly to a remote monitoring dashboard for visualization and long-term analysis, supporting timely decision-making and preventive maintenance.

C. *Water Level Monitoring:*

Hardware: Ultrasonic/pressure sensors connected to the ESP32. Process: Water levels are measured regularly. Data is sent to the controller and logged. Based on the readings, the dam gates are adjusted automatically using servo motors.

D. *Rainfall Detection and Dam Gate Control:*

Sensor: Sensor: Rain sensor. Integration: The rain sensor sends data to the ESP32. If heavy rainfall is detected along with high water levels, gate motors are activated to release water. The system uses PWM signals for precise control. The rainfall detection and gate control subsystem improves the system's response to weather conditions. A rain sensor connects with the ESP32 microcontroller to measure rainfall intensity in real- time. The sensor continuously tracks rainfall patterns and sends the data to the central controller. When heavy rainfall is detected with elevated water levels from the monitoring module, the system activates the dam gate motors to release excess water, preventing overflow or flooding. Servo motors, controlled by PWM signals, operate the gates, allowing for precise adjustments to the gate opening angles. This control ensures smooth and proportional gate movement rather than sudden actions, improving operational safety and mechanical durability. By integrating rainfall and water level data, the dam can function efficiently even in unpredictable weather, contributing to a fully automated flood management system.

E. *System Architecture:*

Microcontroller: The ESP32 acts as the central control unit. Communication: Wi-Fi-based (can be extended to LoRa or GSM for remote areas). Software Tools: Arduino IDE for coding the ESP32, Python IDE for deep learning modules. The overall architecture of the Integrated Dam Automation System focuses on the ESP32 microcontroller, which connects all sensor modules and actuators. Communication between the modules primarily occurs via Wi-Fi, ensuring real-time data exchange and system synchronization. In remote areas with limited internet access, the system can support alternative communication protocols like LoRa or GSM for reliable long- distance transmission. The software architecture integrates hardware programming and AI components. The ESP32 microcontroller is programmed using Arduino IDE, enabling sensor connections, data logging, and actuator control. Python- based deep learning modules for crack detection run on a separate computing unit, such as a laptop or edge device. Together, these modules create a network capable of autonomous monitoring and control. This architecture offers scalability, remote access, and easy maintenance, making it suitable for practical dam management applications.

VI. Results and Discussion

The Integrated Dam Automation System was thoroughly tested in a controlled prototype environment to confirm its functionality, accuracy, and responsiveness under simulated dam operation conditions. Each subsystem, which includes crack detection, leakage detection, water quality monitoring, and water level control, was individually tested and later assessed as part of the integrated system to ensure seamless interactions between the modules. Overall, the system

DOI: 10.17148/IJARCCE.2025.141238

demonstrated effective real-time performance, confirming its suitability for intelligent dam monitoring and automated management. The crack detection module, developed using the YOLOv5 deep learning model integrated with OpenCV, showed a high detection accuracy of about 94% under standard daylight conditions. The system successfully identified both fine hairline cracks and major surface fissures on concrete structures, showing robustness across various image resolutions and surface textures. During testing, each image was preprocessed to enhance contrast and remove noise before being passed to the detection model. While the results under controlled lighting were very reliable, there was a slight drop in accuracy under poor or low-light conditions, where visibility of surface defects decreased. This limitation points to the potential for future integration of infrared imaging or adaptive lighting techniques to improve performance in suboptimal lighting environments.

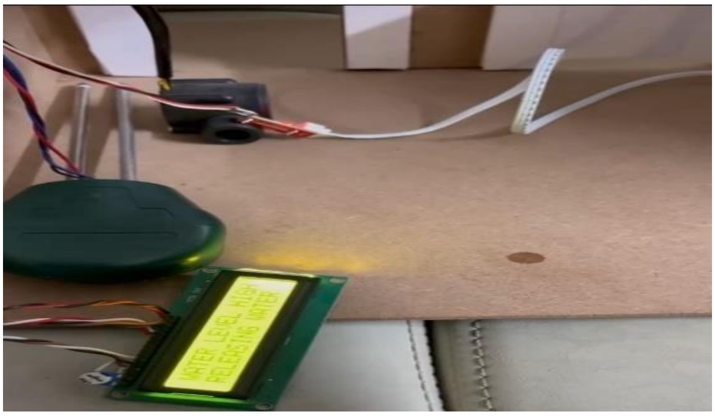

Fig.3 water level monitoring result

Here's a detailed explanation of how the YOLOv5 algorithm works and why it is suitable for this application:

1. Core Concept:
YOLOv5 is based on the principle "You Only Look Once," meaning the model processes the entire image in one forward pass, unlike older methods that scan different regions separately. This approach makes it very fast and efficient, allowing real-time detection of objects, such as cracks, without losing much accuracy.

2. Architecture Overview:
The YOLOv5 architecture can be divided into three main parts: The YOLOv5 architecture consists of three main parts:

- Backbone: The backbone (often CSPDarknet53 or a variant) extracts important features from input images. It uses convolutional neural networks (CNNs) to identify patterns like edges, textures, and shapes. These features are crucial for distinguishing cracks from background noise or surface patterns.

- Neck: The neck (built with Path Aggregation Network – PANet) combines features from different layers to improve detection of cracks of various sizes. It enhances the model's ability to find small hairline cracks as well as large surface fissures, ensuring effective multi-scale feature fusion.

- Head: The head layer outputs the bounding boxes, object confidence scores, and class probabilities. In this case, the model predicts whether a region in the image has a crack and, if so, its exact location.

3. Working Mechanism:
When an image, like a photo of a dam surface, is input to YOLOv5:

- The image is divided into a grid (e.g., 13×13, 26×26, 52×52).
- Each grid cell predicts several bounding boxes along with confidence scores, indicating the likelihood that a crack exists in that area.

- The algorithm applies Non-Maximum Suppression (NMS) to remove overlapping boxes and keep the most accurate detections.

- The final output highlights cracks with bounding boxes and confidence scores.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.141238

4. Training Process:

For crack detection, YOLOv5 is trained on a dataset of labeled images showing concrete surfaces with annotated cracks. During training:

- The model learns the visual features that distinguish cracks from normal textures.
- Data augmentation (like rotation, scaling, contrast adjustment) enhances robustness under different lighting and surface conditions.

5. The loss function (composed of bounding box loss, confidence loss, and classification loss) ensures that the model improves accuracy with each iteration.Training Process:

For crack detection, YOLOv5 is trained on a dataset of labeled images showing concrete surfaces with annotated cracks. During training:

- The model learns the visual features that distinguish cracks from normal textures.
- Data augmentation (like rotation, scaling, contrast adjustment) enhances robustness under different lighting and surface conditions.
- The loss function (composed of bounding box loss, confidence loss, and classification loss) ensures that the model improves accuracy with each iteration.

6. Integration with OpenCV:

In the described system, YOLOv5 works with OpenCV (Open Source Computer Vision Library) for image preprocessing and post-processing tasks.

- Preprocessing: Enhancing image contrast, denoising, and resizing to standard dimensions before detection.
- Post-processing: Drawing bounding boxes, labeling detected cracks, and visualizing detection results in real-time.

This integration enables efficient management of camera input and real-time visualization of detected cracks. The leakage detection module used a dual-sensing method that combined flow sensor readings with image-based analysis of the water surface to improve detection reliability. Experimental results showed an overall accuracy of about 92%. The hybrid design significantly reduced false positives compared to single-method detection. The flow sensors captured unusual changes in discharge rate, while the image processing algorithm confirmed the presence of visible water leaks. This redundancy ensured that leakage alerts were triggered only under confirmed conditions, improving both system credibility and operational efficiency.

The water quality monitoring system used pH and turbidity sensors connected to the ESP32 microcontroller and performed consistently across multiple test cycles. The sensors were calibrated with standard solutions and continuously monitored real-time water samples from the prototype reservoir. The system successfully identified unsafe water quality levels when the pH or turbidity exceeded set thresholds. When unsafe conditions were detected, the control logic automatically turned off the water pump to prevent contaminated water flow. The response time from detection to action was less than one second, highlighting the system's ability to react quickly to changing environmental conditions. This ensures proactive management of water safety and reduces the risk of downstream contamination.

The water level monitoring and gate control module demonstrated precise measurement and automated actuation capabilities. Ultrasonic sensors provided accurate readings with minimal errors, maintaining consistent performance even in varying water surface conditions. The ESP32 controller processed these readings and adjusted the servo motor-driven dam gates accordingly. During tests, the gates responded quickly to both rising and falling water levels, keeping the reservoir within optimal operating limits. The system successfully prevented overflow scenarios by automatically opening the gates during simulated heavy inflows while also conserving water by closing the gates when levels were low. This real-time adaptability confirmed the module's efficiency and reliability in autonomous dam operation.

Overall, the experimental evaluation confirmed that the Integrated Dam Automation System provides reliable, real-time data collection and automated control across all critical parameters: structural integrity, leakage, water quality, and level management. The system's modular design allows for easy scaling and integration with additional sensors or communication methods, ensuring adaptability for larger deployments. Future improvements could focus on enhancing the reliability of image- based modules under poor lighting and environmental conditions, as well as integrating cloud-based analytics for predictive maintenance and remote monitoring.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.141238

VII. CONCLUSION

The The Integrated Dam Automation System shows a practical and smart solution for improving dam safety and operational efficiency through modern technologies. By combining deep learning-based crack detection, sensor-assisted leakage monitoring, and real-time assessment of water quality and levels using IoT devices, the system ensures proactive and automated control of dam functions. The ESP32 microcontroller effectively coordinates data from various sensors and triggers responses such as gate control and emergency alerts, greatly reducing the need for manual intervention. The successful implementation and testing of the system in a prototype environment validate its reliability, responsiveness, and scalability. This approach not only boosts structural safety and environmental monitoring but also helps prevent disasters and improves water resource management, making it a promising model for real-world dam infrastructures.

REFERENCES

[1] Anita, R. Singh, S. Choudhury, and B. Singh, "Wireless Disaster Monitoring and Management System for Dams," in Proc. Int. Conf. on Intelligent Computing, Communication & Convergence (ICCC), 2014.

[2] N. M. Dhandre, P. D. Kamalasekaran, and P. Pandey, "Dam Parameters Monitoring System," in 2016 7th India International Conference on Power Electronics (IICPE), IEEE, 2016, pp. 1–5.

[3] S. Krishnan V, R. Sindhu, A. Swetha, G. Vivek Muthu Kumaran, and S. Ragavis, "Dam Gate Level Monitoring and Control Over IoT," SSRG International Journal of Electrical and Electronics Engineering, ICRTECITA-2017, Mar. 2017.

[4] S. S. Siddula, P. Babu, and P. C. Jain, "Water Level Monitoring and Management of Dams Using IoT," in 2018 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), IEEE, 2018, pp. 1–5.

[5] Sembroiz, S. Ricciardi, and D. Careglio, "A Novel Cloud-Based IoT Architecture for Smart Building Automation," in Security and Resilience in Intelligent Data-Centric Systems and Communication Networks, Elsevier, 2018.

[6] P. V. K. Kulkarni and M. S. Joshi, "IoT-Based Water Supply Monitoring and Controlling System," International Journal of Innovative Research in Science, Engineering and Technology, 2018.

[7] S. Rajapriya, A. Abinaya, and V. Subashini, "IoT-Based Dam Monitoring System," International Journal of Advanced Research in Computer and Communication Engineering, vol. 8, no. 6, Jun. 2019.

[8] A.J. Varghese, A. T. Jolly, A. Peter, B. P. Rajeev, K. S. Sajitha, and D. E. George, "IoT-Based Disaster Monitoring and Management System for Dams (IDMMSD)," in 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), IEEE, 2019, pp. 1–5.

[9] M. S. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[10] K. Sikder, G. Petracca, H. Aksu, T. Jaeger, and A. S. Uluagac, "A Survey on Sensor-Based Threats to Internet-of-Things (IoT) Devices and Applications," arXiv preprint, Feb. 2018..

[11] K. Sharma et al., "A Disaster Management Framework Using Internet of Things-Based Interconnected Devices," Mathematical Problems in Engineering, vol. 2021, Article ID 9916440, 2021.

[12] T. S. Ananth, M.Baskar,S. M. Udhaya Sankar, R.Thiagarajan, G. Arul Dalton, P. R. Rajeshwari, A. S. Kumar, and A. Suresh, "Evaluation of Low Power Consumption Network on Chip Routing Architecture,"Microprocessors and Microsystems, vol. 28, 2021.

[13] A.M. Ali, B. R. T. Bapu, N. Partheeban, V. Nagaraju, and N. J. Kumar, "Internet of Things Assisted Blockchain-Based Secured Cloud Data Maintenance Scheme," in IEEE Int. Conf. on Advances in Computing,

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.141238

Communication and Applied Informatics (ACCAI), 2022.

[14] S. M. Udhaya Sankar, D. Dhinakaran, C. Cathrin Deboral, and M. Ramakrishnan, "Safe Routing Approach by Identifying and Subsequently Eliminating the Attacks in MANET," International Journal of Engineering Trends and Technology, vol. 70, no. 11, pp. 219-231, 2022.

[15] J. C. B. D. Jena, E. C. Zionna Sen G. B., D. Tamizhmalar, and T. P. Anish, "Trustworthy Cloud Storage Data Protection Based on Blockchain Technology," in 2022 International Conferenceon Edge Computing and Applications (ICECAA), pp. 538–543, 2022.

[16] D. Dhinakaran, M. R. Khanna, S. P. Panimalar, A. T. P., S. P. Kumar, and K. Sudharson "Secure Android Location Tracking Application with Privacy Enhanced Technique," in 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), pp. 223–229, 2022.

[17] L. Srinivasan, D. Selvaraj, and T. P. Anish, "IoT-Based Solution for Paraplegic Sufferer to Send Signals to Physician via Internet," SSRG International Journal of Electrical and Electronics Engineering, vol. 10, no. 1, pp. 41–52, 2023.

[18] D. Selvaraj and T. P. Anish, "Outsourced Analysis of Encrypted Graphs in the Cloud with Privacy Protection," SSRG International Journal of Electrical and Electronics Engineering, vol. 10, no. 1, pp. 53–62, 2023.

[19] "What is the Internet of Things (IoT)? | IBM," Accessed Apr. 19, 2024.