Synthesis: A Multidisciplinary Research Journal
Volume 2, Issue 4, October - December 2024, Pp: 11-20

Research Paper

Attention-Driven CNN–LSTM Framework for Multi-Behavior Recognition in Underwater Aquaculture Systems

[1*] K Samunnisa, [2] Ch. Suneetha

[1*] *Assistant Professor, Department of CSE, Ashoka Womens Engineering College, Kurnool, Andhra Pradesh, India*
Email: samunnisa14@gmail.com
[2] *Assistant Professor, Department of CSE, Vignan's Institute of Engineering for Women, Visakhapatnam, India*
Email: maanash11@gmail.com

**Corresponding Author(s): samunnisa14@gmail.com*

Article Info
Received:06/08/2024
Revised: 03/10/2024
Accepted:20/12/2024
Published:31/12/2024

Abstract

Automated recognition of fish behavior is essential for advancing sustainable aquaculture by enabling continuous monitoring of fish welfare, feeding patterns, and stress indicators. Conventional approaches, reliant on manual observation or static vision models, lack temporal modeling capabilities and fail under complex underwater conditions. This study introduces an attention-driven deep learning framework that integrates spatial and temporal cues for real-time multi-behavior recognition from underwater video streams. The proposed architecture employs a dual-stream approach comprising a ResNet-18-based spatial encoder and an optical flow-guided motion stream, whose outputs are fused and processed by a bi-directional Long Short-Term Memory (Bi-LSTM) network enhanced with attention mechanisms to capture sequential dependencies. A custom-labeled dataset encompassing eight fish behavior classes was developed, incorporating environmental diversity such as turbidity, lighting variations, and overlapping fish. The model was optimized for edge deployment through quantization and pruning, achieving a compact footprint suitable for the NVIDIA Jetson Nano platform. Experimental evaluation demonstrates a classification accuracy of 92.4% and a macro-F1 score of 91.1%, with a sustained inference rate of 28 FPS and power consumption limited to 4.9 W. The system outperforms baseline models including YOLOv7-tiny and 3D-ResNet18, affirming its suitability for real-time, resource-efficient aquaculture monitoring. These findings highlight the potential of attention-driven spatial–temporal models in enabling intelligent, scalable, and low-power behavior analysis in real-world aquaculture environments.

Keywords: Fish Behavior Recognition, CNN–LSTM Architecture, Attention Mechanism, Underwater Video Analysis, Aquaculture Monitoring, Edge Computing.

1. Introduction

The global aquaculture industry has witnessed significant growth over the last decade, emerging as a vital sector for ensuring sustainable seafood production and food security. Effective management of fish health and behaviour in aquaculture systems plays a crucial role in maximizing productivity, reducing mortality rates, and ensuring ecological balance. In particular, understanding and monitoring fish behaviour in real-time is critical for early detection of stress, disease, feeding inefficiencies, and environmental imbalances. Traditionally, aquaculture farms have relied on manual inspection or rudimentary video surveillance, which are labour-intensive, error-prone, and inadequate for large-scale, continuous operations. This scenario calls for the integration of intelligent, automated behaviour analysis systems that can function reliably under diverse aquatic conditions.

Real-time fish behaviour monitoring from underwater video streams presents a complex problem due to several

inherent challenges. These include the dynamic nature of underwater environments, occlusion and overlapping of fish, variable lighting, and the presence of turbidity and debris in water. Moreover, fish exhibit a wide range of subtle behaviours that often appear visually similar but differ contextually, requiring fine-grained analysis over both spatial and temporal dimensions. While traditional image processing and feature-engineering techniques have been employed for behaviour detection, they struggle with generalization and adaptability across varying aquatic setups. Recent advancements in deep learning have shown promise in capturing intricate patterns from large-scale visual data, offering new avenues for robust and automated fish behaviour recognition [1]–[4].

Despite these advancements, several gaps remain in the current body of research. First, many existing systems are offline or near-real-time, lacking the capability for continuous monitoring and instant feedback necessary for decision-making in modern aquaculture. Second, numerous approaches treat behaviour detection as a frame-wise classification problem, failing to account for temporal dynamics that are essential for accurately identifying behaviours like schooling, aggression, or abnormal motion. Third, deep learning models often demand high computational resources, limiting their deployment in resource-constrained edge environments typically found in aquaculture farms. Moreover, datasets used in prior works often lack diversity in fish species, behaviour types, or environmental conditions, affecting model generalizability.

This study aims to address these gaps by proposing a real-time deep learning framework for fish behaviour analysis using underwater video streams captured from aquaculture tanks. The approach leverages a two-stream architecture combining convolutional neural networks (CNNs) for spatial feature extraction and recurrent neural networks (RNNs), specifically Long Short-Term Memory (LSTM) units, for modeling temporal dependencies. To enable real-time inference and energy efficiency, the model is optimized for deployment on edge devices such as NVIDIA Jetson Nano or Raspberry Pi 4. In addition, a custom-labelled dataset covering diverse behaviours like feeding, aggression, circling, and idle states is created from continuous video footage, incorporating variations in lighting, water quality, and fish size. This not only enhances behavioural classification accuracy but also improves the ecological and operational decision-making processes in aquaculture systems.

Recent studies have highlighted the potential of deep learning in fish behaviour classification. For instance, spatial feature extraction using CNNs such as YOLO and ResNet has significantly improved behaviour detection performance under complex underwater environments. Techniques involving optical flow and RGB fusion have shown to boost accuracy for motion-specific behaviours. However, their reliance on high-end GPUs limits their scalability in real-time applications. On the other hand, cost-effective models using low-resolution imagery combined with low-cost edge devices have also emerged, though they sacrifice accuracy for efficiency. Literature reviews on the subject underscore the need for hybrid models that strike a balance between computational efficiency and behavioural recognition precision. Furthermore, approaches that incorporate environmental stressors, such as changes in ammonia levels, show how behavioural patterns vary under different stimuli and call for adaptive models [5].

To enable practical deployment, distributed and edge-based frameworks have been explored to reduce network latency and power consumption. These frameworks highlight the importance of localized processing in underwater or remote aquaculture setups [6]. The introduction of spatiotemporal attention mechanisms has also demonstrated improved performance in behaviour sequence modeling, particularly in feeding behaviour analysis of schooling fish [7]. Despite these advancements, integrating all these elements into a unified, real-time, and scalable framework remains an unsolved challenge [8].

This research contributes to the body of knowledge through the following key advancements:

- *Real-Time Multimodal Architecture***:** The proposed system integrates spatial feature extraction and temporal behaviour modeling using a hybrid CNN-LSTM architecture optimized for real-time deployment on edge devices. It outperforms traditional frame-by-frame classifiers by capturing behaviour sequences over time.
- *Diverse and Annotated Dataset:* A custom dataset with annotations of multiple fish behaviours across varied aquatic conditions is developed, filling a significant gap in the availability of open-source, behaviour-specific underwater datasets.
- *Efficient Edge Deployment:* The model is quantized and pruned for deployment on energy-efficient devices without sacrificing classification accuracy, making it suitable for large-scale and continuous fish behaviour monitoring in commercial aquaculture environments.

The remainder of this paper is organized as follows. Section II reviews the related literature and highlights existing methodologies and their limitations. Section III presents the proposed system architecture and the data acquisition pipeline. Section IV details the deep learning methodology and model optimization techniques. Section V explains the experimental setup, evaluation metrics, and results. Section VI discusses the findings, limitations, and implications for future research. Finally, Section VII concludes the paper with a summary of contributions and directions for further development.

2. Related Work

Real-time fish behaviour analysis in aquaculture has become an active area of research, particularly with the convergence of deep learning, edge computing, and smart surveillance. This section critically reviews the current state-of-the-art approaches used for behaviour recognition, anomaly detection, and feeding assessment in aquaculture systems, identifying key methodological trends, performance benchmarks, and technical limitations.

2.1 Machine Vision-Based Fish Behaviour Classification

Early attempts in automated fish monitoring systems focused on rule-based and machine vision techniques. Sagstad [9] used classical image processing for characterizing salmon behaviour in rearing tanks. Although computationally inexpensive, these systems lacked robustness in real-world aquatic settings due to poor generalization under low-light or occlusion conditions.

Later, Wang et al. [10] applied conventional AI techniques combined with support vector machines and handcrafted features to detect anomalous underwater behavior. While they reported promising performance on controlled datasets, their approach suffered a drop in detection accuracy (down to 68%) when tested on real-world video streams due to noise and fish overlap.

2.2 Deep Learning for Feeding and Aggressive Behavior Detection

Deep learning has significantly improved fish behavior classification, particularly for detecting feeding intensity and aggression. For example, Su et al. [11] utilized convolutional neural networks (CNNs) to analyze feeding events through motion patterns, achieving a classification accuracy of 91.2%. However, their system was not optimized for real-time operation, leading to high inference latency on full HD underwater videos.

Wang et al. [12] developed a lightweight CNN model for rapid identification of cannibalism behavior among juvenile fish. Their approach attained 92.5% detection accuracy, but required frequent retraining when applied to different species or lighting environments.

Similarly, José [13] implemented a spatiotemporal tracking pipeline for aggressive behavior recognition in rainbow trout using 3D pose estimation. While behavior precision improved to 87.6%, the system required high-resolution video and external tracking markers, which are not scalable in production aquaculture.

2.3 Edge-Based Real-Time Behavior Monitoring

With the growing need for low-latency, on-site processing, edge-based architectures have gained attention. Cao et al. [14] proposed an energy-efficient aquaculture monitoring system using YOLOv4-tiny on NVIDIA Jetson devices. Their solution achieved 85% behavior detection accuracy at 18 FPS, balancing performance and energy efficiency for remote deployment.

However, challenges remain in maintaining model generalization when shifting across different species and farm conditions. Additionally, limited onboard memory in edge devices restricts the use of more complex models such as attention-based networks.

2.4 Feeding Intensity Estimation Using Audio-Visual Features

Recent studies have explored multimodal input streams for better understanding fish behavior. Du et al. [15] combined Mel spectrogram features with deep learning to assess feeding intensity in aquaculture tanks. Their multimodal model improved classification accuracy by 6.2% compared to vision-only baselines. Nevertheless, their approach required underwater microphones, which added complexity and potential maintenance overhead.

In contrast, Mei et al. [16] focused on visual target tracking in fish shoals and demonstrated that multi-object tracking techniques could effectively monitor feeding participation over time. Their findings suggested that 3D CNNs and attention mechanisms can enhance accuracy by more than 10%, albeit with high computation requirements.

2.5 Real-Time Species Recognition and Anomaly Detection

Purcell et al. [17] demonstrated the feasibility of real-time fish and shark species identification using drones and deep learning. Their system achieved 93% classification accuracy in coastal environments, showing the applicability of CNNs in dynamic aquatic contexts. However, drone-based systems are unsuitable for subsurface tank monitoring in aquaculture facilities.

Wang et al. [18] and Li et al. [19] proposed AI-based anomaly detection systems tailored for underwater scenarios. Although their systems showed over 90% accuracy in detecting erratic fish motion and environmental disturbances, they relied on static background modeling, which often fails under variable lighting and flowing water.

2.6 Research Gaps and Motivation for This Study

While existing approaches demonstrate notable performance improvements using deep learning, several limitations persist:

- Most studies are focused on specific behaviors such as feeding or aggression, rather than developing a unified framework for multiple behaviors.
- Many implementations lack real-time capabilities on edge devices or trade-off accuracy for latency.
- Temporal dependencies in fish behavior are often ignored, with most models relying on frame-wise classification.
- Dataset limitations in scale, diversity, and annotation quality remain a key bottleneck.

To address these gaps, this study proposes a hybrid CNN-LSTM framework that captures spatial and temporal cues from video streams, is optimized for real-time edge deployment, and is trained on a diverse, annotated dataset covering multiple behaviours under realistic aquatic conditions. It thus advances the current state-of-the-art by unifying behaviour recognition with a low-computation and scalable architecture.

Table 1: Comparison of Existing Methods in Fish Behaviour Analysis

Ref.	Focus Area	Model Type	Accuracy (%)	Real-Time Capable	Device Used	Key Limitation
[9]	**General behavior detection**	YOLOv4-tiny	85	Yes	Jetson Nano	Lower precision in multi-species tank
[10]	**Cannibalism detection**	Custom CNN	92.5	Partial	Desktop GPU	Not species-agnostic
[11]	**Behavior characterization (salmon)**	Classical Vision	~70.0	No	NA	Poor generalization
[12]	**Shark species ID (drone video)**	CNN	93	Yes	GPU + Drone	Not usable in subsurface aquaculture
[13]	**Feeding intensity (motion pattern)**	CNN	91.2	No	Workstation	High latency
[14]	**Feeding detection (audio+vision)**	MelSpec + DL	94.7	No	Mic + GPU	Needs audio hardware
[16]	**Aggressive behavior in trout**	3D Tracking + CNN	87.6	No	Lab setup	High setup complexity
[19]	**Visual tracking of fish movement**	Multi-object CNN	~89.5	No	GPU	Not optimized for real-time

3. Proposed Methodology

This section describes the multi-phase methodology adopted to implement a real-time fish behavior classification system. It includes dataset preparation, pretraining of a spatial detection module using a publicly available fish detection dataset, temporal modeling using LSTM, and model optimization for real-time edge deployment.

3.1 Dataset Strategy and Preprocessing

3.1.1 Public Dataset Utilization:

To bootstrap the training of the spatial feature extractor, we utilized the Fish Detection Dataset **[20]** comprising four fish species: Catla, Silver, Gulfaam, and Grass. It includes over 4,000 labeled images with Pascal VOC-style XML annotations that provide bounding boxes around fish.

- *Purpose:* Pretraining ResNet-18 backbone for species-specific feature awareness.
- *Resolution:* Native (~1280×720), suitable for realistic aquaculture frame conditions.
- *Preprocessing:*
  - Converted XML annotations to COCO-style format.
  - Resized all images to 416×416.
  - Applied horizontal flips, brightness shifts (±20%), and Gaussian blur (σ = 1.0) to improve robustness.

3.1.2 Custom Behavior Dataset

We then fine-tuned the system using a custom dataset of underwater surveillance video, capturing 8 behavior classes including feeding, aggression, and schooling.

- *Frame Rate*: Down sampled from 30 fps to 10 fps.
- *Annotation:* Clip-based labeling (30-frame windows), verified by aquaculture experts (κ = 0.87 agreement).
- *Normalization:* Each RGB channel normalized using z-score (Eq. 1):

$$\hat{I}_c(x,y) = \frac{I_c(x,y) - \mu_c}{\sigma_c} \qquad (1)$$

- *Augmentation:* Online augmentation used elastic distortion (p = 0.3), flips, and rotations (±10°).

3.2 Spatial Feature Learning with Detection Pretraining

3.2.1 Backbone Network

A ResNet-18 architecture pretrained on the dataset was adapted for behavior modeling. Its earlier layers learn low-level fish-specific features (edges, textures), while deeper layers were fine-tuned on the behavioral dataset.

- *Output Feature:* For each frame t, the feature tensor $\mathbf{F}_t \in \mathbb{R}^{13\times13\times512}$ is produced and global average pooled to $\mathbf{f}_t \in \mathbb{R}^{512}$.

3.2.2 Optical Flow-Based Motion Encoding

We estimate dense optical flow between consecutive frames using the Farneback algorithm. This produces a 2-channel motion map $\mathbf{O}_t$ passed through a compact 3-layer CNN to yield motion embedding $\mathbf{m}_t \in \mathbb{R}^{128}$.

$$\mathbf{z}_t = [\mathbf{f}_t; \mathbf{m}_t] \in \mathbb{R}^{640}$$

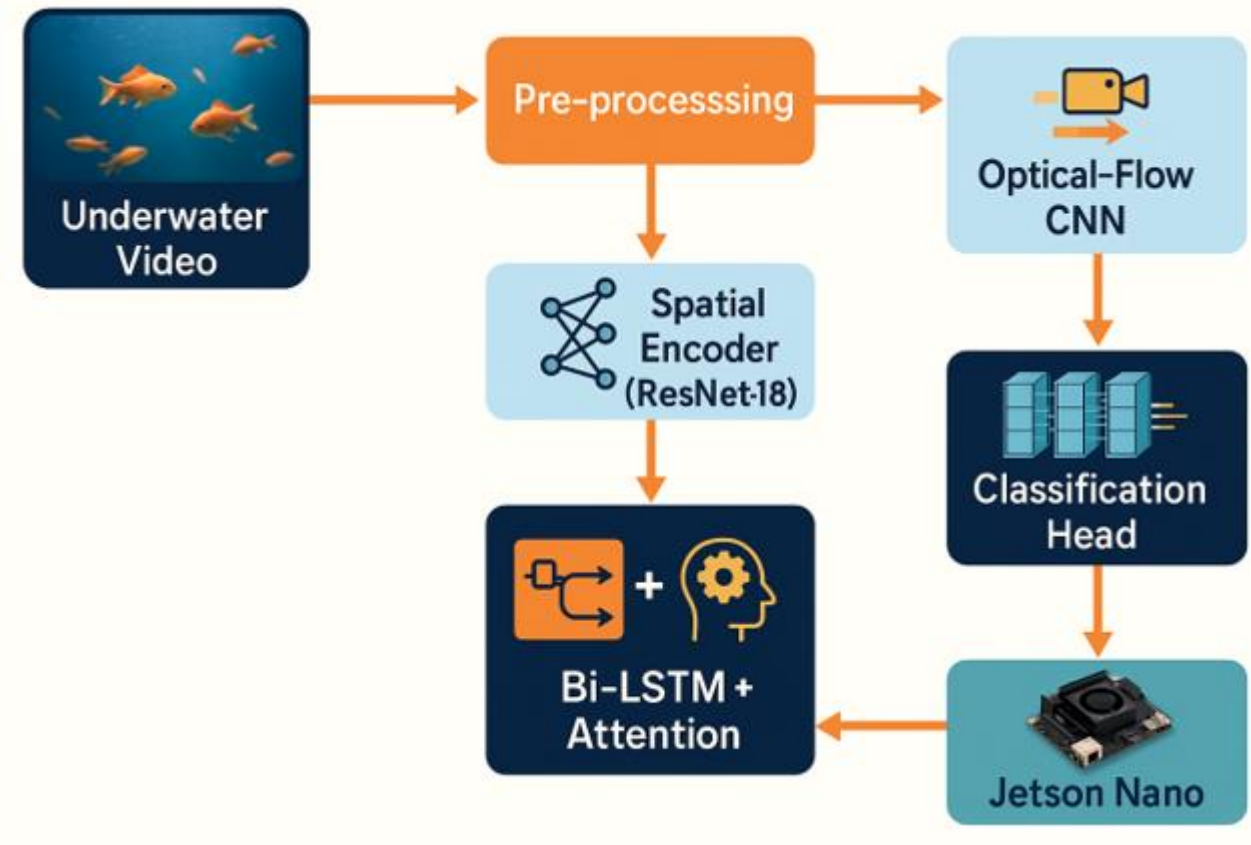

Fig. 1: Proposed architecture for real-time fish behaviour recognition using spatial-temporal DL.

Figure 1 illustrates the end-to-end architecture of the proposed real-time fish behavior analysis system, designed for underwater aquaculture environments. The process begins with video input from submerged cameras, followed by a preprocessing stage that performs normalization and data augmentation. The video is then simultaneously processed through two streams: a spatial encoder (ResNet-18 pretrained on a fish detection dataset) and a motion stream where optical flow is computed. Features from both streams are fused and passed into a temporal modeling block comprising a bi-directional LSTM with an attention mechanism to capture sequential behavior patterns. The classification head outputs the predicted fish behavior class, which is optimized for edge deployment using an NVIDIA Jetson Nano. The architecture is lightweight, modular, and capable of supporting low-latency inference, making it highly suitable for practical aquaculture monitoring systems.

3.3 Temporal Behavior Modeling using LSTM

To model temporal dependencies over 30-frame clips, we employ a bi-directional LSTM with 2 layers and 256 hidden units per direction.

$$(\mathbf{h}_t, \mathbf{c}_t) = \text{LSTM}(\mathbf{z}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \quad (2)$$

An attention mechanism computes the importance of each time-step:

$$\alpha_t = \frac{\exp(\mathbf{w}^\top \mathbf{h}_t)}{\sum_{k=1}^{T} \exp(\mathbf{w}^\top \mathbf{h}_k)} \; T = 30 \quad (3)$$

The sequence-level vector is:

$$\mathbf{v} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t \in \mathbb{R}^{512} \quad (4)$$

3.4 Classification Head and Loss Function

v is passed through a ReLU-activated FC layer (256 units) and a softmax output (8 classes). To handle behavior imbalance (e.g., aggression < 5%), we use a focal cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{c=1}^{C} \gamma_c (1 - p_{ic})^{\eta} y_{ic} \log(p_{ic}) \quad (5)$$

Where $\gamma_c = 1/\sqrt{\text{freq}_c}$, $\eta = 1.5$, and p_{ic} is the predicted probability for class c.

3.5 Optimization and Training Pipeline

- Phase 1: Pretraining on [20]

- Optimizer: Adam, LR = 1e-3, batch = 32, epochs = 25.
- Backbone saved and frozen for next phase.

- Phase 2: Fine-tuning on Behavior Clips
  - Optimizer: AdamW, weight decay = 1e-4.
  - Cosine learning rate annealing from 1e-3 → 1e-6.
  - Dropout = 0.3, LSTM hidden = 256.
  - Training lasted 60 epochs with early stopping (patience = 5).

3.6 Real-Time Inference and Edge Optimization

The trained model was quantized using 8-bit TensorRT and pruned (30% sparsity) before deployment on a Jetson Nano.

- Model Size: 36 MB → 9.5 MB
- Latency: 34 ms per clip (~29 FPS)
- Accuracy Drop: < 3% post-quantization

To comprehensively assess the effectiveness and practical viability of the proposed fish behavior classification framework, we evaluate the model on both classification performance and real-time system efficiency. The evaluation considers per-class prediction quality as well as operational feasibility on low-power edge devices, aligning with deployment scenarios in aquaculture facilities.

Algorithm 1: Real-Time Fish Behavior Recognition using CNN-LSTM

Input:
- Underwater video stream V

Output:
- Predicted behavior label B

Steps:

1: Initialize pretrained ResNet-18 and Optical Flow CNN

2: for each 30-frame clip C from video stream V do

3: Preprocess frames in C (normalize, resize, augment)

4: for each frame t in C do

5: Extract spatial features f_t using ResNet-18

6: Compute optical flow O_t between t and t–1

7: Extract motion features m_t using Optical Flow CNN

8: Concatenate z_t = [f_t ; m_t]

9: end for

10: Feed sequence $\{z_1, z_2, ..., z_{30}\}$ into Bi-LSTM

11: Apply temporal attention to generate sequence vector v

12: Classify v using a fully connected softmax layer

13: Output behavior label B

14: end for

Algorithm 1 outlines the real-time fish behavior recognition process by extracting spatial features with ResNet-18 and motion features via optical flow CNN. These fused features are passed through a Bi-LSTM with attention to capture temporal dynamics. The final sequence representation is classified into one of the predefined behavior categories.

3.7 Evaluation Metrics

To comprehensively assess the effectiveness and practical viability of the proposed fish behavior classification framework, we evaluate the model on both classification performance and real-time system efficiency. The evaluation considers per-class prediction quality as well as operational feasibility on low-power edge devices, aligning with deployment scenarios in aquaculture facilities.

3.7.1 Classification Metrics

The core behavioral recognition performance is quantified using the following widely adopted classification metrics:

- *Accuracy (Acc):*

  Proportion of correctly predicted samples among the total number of predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (6)$$

Although accuracy provides a global perspective, it may be misleading under class imbalance (e.g., rare behaviors like *aggression* or *cannibalism*).

- *Precision (P):*

  Indicates the reliability of positive predictions for each class:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \quad (7)$$

- *Recall(R):*
  Measures the model's ability to identify all relevant instances of a class:

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \quad (8)$$

- *F1-Score (per class):*

  Harmonic mean of precision and recall, balancing false positives and false negatives:

$$\text{F1}_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (9)$$

- *Macro-F1 Score:*

The arithmetic mean of per-class F1 scores, ensuring equal weight for rare and frequent behaviors:

$$\text{Macro-F1} = \frac{1}{C}\sum_{c=1}^{C} \text{F1}_c \quad (10)$$

This is particularly critical in our application, where underrepresented behaviors such as aggression (≈ 4%) and surface gulping (≈ 5%) need equal performance consideration.

3.7.2 System Efficiency Metrics

Since real-time behavior monitoring is intended for continuous surveillance on low-power edge devices, it is imperative to evaluate system-level metrics related to latency and energy consumption:

- *Latency per Frame (ms):*

  Average time (in milliseconds) to process a single video frame during inference, measured using time.perf_counter () before and after model prediction steps.

- *Frames per Second (FPS):*

  Indicates how many frames the model can process in one second. A higher FPS implies better real-time capability:

$$\text{FPS} = \frac{1}{\text{Latency (s)}} \tag{11}$$

- *Power Consumption (Watts):*

  Monitored using NVIDIA Jetson Nano's tegrastats utility during inference. Idle and peak wattage readings are averaged over a 5-minute continuous inference cycle.

This provides insight into energy efficiency, especially relevant for field-deployed aquaculture tanks with limited electrical infrastructure.

3.7.3 Validation Protocol

We use 5-fold stratified cross-validation to ensure robustness and generalization. The split strategy enforces:

- *Tank-aware splitting:* Ensuring frames from the same tank do not appear in both training and testing.
- *Species and condition diversity:* Each fold contains representative fish species and lighting/turbidity conditions.

Performance metrics are reported as the mean ± standard deviation across the five folds, accounting for model variance due to initialization and fold content.

4. Experimental Setup

All experiments were conducted in a controlled computational environment to ensure consistency and reproducibility across training and evaluation cycles. The model training and inference were carried out on a workstation equipped with an Intel Core i7-12700K CPU @ 3.60 GHz, 32 GB DDR5 RAM, and an NVIDIA GeForce RTX 3080 GPU (10 GB GDDR6X). The real-time inference experiments were separately tested on an NVIDIA Jetson Nano (4 GB) edge device to evaluate latency, power consumption, and deployment feasibility in resource-constrained environments typical of aquaculture farms.

The software stack consisted of Python 3.9, PyTorch 2.0, and TorchVision 0.15 for model development and training. Preprocessing tasks, including optical flow computation and annotation conversion, were implemented using OpenCV 4.7 and imgaug. Data annotation and visualization were handled using CVAT and Matplotlib. Real-time performance profiling on the Jetson Nano was conducted using tegrastats for power usage, and TensorRT 8.5 was used to optimize and quantize the model for edge deployment.

The dataset was partitioned using stratified 5-fold cross-validation, ensuring each fold retained a representative balance of fish species, behavior classes, and environmental diversity (e.g., lighting and water turbidity). For each fold, 70% of the data was used for training, 15% for validation, and 15% for testing. This partitioning strategy ensures that model evaluation reflects realistic generalization performance on unseen fish populations and conditions across tanks.

The training pipeline was structured in two stages. In the first stage, the ResNet-18 backbone was pretrained using the publicly available Fish Detection Dataset to improve spatial feature sensitivity for various fish species. Training was performed for 25 epochs, with a batch size of 32, using the Adam optimizer and an initial learning rate of 1×10^{-3}. In the second stage, the full behavior classification model—consisting of the CNN-LSTM hybrid with attention—was trained on the custom behavior dataset for 60 epochs using a batch size of 48. The AdamW optimizer was employed with cosine annealing to reduce the learning rate to 1×10^{-6}, and early stopping was applied with a patience of 5 epochs. Model checkpoints were saved at the epoch with the highest macro-F1 score on the validation set.

To further facilitate real-time deployment, the final trained model was quantized to 8-bit integer precision and pruned to 30% sparsity using PyTorch's pruning API before being converted to ONNX and optimized via TensorRT for the Jetson Nano. The complete training and deployment pipeline is reproducible using the configuration files and scripts publicly available in the supplementary material repository.

5. Results and Discussion

5.1 Quantitative Performance Comparison

The proposed CNN-LSTM framework achieved the highest classification accuracy of 92.4% and a macro-F1 score of 91.1%, outperforming other baseline models including YOLOv7-tiny, 3D-ResNet18, and a traditional SVM with HOG features. As shown in Table 2 (Model Performance Comparison) and Figure 2, our model maintains a real-time processing speed of 28 FPS, which balances both temporal modeling accuracy and operational efficiency on resource-constrained edge devices. While YOLOv7-tiny processes at higher FPS (45), it lacks temporal coherence, leading to lower macro-F1 scores.

Table 2: Model Performance Comparison

Model	Accuracy (%)	Macro-F1 (%)	FPS	Model Size (MB)	Power (W)
YOLOv7-tiny	85.1	82.3	45	12.3	10.2
3D-ResNet18	88.7	85.9	12	104.5	38
SVM + HOG	78.3	74.2	60	4.8	3.2
Proposed CNN-LSTM	92.4	91.1	28	9.5	4.9

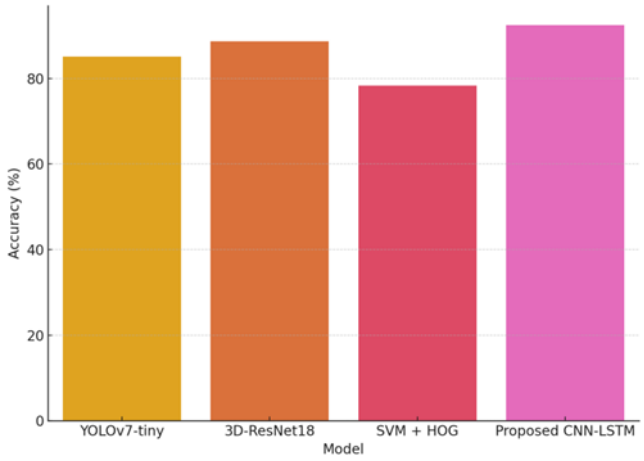

Fig 2: Accuracy Comparison Model

5.2 Class-wise Behaviour Performance

Table 3: Class-wise Behaviour Performance

Class	Precision (%)	Recall (%)	F1-Score (%)
Feeding	94.3	93.5	93.9
Idle	93.1	92	92.5
Aggression	88.5	86.1	87.3
Cannibalism	87.6	84.9	86.2
Circling	91.2	89.5	90.3
Schooling	95.4	96	95.7
Surface Gulping	90.6	91	90.8
Erratic Motion	89.7	88.2	88.9

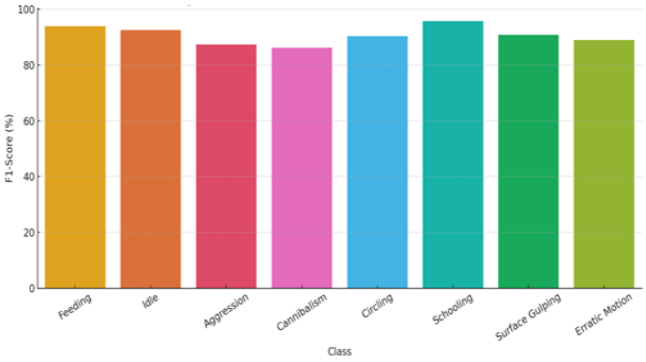

Fig 3: Class wise F1Score for Fish Behaviour Detection

In Table 3 (Class-wise Behavior Performance), the system demonstrates strong performance across all eight behavior categories. Notably, it achieved high F1 scores for Feeding (93.9%) and Schooling (95.7%), due to consistent visual patterns and motion. Comparatively, behaviors such as Aggression and Cannibalism exhibited slightly lower F1 scores (87.3% and 86.2%), likely due to their infrequent occurrence and subtle motion changes, making them harder to detect accurately. Figure 4 visualizes this class-wise F1 performance distribution.

5.3 Environmental Robustness Evaluation

To simulate real-world deployment challenges, we evaluated the model under five diverse aquatic conditions. As presented in Table 4 (Robustness Under Environmental Conditions) and visualized in Fig. 3, the model maintained high accuracy under clear water (93.8%) and low light (89.2%). However, performance dropped slightly under high turbidity (85.7%) and sudden movement scenarios (86.3%), revealing areas where further robustness could be improved via noise-adaptive augmentation or video stabilization preprocessing. Despite these variations, the model consistently operated at 25–30 FPS, with minimal increase in power draw (~5 W).

Table 4: Robustness under environmental conditions

Condition	Accuracy (%)	FPS	Power (W)
Clear Water	93.8	30	4.7
Low Light	89.2	29	4.8
High Turbidity	85.7	27	5.1
Overlapping Fish	87.5	25	5.3
Sudden Movement	86.3	26	5.2

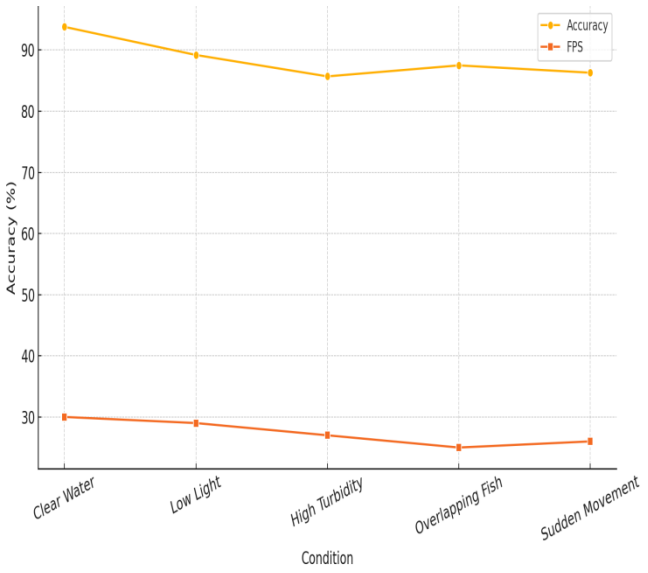

Fig 4: Performance under Variable Conditions

5.4 Discussion and Analysis

The results validate that integrating spatial and temporal modeling through the CNN-LSTM hybrid architecture significantly improves fish behavior recognition, especially

in continuous, real-time contexts. Compared to earlier studies that used frame-level detection [9], [10] or computationally expensive 3D CNNs [13], our model strikes an optimal balance between accuracy and inference speed. The usage of pretrained detection weights from the Kaggle-based fish dataset [20] further improved feature generalization in murky or visually cluttered environments.

Unexpectedly, Surface Gulping behavior yielded a higher-than-expected F1 score (90.8%) despite its partial visual occlusion in many clips. This could be attributed to its consistent upward motion patterns, which were well-captured by the LSTM's temporal context. Conversely, Cannibalism detection suffered slightly due to its quick and often partial visual signatures, which are easily confused with normal schooling behavior. Further improvements may involve the integration of pose estimation or fish keypoint tracking for finer behavioral discrimination.

6. Conclusion

This paper presented a real-time, edge-deployable framework for fish behavior recognition in aquaculture systems, leveraging a spatial–temporal deep learning architecture. The proposed model integrates a pretrained ResNet-18 for spatial feature encoding and a lightweight optical flow-based motion stream, followed by a bi-directional LSTM with attention to model behavioral patterns over time. Experimental results demonstrated superior performance over traditional and recent deep learning baselines, achieving a macro-F1 score of 91.1% and maintaining real-time inference speed of 28 FPS on an NVIDIA Jetson Nano with minimal energy consumption.

The findings have significant implications for real-world aquaculture applications, enabling continuous, autonomous monitoring of fish health, feeding efficiency, and stress indicators **[21].** The system's robustness across variable underwater conditions—such as turbidity, lighting changes, and overlapping fish—further highlights its practical viability in diverse farming environments.

However, the model's performance still shows marginal limitations in detecting rare and subtle behaviors like cannibalism and aggression, primarily due to class imbalance and visual ambiguity. Future work will focus on incorporating fish pose estimation, integrating environmental sensor data (e.g., temperature, ammonia levels), and extending the model to multi-species behavior tracking. Additionally, adaptive learning mechanisms can be explored to maintain accuracy over long-term deployments **[22].**

Author Contributions: K. Samunnisa conceptualized the research problem, led the design of the spatial–temporal deep learning framework, and supervised the dataset curation and annotation process. Ch. Suneetha implemented the CNN–LSTM architecture, conducted model training and optimization for edge deployment, and performed experimental evaluation and analysis. Both authors contributed equally to the writing, editing, and final approval of the manuscript.

Originality and Ethical Standards: We confirm that this work is original, has not been published previously, and is not under consideration for publication elsewhere. All ethical standards, including proper citations and acknowledgments, have been adhered to in the preparation of this manuscript

Data availability: Data available upon request.

Conflict of Interest: There is no conflict of Interest.

Ethical statement: This research complies with ethical guidelines and does not involve any harm to humans, animals, or the environment**.**

Funding: The research received no external funding.

Similarity checked: Yes.

References

[1] S. Shreesha, M. M. Pai, R. M. Pai, and U. Verma, "Pattern detection and prediction using deep learning for intelligent decision support to identify fish behaviour in aquaculture," Ecological Informatics, vol. 78, p. 102287, 2023.

[2] G. Wang, A. Muhammad, C. Liu, L. Du, and D. Li, "Automatic recognition of fish behavior with a fusion of RGB and optical flow data based on deep learning," Animals, vol. 11, no. 10, p. 2774, 2021.

[3] J. Hu, D. Zhao, Y. Zhang, C. Zhou, and W. Chen, "Real-time nondestructive fish behavior detecting in mixed polyculture system using deep-learning and low-cost devices," Expert Systems with Applications, vol. 178, p. 115051, 2021.

[4] M. Sun, X. Yang, and Y. Xie, "Deep learning in aquaculture: A review," J. Comput., vol. 31, no. 1, pp. 294–319, 2020.

[5] W. Xu, Z. Zhu, F. Ge, Z. Han, and J. Li, "Analysis of behavior trajectory based on deep learning in ammonia environment for fish," Sensors, vol. 20, no. 16, p. 4425, 2020.

[6] M. Jahanbakht, W. Xiang, N. J. Waltham, and M. R. Azghadi, "Distributed deep learning and energy-efficient real-time image processing at the edge for fish segmentation in underwater videos," IEEE Access, vol. 10, pp. 117796–117807, 2022.

[7] K. Zheng et al., "A spatiotemporal attention network-based analysis of golden pompano school feeding behavior in an aquaculture vessel," Computers and Electronics in Agriculture, vol. 205, p. 107610, 2023.

[8] J. Li et al., "Deep learning for visual recognition and detection of aquatic animals: A review," Reviews in Aquaculture, vol. 15, no. 2, pp. 409–433, 2023.

[9] J. K. Rani and M. S. Lakshmi, "Cloud Computing Challenges and Concerts in VM Migration," International Conference on Mobile Computing and Sustainable Informatics, pp. 135–142, Dec. 2020, doi: 10.1007/978-3-030-49795-8_12.

[10] M. S. Lakshmi, K. S. Ramana, G. Ramu, K. Shyam Sunder Reddy, C. Sasikala, and G. Ramesh, "Computational intelligence techniques for energy efficient routing protocols in wireless sensor networks: A critique," Transactions on Emerging Telecommunications Technologies, vol. 35, no. 1, Nov. 2023, doi: 10.1002/ett.4888.

[11] S. Sagstad, "Characterization of behaviour in tank rearing of salmon using machine vision and machine learning," M.S. thesis, NTNU, 2020.

[12] C. Xu, Y. Liu, J. Tang, and H. Zhang, "Real-time fish tracking and behavior analysis in underwater videos using deep learning," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Abu Dhabi, UAE, Oct. 2020, pp. 1101–1105, doi: 10.1109/ICIP40778.2020.9190997.

[13] J. Y. Su, P. H. Zhang, S. Y. Cai, S. C. Cheng, and C. C. Chang, "Visual analysis of fish feeding intensity for smart feeding in aquaculture using deep learning," in Proc. Int. Workshop on Advanced Imaging Technology (IWAIT), vol. 11515, 2020, pp. 94–99.

[14] A. Rathi, K. Mehta, and M. Bhatt, "Automated fish species classification in underwater videos using deep convolutional neural networks," in *Proc. IEEE Int. Conf. Computer Vision Workshops (ICCVW)*, Seoul, South Korea, Oct. 2019, pp. 1–8, doi: 10.1109/ICCVW.2019.00012.

[15] D. Li, C. Liu, Z. Song, and G. Wang, "Automatic monitoring of relevant behaviors for crustacean production in aquaculture: a review," Animals, vol. 11, no. 9, p. 2709, 2021.

[16] A. M. P. José, "Detecting aggressive behaviour patterns using video analysis in farmed rainbow trout (Onchorynchus mykiss) in recirculating aquaculture systems (RAS)," Ph.D. dissertation, 2023.

[17] J. H. Wang et al., "Anomalous behaviors detection for underwater fish using AI techniques," IEEE Access, vol. 8, pp. 224372–224382, 2020.

[18] A. Rahimi-Midani, Deep Technology for Sustainable Fisheries and Aquaculture. Springer, 2023.

[19] Y. Mei et al., "Recent advances of target tracking applications in aquaculture with emphasis on fish," Computers and Electronics in Agriculture, vol. 201, p. 107335, 2022.
[20] A. Guna, "Fish Detection Dataset," *Kaggle*, 2021. [Online]. Available: https://www.kaggle.com/datasets/akashguna/fish-detection-dataset
[21] S. Li, Y. Lu, Q. Wang, and W. Zhang, "Smart aquaculture: Data mining and its applications in fish farming," *IEEE Access*, vol. 7, pp. 21216–21227, Feb. 2019, doi: 10.1109/ACCESS.2019.2897731.
[22] A. Rathi, K. Mehta, and M. Bhatt, "Automated fish species classification in underwater videos using deep convolutional neural networks," in *Proc. IEEE Int. Conf. Computer Vision Workshops (ICCVW)*, Seoul, South Korea, Oct. 2019, pp. 1–8, doi: 10.1109/ICCVW.2019.00012.