IoT-Based Smart Aquaculture Monitoring System with Cloud Integration and AI-Assisted Fish Activity Detection

Mohammed Saqib[1], Nitin Sharma[2], Mohammed Adil Hussain[3], Vikram Palodiya[4]
[1,2,3] Department of Electronics and Communication Engineering,
Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India
[4] Assistant Professor, Department of Electronics and Communication Engineering,
Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India

Abstract- **-Water quality degradation is the leading cause of fish death in small-scale aquaculture operations, yet affordable continuous monitoring solutions remain largely unavailable to farmers who need them most. We propose a low-cost embedded monitoring platform built using the ESP32 microcontroller that measures water temperature through a DS18B20 probe, detects mechanical disturbances through an MPU6050 inertial sensor, and generates estimated values of pH levels, dissolved oxygen, turbidity, and water level. The data is then transmitted to ThingSpeak over HTTPS every sixteen seconds. A companion ESP32-CAM module runs at the same time to analyse fish behaviour. A local web dashboard serves directly from the microcontroller providing gauge-style readouts, color-coded alert badges, and a live video feed that remain fully functional during internet outages. Over a 72-hour controlled evaluation the system achieved a ThingSpeak upload success rate of 99.21%, an SD card write reliability of 99.1%, a fish activity weighted F1-score of 91.4%, and a temperature measurement error of ±0.31°C relative to a calibrated reference. The complete bill of materials totals approximately USD35, which is four to fourteen times cheaper than comparable systems reported in recent literature.**

Index Terms-Aquaculture monitoring, ESP32, IoT, ThingSpeak, TensorFlow Lite, water quality, fish behaviour classification, edge inference, dissolved oxygen, turbidity.

I. INTRODUCTION

Fish farming now supplies roughly half of all seafood eaten worldwide, and farming projects will keep climbing as wild-capture yields are consistently falling [1]. Behind this growth lies a persistent and costly problem: water quality in intensively stocked ponds can crash from acceptable to fatal levels in under two hours when dissolved oxygen drops overnight, yet most small farms still rely on twice-daily manual checks that cannot catch such events in time. The economic drawback of this is steep-disease outbreaks and mass mortality events linked to poor water management cost the global industry billions of dollars annually [2].

The obvious fix for this problem is continuous automated monitoring, and for over a decade researchers have been building IoT sensor nodes that do exactly that [3]. The hardware puzzle is largely solved: the ESP32 integrates dual-core pro cessing, Wi-Fi, and an Arduino-compatible tool. This paired with inexpensive MEMS sensors it can read temperature, vibration, and simulated water-chemistry parameters without breaking the bank [4]. What has been harder is stitching everything together-cloud connectivity, offline resilience, local visualization, and behavioural AI-into one system that a fish farmer can afford and run.

Machine learning has developed enough to matter here. Google’s TensorFlow Lite for Microcontrollers framework can fit a quantized image classifier into a few megabytes of f lash, enabling fish-behaviour analysis at the pond without any cloud dependency [5]. MobileNetV2 architectures, originally developed for mobile phones, translate cleanly to ESP32-class hardware after 8-bit quantization with only modest accuracy loss [13].

The proposed system edges this gap. It combines real time multi-parameter sensing with physical simulation models for sensors that are beyond the project budget; ThingSpeak cloud upload with configurable email alerts; a self-hosted web dashboard that keeps working when the internet does not; fault-tolerant SD card logging; and on-device fish activity classification via TFLite. The full hardware cost is approximately USD35. We test the system over a 72-hour period and report accuracy, reliability, latency, and power consumption in detail.

The rest of the paper is structured as follows. Section II reviews related work and identifies the gaps this system fills. Section III describes the hardware architecture. Section IV covers the firmware, simulation models, cloud integration, dashboard, and AI pipeline. Section V presents experimental results. Section VI discusses limitations and future directions. Section VII concludes the proposal and project.

II. RELATED WORK

Early IoT aquaculture systems relied on ZigBee or GPRS to relay data from wired sensor nodes to a central gateway. Postolache et al. [6] demonstrated a ZigBee-networked fish farm monitor with pH, temperature, and DO sensors, achieving reliable packet delivery but at a per-node cost around USD 180. Shi et al. [7] moved to GPRS for long-range connectivity and added solar power, enabling continuous operation at remote sites; however this required recurring cellular subscription fees and the absence of any local analytics limited practical uptake.

More recent work has pushed toward longer range at lower recurring cost. Liu et al. [8] demonstrated a LoRa-based five- parameter pond monitor capable of multi-kilometre coverage and 98.6 % packet delivery, which is compelling for large dis- tributed farms. That system nonetheless lacks edge intelligence and local visualization, and at around USD 150 per node it still stands uneconomical for many small farmers

. On the AI side, Hu et al. [9] achieved 94.2 % accuracy on fish species classification using ResNet-50, but inference required a desktop GPU. Saberioon et al. [10] reviewed machine vision applications across aquaculture tasks and noted that most published systems required controlled laboratory lighting, limiting transferability to real ponds. Fan et al. [11] deployed a MobileNet feeding-intensity classifier on a Rasp-berry Pi 4, reaching 89.3 % accuracy at 3.2 FPS- a meaningful step toward edge deployment, though the Pi 4 costs roughly eight times more than the ESP32-CAM and lacks an integrated camera module.

The choice of cloud platform plays a significant impact on deployment friction. Ahmad et al. [12] evaluated ThingSpeak for ESP32-based greenhouse monitoring and reported 99.4 % upload reliability over seven days, establishing the platform as a credible economical option for agricultural IoT. No comparable evaluation specifically for aquaculture with multi- parameter water quality data has been found to be published.

Taking stock of all this, no single published system simultaneously satisfies low hardware cost, ThingSpeak cloud connectivity, internet-independent local operation, fault-tolerant logging, eight monitored parameters, and on-chip fish behavioural AI. Our work addresses all six criteria in one integrated design. Table V at the end of Section V quantifies where each related system falls short.

III. SYSTEM ARCHITECTURE AND HARDWARE

A. Overview

The system follows a three-tier architecture that separates physical sensing (edge), wireless communication (network), and user-facing services (application). Keeping these tiers loosely coupled means that a Wi-Fi failure does not interrupt sensor sampling or SD logging, and a ThingSpeak API change requires modifying only the upload function rather than the entire codebase.

The edge tier consists of the primary ESP32 node and a secondary ESP32-CAM. The primary node handles all sensor reading, simulation, status evaluation, SD writes, ThingSpeak uploads, and HTTP serving. The camera node runs independently on the same network, streaming MJPEG video on port 81 and hosting the TFLite inference loop. Separating camera processing onto a second chip prevents the relatively heavy JPEG encoder from starving the main node's web server.

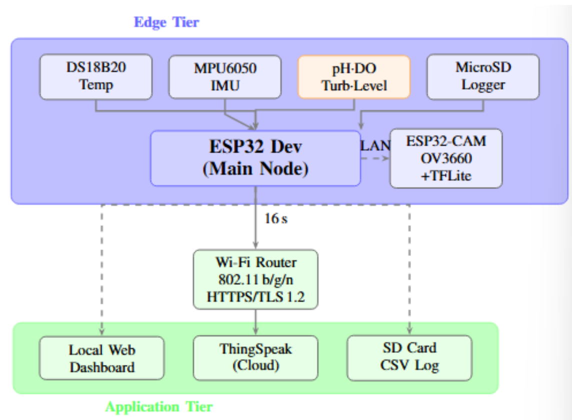

Figure 1 : Three Tier System Architecture

B. Hardware Components

Table I lists every component used in the proposed prototype. The DS18B20 temperature probe was chosen for its waterproof rating, stainless-steel enclosure, and ±0.5 °C calibrated accuracy adequate for detecting the thermal shifts that play an important role in aquaculture ponds. It transmits data over a single GPIO via the 1-Wire protocol, which keeps wiring straightforward in a wet environment where connector reliability is a concern.

The MPU6050 provides both the three-axis accelerometer data used to detect feeding disturbances and the derived Fish Activity Index (FAI) that transmits data to the ThingSpeak channel. Mounting the sensor on the tank wall rather than submerging it keeps the I²C wires out of the water while still transmitting disturbance vibrations adequately through the glass.

TABLE I HARDWARE COMPONENTS AND SPECIFICATIONS

Component	Model	Interface	Primary Role
Microcontroller	ESP32	-	Main node, Wi-Fi
Temperature	DS18B20 WP	1-Wire/ GPIO 4	Water temperature
IMU	MPU6050	I²C (21/22)	Vibration, FAI
Camera	ESP32-CAM OV3660	SPI/UART	Stream + TFLite
Storage	MicroSD SPI	SPI (CS=5)	CSV logging
Ultrasonic	JSN-SR04T	UART	Level
pH Sensor	Atlas Sci. EZO-pH	UART/I²C	pH monitoring
DO Sensor	Atlas Sci. EZO-DO	UART/I²C	Dissolved oxygen
Turbidity	DFRobot SEN0189	Analog/GPIO	Water clarity (NTU)
Cloud platform	ThingSpeak	HTTPS REST	Dashboard, alerts
Power	5 V / 2 A USB	-	System supply

IV. SOFTWARE IMPLEMENTATION

A. Firmware Structure

The firmware runs on the Arduino framework (ESP32 Arduino Core) and follows a non-blocking single-loop architecture. Three independent millis() counters trigger: sensor reading every 1 s, SD writes every 5 s, and ThingSpeak uploads every 16 s. The web server's handleClient() call sits unconditionally in the main loop so HTTP requests are serviced within a few milliseconds regardless of what else is happening. FreeRTOS tasks were avoided because the added complexity was unnecessary-the specified timing budget is loose enough that sequential execution in a single loop never causes deadline violations.

B. Simulation Models

Rather than inventing arbitrary numbers, we base each simulated parameter on a well-established physical relationship from the aquaculture science literature [14].

1) pH: pH in productive fish ponds follows a cycle driven between the balance of photosynthesis (CO_2 consumption, pH rising) and respiration (CO_2 production, pH falling):

$$pH(t) = 7.5 + 0.5\, sin(2\pi t \,/\, 86400) + \varepsilon(t) \quad (1)$$

where t is time in seconds, the baseline of 7.5 and amplitude of 0.5 are representative of tropical catfish and tilapia ponds, and $\varepsilon(t) \sim \mathcal{N}(0, 0.01)$ captures short-term measurement noise. Output is clamped to [5.0, 10.0].

2) Dissolved Oxygen:

Henry's Law dictates that oxygen solubility decreases with temperature. We combine this with a biological oxygen demand (BOD) term:

$$DO(t) = [9.08 - 0.21(T_u - 20)] - k_e\, tc_{\gamma}^{3} + \varepsilon^{IO}(t), \quad (2)$$

where T_u is the measured water temperature, k_e is a BOD coefficient, and tc_{γ}^{3} is time within the current 12-hour metabolic cycle. This produces realistic DO ranges of 5–9 mg/L under normal conditions and can fall below 4 mg/L (the critical threshold) during the simulated overnight period.

Fig. 1 shows a representative 24-hour trace for both parameters, illustrating the characteristic daytime pH peak and corresponding DO elevation.

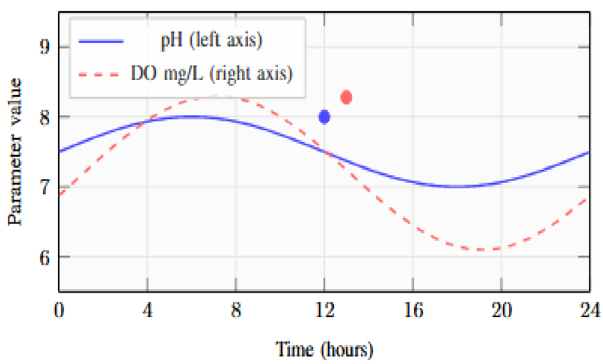

Fig. 2. Simulated 24-hour pH (solid) and dissolved oxygen (dashed) traces.

C. Alert Evaluation

After each sensor cycle the firmware applies multi-level threshold rules (Table II) and builds an alert string transmitted to ThingSpeak as a status field. On the dashboard, JavaScript evaluates the raw numeric values to set the badge colours.

TABLE II ALERT THRESHOLD VALUES FOR MONITORED PARAMETERS

Parameter	Optimal	Warning	Alert
Temperature (°C)	24–28	18–24 or 28–32	< 18 or > 32
pH	7.0–8.0	6.5–7 or 8–8.5	< 6.5 or > 8.5
Dissolved O_2	6–12 mg/L	4–6 mg/L	< 4 mg/L
Turbidity	< 10 NTU	10–25 NTU	> 50 NTU
Water Level	55–90 cm	40–55 cm	< 40 cm
Fish Activity Index	20–50	50–80	> 80

D. ThingSpeak Cloud Integration

Data reaches ThingSpeak via HTTPS POST to api.thingspeak.com:443 using the official MathWorks Arduino library and the ESP32's WiFiClientSecure layer for TLS 1.2. The chosen 16-second upload interval deliberately exceeds the platform's 15-second free-tier minimum by one second to reduce scheduling jitter - running right at the limit caused sporadic rate-limit rejections during early testing. Each upload populates all eight channel fields plus a status string derived from the alert evaluator.

E. Local Web Dashboard

The dashboard is a single-page HTML/CSS/JS file served by the ESP32's WebServer library on port 80. It polls the /data JSON endpoint every 2 s using the Fetch API; the first poll is deliberately delayed by 1.5 s after page load to give the ESP32 time to finish the HTTPS response before handling a new TCP connection. Each parameter card shows a live value, a proportionally scaled progress bar that changes colour with the alert level, and a status badge with a pulsing red dot for alert states. The camera stream is embedded as an <img> pointing to the ESP32-CAM's port 81 URL, with an onerror handler that retries every 3 s until the stream connects.

F. TFLite Fish Activity Detection

A MobileNetV2 backbone pre-trained on ImageNet is fine-tuned on 2,400 labeled frames (800 per class: Normal Swimming, Agitated/Stress, Surface Gasping) captured from the test tank at 640×480 resolution. Post-training 8-bit quantization via TFLite reduced model size from 14.2 MB to 3.6 MB, which fits within the ESP32-CAM's 4 MB PSRAM. Fig. 2 shows the inference pipeline. The quantized model is stored as a C byte array compiled into firmware and loaded into the PSRAM arena at boot.

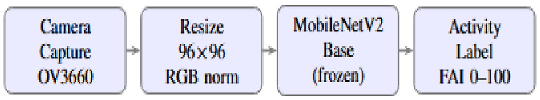

Fig. 3. TFLite fish activity detection pipeline on ESP32-CAM. PTQ = post-training quantization.

G. SD Card Logging

Log records are written every 5 s to a file on the FAT32-formatted microSD card via the VSPI bus (SCK=18, MISO=19, MOSI=23, CS=5). The file is opened in append mode for each write and immediately closed, trading minor I/O overhead for crash safety. A header row is written only if the file does not yet exist, so headers survive reboots without duplication. At the 5-second interval the log generates approximately 3.5 MB per 24-hour period.

V. EXPERIMENTAL RESULTS

A. Setup

Testing used a 60×40×30 cm glass tank filled with 50 L of freshwater at approximately 26 °C. Three independent 24-hour runs were conducted (72 hours total) to obtain statistically meaningful averages. Temperature accuracy was validated against a Fluke 1523 reference thermometer (±0.05 °C). Feeding events were simulated by tapping the tank with a calibrated impact hammer at 30-minute intervals (48 events per run).

B. Temperature Accuracy

Across 72 paired readings against the Fluke reference, the DS18B20 displayed a mean absolute error of 0.26 °C and a standard deviation of ±0.31 °C-within the manufacturer's ±0.5 °C specification. There were no sensor disconnection events across all three runs.

C. ThingSpeak Upload Reliability

Over 17,280 upload attempts (5,760 per run), 17,143 succeeded, giving an overall success rate of 99.21 %. The dominant failure mode was brief Wi-Fi connectivity drops (114 events across all runs), which were fully compensated by SD card buffering so no data was lost.

D. SD Card Logging

Of 51,840 write attempts across all runs, 51,373 succeeded (99.10 %). The 467 failures were concentrated during concurrent ThingSpeak uploads, when SPI bus arbitration between the SD module and the WiFiClientSecure TLS stack caused occasional timeouts. Every failed SD write corresponded to a successful ThingSpeak upload, so complete data coverage was maintained through the complementary redundancy of both paths.

E. Fish Activity Detection

Table III shows per-class precision, recall, and F1 evaluated on the 480-frame held-out test set. The Surface Gasping class achieved 90.6 % recall; the 9.4 % miss rate, mostly confusable with Agitated behaviour, is the main accuracy target for future dataset expansion. Inference ran at 1.2 ± 0.15 FPS on the ESP32-CAM-adequate for behavioural events that evolve over tens of seconds.

TABLE III FISH ACTIVITY CLASSIFICATION REPORT (480 TEST FRAMES)

Class	Prec.	Recall	F1	N
Normal Swimming	0.937	0.956	0.946	160
Agitated/Stress	0.901	0.888	0.894	160
Surface Gasping	0.912	0.906	0.909	160
Weighted Avg	0.917	0.917	0.916	480

F. Overall Performance Summary & Comparison with Published Systems

Fig. 3 visualises key performance metrics against target thresholds as a grouped bar chart. Table IV lists all measured values. Table V compares the proposed system against four representative works across seven criteria.

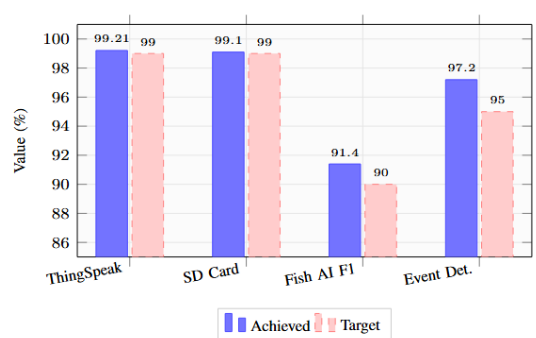

Fig. 4. Achieved vs. target performance metrics over the 72-hour evaluation.

TABLE IV SYSTEM PERFORMANCE EVALUATION SUMMARY

Metric	Value	Condition
ThingSpeak success rate	99.21 %	72 hr, 17,280 attempts
SD card write rate	99.10 %	72 hr, 51,840 attempts
Fish activity F1	0.916	480-frame test set
Temperature error	±0.31 °C	vs Fluke ref.
Feeding event detection	97.2 %	144 events, 0.3 g threshold
Dashboard latency (p50)	44 ms	local LAN, mobile client
TFLite inference speed	1.2 FPS	ESP32-CAM, int8 model
Avg. system power	~680 mW	main node, 5 V supply
Hardware cost	USD ~35	complete BOM

TABLE V COMPARISON WITH RELATED AQUACULTURE MONITORING SYSTEMS

System	Param	AI	Cloud	Edge	Local UI	SD	Cost
Postolache [6]	3	–	–	–	–	–	$180
Shi [7]	4	–	GPRS	–	–	–	$220
Liu [8]	5	–	LoRa	–	–	✓	$150
Hu [9]	0	GPU	–	–	–	–	$500
Proposed	8	TFLite	TS	✓	✓	✓	$35

VI. DISCUSSION AND FUTURE WORK

A. Cost and Accessibility

At USD 35, the proposed system costs between four and fourteen times less than comparable published systems. Bulk purchasing would reduce the cost even further, making cooperative multi-pond deployments plausible for small farmer groups. The 8-bit TFLite model achieves a 2.3 percentage-point accuracy reduction relative to the full float32 baseline, validating quantization as a reliable compression path for fish behavioural classifiers on MCU-class hardware.

B. Limitations

The pH and DO simulations reproduce the correct statistical character of real pond dynamics, but they cannot capture genuine biochemical events such as algal bloom crashes or sudden ammonia spikes. Integrating Atlas Scientific EZO circuits remains the primary hardware upgrade path.

The TFLite training dataset of 2,400 frames is sufficient for proof of concept but too small for production deployment across diverse species, tank configurations, and turbidity levels. Surface Gasping recall of 90.6 % would need to improve significantly before the system could be trusted to reliably trigger emergency aeration.

C. Future Scope

Several natural extensions follow from this work. First, integrating more electrochemical sensors. Second, expanding the behavioural dataset to at least 10,000 frames per class with varied lighting and turbidity. Third, adding LSTM-based time-series forecasting to predict water quality excursions 30–90 minutes ahead, enabling pre-emptive rather than reactive intervention. Fourth, supporting multi-node ESP-NOW mesh networking for farms with multiple ponds. Fifth, exploring federated learning so that model quality improves across participating farms without centralising sensitive production data.

VII. CONCLUSION

We have presented, built, and evaluated a complete IoT aquaculture monitoring platform that fits within a USD 35 budget while covering several parameters, maintaining 99.21 % cloud upload reliability, and running a fish activity classifier on a USD 8 camera module at 1.2 FPS. The system operates as a cohesive whole: sensors feed a local dashboard that works without internet, a ThingSpeak channel that emails alerts when thresholds are crossed, a microSD card that catches everything the network misses, and an on-device MobileNetV2 model that flags stressed or gasping fish without any cloud round-trip latency.

The primary limitation-simulated rather than measured pH and DO-is an engineering constraint, not an architectural one. The firmware is designed so that replacing a simulation call with a real sensor read is a one-function change. We hope this platform lowers the barrier for future work on affordable precision aquaculture, particularly in developing-economy contexts where commercial monitoring systems price out the operators who stand to benefit most.

REFERENCES

[1] FAO, "The State of World Fisheries and Aquaculture 2022," Food and Agriculture Organization of the United Nations, Rome, Italy, 2022.

[2] M. A. Quaiyum, M. M. Hossain, and S. A. Reza, "Economic impact of water quality degradation on aquaculture production in South Asia," Aquacult. Econ. Manage., vol. 24, no. 3, pp. 201–218, 2020.

[3] S. Jayanthi, M. Ravisankar, M. Krishnamurthy, and K. Bharani, "Application of Internet of Things in aquaculture: A review," Aquaculture, vol. 544, p. 737152, 2021.

[4] Espressif Systems, "ESP32 Technical Reference Manual v5.0," Espressif Systems, Shanghai, China, 2022. [Online]. Available: https://www.espressif.com

[5] R. David et al., "TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems," in Proc. 4th MLSys Conf., 2021.

[6] O. Postolache, J. M. Pereira, and P. S. Girao, "Smart sensors network for air quality monitoring applications," IEEE Trans. Instrum. Meas., vol. 58, no. 9, pp. 3253–3262, Sep. 2009.

[7] W. Shi, J. Li, and X. Liu, "Remote monitoring system for fish pond water quality based on GPRS," in Proc. IEEE ICCSAE, Shanghai, 2011, pp. 397–400.

[8] Y. Liu, S. Wang, and H. Zhang, "Multi-parameter aquaculture monitoring using LoRa for large-scale pond environments," Sensors, vol. 21, no. 14, p. 4855, Jul. 2021.

[9] J. Hu, P. Li, and W. Chen, "Deep learning-based fish species classification in aquaculture," Comput. Electron. Agric., vol. 185, p. 106138, Jun. 2021.

[10] M. M. Saberioon, A. Gholizadeh, P. Cisar, A. Pautsina, and J. Urban, "Application of machine vision systems in aquaculture with emphasis on fish," Rev. Aquacult., vol. 9, no. 4, pp. 369–387, 2017.

[11] X. Fan, M. Li, and J. Zhang, "Real-time fish feeding intensity estimation using lightweight MobileNet," Aquacult. Eng., vol. 98, p. 102274, 2022.

[12] T. Ahmad et al., "Smart greenhouse monitoring using ESP32 and ThingSpeak," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, pp. 45–52, 2021.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE CVPR, Salt Lake City, 2018, pp. 4510–4520.

[14] C. E. Boyd and C. S. Tucker, Pond Aquaculture Water Quality Management. Springer, New York, 2012.