EXPLORING THE POTENTIAL OF USING ARUCO MARKERS TO MONITOR FISH FEEDING STATUS

BY

GOH KEN HOW

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

FEBRUARY 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to all those who have supported and guided me throughout the course of this project.

First and foremost, I would like to thank my former supervisor, Ts Dr. Ooi Boon Yaik, for his invaluable guidance, encouragement, and insightful feedback throughout this project. His expertise and advice have been instrumental in shaping the direction of my research and ensuring its successful completion. I am particularly grateful for his provision of essential tools and resources, which significantly contributed to the successful execution of this work.

Secondly, I would like to express my sincere thanks to my current supervisor, Mr. Tan Teik Boon, for his continued support, constructive advice, and dedicated supervision. His guidance has been especially valuable during the critical stages of the project, where his insights helped me overcome challenges, refine my methodology, and strengthen the overall quality of my work. His encouragement and clear direction enabled me to stay focused and confident throughout the research process.

I am also grateful to the Faculty of Information and Communication Technology (FICT) for providing the resources and facilities necessary for conducting this research. Special thanks to the technical support team for their assistance and expertise during the experimental phases of the project.

In addition, I would like to extend my appreciation to Mr. Colin Kiu Qi Song from the Department of Agricultural and Food Science for his generous assistance with my Final Year Project. He played a vital role in supporting the data collection process by feeding the fish using appropriate and professional aquaculture methods, as well as operating fish farming-related equipment. His practical expertise greatly contributed to the reliability and accuracy of the data collected for this research.

My heartfelt thanks also go to my family and friends for their unwavering encouragement, patience, and emotional support throughout this journey. Their belief in me has been a constant source of strength and motivation, especially during challenging times.

To all who have contributed to the success of this project, directly or indirectly—thank you very much.

ABSTRACT

Efficient feeding management is a cornerstone of sustainable aquaculture, directly influencing fish growth, health, and resource utilization. Traditional feeding methods, which rely on manual observation to determine satiety, are labour-intensive, subjective, and prone to human error—often resulting in overfeeding and operational inefficiencies. This project presents a novel approach for monitoring fish feeding status by leveraging ArUco marker tracking. Pose estimations of floating markers are analysed to extract movement intensity, which is then interpreted using a time-series LSTM classification model to detect fish activity and infer satiety levels. The system was developed using a combination of Python, Keras, and OpenCV, and deployed in a real aquaculture setting using red hybrid tilapia (Oreochromis sp.). A web-based interface provides real-time pose data, fish activity classification, feeding recommendations, and status tracking. Model performance was validated through cross-validation and real-world testing, achieving high accuracy and practical reliability. Beyond monitoring fish feeding status, the system also detects air pump operation and tracks water level variations, offering a broader view of tank conditions. It supports multi-tank monitoring using a single camera, making the solution cost-effective, scalable, and non-invasive. The results affirm the system's potential to improve feed management, reduce labour dependency, and support more intelligent and sustainable aquaculture practices.

Area of Study: Internet of Things, Computer Vision

Keywords: Precision Aquaculture, Internet of Things, Monitoring, ArUco Markers, Fish Feeding Monitoring, Non-Invasive Tracking, Scalable Aquaculture System, Pose Estimation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Symbol	**Definition**
°	degree

LIST OF ABBREVIATIONS

Abbreviation	Definition
CNN	Convolutional Neural Network
GPU	Graphics Processing Unit
LSTM	Long Short-Term Memory
AUC	Area Under Curve
ROC	Receiver-Operating Characteristic

CHAPTER 1 INTRODUCTION

1.1 Project Background

Aquaculture, serving as the vital component of the global food industry, does provide a sustainable and continuous source of protein through fish farming. Historically, the core of optimizing aquaculture practices is the effective and optimal monitoring of fish feeding behaviour. Accurate monitoring of fish feeding behaviour and managing feeding times are crucial, as these factors directly influences the growth rates, health of the fish populations, as well as impacts the resource management such as feed.

Traditionally, fish feeding behaviour has depended on visual observation. Precisely, fish farmers tend to provide feed to fish in tanks until the fish stop consuming the feed, to ensure that the fish are fully satiated. Yet, such feeding approach has its drawbacks. It is labour-intensive, subjective as well as intrusive. More importantly, reliance on human observation greatly possesses human errors which potentially leads to overfeeding–problem that can interfere with the sustainability and profitability of aquaculture operations.

Our project proposes an innovative solution by exploring the potential of using ArUco markers to monitor fish feeding status. Instead of proprietary and specialized sensors, we aim to capture data that reflects the activity status of fish within the water using ArUco marker system, which are indicative of fish feeding activities. By analysing the pose estimations of ArUco markers affixed to floats on the water's surface, we can gain insights into the fish's activity levels and feeding patterns. This novel approach not only enhances the accuracy of monitoring fish feeding status but also strives to improve the overall efficiency of aquaculture operations.

Beyond monitoring fish feeding status, the system also classifies air pump operation based on surface motion patterns and tracks water level fluctuations through vertical marker displacement. These extended features offer a broader understanding of tank conditions and improve environmental monitoring. Furthermore, the system supports multi-tank monitoring using a single overhead camera, significantly reducing hardware requirements and making the solution scalable for larger aquaculture facilities. Together, these capabilities form a comprehensive, non-invasive, and intelligent monitoring platform designed to promote sustainability and productivity in modern fish farming.

Leveraging advanced computer vision technology and data analytics, our project seeks to establish a low-cost, non-invasive, and reliable method for monitoring fish feeding status. This method can reduce dependency on human observation, optimizing feeding schedules and preventing overfeeding. Ultimately, our solution aims to improve the health and growth rates of fish populations, contributing to more sustainable and profitable aquaculture practices.

1.2 Motivation

Fish farmers aim to increase the size and enhance the value of their fish by continuously feeding them, as this is the quickest and most straightforward method to promote growth of the fish. However, farmers feed the fish continuously until the fish stop consuming the feed to ensure that the fish are fully satiated. Consequently, the excessive amounts of uneaten feed remain in the fish tanks, leading to issues of feed waste and a polluted environment in the fish tanks. This scenario underscores a need for more precise and controlled feeding methods that can prevent overfeeding, reduce waste, and maintain optimal water quality. Therefore, addressing this need motivates our exploration into the use of ArUco marker as a novel approach to monitor fish feeding status more accurately and non-invasively.

1.3 Problem Statements

We aim to address and solve two primary problem statements as follows:

1. **Lack of effective and non-invasive method to capture data to monitor fish feeding status.**

   This problem statement highlights the difficulty of capturing accurate data to monitor fish feeding status without being affected by the environmental factors of fish tanks and without disrupting the natural behaviours of the fish during feeding activity. It is important as an effective and non-invasive method to capture data not only preserves the nature of the aquatic system in the fish tank but also enhances the accuracy of the captured data.

2. **Uncertainty in interpreting movement intensity to assess fish satiety level.**

   This problem statement highlights the difficulty in analysing activity status of fish to draw meaningful conclusions about the satiety levels of fish. In addition, it highlights the challenge in determining fish satiety level using an effective and non-invasive method. It is important as determining fish satiety level helps optimize feed usage, preventing fish farmers from providing excessive feeds to the fish during feeding activities.

1.4 Objectives

We aim to achieve two primary objectives in this project to address the problem statements. Two primary project objectives are as follows:

1. To capture movement intensity data that reflects fish feeding status.
2. To interpret movement intensity data to indicate the fish satiety levels.

1.5 Project Scope

This project explores the potential of using ArUco markers to monitor fish feeding status by capturing and analysing the movement intensity of fish within a tank through an ArUco marker-based vision system. Specifically, it investigates the feasibility of affixing ArUco markers to floating objects on the water surface to indirectly measure feeding activity based on changes in marker pose during fish feeding periods.

The feasibility of this approach was initially assessed in Project I, where preliminary tests validated the accuracy and reliability of pose estimations obtained from ArUco markers in controlled environments. Building upon this foundation, Project II focuses on developing a functional monitoring system capable of capturing real-time pose estimations data and interpreting it to infer fish feeding status. The aim is to determine whether feeding should be continued or stopped based on movement intensity of the marker, contributing to more efficient and responsive feed management.

In this project, all experiments are conducted using red hybrid tilapia (Oreochromis sp.), with a total sample size of around 80 individuals. The testing is carried out at the campus aquaculture facility, where the fish are maintained under standard rearing conditions by students specializing in aquaculture. This environment closely reflects real-world farming practices, making it a suitable setting for evaluating the practical effectiveness of the proposed monitoring system.

Beyond feeding status detection, the project also explores the potential to extend the system’s functionality to monitor additional tank conditions, such as air pump status and water level fluctuations, using similar visual indicators. The final deliverable includes a user-accessible interface that allows fish farmers to view real-time data for multiple tanks. This includes actionable insights such as whether to continue feeding, and alerts related to water level and aeration status. Notably, the system is designed for multi-tank monitoring using a single camera, significantly reducing hardware costs while enhancing scalability and ease of deployment in commercial aquaculture settings.

1.6 Contributions

Our project offers a meaningful contribution to the aquaculture field globally by introducing a novel, computer vision–based approach for monitoring fish feeding status using ArUco markers. Through the analysis of pose estimations from floating markers, the system provides real-time insights into fish activity levels, allowing for accurate detection of feeding behaviour and satiety. This enables fish farmers to optimize feed usage, prevent overfeeding, and maintain healthier fish populations—ultimately leading to improved growth performance and reduced feed waste, which is one of the largest operational costs in aquaculture.

Beyond individual tank monitoring, the system is designed with multi-tank scalability in mind. By dividing the camera frame into predefined regions and associating each with a different ArUco marker, a single overhead camera can simultaneously monitor multiple tanks. This dramatically reduces hardware requirements and installation complexity, making the system cost-effective for both small-scale and large-scale aquaculture facilities.

Furthermore, the project deliverables lay the groundwork for future automation. The ability to classify feeding activity and generate feeding recommendations can be extended to trigger automated feeding mechanisms, forming the foundation for a fully automated smart feeding system. Such a system would not only streamline daily operations but also minimize labour dependency and reduce human error—two common challenges in traditional fish farming. In the long term, adopting this intelligent monitoring and automation approach promotes more efficient, scalable, and sustainable aquaculture practices, supporting food security and responsible resource use across the industry.

1.7 Report Organization

This report is organized into seven chapters, each structured to reflect the development process and outcomes of the project as follows:

- Chapter 1 introduces the background, problem statements, project objectives, scope, and significance of the study.
- Chapter 2 presents a literature review of relevant research, including fish feeding behaviour, movement analysis techniques, and ArUco marker-based tracking systems.
- Chapter 3 outlines the system methodology, including use case, system architecture, and activity diagrams that illustrate the overall workflow of the system.
- Chapter 4 details the system design, covering hardware and software specifications, model training, data preprocessing, and the interaction between system components.
- Chapter 5 describes the implementation process, including the deployment environment, training pipeline, and operation of the system in real time.
- Chapter 6 presents the system evaluation, including model performance metrics, real-world testing results, challenges faced, and how each objective was evaluated.
- Chapter 7 concludes the report by summarizing the project outcomes and proposing potential improvements and future work.

CHAPTER 2 LITERATURE REVIEW

This chapter reviews existing literature to establish the importance of monitoring fish feeding status, with particular attention to the implications of overfeeding in aquaculture. It begins by discussing the biological behaviour of fish during feeding activities, followed by a critical evaluation of existing technological solutions, including deep learning, computer vision, acoustic, and vibrational methods. The chapter also explores the use of ArUco marker-based tracking across various fields to highlight its potential as a novel, non-invasive approach for fish feeding status monitoring. Finally, the chapter concludes by summarizing key insights and justifying the relevance of the proposed method in this project.

2.1 Necessity to Monitor Fish Feeding Status

Inappropriate feeding methods, such as feeding the fish until they stop consuming the feed to ensure that they are fully satiated, can lead to overfeeding. Overfeeding not only decreases the efficiency of the feed but also causes feed waste, bacterial growth, and environmental degradation in the fish tank [1], [2]. According to [1], when excessive amounts of nutrients are released into the aquatic system of the fish tank, particularly in a continuous timeline, it promotes the growth of algae, leading to a phenomenon known as algal blooms. This phenomenon threatens the life of the fish in the fish tank. Typically, the algae might consume a higher amount of oxygen compared to the amount they produce during photosynthesis, leading to oxygen depletion in the aquatic system of the fish tank. Consequently, without sufficient oxygen in the fish tank, the fish face the risk of death.

2.2 Fish Behavioural Patterns Related to Feeding Activity

Fish exhibit distinct changes in swimming behaviour before, during, and after feeding events. When hungry, fish typically display increased locomotor activity characterized by higher swimming speeds and broader movement ranges as they actively search for food. Following feeding, their movement becomes more restrained; well-fed fish tend to swim slower and within a smaller spatial range, indicating a state of satiety. These behavioural shifts are strongly linked to feeding motivation and can serve as useful indicators for assessing the overall feeding status of a fish group [3].

2.3 Existing Solutions on Monitoring Fish Feeding Status

Numerous technological solutions have been developed to monitor fish feeding status, aiming to improve feed efficiency, reduce waste, and support sustainable aquaculture practices. These solutions range from advanced artificial intelligence techniques to sensor-based and computer vision methods. Each approach offers unique advantages but also faces limitations related to cost, accuracy, scalability, or environmental dependency. This section presents a review of the most prominent methods currently used in the field, including the application of deep learning, computer vision, acoustic analysis, and vibrational pattern recognition, with a focus on their effectiveness and practicality for real-world aquaculture environments.

2.3.1 Application of Deep Learning

Overview of Deep Learning Application

To address the challenge associated with inaccurately assessing fish hunger and satiety levels during feeding, deep learning techniques has been integrated into automatic fish feeding systems. Deep learning frameworks—like deep neural networks—have been utilized to analyse the video footage from aquaculture environments. These deep learning models process images to detect and interpret the feeding behaviours of fish. More precisely, they recognise the size of the waves caused by fish eating to decide whether to continue feeding or not [4].

In addition to recognising wave size, deep CNN models have been integrated into automatic fish feeding systems to determine the fish feeding intensity levels. According to [5], two CNNs have been combined to expedite evaluation of fish feeding intensity levels. These networks analyse both the optical flow of fish in the water and the water movement intensity to accurately assess feeding levels, surpassing the accuracy of human observations. Similarly, as demonstrated in [6] and [7], neural networks are widely adopted in automatic fish feeding systems to predict both feeding and non-feeding behaviours.

Capabilities of Deep Learning Application

The application of artificial intelligence in determining fish feeding behaviours achieves high levels of accuracy. For instance, the use of a deep neural network has been shown to determine fish feeding behaviours with an accuracy of 93.2% [4]. Moreover, employing CNNs to analyse the optical flow in video inputs has reached accuracy of 95% [5]. The process of evaluating fish feeding intensity levels has been accelerated by using a combination of two CNNs. According to [6] and [7], the accuracy of using various neural networks to decide whether to continue or stop feeding fish also exceeds 80%. In brief, applying deep learning techniques ensures not only high accuracy in determining fish feeding behaviours but also a faster evaluation process.

Limitations of Deep Learning Application

However, applying deep learning techniques in automatic feeding systems requires high computational costs, necessitating high performance level GPUs and large amounts of memory spaces. This need increases as more feature combinations are considered with larger input sizes into the models [8]. Consequently, performance in evaluating fish feeding behaviours may decline if the system lacks these robust capabilities.

Moreover, the accuracy of deep learning models is sometimes questioned due to the “black box” nature of these systems, as outlined in [9]. Researchers understand the input features and the output but lack visibility into the internal processes that evaluate fish feeding behaviours. This lack of transparency can lead to doubts about the interpretability of the models, potentially undermining trust in the accuracy of the results produced by the deep learning models.

2.3.2 Application of Computer Vision

Overview of Computer Vision Application

Computer vision techniques have been applied in previous works to monitor fish feeding behaviour. This technology involves installing cameras to capture videos of fish activity, which are then analysed to determine their feeding behaviours [10], [11]. Different computer vision algorithms are implemented to perform analysis work on the captured data [12]. The primary goal is to capture the movement and motion of fish using image processing methods during feeding activities.

According to [10], a camera is mounted above the fish tank to record the fish's movements during feeding times. The data from these recordings are analysed by subtracting consecutive video frames to identify the intensity of movements, highlighting areas of motion, and thus determining the level of fish activity. Furthermore, researchers have enhanced the accuracy of these measurements by introducing an overlap coefficient, which helps correct calculation errors caused by the overlapping of fish in the video frames.

Capabilities of Computer Vision Application

Integrating computer vision technology into automatic fish feeding systems offers advantages such as low computational costs and high effectiveness and accuracy in identifying fish feeding behaviours. The relatively low computational demand of computer vision addresses some limitations of deep learning models used for similar purposes. Additionally, as noted in [10], there is a high correlation between indices derived from computer vision-based feeding activity and those observed manually, underscoring the effectiveness of this method.

Limitations of Computer Vision Application

The effectiveness of the computer vision technique depends heavily on clear visibility of the captured video. As mentioned in [10], while a camera mounted above the tank does capture fish movements, the visibility of these videos can be affected by various factors such as water quality, the colour of the tank glass, ambient lighting conditions and size of fish targets. For instance, if the fish and the tank glass are similarly coloured, particularly if both are dark, it can reduce the visibility of the fish's movements. Additionally, low light conditions surrounding the tank can further reduce video clarity. The density of the fish population also plays a role; in densely populated tanks, it becomes challenging to capture and accurately analyse the movement of individual fish.

2.3.3 Application of Acoustic Techniques

Overview of Applying Acoustic Techniques

Acoustic techniques are also implemented to monitor fish feeding behaviour. Fish produce sound as acoustic signals while consuming feed during feeding activities [13], [14], [15]. These acoustic signals could aid in monitoring fish feeding behaviour. Hence, a hydrophone is installed in the water to capture these acoustic signals for data processing purposes to determine fish feeding behaviours. Peixoto, Soares and Allen Davis in [13] have discovered that the acoustic signal variances are directly proportional to the amount of feed consumed by the fish. These findings suggest that acoustic signals can be utilized as indicators to monitor the hunger and satiety levels of fish [13].

Capabilities of Applying Acoustic Techniques

Applying acoustic techniques to monitor fish feeding behaviour addresses some challenges encountered with the use computer vision. Acoustic techniques allow automatic fish feeding systems to work smoothly and effectively in fish tanks with low water quality and high fish density. The reason is acoustic techniques rely solely on the acoustic signals produced by the fish during feeding Importantly, such technique can be applied effectively in dark environments or in fish tanks with unclear water [16].

Limitations of Applying Acoustic Techniques

However, there are challenges associated with applying acoustic techniques to determine fish feeding behaviour. A primary weakness is the disturbance caused by surrounding noise in the water. For instance, oxygen supplies in fish tanks often produce micro bubbles that can weaken the transmission of acoustic signals to the hydrophone, hence reducing the effectiveness of this technique. Moreover, the presence of surrounding noise indicates that acoustic techniques are most effective in quieter settings such as external ponds or rearing cages in water, where noise sources are minimal [13].

2.3.4 Application of Vibrational Pattern Analysis

Overview of Applying Vibrational Pattern Analysis

Previous studies have explored the use of vibration data to monitor fish feeding behaviour, focusing particularly on the vibrational patterns of the fish rather than the water. According to [17], an accelerometer is attached to the fish to capture the movements of the fish during feeding activities. The captured accelerometer data is then encoded using an 8-directional Chain Code algorithm to discover the direction and magnitude of the fish movements. Discrete Fourier Transform and Fourier Descriptors are implemented to analyse and summarise the vibrational patterns into a form that is easily classified. As a result, fish feeding behaviours can be monitored and automated based on the vibration analysis and classification results [17].

Capabilities of Applying Vibrational Pattern Analysis

Utilizing accelerometer data to capture the fish movements for analysis purposes provides robust results. This method directly measures the physical activities of the fish, which provides an accurate and immediate reflection of the fish's movements during feeding activities. Furthermore, the application of the Chain Code algorithm and Discrete Fourier Transform allows for advanced processing and complex analysis of movement data in detail. Hence, specific behavioural patterns performed by the fish during feeding activities can be identified precisely.

Limitations of Applying Vibrational Pattern Analysis

One potential weakness of the vibrational analysis system that relies on accelerometers for monitoring fish feeding behaviour is the requirement to physically attach the accelerometer devices to the fish. Attaching accelerometers directly to the fish is an invasive procedure that may cause stress or harm to the fish. This stress can potentially alter their natural behaviours, including feeding, swimming, and interaction with the environment and other fish, which in turn could lead to skewed data. In addition. the extra weight and presence of the device might affect the natural movements and behaviours of the fish.

2.4 Existing Solutions on ArUco Marker-Based Tracking

While ArUco marker-based tracking has not been widely applied in aquaculture, it has gained significant traction in various other fields due to its accuracy, low cost, and ease of implementation. These markers have been extensively used for real-time pose estimation and object tracking in robotics, augmented reality, and drone navigation systems. This section explores the core technology behind ArUco markers and reviews their applications in different domains to highlight their effectiveness. By examining these existing solutions, the potential of adapting ArUco marker-based tracking for fish feeding status monitoring is evaluated as a promising direction for this project.

2.4.1 Overview of ArUco Marker

ArUco markers are a type of binary square fiducial marker widely used in computer vision applications for object detection and pose estimation. Each marker contains a unique ID encoded in a black-and-white grid pattern, allowing it to be identified and distinguished from others in a camera frame. These markers are particularly known for enabling six degrees of freedom (6-DoF) pose estimation, which includes three translational and three rotational parameters relative to a calibrated camera. With the aid of computer vision libraries such as OpenCV, ArUco markers can be detected and processed in real time with high accuracy and minimal computational requirements [18], [19].

2.4.2 Applications in Robotics and Automation

In the domain of robotics, ArUco markers play a crucial role in enabling spatial awareness and precise control. They are commonly employed in mobile robotics for tasks such as localization and navigation, where the robot uses markers placed in the environment to determine its position and orientation. This allows autonomous robots to traverse indoor environments and interact with their surroundings with greater accuracy. In robotic arms and manipulators, ArUco markers are often attached to tools or target objects to facilitate alignment and calibration processes. This ensures that robotic systems can perform actions such as picking, placing, or assembling with high precision [20], [21]. These applications demonstrate the ArUco marker's capability in precise real-time pose estimation and spatial tracking under dynamic conditions, which is directly applicable to this project's goal of monitoring fish movement and surface activity. The robustness and accuracy in robotic calibration align well with tracking floating markers to infer fish behaviour, especially when subtle movements are involved.

2.4.3 Applications in Augmented Reality and Virtual Environments

ArUco markers have also found extensive use in augmented reality (AR) systems, where they function as visual anchors that link digital content to physical spaces. When a marker is detected by a device's camera, the system computes its pose and uses that information to accurately position and render virtual elements, such as 3D models, labels, or animations, within the user's view. This allows for seamless integration of virtual content into the real world, maintaining correct scale and orientation relative to the marker [22]. ArUco markers are particularly valuable in mobile AR applications and head-mounted display systems, where they provide fast, stable tracking even in suboptimal lighting or partially occluded conditions [23]. Their efficiency and reliability make them ideal for interactive educational tools, museum exhibits, design visualization, and other immersive applications that depend on precise visual alignment between digital and real-world objects. AR applications showcase ArUco markers' strengths in pose estimation with varying viewpoints and lighting conditions, which is essential for this project, where markers may move unpredictably on the water surface. The stable tracking ensures consistent pose readings, which are critical for accurately calculating fish movement intensity from the floating markers.

2.4.4 Applications in Drone Navigation and Control

ArUco markers have proven highly effective in drone navigation systems, especially in environments where GPS signals are unreliable or unavailable, such as indoors or in densely built areas. By using onboard cameras, drones can detect ArUco markers strategically placed in the environment to estimate their relative position and orientation in real time. This visual-based localization enables drones to perform complex autonomous maneuvers with precision, such as navigating through pre-defined waypoints, maintaining stable hovering near targets, or adjusting their orientation to align with a landing pad. One of the most notable applications is precision landing, where the marker serves as a visual beacon to guide the drone during descent and ensure accurate placement [24]. Additionally, marker tracking supports tasks like inspection, package delivery, and spatial mapping by enhancing positional awareness. The real-time detection, combined with low hardware requirements and high reliability, makes ArUco markers an accessible and effective solution for enabling intelligent control in modern aerial robotics [25]. Drone navigation emphasizes the use of ArUco markers in unpredictable, real-time environments, similar to the aquatic setting of this project. Just as drones rely on visual markers for precise descent and orientation, this system uses marker movement to determine feeding activity and environmental status. The proven success in drones also highlights ArUco markers' effectiveness in GPS-denied, camera-only systems, which parallels the single-camera, marker-based setup used for multi-tank fish monitoring.

2.4.5 Summary of Advantages of ArUco Marker-Based Tracking

ArUco marker-based tracking systems offer a combination of simplicity, reliability, and cost-effectiveness that makes them attractive for a wide range of real-time monitoring applications. These systems require only standard camera equipment and printed markers, eliminating the need for expensive or specialized hardware. Despite their simplicity, they deliver high accuracy in pose estimation, capturing both spatial position and orientation with precision sufficient for many engineering and research applications. A major advantage of ArUco systems is their scalability; multiple unique markers can be tracked simultaneously within a single camera frame, allowing for the monitoring of several objects or environments concurrently. Their detection remains robust even under changing lighting conditions or when markers are partially occluded or viewed from oblique angles. Moreover, the widespread availability of open-source libraries, such as those offered by OpenCV, facilitates easy integration and implementation without licensing costs. ArUco tracking is also non-intrusive, relying solely on visual input, which makes it ideal for applications where minimal interference is essential, such as biological observation or sensitive manufacturing processes. These characteristics collectively position ArUco marker-based tracking as a versatile and practical solution for real-time monitoring across diverse fields. In the context of this project, their reliability, low cost, and non-intrusiveness make them particularly well-suited for fish feeding status detection and multi-tank monitoring in aquaculture environments.

2.5 Literature Review Summary and Project Justification

The literature reviewed in this chapter has highlighted the critical importance of accurately monitoring fish feeding status in aquaculture settings. It is well-established that fish exhibit observable changes in movement patterns during feeding, and these behaviours can serve as valuable indicators for optimizing feed management and improving overall fish welfare.

Existing monitoring solutions — including deep learning algorithms, computer vision techniques, acoustic systems, and vibrational analysis — demonstrate promising results but also present notable limitations. Deep learning and optical flow-based methods often require significant computational resources and can be affected by environmental noise, such as lighting changes or water turbidity. Acoustic and vibration-based approaches, while suitable for low-visibility conditions, can be invasive or unreliable in noisy aquatic environments. Furthermore, many of these methods struggle with scalability and cost-effectiveness, especially when applied across multiple tanks or large-scale fish farms.

In parallel, ArUco marker-based tracking systems have shown considerable success in other technical fields such as robotics, augmented reality, and drone navigation. These systems provide accurate pose estimation using simple, low-cost hardware and offer real-time performance with minimal computational overhead. Their use in tracking movement and orientation makes them a compelling candidate for aquatic behaviour monitoring.

Based on the limitations of existing approaches and the proven capabilities of marker-based tracking, this project proposes a novel application of ArUco markers for monitoring fish feeding status. By attaching markers to floating objects and analysing their pose changes caused by surface disturbances during feeding, the system aims to deliver a non-invasive, scalable, and cost-efficient monitoring solution. The literature supports both the need for improved feeding monitoring techniques and the technical feasibility of the proposed approach, thereby forming a strong foundation for our solution.

CHAPTER 3 SYSTEM METHODOLOGY

This chapter outlines the system methodology by presenting an overview of the system, along with the use case diagram, system architecture diagram, and activity diagram. These visual models collectively describe how the system operates, from user interaction to backend processing and final output generation.

3.1 Overview of the System

The proposed system is designed to monitor fish feeding status in multiple aquaculture tanks by analysing water surface movement using ArUco marker tracking combined with machine learning-based classification. ArUco markers are affixed to floating objects, and their motion patterns reflect the water disturbances caused by fish activity during feeding. By tracking pose changes over time, the system can determine whether the fish are actively feeding or have reached satiety.

The system architecture includes four main components: a camera, a backend server, a trained deep learning model, and a web-based frontend interface. The camera captures live video of multiple tanks. The backend processes each frame to detect ArUco markers and estimate their pose in the form of Euler angles (yaw, pitch, and roll) and translation values (x, y, z). These pose values and translation values are collected in a sliding time window and passed to an LSTM model, which classifies fish activity as active or inactive. Based on the duration of the activity state, the system generates a feeding recommendation — to continue or stop feeding.

Beyond monitoring feeding behaviour, the system also detects air pump status by analysing specific marker movement patterns generated by water turbulence from the pump. In addition, the system is capable of tracking water level changes by analysing the vertical position of the marker in the camera view, allowing users to detect abnormal drops or rises in water level.

All predictive outputs — including fish activity status, duration of fish activity status, feeding recommendation, air pump status, and water level — are sent to a React-based frontend interface where users can monitor multiple tanks in real time. The system is scalable and operates using a single camera, offering a cost-effective, non-invasive solution to support intelligent decision-making in aquaculture management.

3.2 Use Case Diagram

This section presents the use case diagram, which illustrates the interactions between the user and the system, including key functionalities such as monitoring fish activity status, monitoring duration of fish activity status, receiving feeding recommendations, monitoring air pump status, monitoring water level and accessing real-time tank data. Figure 3.2.1 below shows the use case diagram.

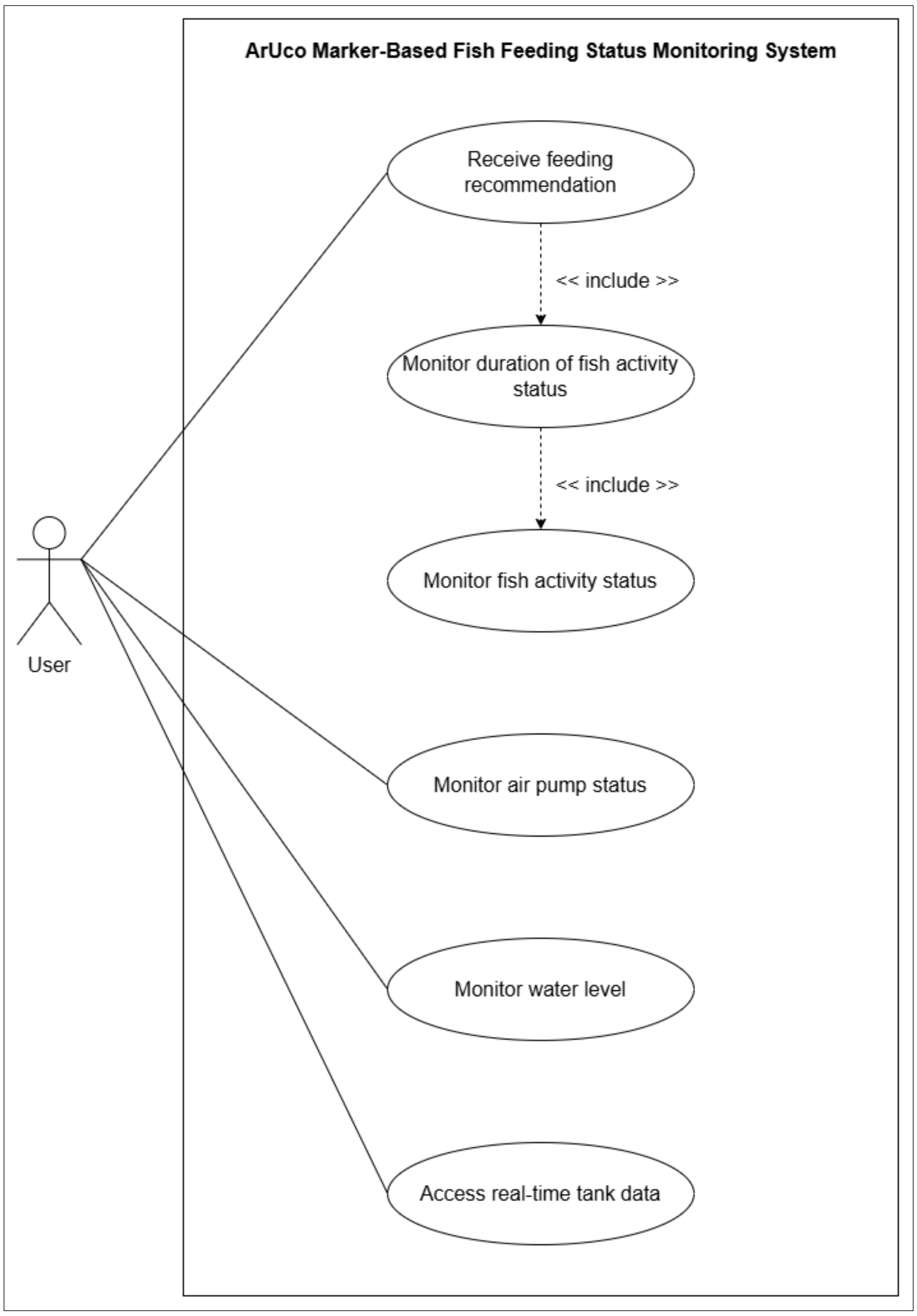

ArUco Marker-Based Fish Feeding Status Monitoring System
Receive feeding recommendation
<< include >>
Monitor duration of fish activity status
<< include >>
Monitor fish activity status
User
Monitor air pump status
Monitor water level
Access real-time tank data

Figure 3.2.1. Use case diagram of the ArUco marker-based fish feeding status monitoring system

Figure 3.2.1 above presents the use case diagram, which models the interaction between the primary actor (user) and the system. The user, typically a fish farm operator, interacts with the system through a web-based frontend that communicates with a Python-based backend, where real-time data processing and classification take place.

The system is designed around key use cases that represent the external functionalities experienced by the user, while also implying complex internal operations handled autonomously by the system:

- **Monitor fish activity status**

  The user is able to view whether fish in a tank are active or inactive. This status is derived from the movement of floating ArUco markers, whose pose (Euler angles and translation values) is estimated in real time. The backend processes these pose values in a sliding window and applies a trained LSTM model to classify fish activity status.

- **Monitor duration of fish activity status**

  The system continuously tracks the duration of each detected activity state (active or inactive). This temporal tracking is a key step toward feeding decision-making and is not directly controlled by the user. It is included as an internal component of the higher-level feeding recommendation process.

- **Receive feeding recommendation**

  Based on the classification results and the duration of fish activity states, the system infers whether fish are still feeding or have reached satiety. Once the predefined threshold is met, the system generates a recommendation to stop or continue feeding. This output is then made visible to the user via the frontend.

- **Monitor air pump status**

  The system evaluates movement patterns specific to surface turbulence caused by air pump activity. The motion of the ArUco marker, influenced by water agitation, is tracked and converted into pose data. These pose sequences are buffered into a sliding window and passed into an LSTM model trained specifically to classify air pump status. The model predicts whether the air pump is switched on or off based on the temporal patterns of marker movement. The result is displayed to the user alongside other real-time tank data via the frontend interface.

- **Monitor water level**

  The vertical position of the ArUco marker in the camera frame is used to passively monitor the water level in each tank. As the marker floats on the water surface, its position reflects changes in water height. The system continuously tracks this vertical position and displays the real-time water level to the user through the frontend interface. Unlike fish activity and air pump status, this feature does not rely on predefined thresholds or classification but instead provides live positional data to support manual observation and decision-making.

- **Access real-time tank data**

  All predictions and raw values – Euler angles, activity status, duration of fish activity status, water level – are made available through a centralized frontend interface. This allows the user to monitor multiple tanks simultaneously and make informed management decisions.

Also, Figure 3.2.1 illustrates internal dependencies using <<include>> relationships, notably between “Receive feeding recommendation”, “Monitor duration of fish activity status”, and “Monitor fish activity status”. These inclusions reflect the layered processing pipeline in the backend system. While the user does not initiate or control these processes directly, they benefit from their outputs through a seamless frontend interface.

3.3 System Architecture Diagram

This section presents the system architecture of the ArUco marker-based fish feeding status monitoring system. The architecture outlines how the hardware and software components interact to enable real-time monitoring of fish feeding status, air pump operation, and water level. Figure 3.3.1 below shows the system architecture diagram.

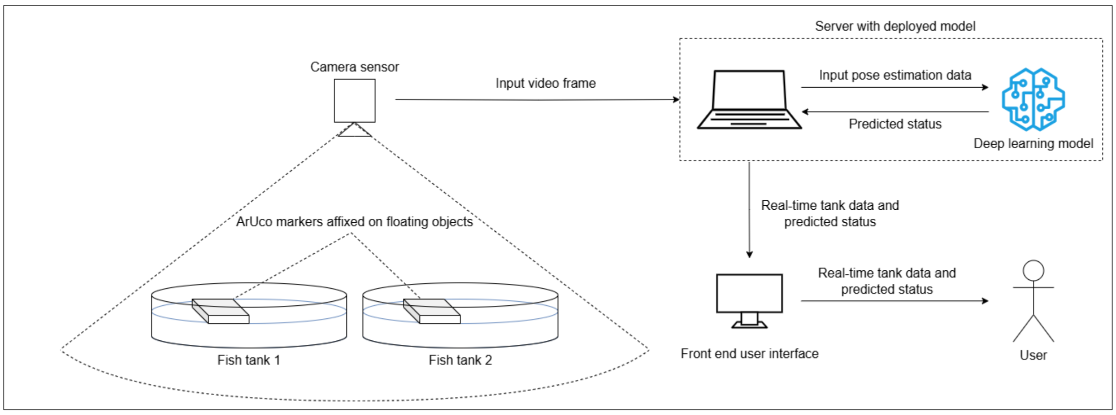

Server with deployed model
Camera sensor
Input video frame
Input pose estimation data
Predicted status
Deep learning model
Real-time tank data and predicted status
ArUco markers affixed on floating objects
Real-time tank data and predicted status
Fish tank 1
Fish tank 2
Front end user interface
User

Figure 3.3.1. System architecture diagram of ArUco marker-based fish feeding status monitoring system

As shown in Figure 3.3.1, the system comprises four major components:

- Camera sensor
- Server with deployed model
- Frontend user interface
- User

The camera is mounted above the tanks and is responsible for capturing continuous video footage that includes floating ArUco markers placed in each tank. These markers are affixed to floating objects that move in response to fish activity, air pump turbulence, or water level changes.

The server receives the video stream and performs several processing steps:

1. Marker detection and pose estimation.
2. Conversion of marker orientation into Euler angles (yaw, pitch, roll) and translation values (x, y, z).
3. Buffering of pose data into sliding windows for time-series modelling
4. Classification using LSTM models to predict both fish activity status and air pump status based on temporal motion patterns.
5. Analysis of duration of fish activity status using a predefined threshold to recommend stop feeding or continue feeding.

In parallel, the vertical position of the marker in the camera frame is tracked to calculate real-time water level readings.

Once predictions and pose data are generated, they are sent to the frontend user interface, developed using React. The frontend displays all relevant tank metrics, including:

- Fish activity status (Active/Inactive)
- Duration of fish activity status
- Feeding recommendation (Continue feeding/Stop feeding)
- Air pump status (On/Off)
- Real-time water level
- Euler angles from pose estimation

The user accesses this data through the interface to make informed management decisions. The system supports multi-tank monitoring using a single camera, reducing hardware cost and improving scalability for commercial aquaculture environments.

3.4 Activity Diagram

This section presents the activity diagram of the ArUco marker-based fish feeding status monitoring system. The diagram illustrates the sequential flow of operations carried out by the system; from the moment a video frame is captured to the point where monitoring results are displayed to the user. The system handles both fish activity detection and air pump status classification using the same input data pipeline, while also monitoring water level as a continuous visual metric. Figure 3.4.1 below shows the activity diagram.

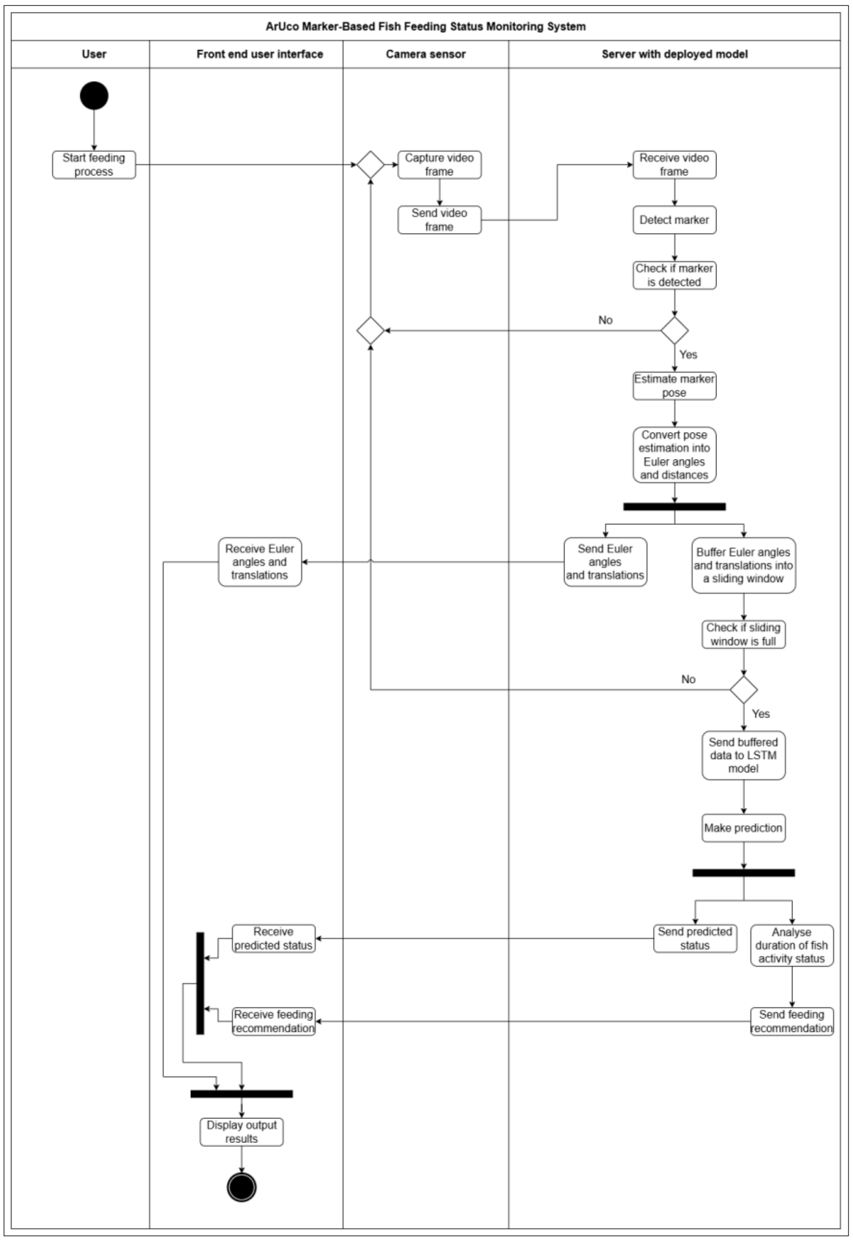

ArUco Marker-Based Fish Feeding Status Monitoring System
User
Front end user interface
Camera sensor
Server with deployed model
Start feeding process
Capture video frame
Send video frame
Receive video frame
Detect marker
Check if marker is detected
No
Yes
Estimate marker pose
Convert pose estimation into Euler angles and distances
Send Euler angles and translations
Buffer Euler angles and translations into a sliding window
Receive Euler angles and translations
Check if sliding window is full
No
Yes
Send buffered data to LSTM model
Make prediction
Send predicted status
Analyse duration of fish activity status
Send feeding recommendation
Receive predicted status
Receive feeding recommendation
Display output results

Figure 3.4.1. Activity diagram of ArUco marker-based fish feeding status monitoring system

The process begins when the user initiates the feeding process. The camera sensor captures video frames of the fish tanks, which include floating ArUco markers. These frames are transmitted to the backend server, where pose estimation is performed using computer vision techniques.

If a marker is successfully detected in a frame, its 6-DoF pose is estimated. The pose is then converted from quaternion format into Euler angles (yaw, pitch, roll), and the translation values (x, y, z) are extracted. The Z-distance specifically corresponds to the marker’s vertical position relative to the camera, which is used to monitor the water level in real time.

The Euler angles and Z-distance are then transmitted back to the frontend for real-time display. At the same time, the Euler angels and the translation values are buffered into a sliding window on the server side. When the buffer is filled, the sequence is passed into a trained LSTM model, which generates predictions for both fish activity status (Active/Inactive) and air pump status (On/Off) based on the motion patterns captured in the time-series data.

The output fish activity status is analysed to track how long the fish have remained inactive or active. If the duration exceeds a predefined threshold, the system recommends stopping feeding or continuing feeding. Both the predicted statuses and the feeding recommendation are sent to the frontend.

Finally, the frontend interface presents all relevant data to the user, including:

- Real-time Euler angles and Z- distance to represent water level
- Fish activity status
- Air pump status
- Duration of fish activity status
- Feeding recommendation

This process runs continuously in a loop, enabling real-time monitoring and intelligent feeding decision support during aquaculture operations.

CHAPTER 4 SYSTEM DESIGN

This chapter presents the detailed system design of the ArUco marker-based fish feeding status monitoring system. It describes the hardware and software components used, how they are integrated, and the overall architecture of the system. Key subsystems — including pose estimation, data preprocessing, and time-series classification using an LSTM model — are explained in depth. Special design considerations such as throttling, quaternion-to-Euler angle conversion, and sliding window buffering are also discussed, as they play critical roles in ensuring accurate and efficient predictions. The interaction between the backend and frontend is outlined to illustrate how real-time data is processed and delivered to the user interface. This chapter aims to provide the necessary design insights to replicate or extend the system in similar applications.

4.1 Hardware and Software Components

This section outlines the hardware and software components used to implement the ArUco marker-based fish feeding status monitoring system. The system combines video capture, computer vision, deep learning, and a web-based user interface to provide real-time monitoring of fish feeding status, air pump operation, and water level in multiple aquaculture tanks.

4.1.1 Hardware Components

To implement the system, three main hardware components are required as follows:

- **Camera sensor with mounting setup**

  A smartphone is mounted above the fish tanks and acts as the primary video capture device. The phone uses the DroidCam application to stream live video to the backend server over the local network. This setup provides a stable top-down view of the tank, suitable for consistent ArUco marker detection and pose estimation. The video stream is accessed in real time, with resolution and frame rate balanced for clarity and performance.

- **Floating ArUco markers**

  ArUco markers are printed and affixed to lightweight floating objects made of foam. These floats are designed to remain on the water surface and respond naturally to disturbances caused by fish movement, aeration from the air pump, and changes in water level. The motion of the marker serves as the observable input for determining fish behaviour and tank conditions.

- **Local server**

  A personal computer is used to host the backend system. It runs the pose estimation algorithm, buffers time-series data, and executes the LSTM classification model to predict fish activity and air pump status. The server also handles real-time communication with the frontend via WebSocket. The machine requires moderate processing capabilities to support live video frame analysis and continuous inference without performance bottlenecks.

4.1.2 Software Components

The software architecture of the system consists of multiple modules that handle video processing, pose estimation, time-series classification, real-time communication, and frontend visualization. The system is primarily developed using Python for backend processing and React for the user interface. The software components and modules that are included are as follows:

- **React (Frontend user interface)**

  The frontend is built using the React JavaScript framework. The interface is designed for clarity and usability, enabling the user to quickly assess tank conditions and make informed decisions. It receives real-time data from the backend and dynamically renders:

  - Euler angles (yaw, pitch, roll)
  - Z-distance (water level)
  - Fish activity status
  - Air pump status
  - Duration of current fish activity status
  - Feeding recommendation

- **Python (Backend)**

  Python serves as the core backend language. It manages video frame handling, ArUco marker detection, pose estimation, data buffering, LSTM model inference, and real-time communication with the frontend. Python’s extensive ecosystem enables rapid development and integration of computer vision and machine learning modules.

- **OpenCV**

  OpenCV is used for ArUco marker detection and pose estimation. The module identifies the marker in each video frame and computes its pose, which includes both rotation (as a rotation vector or quaternion) and translation (x, y, z position) relative to the camera. In this system, rotation data is initially obtained in quaternion format, a four-dimensional representation of orientation. Quaternions are preferred over Euler angles or rotation matrices in the pose estimation stage because they are mathematically stable, do not suffer from gimbal lock, and can smoothly interpolate rotations. This makes them highly suitable for tracking orientation changes over time, especially when working with small but continuous rotational movements — as seen with floating ArUco markers on water. Once obtained, the quaternions are converted into Euler angles (yaw, pitch, and roll) for easier interpretation, display, and input into the classification model. Additionally, the Z-distance component of the translation vector is extracted and used to estimate water level, as it directly reflects the vertical displacement of the marker within the camera's coordinate system.

- **NumPy & SciPy**

  These libraries are used for numerical operations, matrix manipulation, and sliding window buffering of pose data sequences. They provide efficient tools for preparing model input and post-processing model output.

- **LSTM Model (Keras with TensorFlow Backend)**

  The system uses a Long Short-Term Memory (LSTM) model developed using Keras, a high-level neural networks API running on top of TensorFlow. The model is trained on time-series pose data to classify two outputs: fish activity status (Active/Inactive) and air pump status (On/Off). Keras provides a streamlined interface for defining, training, and evaluating deep learning models, making it well-suited for rapid development and experimentation.

- **Throttling Logic**

  A throttling mechanism is implemented in the backend to control the frequency of model predictions. Instead of performing inference on every frame, the system triggers prediction only at fixed intervals to reduce computational load and avoid redundant processing.

- **WebSocket Communication**

  A WebSocket protocol is used for efficient, real-time communication between the backend and frontend. It streams a continuous feed of pose data (Euler angles and Z-distance), classification results, and feeding recommendations to the user interface.

4.2 System Block Diagram

This section presents the system block diagram of the ArUco marker-based fish feeding status monitoring system. The system captures and processes real-time motion data from ArUco markers floating on the tank surface to determine fish activity status, air pump operation, and feeding recommendations. It consists of several modular components, each performing a specific task within the data processing and monitoring pipeline. Figure 4.2.1 below illustrates the system block diagram of the ArUco marker-based fish feeding status monitoring system.

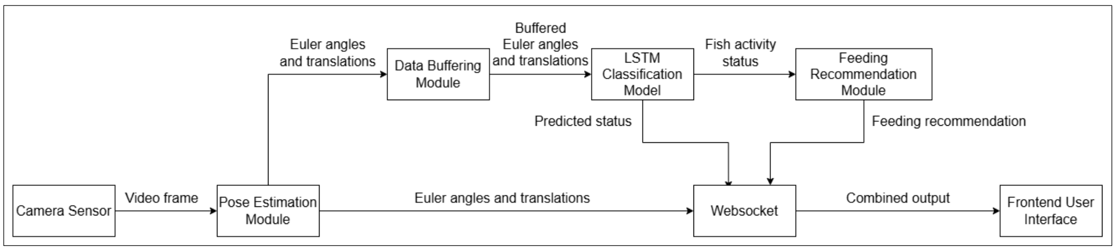


Figure 4.2.1. System block diagram of ArUco marker-based fish feeding status monitoring system

Based on Figure 4.2.1, the process begins with a camera sensor, which continuously captures video frames of the water surface. These frames are processed by the Pose Estimation Module, which detects ArUco markers and estimates their 6-DoF pose. The pose is calculated using the quaternion method to ensure rotational stability, and is then converted to Euler angles (yaw, pitch, roll) and translation values (x, y, z) for feature extraction. The Z-axis translation is also extracted from the pose data to represent the vertical distance, which reflects the water level.

The Euler angles and translation values are sent along two paths:

- Directly to the Frontend User Interface via WebSocket, allowing users to observe live movement and water level.
- Simultaneously to the Data Buffering Module, where the data is accumulated in a sliding window format.

Once the buffer reaches a sufficient length, it is passed into the trained LSTM Classification Model, which predicts:

- Fish activity status (Active or Inactive)
- Air pump status (On or Off)

To reduce computational load and noise, a throttling strategy is implemented, where predictions are only triggered at fixed intervals rather than on every frame.

The predicted fish activity status is further processed by the Feeding Recommendation Module, which monitors the duration of fish activity status and determines whether feeding should be continued or stopped based on predefined thresholds. This logic helps prevent overfeeding based on sustained behavioural trends.

The WebSocket serves as the central communication channel between the backend and frontend. It transmits all relevant outputs — Euler angles, vertical distance, predicted statuses, duration of fish activity status, and feeding recommendations — as a combined data stream to the Frontend User Interface. The frontend presents these metrics in real time, enabling the user to monitor and manage multiple tanks efficiently.

This architecture enables reliable, real-time monitoring of fish behaviour and tank conditions using only a single camera, making it suitable for scalable deployment in commercial aquaculture environments.

4.2.1 Breakdown of System Block Diagram

To better understand how each component functions within the system, the key processing modules are summarized below:

- **Camera Sensor**

  A smartphone camera, mounted directly above the fish tanks, is used as the primary video capture device. It streams live video to the backend server using the DroidCam application over a local network. This setup provides a stable, top-down perspective of the tank surface, ensuring consistent visibility of the floating ArUco markers. The use of DroidCam enables a cost-effective and easily deployable alternative to conventional IP or USB cameras.

- **Pose Estimation Module**

  The video stream from the camera is processed frame by frame by the pose estimation module. This module detects ArUco markers using OpenCV and computes their full 6-DoF pose, which includes both rotation and translation vectors. The rotation is initially calculated in quaternion form to ensure numerical stability and avoid issues like gimbal lock. It is then converted into Euler angles — yaw, pitch, and roll — for further analysis and interpretability. In addition to orientation, the module extracts translation values (x, y, z), where the Z-distance (vertical displacement) is specifically used to monitor water level.

- **Data Buffering Module**

  To prepare time-series data for classification, a fixed-length sliding window is used to buffer sequences of pose data — specifically the Euler angles and translation values. This module ensures that only structured, temporal input is passed into the classification model. The use of buffering helps capture movement trends over time, rather than analysing isolated frames.

- **LSTM Classification Model**

  The buffered sequences are fed into a deep learning model based on Long Short-Term Memory (LSTM) architecture, developed using Keras. The model is trained to predict two classification outputs: fish activity status (Active or Inactive) and air pump status (On or Off). To avoid excessive computation, a throttling mechanism is applied so that predictions are only made at defined frame intervals, rather than continuously on every frame. This improves efficiency while preserving temporal accuracy.

- **Feeding Recommendation Module**

  Once fish activity predictions are obtained from the classification model, they are passed to the Feeding Recommendation Module, which operates based on time-based behavioural analysis. This module continuously monitors the duration of the predicted activity status and compares it to a predefined threshold. If the active state persists beyond the threshold, the system automatically recommends continuing the feeding process. Conversely, if the inactive state persists for longer than the threshold, a recommendation to stop feeding is issued. This decision-making logic helps reduce the risk of overfeeding by aligning feed delivery with real-time fish behaviour, thus enabling more efficient and responsive feed management.

- **WebSocket Communication Layer**

  To enable real-time interaction between the backend and frontend, the system uses a WebSocket protocol, which operates over TCP. This communication layer streams all relevant outputs — including Euler angles, Z-distance, prediction results, duration of fish feeding status and feeding recommendations — as a continuous data stream to the frontend interface. WebSocket ensures low-latency, bidirectional communication, allowing the user to monitor tank status without delay.

- **Frontend User Interface (Real-Time Display)**

  The frontend, developed using React, receives Euler angles, Z-distance, classification results, duration of fish activity status and feeding recommendation from the backend via WebSocket. It displays real-time values for marker orientation and water level, as well as system predictions including fish activity status, air pump status, and feeding recommendations. The interface also visually tracks and displays the duration of the current fish activity state, providing users with contextual insight into the system's decision-making logic.

4.3 LSTM Model Development

This section details the development process of the classification model used to detect fish activity status and air pump status. It begins with data collection, where pose information is extracted from floating ArUco markers during real feeding sessions. The data is then labelled, pre-processed, and formatted into time-series sequences using a sliding window technique. These sequences are used to train a deep learning model based on Long Short-Term Memory (LSTM) architecture, which is well-suited for temporal behaviour recognition. The section also discusses model architecture, training configurations, and evaluation strategies, culminating in the selection of the most accurate and efficient model for real-time deployment in the system.

4.3.1 Data Collection and Labelling

To train the classification model, two types of behaviour were targeted: fish feeding activity and air pump operation status. Each was collected and processed separately using the same data acquisition pipeline, which extracted pose information from a floating ArUco marker in the tank. The recorded data includes Euler angles (yaw, pitch, roll) and translation values (x, y, z), derived through a quaternion-based pose estimation process using OpenCV.

Fish feeding data was collected by initiating a controlled feeding session. A batch script (.bat file) was executed to start real-time collection of pose data (Euler angles and translation vectors) immediately before, during, and after the feeding process. This ensured that the dataset captured the full behavioural transition from active feeding to post-feeding inactivity.

The movement of the floating ArUco marker, caused by fish surfacing or swimming near the surface during feeding, was reflected in the recorded pose values. The resulting time-series data served as a proxy for fish activity levels during feeding events.

For air pump activity, data collection followed a similar approach. Pose data was collected while the air pump was manually turned on and off at controlled intervals. When the pump was running, the resulting turbulence on the water surface caused distinctive movement patterns in the floating marker, which were captured as fluctuations in Euler angles and translation values. In contrast, the stillness during pump-off periods produced noticeably stable marker motion.

4.3.2 Manual Data Labelling

The collected pose data was then manually labelled based on synchronized observation of the recorded video footage. During review, the elapsed time in the pose data was matched with the corresponding timestamps in the video. Binary labels were then assigned as follows:

- Fish Activity Status
    - 1 for Active (during observable feeding behaviour)
    - 0 for Inactive (before or after feeding when fish were calm)
- Air Pump Status
    - 1 for On (when pump was operating)
    - 0 for Off (when pump was not operating)

This labelling method ensured that each window of pose data used in model training had an accurate and consistent ground truth label derived from visual validation.

4.3.3 Data Preprocessing

To prepare the collected pose data for time-series classification, a sliding window approach was applied. Rather than analysing individual frames in isolation, this method groups consecutive pose data points into fixed-length sequences, enabling the model to capture temporal dependencies and learn motion dynamics over time. This temporal structuring is essential for modelling real-world behaviour, where feeding activity and aeration patterns unfold gradually rather than appearing in isolated snapshots.

Each window contains a sequence of consecutive pose samples — specifically the Euler angles (yaw, pitch, roll) and the translation values (x, y, z) — collected at consistent intervals. These sequences serve as input features to the LSTM model, which is designed to process sequential data by retaining memory of previous time steps. This allows the model to identify patterns of change and movement trends, rather than relying solely on individual frame characteristics.

The window advances by a predefined step size to create overlapping segments, ensuring continuity between sequences and improving training diversity. The following hyperparameters were used:

- Window size: The number of consecutive frames (time steps) per sequence.
- Step size: The number of frames the window shifts forward to form the next sequence.

These parameters were tuned experimentally. A larger window captures more context, allowing the model to detect longer behavioural trends, while a smaller window increases responsiveness but may be more sensitive to noise. The selected configuration represents an optimal balance between accuracy and real-time prediction.

Each sequence was labelled according to the status of the final frame (either fish activity or air pump state), preserving temporal alignment between inputs and observed ground truth. Through this process, raw continuous pose data was transformed into structured, labelled time-series sequences. After normalization, these were used to train and evaluate the LSTM model, allowing it to effectively learn motion patterns and behaviour transitions over time.

4.3.4 Model Architecture and Training

The classification model used in this project is based on a Long Short-Term Memory (LSTM) architecture, implemented using the Keras API on top of TensorFlow. The model is designed to classify short sequences of pose data — specifically Euler angles and translation vectors — into binary states representing either fish activity status (Active/Inactive) or air pump status (On/Off), using time-series motion patterns.

The final LSTM model consists of the following layers:

- Masking Layer: Masks padded values in input sequences to prevent them from influencing learning.
- First LSTM Layer: 64 units, configured to return sequences for deeper temporal representation.
- Second LSTM Layer: 32 units, used to extract higher-level temporal features.
- Dense Layer: 16 neurons with ReLU activation to learn abstract representations.
- Output Layer: A single neuron with a sigmoid activation function to produce binary classification output.

The model was compiled with configurations tailored for binary classification. Binary cross-entropy was used as the loss function, which is appropriate for tasks involving two output classes, such as distinguishing between active and inactive fish behaviour or air pump On and Off states. The Adam optimizer was selected for its adaptive learning rate capabilities and efficient convergence properties, making it well-suited for training deep learning models on time-series data.

Model training was then performed using 5-fold Stratified Cross-Validation to ensure balanced representation of both classes in each fold. For each fold:

- The model was trained for 100 epochs with a batch size of 32.
- Validation performance was evaluated after training.
- Predictions were thresholded at 0.5 to convert probabilities into binary class labels.

4.3.5 Model Evaluation

For each fold after training, the following evaluation metrics were recorded:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- ROC Curve and AUC Score

The model that achieved the highest accuracy across all folds was selected as the final model for deployment. A confusion matrix and ROC curve were generated using predictions aggregated across all folds to visualize classification performance and error distribution.

Prior to training, class distributions were visualized using bar plots to ensure that the dataset was sufficiently balanced between the binary classes for both fish activity status and air pump status. This step was crucial to avoid training a biased model and to ensure fair evaluation metrics.

To identify the optimal temporal window for classification, the model was trained and evaluated using several combinations of window size (in seconds) and step size (in number of frames). These hyperparameters control how many frames are included in each time-series input and how much overlap exists between consecutive samples. Separate sets of combinations were used for each classification task:

- Fish Activity Status:
    - (30, 30), (30, 15), (15, 30), (15, 15), (10, 30), and (10, 15)
- Air Pump Status:
    - (30, 30), (30, 15), (15, 30), and (15, 15)

These combinations were chosen to evaluate the impact of both longer and shorter sequences on the model's performance. Larger windows provide more context and are expected to improve classification accuracy for subtle patterns such as gradual slowing down of feeding, while smaller windows offer faster responsiveness, which is important for real-time prediction. Likewise, smaller step sizes introduce greater overlap between sequences, which increases the data volume and potentially improves generalization, while larger steps reduce redundancy and speed up inference. By comparing performance across these combinations, the most suitable configuration for each classification task was identified and used in the final deployed model.

4.3.6 Model Deployment

Once the optimal LSTM model was selected through cross-validation, it was saved and integrated into the backend system for real-time deployment. The final model was stored in .keras format, while the corresponding data scaler used for feature standardization was saved separately using joblib to ensure consistent preprocessing during inference.

At runtime, the backend continuously collects pose data from the pose estimation module in real time. The Euler angles and translation values are buffered using a sliding window mechanism. Once the window is filled with sufficient data points, the sequence is reshaped and normalized using the saved StandardScaler, ensuring that the live input matches the format of the training data. The normalized sequence is then passed into the deployed LSTM model to generate predictions. The model outputs a probability between 0 and 1, which is thresholded at 0.5 to determine the final binary classification as follows:

- Fish activity status: Active (1) or Inactive (0)
- Air pump status: On (1) or Off (0)

Predictions are made at predefined intervals according to the throttling strategy implemented in the backend, which controls how frequently inference is triggered to maintain system efficiency.

The predicted status is transmitted to the frontend interface via WebSocket, along with the corresponding pose data and water level reading. This enables the user to monitor classification results in real time and receive timely feeding recommendations based on the inferred fish behaviour.

This deployment approach ensures that the system remains lightweight, responsive, and consistent with the training environment, while supporting live decision-making in aquaculture management.

4.4 System Components Interaction Flow

This section explains how the hardware and software components interact within the complete system pipeline after the trained LSTM model has been finalized and deployed. Each component plays a defined role in facilitating real-time monitoring of fish feeding activity, air pump operation, and tank conditions.

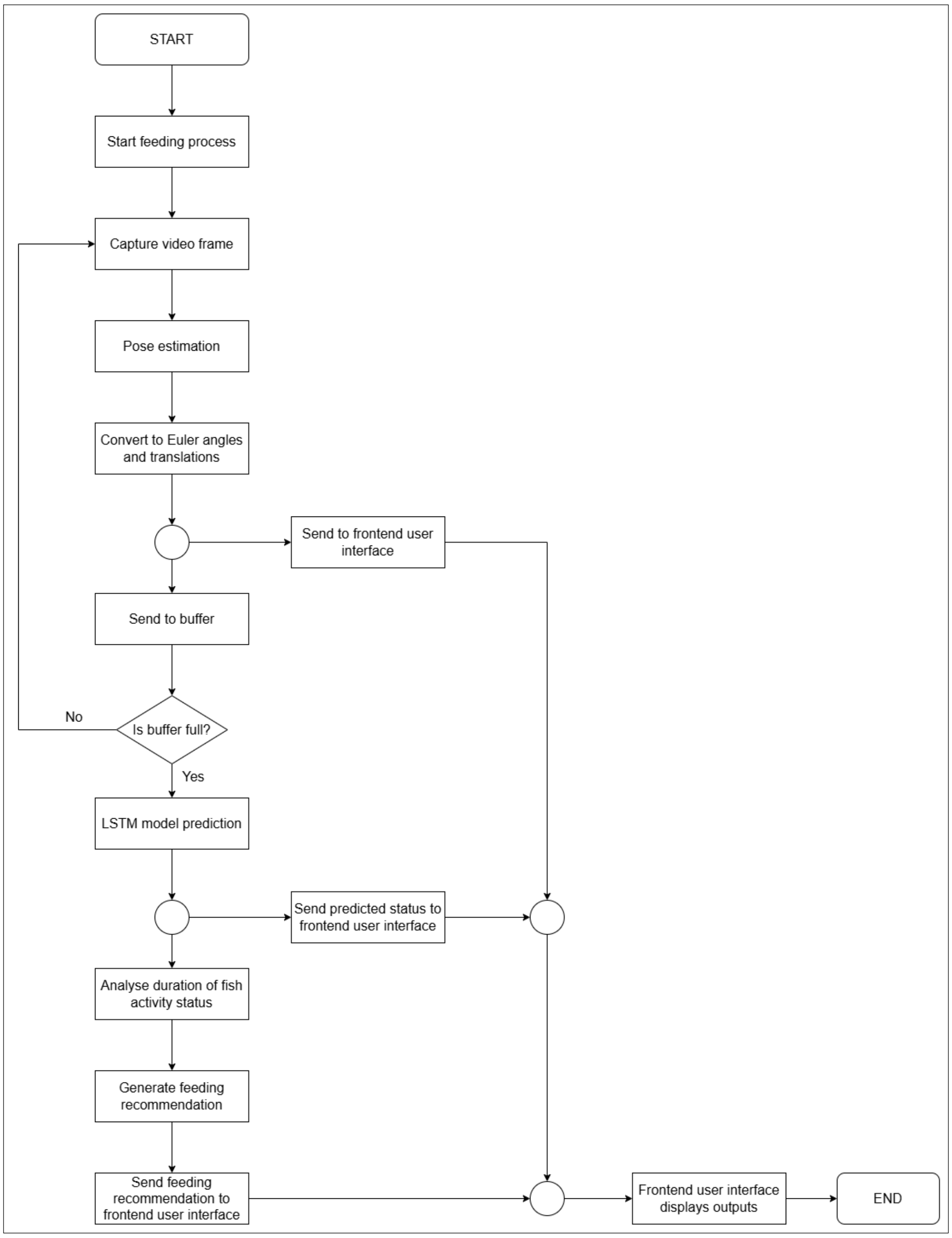

START
Start feeding process
Capture video frame
Pose estimation
Convert to Euler angles and translations
Send to frontend user interface
Send to buffer
No
Is buffer full?
Yes
LSTM model prediction
Send predicted status to frontend user interface
Analyse duration of fish activity status
Generate feeding recommendation
Send feeding recommendation to frontend user interface
Frontend user interface displays outputs
END

Figure 4.4.1. System flowchart of ArUco marker-based fish feeding status monitoring system

Referring to Figure 4.4.1, The system begins with a smartphone mounted above the tanks, which functions as the camera sensor. Using the DroidCam application, the smartphone streams live video frames to the backend server. Upon receiving the video stream, the pose estimation module processes each frame using OpenCV to detect ArUco markers and estimate their 6-DoF pose. The orientation is initially calculated in quaternion format for rotational stability and then converted to Euler angles (yaw, pitch, roll), along with translation values (x, y, z). The Z-axis translation specifically serves as a proxy for estimating water level. These pose values are sent via WebSocket to the frontend user interface, built using React, which displays real-time marker orientation and water level data.

Simultaneously, the pose data is routed to the data buffering module, where sequences are accumulated in a fixed length sliding window. Once the buffer is filled, the sequence is normalized using the previously saved scaler and passed into the LSTM classification model, developed using Keras. This model predicts the fish activity status (Active or Inactive) and air pump status (On or Off) based on temporal motion patterns. To ensure computational efficiency, a throttling mechanism is implemented to control the prediction frequency.

The predicted fish activity status is further processed by the feeding recommendation module, which tracks the duration of fish activity status and generates an output indicating whether feeding should continue or stop. All relevant outputs — including Euler angles, Z-distance, predicted statuses, duration of fish activity status and feeding recommendation — are packaged into a unified data stream and transmitted back to the frontend interface via WebSocket, enabling the user to observe tank behaviour and receive actionable insights in real time.

This interaction flow ensures tight coordination between modules, supporting a scalable, lightweight, and responsive solution for aquaculture tank monitoring using a single camera.

CHAPTER 5 SYSTEM IMPLEMENTATION

This chapter outlines the practical implementation of the ArUco marker-based fish feeding status monitoring system. It provides a detailed account of the hardware and software configurations required to develop and deploy the system, including camera positioning, data acquisition, and backend integration. The chapter begins by describing the setup of both the model training environment and the real-time monitoring system. It then covers the full pipeline for training and evaluating the LSTM classification model — from manual data labelling to preprocessing, training, and visualization of performance metrics.

Following model development, the deployment process is discussed, highlighting how the trained model was integrated into the live monitoring system and connected to the frontend interface. System operation procedures are described, including how feeding sessions are initiated and monitored using the interface. Screenshots and descriptions of the user interface are included to demonstrate how users interact with the system and view predictions in real time. Finally, this chapter addresses key implementation challenges encountered during development and the strategies employed to overcome them.

5.1 Hardware Setup

This section presents the physical setup of the hardware to implement the ArUco marker-based fish feeding status monitoring system. The physical setup of the system is designed to be simple, low-cost, and practical for real aquaculture environments. The key hardware components include a smartphone camera, floating ArUco markers, and a controlled tank environment located at the campus aquaculture facility.

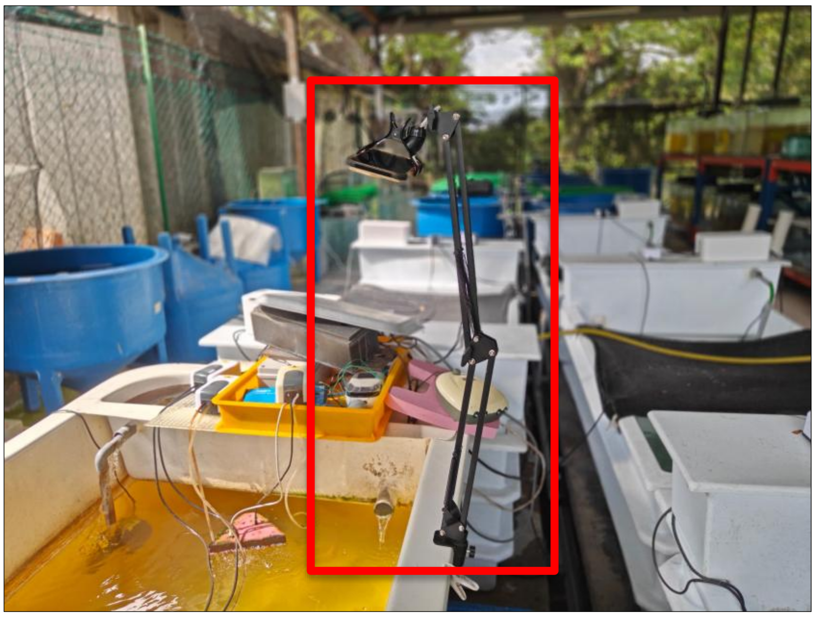


Figure 5.1.1. Mounted smartphone above fish tank

As shown in Figure 5.1.1, a smartphone is mounted directly above the tanks to function as the camera sensor. The phone is positioned at a fixed height to provide a consistent, top-down view of the water surface. This positioning ensures that the floating ArUco markers always remain within the camera's field of view, even as they move due to fish activity or aeration. The smartphone uses the DroidCam application to stream live video frames to the backend system over the local network. This method provides a reliable and cost-effective alternative to conventional webcams or surveillance cameras, requiring no special hardware interfaces.

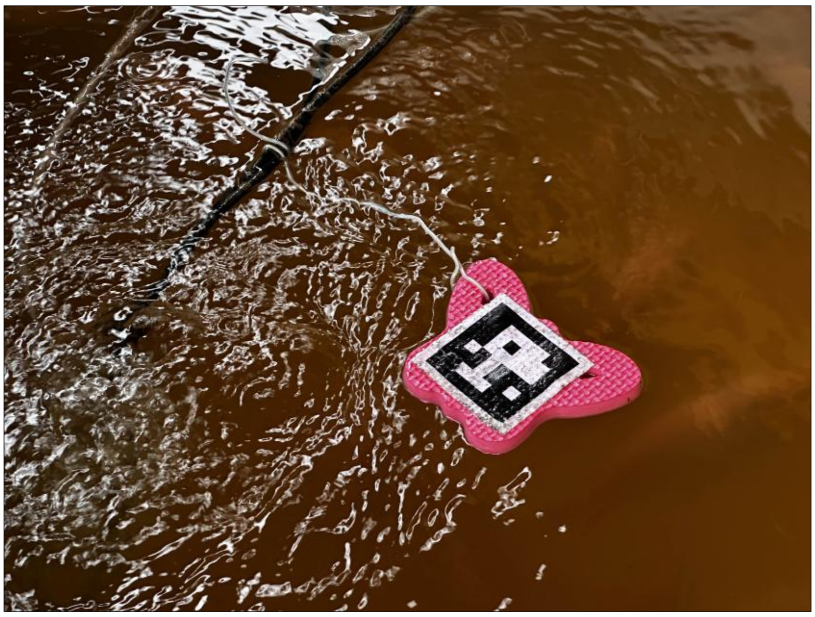


Figure 5.1.2. Printed ArUco marker affixed to float placed on water surface

As shown in Figure 5.1.2, to detect fish activity and surface movement, printed ArUco markers are affixed to lightweight foam floats and placed on the surface of each tank. These markers move in response to surface agitation caused by either feeding behaviour or aeration, allowing the system to indirectly interpret environmental changes through marker motion.

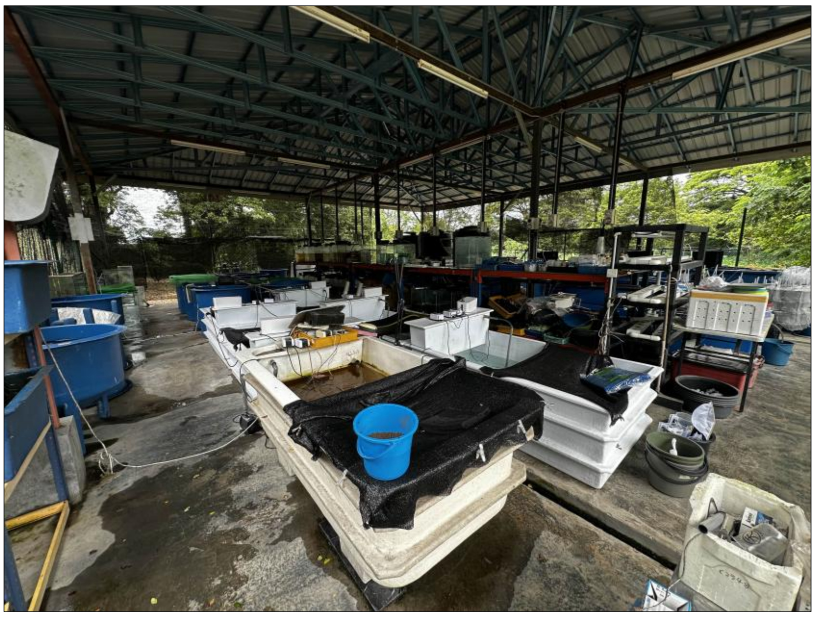


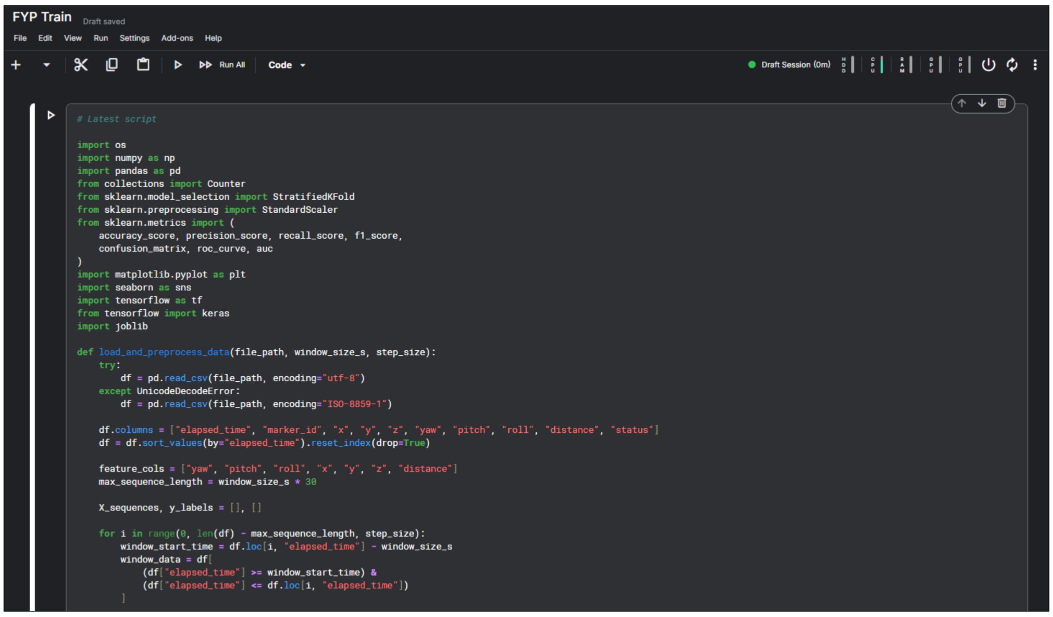
Figure 5.1.3. Environment of controlled aquaculture facility

As illustrated in Figure 5.1.3, the tanks themselves are part of a controlled aquaculture facility and house red hybrid tilapia (Oreochromis sp.), selected for their commercial relevance and consistent behaviour. Lighting conditions are kept stable to minimize detection errors, and the system is calibrated to operate reliably under normal daytime illumination.

5.2 Software Setup

This section describes the software environments used for both training the LSTM classification model and deploying it within the real-time monitoring system. The implementation was divided into two stages: model development and system integration.

5.2.1 Model Training Environment Setup



```
# Latest script

import os
import numpy as np
import pandas as pd
from collections import Counter
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    confusion_matrix, roc_curve, auc
)
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow import keras
import joblib

def load_and_preprocess_data(file_path, window_size_s, step_size):
    try:
        df = pd.read_csv(file_path, encoding="utf-8")
    except UnicodeDecodeError:
        df = pd.read_csv(file_path, encoding="ISO-8859-1")

    df.columns = ["elapsed_time", "marker_id", "x", "y", "z", "yaw", "pitch", "roll", "distance", "status"]
    df = df.sort_values(by="elapsed_time").reset_index(drop=True)

    feature_cols = ["yaw", "pitch", "roll", "x", "y", "z", "distance"]
    max_sequence_length = window_size_s * 30

    X_sequences, y_labels = [], []

    for i in range(0, len(df) - max_sequence_length, step_size):
        window_start_time = df.loc[i, "elapsed_time"] - window_size_s
        window_data = df[
            (df["elapsed_time"] >= window_start_time) &
            (df["elapsed_time"] <= df.loc[i, "elapsed_time"])
        ]
```

Figure 5.2.1.1. Jupyter Notebook environment hosted on Kaggle Notebook

As shown in Figure 5.2.1.1, the model training was performed using Python 3.10 in a Jupyter Notebook environment hosted on Kaggle Notebooks. This platform was chosen for its free GPU support and ease of experimentation. The following major libraries were used:

- Pandas and NumPy for data handling and transformation
- scikit-learn for preprocessing such as standardization and evaluation metrics
- Matplotlib and Seaborn for data visualization and metric plotting
- TensorFlow/Keras for defining, training, and saving the LSTM classification models
- Joblib for saving the scaler used in data normalization

The data pipeline involved loading raw pose data (Euler angles and translation vectors), applying a sliding window to generate time-series samples, standardizing the input features, training the LSTM model, and evaluating its performance using 5-fold stratified cross-validation. Metrics such as accuracy, precision, recall, F1-score, and AUC were computed and visualized to select the best-performing configuration. Both the trained model and the fitted scaler were saved to disk in .keras and .pkl formats respectively, for later use in the deployment phase.

5.2.2 Model Deployment Environment Setup

After training, the model was deployed in a real-time backend system written in Python. The backend runs on a local machine and handles live video input, pose estimation, prediction, and communication with the user interface.

Key software components used in the deployment environment include:

- OpenCV for detecting ArUco markers and estimating 6-DoF pose
- NumPy for buffering and formatting pose data in real time
- Keras for loading and executing the trained LSTM model
- Joblib for loading the saved data scaler
- WebSocket for real-time communication with the frontend
- React.js for rendering the user interface dynamically in the browser

The pose estimation module extracts Euler angles and translation values from each video frame. These values are streamed to the frontend for live visualization and simultaneously sent to a buffering module for classification. Once the buffer is full, data is normalized using the saved scaler and passed to the LSTM model for inference. The predicted statuses and feeding recommendation are then sent back to the frontend.

This modular deployment architecture allows the system to be scalable, efficient, and responsive for real-time aquaculture monitoring.

5.3 Model Training and Evaluation Pipeline

This section details the training pipeline used to develop the LSTM classification model, including data preparation, training configuration, and performance evaluation. The goal of the pipeline was to build a robust model capable of accurately classifying both fish activity status and air pump status from time-series pose data collected from ArUco markers.

5.3.1 Data Collection

To implement the data acquisition process, a batch script (.bat file) was created to initiate pose data logging during each test session. This script activated real-time capture of Euler angles and translation vectors from the floating ArUco marker, using the system's pose estimation module. A total of 5 datasets were collected — including feeding sessions (capturing pre-, during-, and post-feeding activity) and air pump sessions (covering controlled On/Off cycles). These sessions were conducted in the actual aquaculture test tanks. Each pose data log was synchronized with a video recording to support later manual labelling. The recorded pose data reflected the movement intensity of the marker caused by either fish behaviour or aeration. This raw time-series data was stored in CSV format and used in subsequent preprocessing and training stages.

5.3.2 Manual Data Labelling

Before training, all collected pose data was manually labelled by reviewing synchronized video recordings of the feeding sessions and air pump operation. During data collection, pose data (Euler angles and translation vectors) was saved with timestamps, while video recordings were used to observe fish behaviour and water surface turbulence. Each pose sample was manually assigned a binary label based on visual observation and time alignment as follows:

- Fish activity status: labelled as 1 (Active) when fish were observed splashing or swimming actively near the surface, and 0 (Inactive) when the fish were calm or little to no surface activity was present.
- Air pump status: labelled as 1 (On) when the pump was operating, and the water surface is rippling or bubbling, and 0 (Off) when the pump was not operating, and the water surface was calm.

This manual labelling process ensured that the ground truth was accurately captured, even in the absence of automated sensors.

5.3.3 Data Input and Preprocessing

The labelled dataset, stored in CSV format, was loaded and sorted chronologically. Feature columns included yaw, pitch, roll, x, y, z, and distance, extracted from the pose estimation process. A sliding window technique was used to convert the continuous data stream into fixed-length time-series sequences, each representing a short temporal snapshot of marker motion. Each window was padded with zeros if needed and reshaped to match the expected input format for LSTM. The data was then standardized using StandardScaler from scikit-learn.

5.3.4 Model Training

The model was built using Keras with the following architecture:

- A masking layer to ignore padded values
- Two LSTM layers (64 and 32 units)
- A dense layer with 16 ReLU-activated neurons
- A sigmoid output layer for binary classification

The model was trained using 5-fold Stratified Cross-Validation to ensure class balance. Each fold was trained for 100 epochs with a batch size of 32, and predictions were thresholded at 0.5.

To determine the most effective input structure for classification, multiple combinations of window size and step size were tested. For fish activity classification, six configurations were evaluated: (30s, 30), (30s, 15), (15s, 30), (15s, 15), (10s, 30), and (10s, 15). For air pump classification, four configurations were used: (30s, 30), (30s, 15), (15s, 30), and (15s, 15). Each combination was trained using the same model pipeline, and the results of each setup were recorded for comparison. Detailed performance metrics for all configurations are presented in Chapter 6.

5.3.5 Evaluation Metrics and Visualizations

After training, performance was evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion matrix
- ROC curve and AUC score

Visualization tools such as Matplotlib and Seaborn were used to generate performance graphs. An initial class distribution plot was created to verify that the dataset was not imbalanced, helping avoid biased training.

5.3.6 Artifacts Saving

The pipeline saved the following outputs:

- The best model (.keras)
- The scaler (.pkl)
- Plots of confusion matrix, ROC curve, and class distribution (.png)

These were later used during model evaluation and selection, as well as system deployment to ensure consistency and maintain inference quality.

5.4 Model Deployment in Live Monitoring System

Once the best-performing LSTM model and corresponding scaler were selected through cross-validation, they were exported and integrated into the real-time backend system for live inference. This section outlines how the trained model was deployed, how it operates in real time, and how its predictions support feeding decisions and system monitoring.

The trained model was saved in .keras format, and the StandardScaler used during training was saved as a .pkl file using joblib. These files were loaded into the backend system during initialization to ensure consistent preprocessing and inference. The deployment code was written in Python, using the Keras API for model loading and prediction, and NumPy for real-time data buffering and reshaping.

During system operation, pose data (Euler angles and translation vectors) is continuously extracted from incoming video frames using OpenCV’s ArUco marker detection. Each new data point is appended to a sliding window buffer. When the buffer reaches the required length, the data is reshaped and standardized using the saved scaler from the training phase.

The normalized sequence is then passed to the LSTM model for classification. The model outputs a probability between 0 and 1, which is thresholded at 0.5 to determine the final class:

- 1 for Active fish or Pump On
- 0 for Inactive fish or Pump Off

To optimize performance, a throttling mechanism is implemented to trigger inference only at fixed intervals – every 15 frames – rather than on every frame. This reduces computational load and avoids redundant predictions while maintaining timely updates.

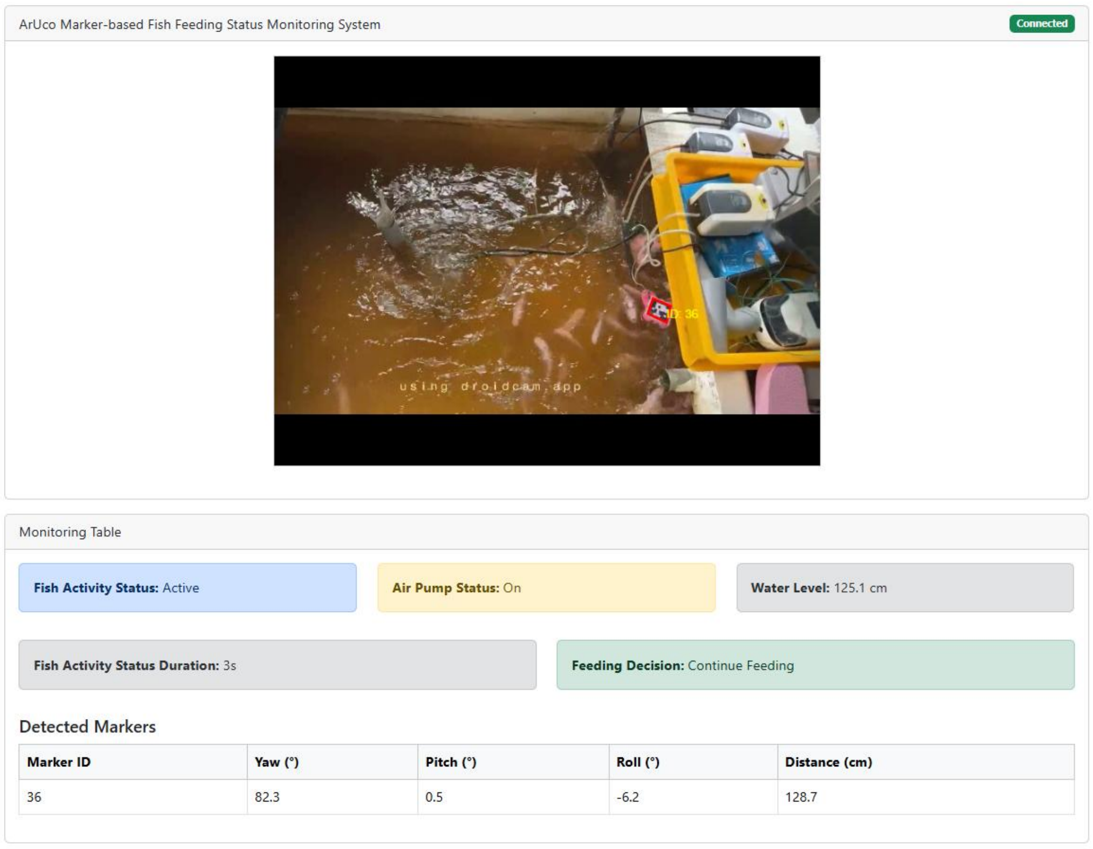
The predicted fish activity status is then passed to the Feeding Recommendation Module, which tracks the duration of the fish activity status. If the duration of active status exceeds 3 seconds, the system generates a recommendation to continue feeding. Else, if the duration of inactive status exceeds 5 seconds, the system generates a recommendation to stop feeding, helping prevent overfeeding and waste.

All outputs — including pose data, prediction results, duration of fish activity status and feeding recommendations — are sent to the frontend interface via a WebSocket communication layer. The React-based UI displays these results in real time, allowing users to observe marker motion, track tank conditions, and make informed decisions. This lightweight deployment pipeline enables efficient multi-tank monitoring using only a single camera and ensures that the system responds accurately and consistently to real-world conditions.

5.5 System Operation and User Interface

This section describes how the system is operated in real time and how users interact with the monitoring interface.

Once the system is initialized, the camera begins streaming live video to the backend. The user initiates the feeding session through the interface, after which pose estimation and model inference start automatically. The backend continuously processes incoming frames to extract Euler angles and translation data from the floating ArUco marker. This data is then sent to the frontend interface via WebSocket in real time.



Fish Activity Status: Active

Air Pump Status: On

Water Level: 125.1 cm

Fish Activity Status Duration: 3s

Feeding Decision: Continue Feeding

Detected Markers

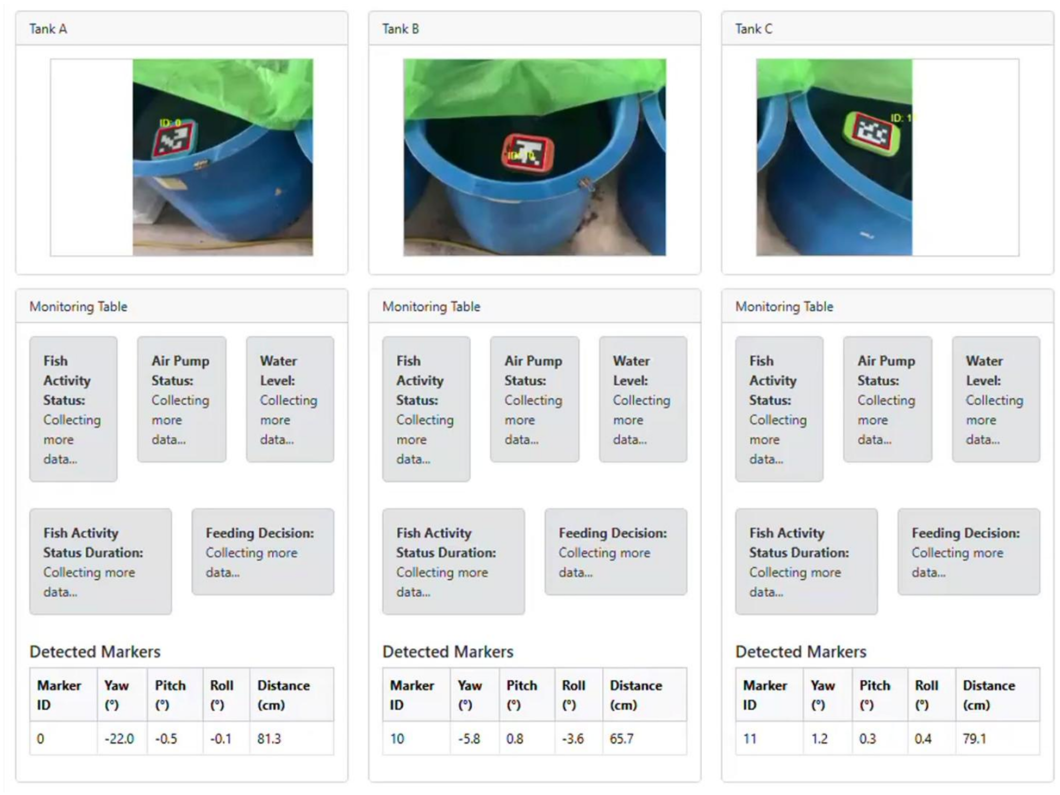
Marker ID	Yaw (°)	Pitch (°)	Roll (°)	Distance (cm)
36	82.3	0.5	-6.2	128.7

Figure 5.5.1. User interface of ArUco marker-based fish feeding status monitoring system

As presented in Figure 5.5.1, the frontend user interface, developed in React, displays the following information:

- Euler angles: yaw, pitch, roll
- Z-axis translation: used as a proxy for water level
- Fish activity status: Active or Inactive
- Air pump status: On or Off
- Duration of current fish activity status
- Feeding recommendation: Continue or Stop Feeding

These values are updated live and allow users to monitor behavioural trends over time.



Tank A

Monitoring Table

Fish Activity Status: Collecting more data...

Air Pump Status: Collecting more data...

Water Level: Collecting more data...

Fish Activity Status Duration: Collecting more data...

Feeding Decision: Collecting more data...

Detected Markers

Marker ID	Yaw (°)	Pitch (°)	Roll (°)	Distance (cm)
0	-22.0	-0.5	-0.1	81.3

Tank B

Monitoring Table

Fish Activity Status: Collecting more data...

Air Pump Status: Collecting more data...

Water Level: Collecting more data...

Fish Activity Status Duration: Collecting more data...

Feeding Decision: Collecting more data...

Detected Markers

Marker ID	Yaw (°)	Pitch (°)	Roll (°)	Distance (cm)
10	-5.8	0.8	-3.6	65.7

Tank C

Monitoring Table

Fish Activity Status: Collecting more data...

Air Pump Status: Collecting more data...

Water Level: Collecting more data...

Fish Activity Status Duration: Collecting more data...

Feeding Decision: Collecting more data...

Detected Markers

Marker ID	Yaw (°)	Pitch (°)	Roll (°)	Distance (cm)
11	1.2	0.3	0.4	79.1

Figure 5.5.2. Multi-tank monitoring user interface of ArUco marker-based fish feeding status monitoring system

On the other hand, as shown in Figure 5.5.2, the interface also supports multi-tank monitoring, allowing data from multiple ArUco markers to be tracked and interpreted using a single camera view. To distinguish between tanks, the video frame is cropped using predefined coordinate boundaries, with each region mapped to a specific marker representing a different tank. This approach enables parallel tracking of multiple tanks without the need for additional cameras, significantly improving scalability while minimizing hardware requirements and system complexity.

5.6 Implementation Issues and Challenges

This section discusses the issues and challenges met during implementation phase. During the system development and deployment process, several technical and operational challenges were encountered. Each issue required iterative testing, debugging, or creative workarounds to ensure the system could function reliably in a real aquaculture environment as follows:

- **Inconsistent Pose Estimation due to Marker Detection Fluctuations**

  The accuracy of pose estimation was initially affected by unstable rotation measurements, particularly when markers were tilted or partially occluded. This was addressed by applying the quaternion-based rotation method, which provided a more stable and robust representation of orientation compared to traditional rotation vectors or Euler angles directly. Quaternion data was then converted to Euler angles after stabilization, improving consistency in motion tracking across frames.

- **Real-Time System Synchronization**

  Initially, syncing live pose data, predictions, and frontend rendering caused occasional delays or mismatched updates. The implementation of a WebSocket-based communication layer helped enable low-latency, bidirectional data transfer. A throttling mechanism was added to limit prediction frequency and ensure smooth frontend updates.

- **Model Overhead and Prediction Frequency**

  Frequent calls to the LSTM model caused processing delays and added load on the backend. This was addressed by introducing a throttling strategy where predictions were triggered only after a set number of frames, reducing computational overhead while maintaining real-time responsiveness.

- **Multi-Tank Region Mapping**

  In the multi-tank setup, the frame had to be divided into predefined regions to track separate ArUco markers accurately. This required calibration of cropping coordinates and testing with multiple markers to ensure the system correctly associated each marker with its corresponding tank. Static region coordinates were defined based on pixel boundaries.

These challenges helped refine the system architecture, improve its resilience in real-world conditions, and enhance its overall performance and usability.

CHAPTER 6 SYSTEM EVALUATION

This chapter presents a comprehensive preliminary analysis, evaluation of the system's performance, covering both the classification model and the live deployment environment. The first part focuses on the preliminary analysis, analysis of class distribution, and performance of the LSTM model using standard machine learning metrics such as accuracy, precision, recall, F1-score, and AUC, based on cross-validation results. Visualizations such as confusion matrices and ROC curves are also analysed to highlight the model's classification capabilities.

The second part discusses the system's real-time behaviour during actual usage. This includes validation of prediction accuracy during feeding sessions and air pump operation, as well as how well the system's feeding recommendations matched observed fish behaviour. The discussion also considers the system's responsiveness, reliability, and practical effectiveness in an aquaculture setting.

Finally, this chapter reflects on challenges faced during evaluation and examines whether the project successfully achieved its stated objectives.

6.1 Preliminary Analysis

This section presents the preliminary analysis of the collected dataset before training the classification model. Before training the classification model, an exploratory analysis was conducted to examine whether measurable differences existed in motion patterns between behavioural states. To quantify surface activity, a movement intensity metric was defined by summing the absolute changes in yaw, pitch, and roll across frames. Specifically, a new feature called Total Tilt Change (°) was computed as the sum of absolute frame-to-frame differences in yaw, pitch, and roll. A rolling mean was then applied to smooth out short-term fluctuations, resulting in the final movement intensity.

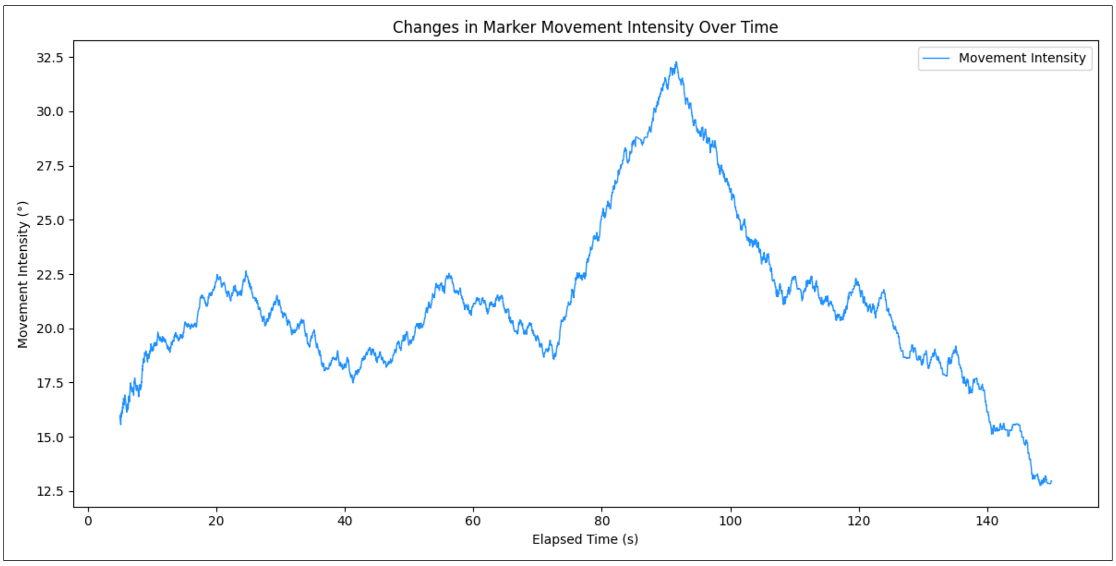


Figure 6.1.1. Plot of changes in marker movement intensity over time during feeding activities

As shown in Figure 6.1.1, movement intensity sharply increases during the active feeding period, which begins at approximately 80 seconds and peaks around 90 seconds, followed by a gradual decline as fish become satiated. The intervals before and after this peak, where intensity remains low and stable, correspond to inactive feeding states. This clear distinction between phases validates the use of motion intensity for inferring fish feeding status.

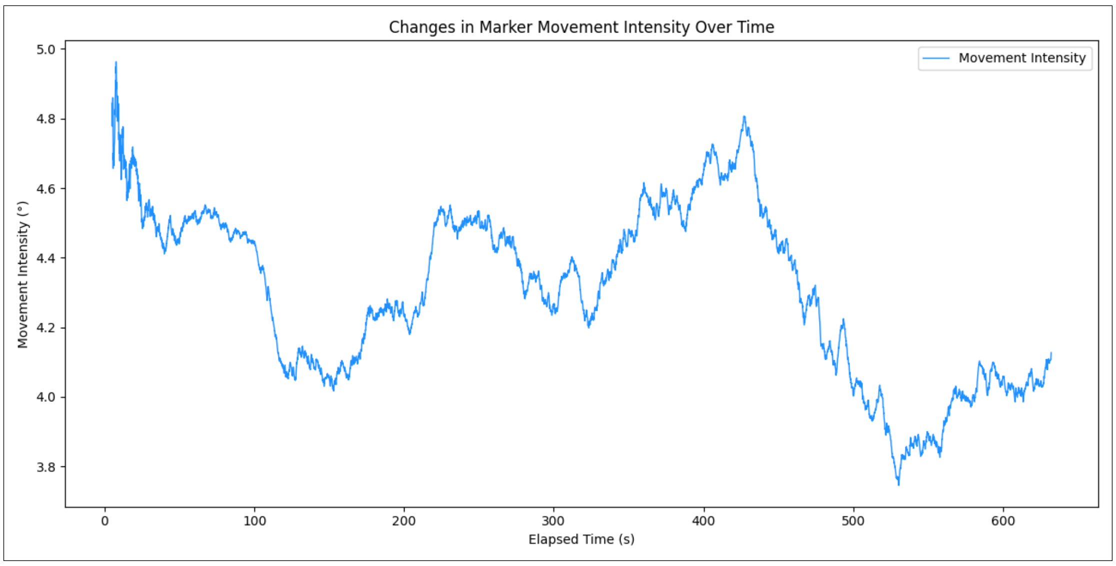


Figure 6.1.2. Plot of changes in marker movement intensity over time during air pump operation

Similarly, Figure 6.1.2 depicts movement intensity during air pump operation. The pump is powered on during the intervals 0s–120s, 200s–300s, and 400s–500s, which correspond to higher and consistent motion intensity caused by aeration-induced surface turbulence. In contrast, during 120s–200s, 300s–400s, and from 500s onward, the pump is powered off, resulting in a noticeable drop in motion intensity and a visibly calmer water surface. These distinct transitions between On and Off phases are clearly reflected in the movement intensity signal, reinforcing the feasibility of using pose-derived motion patterns to classify air pump status through time-series analysis.

6.2 Class Distribution Analysis

Before training the LSTM classification model, it is essential to understand the distribution of classes within the dataset, as class imbalance can bias the model's learning process and affect its predictive performance. During preprocessing, the data for both fish activity status and air pump status was windowed into fixed-length sequences, and each sequence was labelled based on the status of the final frame. This transformation was repeated for multiple configurations of window size and step size to evaluate which parameters produced the most balanced and effective datasets for classification.

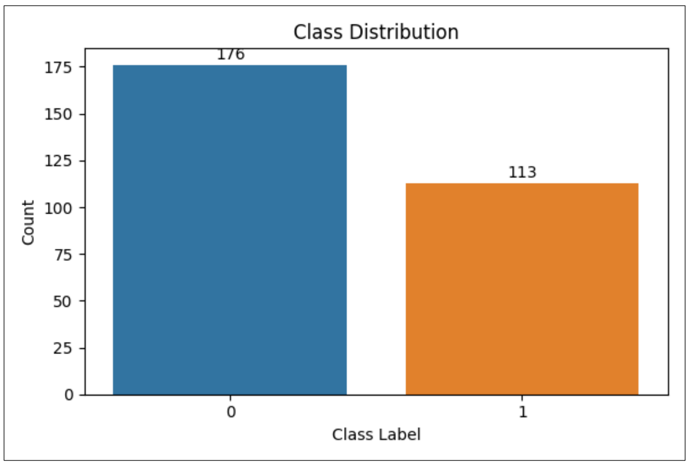


Figure 6.2.1. Class distribution of fish activity status for window size 30s and step size 30 frames

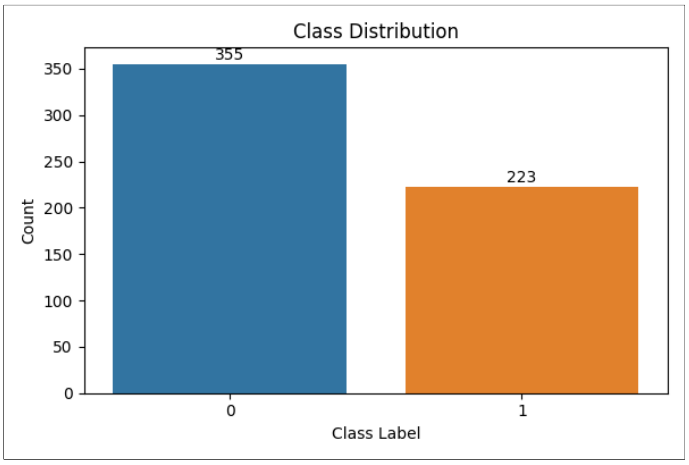


Figure 6.2.2. Class distribution of fish activity status for window size 30s and step size 15 frames

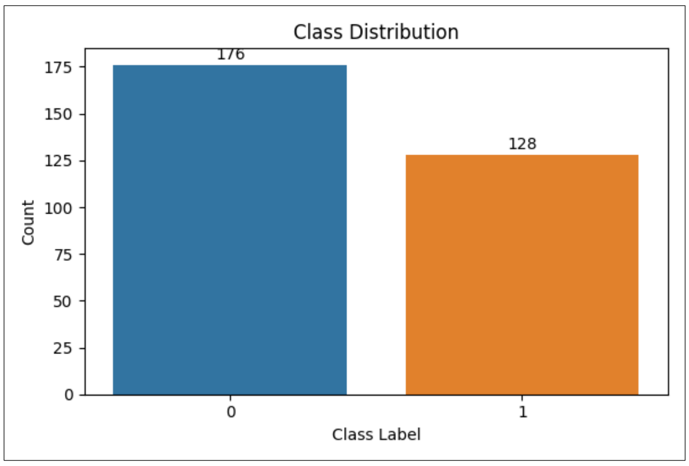


Figure 6.2.3. Class distribution of fish activity status for window size 15s and step size 30 frames

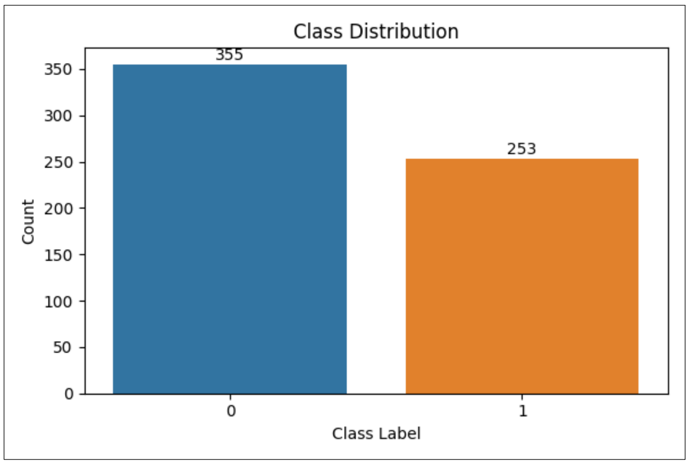


Figure 6.2.4. Class distribution of fish activity status for window size 15s and step size 15 frames

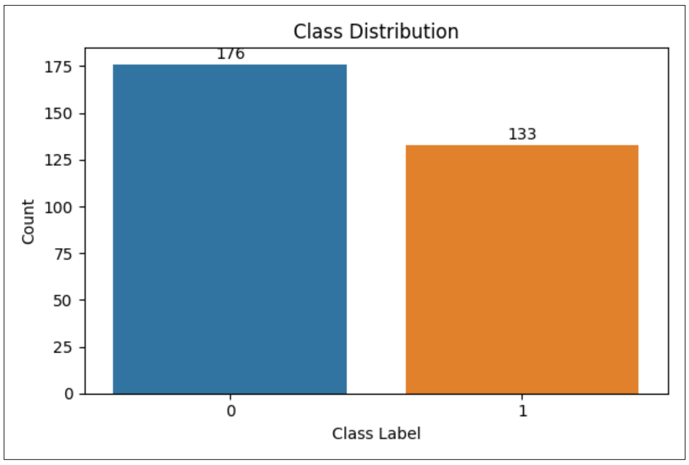


Figure 6.2.5. Class distribution of fish activity status for window size 10s and step size 30 frames

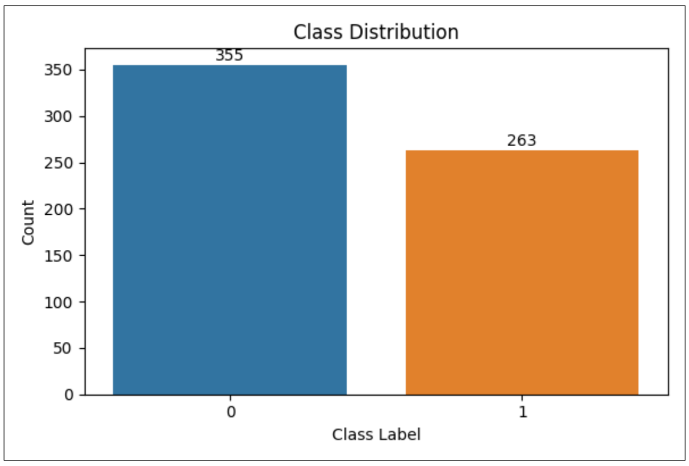


Figure 6.2.6. Class distribution of fish activity status for window size 10s and step size 15 frames

Figures 6.2.1 to 6.2.6 illustrate the distribution of class labels—Inactive (0) and Active (1)—across six different window size and step size configurations: (30s, 30), (30s, 15), (15s, 30), (15s, 15), (10s, 30), and (10s, 15). As observed in Figure 6.2.1, the (30s, 30) configuration yields a more imbalanced dataset with 176 Inactive samples and 113 Active samples. This trend continues across other configurations. In Figure 6.2.2, the (30s, 15) setting shows 355 Inactive and 223 Active samples. Figures 6.2.3 and 6.2.4 further highlight the class differences in the (15s, 30) and (15s, 15) configurations, where Active labels consistently fall behind Inactive ones. The most noticeable imbalance is shown in Figure 6.2.5 for the (10s, 30) configuration, with 176 Inactive and only 113 Active samples.

Despite the skew, the datasets still contain sufficient samples for each class to facilitate effective binary classification. Moreover, stratified cross-validation was used during training to mitigate the impact of imbalance and ensure fair performance evaluation. Understanding these distributions is essential to interpreting the model’s behaviour and identifying potential bias in predictive outputs.

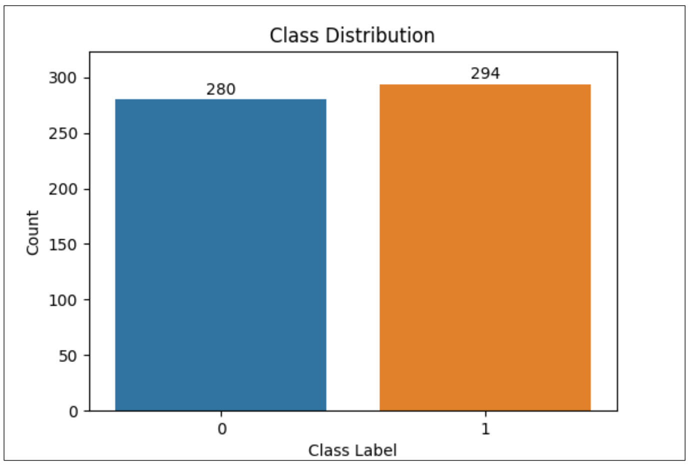


Figure 6.2.7. Class distribution of air pump status for window size 30s and step size 30 frames.

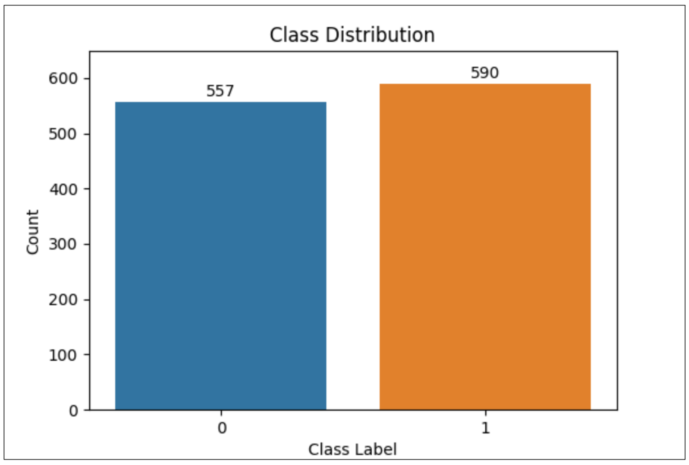


Figure 6.2.8. Class distribution of air pump status for window size 30s and step size 15 frames.

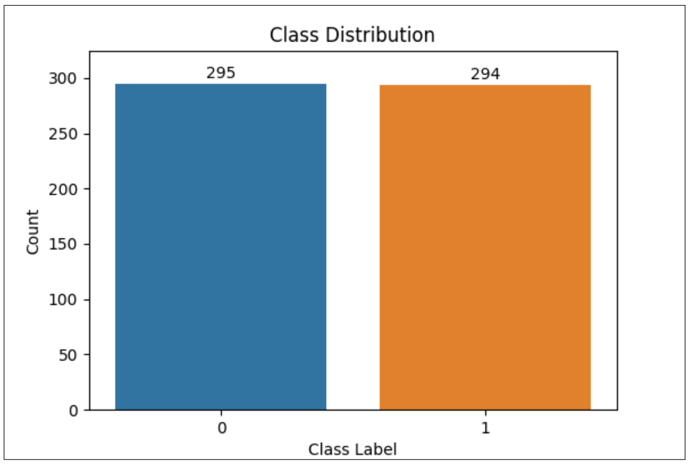


Figure 6.2.9. Class distribution of air pump status for window size 15s and step size 30 frames.

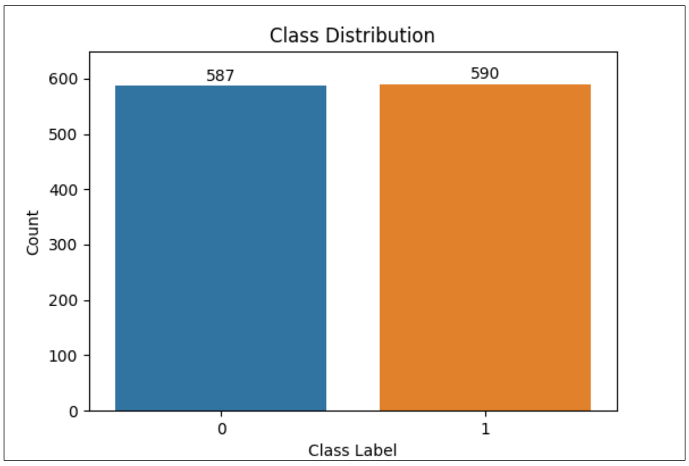


Figure 6.2.10. Class distribution of air pump status for window size 15s and step size 15 frames.

Similarly, Figures 6.2.7 to 6.2.10 present the class distributions for air pump status (Off = 0, On = 1). Four configurations were tested: (30s, 30), (30s, 15), (15s, 30), and (15s, 15). In general, the datasets are relatively well-balanced. The (15s, 15) configuration yields nearly identical counts for both classes (587 Off vs. 590 On), providing a strong foundation for unbiased learning. Slight variations exist across other configurations, but the differences are minor (e.g., 280 Off vs. 294 On in the (30s, 30) configuration). These distributions validate the suitability of the datasets for binary classification and help ensure that high evaluation scores are not merely the result of class majority dominance.

This class distribution analysis provides transparency into the structure of the model training data and supports the validity of the evaluation results presented in the following sections.

6.3 Model Performance Evaluation

This section presents the evaluation results of the LSTM classification model using metrics obtained during the cross-validation phase of training. The objective was to assess the model's ability to accurately classify two types of binary labels: fish activity status (Active or Inactive) and air pump status (On or Off). Following the preliminary analysis, both classification models were evaluated using a 5-fold stratified cross-validation approach. The input data was processed using a sliding window technique, and each fold involved training on 80% of the data while validating on the remaining 20%. To assess model performance, several key metrics were recorded across all folds:

- Accuracy – the proportion of correct predictions
- Precision – the proportion of true positives among predicted positives
- Recall – the proportion of true positives among actual positives
- F1-score – the harmonic mean of precision and recall
- AUC – area under the ROC curve

To determine the optimal configuration for time-series classification, several combinations of window size and step size were systematically evaluated. The window size determines the number of seconds of pose data used in each input sequence, while the step size controls how far the window advances between each sample, thereby affecting sequence overlap. These parameters were varied to investigate how temporal context and sampling frequency influence model performance. The average metrics for each configuration are summarized in the tables below.

Table 6.3.1. Model performance across configurations for fish activity status prediction

Window size (s)	Step size (frame)	Accuracy	Precision	Recall	F1-Score	AUC
30	30	0.87	0.97	0.70	0.81	0.90
30	15	0.92	0.90	0.90	0.89	0.97
15	30	0.83	0.96	0.52	0.75	0.89
15	15	0.85	0.96	0.66	0.78	0.91
10	30	0.84	0.86	0.76	0.80	0.90
10	15	0.86	0.91	0.75	0.81	0.93

Referring to Table 6.3.1, the model's performance in classifying fish activity status varied across configurations. Notably, the configuration with a 30-second window and a 15-frame step size produced the highest scores overall, with an F1-score of 0.89 and an AUC of 0.97. These metrics reflect a strong balance between precision and recall, making this configuration well-suited for applications where both false positives (mistakenly identifying inactive fish as active) and false negatives (failing to detect active fish) are critical to avoid. The long temporal window provides sufficient motion context for the LSTM to recognize feeding behaviour patterns, while the smaller step size ensures better data granularity.

In contrast, configurations with shorter windows (e.g., 10s) or larger step sizes (e.g., 30) tended to exhibit slightly lower recall, suggesting reduced sensitivity in capturing brief or sporadic feeding activities.

Table 6.3.2. Model performance across configurations for air pump status prediction

Window size (s)	**Step size (frame)**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
30	30	0.97	0.98	0.96	0.97	0.98
30	15	0.99	0.99	0.98	0.99	0.98
15	30	0.99	0.99	0.98	0.99	0.99
15	15	0.99	0.99	0.99	0.99	0.99

Based on Table 6.3.2, air pump status classification yielded excellent performance across all tested configurations. The 15s window with 15-frame step size achieved near-perfect results in all metrics, with accuracy, precision, recall, and F1-score of 0.99, and an AUC of 0.99. These results demonstrate the LSTM model's strong ability to distinguish aeration-induced surface motion from natural water calmness, aided by the clearer and more consistent motion patterns generated by the air pump's operation.

To further analyse the model behaviour, confusion matrices and ROC curves for the selected configurations are presented in the following figures.

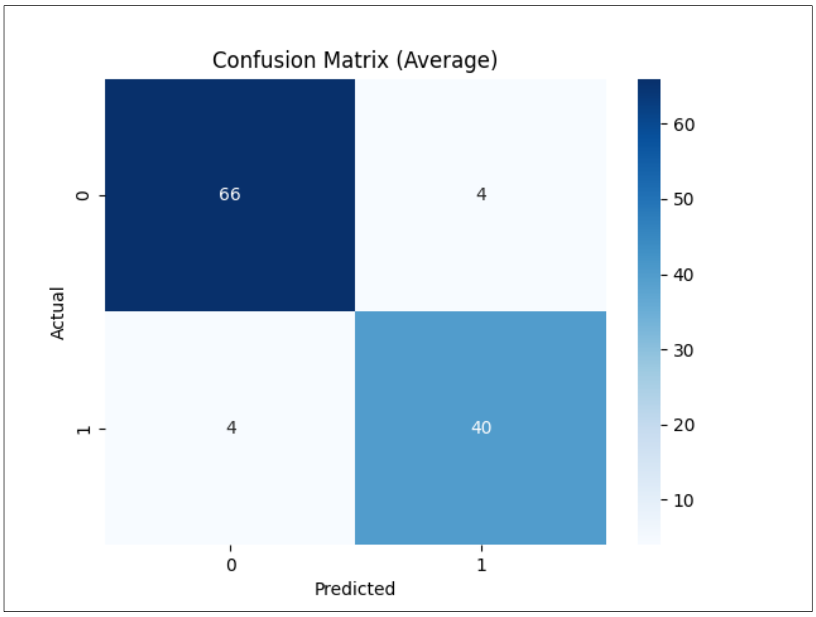


Figure 6.3.1. Confusion matrix for fish activity status model

Figure 6.3.1 presents the average confusion matrix for the selected fish activity model (30s window, 15-step). The matrix shows:

- 66 true negatives (TN): Inactive fish correctly predicted as inactive.
- 40 true positives (TP): Active fish correctly predicted as active.
- 4 false positives (FP): Inactive fish incorrectly predicted as active.
- 4 false negatives (FN): Active fish incorrectly predicted as inactive.

This balanced distribution of correct and incorrect predictions confirms the model's capacity to distinguish feeding states effectively while keeping the error rate low. The symmetry in false positive and false negative counts also reinforces the model's reliability.

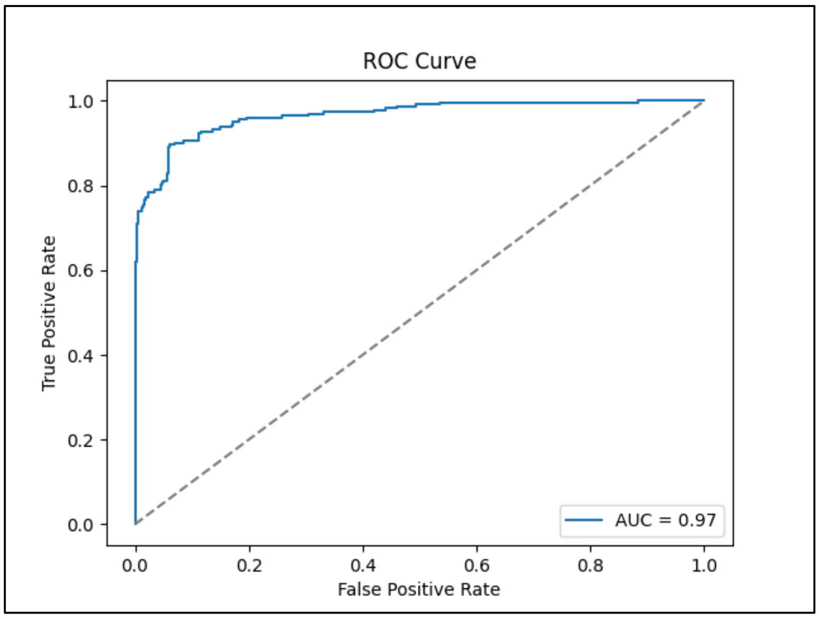


Figure 6.3.2. ROC curve for fish activity status model

Figure 6.3.2 shows the ROC curve for the same fish activity model. With an AUC of 0.97, the curve approaches the top-left corner, indicating excellent sensitivity and specificity. This suggests the model consistently makes confident and accurate predictions, even under varied motion scenarios.

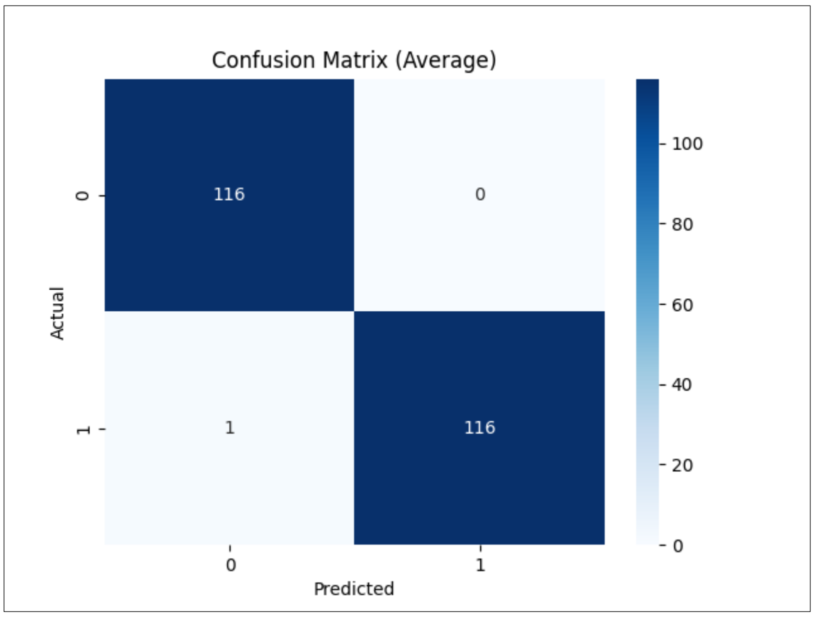


Figure 6.3.3. Confusion matrix for air pump status model

Based on Figure 6.3.3, the confusion matrix for the air pump status model, evaluated using the optimal configuration (15s window and 15-frame step), demonstrates outstanding prediction accuracy. The matrix reveals:

- 116 true negatives (TN): Periods when the pump was off and correctly predicted as off.
- 116 true positives (TP): Periods when the pump was on and correctly predicted as on.
- 1 false negative (FN): A single instance where the pump was on but predicted as off.
- 0 false positives (FP): No incorrect predictions where the pump was predicted on while actually being off.

This nearly perfect classification result highlights the robustness of the LSTM model in recognizing the surface turbulence generated by the air pump. The very low number of false negatives and complete absence of false positives indicate the model's high confidence and precision when detecting aeration activity. This is particularly beneficial for real-time systems where prompt and reliable detection is required to assess aeration status.

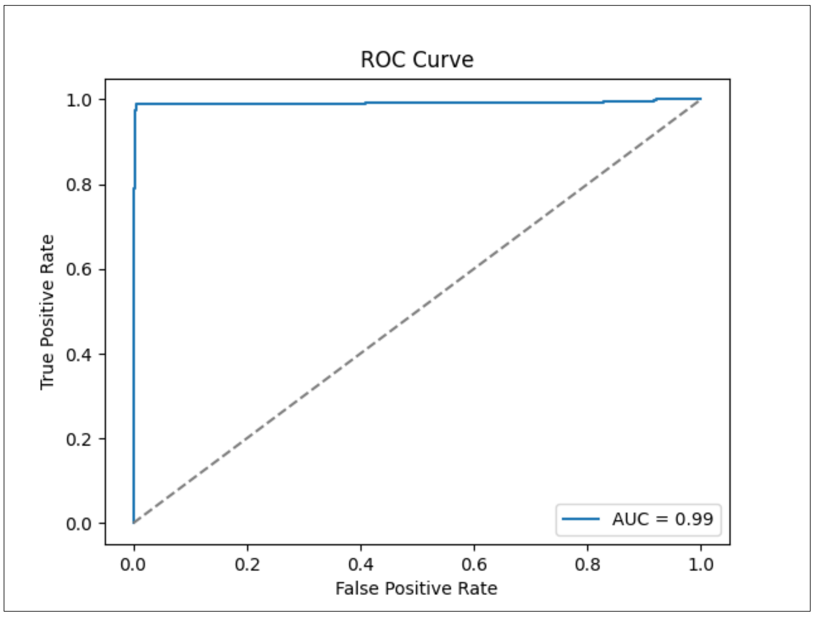


Figure 6.3.4. ROC curve for air pump status model

Based on Figure 6.3.4, the ROC curve for the air pump model supports the confusion matrix findings, with an AUC score of 0.99, approaching the ideal value of 1.0. The curve closely hugs the top-left corner of the graph, indicating that the classifier achieves high true positive rates while keeping the false positive rate extremely low. This level of performance is expected due to the clear distinction between water surface motion when the pump is operating versus when it is off. Since the surface turbulence generated by aeration is strong and consistent, it allows the time-series model to learn well-defined patterns for the On and Off states.

In summary, the LSTM classification models demonstrated strong performance in classifying both fish activity status and air pump status using time-series pose data. Multiple configurations were tested, and the optimal setups were selected based on F1-score and AUC. Visual analyses, including confusion matrices and ROC curves, confirmed the models' ability to accurately distinguish behavioural states. These results validate the feasibility of using ArUco marker motion data as a reliable input for intelligent aquaculture monitoring.

6.4 Real-World System Performance Validation

This section presents the testing methodology and real-world results of the deployed ArUco marker-based fish feeding status monitoring system. After training and selecting the best-performing models, the system was deployed and tested under actual aquaculture conditions to evaluate its effectiveness in practical scenarios. The tests were conducted at the UTAR campus aquaculture facility using red hybrid tilapia (Oreochromis sp.) as the target species.

6.4.1 Testing Procedure and Validation Setup

Two key validation strategies were employed to assess real-time system performance:

- Fish feeding status validation
- Air pump status validation

During scheduled feeding sessions, the system monitored fish behaviour using pose data from floating ArUco markers. When the model predicted an active state, the fish were visibly responsive and consumed the feed, confirming accurate detection of feeding interest. Conversely, when the model predicted an inactive state and recommended stopping the feeding process, feed was still manually dispensed to observe behavioural response. In most cases, the fish showed little or no interest in the additional feed, validating the system’s ability to detect satiety based on motion intensity. These tests confirmed that the system can deliver accurate, behaviour-driven feeding recommendations in real time.

To verify air pump classification, the pump was manually toggled on and off at known intervals. The system was expected to detect the corresponding changes in surface motion caused by aeration. Throughout testing, the predicted pump status consistently matched the actual On/Off control state. These results demonstrated that the model could reliably classify aeration status from marker movement patterns, even in real-time operation.

6.4.2 Observations and System Behaviour During Testing

During real-world testing, the system demonstrated consistent and reliable performance aligned with expected outcomes. The fish activity classification model accurately detected transitions between active and inactive feeding states. When the system predicted an active state, the fish were observed responding to feed with visible surface movement. Conversely, during predicted inactive states, the fish showed little or no interest in additional feed, confirming satiety. These observations validate the effectiveness of the feeding recommendation logic.

The air pump classification model also performed exceptionally well during controlled toggling tests. The system was able to detect the presence or absence of surface turbulence caused by aeration and consistently predicted the pump’s operational status with high accuracy. All predicted statuses matched the actual On/Off states during multiple cycles, demonstrating the model’s robustness in distinguishing subtle differences in surface motion patterns.

In terms of user interaction, the frontend interface provided responsive and intuitive visual feedback. Real-time pose data, including Euler angles and Z-axis displacement, were streamed to the interface along with predicted statuses and feeding recommendations. These updates occurred with minimal latency, allowing users to monitor tank conditions and system outputs continuously and effectively.

Additionally, the system’s ability to support multi-tank monitoring was successfully validated. By assigning fixed pixel regions within the camera frame to represent different tanks, the backend system was able to isolate, and process pose data from multiple ArUco markers simultaneously through distinct marker IDs. This approach allowed a single camera to monitor multiple tanks in parallel, reducing hardware requirements and improving scalability.

In short, the real-world testing of the system confirmed its practical effectiveness in detecting fish feeding behaviour and air pump operation with high accuracy. The validation procedures demonstrated that the model predictions aligned closely with actual observations, and the frontend interface successfully delivered real-time updates and recommendations. The system’s ability to support multi-tank monitoring using a single camera further highlights its scalability and applicability in commercial aquaculture environments.

6.5 Project Challenges

Throughout the development and testing of the system, several challenges were encountered across both the model training and deployment phases. These challenges provided valuable learning opportunities and informed refinements that improved the final system's robustness and performance.

One of the primary challenges was ensuring stable and reliable pose estimation under real-world conditions. Water surface reflection, lighting variability, and partial marker occlusion occasionally disrupted marker detection. To address this, the quaternion method was adopted during pose estimation to enhance rotational stability before converting to Euler angles. This significantly reduced noise and improved consistency in orientation data across frames.

In the deployment phase, ensuring smooth real-time synchronization between backend predictions and frontend display was initially problematic due to frame processing delays and data bursts. To mitigate this, a WebSocket-based communication channel was implemented, allowing low-latency, bidirectional streaming. Additionally, a throttling mechanism was introduced to trigger predictions at fixed intervals, balancing computational load with responsiveness.

Finally, implementing multi-tank tracking using a single camera introduced complexity in isolating pose data from different tanks. This was solved by defining static cropping regions within the frame to associate each ArUco marker with a specific tank, allowing parallel processing of multiple marker streams.

Despite these challenges, the project achieved its objectives through iterative testing and continuous refinement. The solutions applied contributed to a more robust and scalable system, suitable for real-time aquaculture monitoring.

6.6 Objectives Evaluation

This section reviews the outcomes of the project in relation to the two primary objectives defined at the outset:

1. To capture movement intensity data that reflects fish feeding status.
2. To interpret movement intensity data to indicate the fish satiety levels.

The first objective was successfully achieved. The system employed a smartphone camera and floating ArUco markers to collect real-time pose data from the surface of the water. Movement intensity was derived from frame-to-frame changes in yaw, pitch, and roll angles, using a total tilt change formula combined with a rolling average. This approach enabled the system to quantify surface disturbance caused by fish during feeding. The resulting movement intensity signal clearly reflected changes in feeding status, as validated through exploratory analysis and real-world observation.

The second objective was also successfully fulfilled. The derived movement intensity data was used as input to a time-series LSTM model, which was trained to classify fish activity status as either active or inactive. A feeding recommendation logic was applied based on the duration of the predicted activity state. During live testing, when the system predicted an active state, the fish consistently responded to feed, indicating ongoing feeding interest. Conversely, when the system predicted inactivity and recommended stopping the feeding process, the fish showed little or no interest in additional feed. These outcomes confirm the system's ability to infer fish satiety levels indirectly through behavioural motion patterns, supporting accurate, real-time feeding decisions.

In conclusion, both objectives were fully met, and the system proved capable of monitoring fish feeding status using visual motion data while offering meaningful insights for feed management.

CHAPTER 7 CONCLUSION AND FUTURE WORKS

This chapter summarizes the overall outcomes of the project, reflecting on key achievements in relation to the original objectives. It also outlines potential directions for future enhancements, addressing current limitations and exploring opportunities for scaling and improving the system further. The goal is to consolidate what has been accomplished and to suggest how the project can be extended or refined in future iterations.

7.1 Conclusion

This project successfully explored the potential of using ArUco markers to monitor fish feeding status through pose-based motion analysis. By capturing real-time movement intensity from floating markers, the system was able to distinguish between active and inactive feeding states with high reliability. The LSTM-based classification model, trained on pose data derived from Euler angles and translation values, demonstrated strong performance during both cross-validation and real-world testing. Feeding recommendations generated based on predicted status were consistent with observable fish behaviour, validating the model's practical effectiveness in detecting satiety.

Beyond monitoring fish feeding status, the system also successfully integrated additional monitoring features, including air pump status classification and water level tracking, all accessible through a responsive web-based interface. Multi-tank monitoring was achieved using a single overhead camera, making the system both cost-efficient and scalable.

Overall, the project achieved its stated objectives, offering a functional and data-driven solution for intelligent aquaculture feed management.

7.2 Future Work

While the current system successfully meets its core objectives, several enhancements could be explored to further improve its accuracy, usability, and scalability in real-world aquaculture applications. One of the key areas for improvement is in the data labelling process. The current method relies on manual observation and timestamp alignment, which is time-consuming and limits dataset scalability. Future development could explore semi-supervised learning methods or video-assisted labelling tools to automate and accelerate this process, enabling larger and more diverse training datasets.

Another area of potential enhancement lies in the feeding recommendation logic. Currently, the system uses fixed duration thresholds to determine when to stop feeding or continue feeding based on detected fish activity status. This rule-based approach, while effective, could be improved by introducing adaptive thresholds that adjust based on behavioural trends, tank-specific feeding responses, or historical data. Such personalization could lead to more precise and efficient feed management strategies.

Additionally, integration with IoT-based actuators such as automated feeders and smart pump controllers would allow the system to not only provide recommendations but also execute actions autonomously. This would close the feedback loop and enable a fully automated feeding and monitoring system. Expanding the system to monitor additional environmental parameters—such as temperature, pH, and dissolved oxygen—could also provide a more holistic view of tank conditions and further refine feeding decisions.

Lastly, optimizing the system for edge computing would be beneficial for deployment in environments with limited infrastructure. Converting the current backend model into a lightweight version that can run on low-power hardware such as Raspberry Pi or ESP32 would improve portability and reduce the reliance on centralized servers, making the system more accessible to small-scale fish farms.

These future enhancements have the potential to transform the current prototype into a comprehensive and scalable smart aquaculture monitoring platform.

REFERENCES

[1] C. E. Boyd, R. P. Davis, A. A. McNevin, and V. Kumar, “Water quality and its impacts on feeding practices,” Academic Press, 2025, pp. 383–401. doi: https://doi.org/10.1016/B978-0-443-21556-8.00006-5.

[2] H. Roh et al., “Overfeeding-Induced Obesity Could Cause Potential Immuno-Physiological Disorders in Rainbow Trout (Oncorhynchus mykiss),” Animals, vol. 10, no. 9, p. 1499, Aug. 2020, doi: https://doi.org/10.3390/ani10091499.

[3] Y. Xiao et al., “Feeding Behavior Quantification and Recognition for Intelligent Fish Farming Application:A Review,” Applied Animal Behaviour Science, vol. 285, pp. 106588–106588, Mar. 2025, doi: https://doi.org/10.1016/j.applanim.2025.106588.

[4] W.-C. Hu, L.-B. Chen, B.-K. Huang, and H.-M. Lin, “A Computer Vision-Based Intelligent Fish Feeding System Using Deep Learning Techniques for Aquaculture,” IEEE Sensors Journal, vol. 22, no. 7, pp. 7185–7194, Apr. 2022, doi: https://doi.org/10.1109/jsen.2022.3151777.

[5] N. A. Ubina, S.-C. Cheng, C.-C. Chang, and H.-Y. Chen, “Evaluating fish feeding intensity in aquaculture with convolutional neural networks,” vol. 94, pp. 102178–102178, Aug. 2021, doi: https://doi.org/10.1016/j.aquaeng.2021.102178.

[6] S. Zhao et al., “Application of machine learning in intelligent fish aquaculture: A review,” Aquaculture, vol. 540, p. 736724, Jul. 2021, doi: https://doi.org/10.1016/j.aquaculture.2021.736724.

[7] W.-C. Hu, L.-B. Chen, B.-K. Huang, and H.-M. Lin, “A Computer Vision-Based Intelligent Fish Feeding System Using Deep Learning Techniques for Aquaculture,” IEEE Sensors Journal, vol. 22, no. 7, pp. 7185–7194, Apr. 2022, doi: https://doi.org/10.1109/jsen.2022.3151777.

[8] H. Choi and J. Lee, “Efficient Use of GPU Memory for Large-Scale Deep Learning Model Training,” *Applied Sciences*, vol. 11, no. 21, p. 10377, Nov. 2021, doi: https://doi.org/10.3390/app112110377.

[9] S. Tsimenidis, “Limitations of Deep Neural Networks: a discussion of G. Marcus’ critical appraisal of deep learning,” *arXiv:2012.15754 [cs]*, Dec. 2020, Available: https://arxiv.org/abs/2012.15754

[10] S. Al-Abri, S. Keshvari, K. Al-Rashdi, R. Al-Hmouz, and H. Bourdoucen, “Computer vision based approaches for fish monitoring systems: a comprehensive study,” *Artificial Intelligence Review*, vol. 58, no. 6, Mar. 2025, doi: https://doi.org/10.1007/s10462-025-11180-3.

[11] L. Yang et al., “Computer Vision Models in Intelligent Aquaculture with Emphasis on Fish Detection and Behavior Analysis: A Review,” Archives of Computational Methods in Engineering, vol. 28, no. 4, pp. 2785–2816, Sep. 2020, doi: https://doi.org/10.1007/s11831-020-09486-2.

[12] Shreesha S, M. Pai, U. Verma, and R. M. Pai, “Computer Vision Based Fish Tracking And Behaviour Detection System,” Oct. 2020, doi: https://doi.org/10.1109/discover50404.2020.9278101.

[13] S. Peixoto, R. Soares, and D. Allen Davis, “An acoustic based approach to evaluate the effect of different diet lengths on feeding behavior of Litopenaeus vannamei,” *Aquacultural Engineering*, vol. 91, pp. 102114–102114, Nov. 2020, doi: https://doi.org/10.1016/j.aquaeng.2020.102114.

[14] Y. Zeng et al., “Fish school feeding behavior quantification using acoustic signal and improved Swin Transformer,” Computers and electronics in agriculture, vol. 204, pp. 107580–107580, Jan. 2023, doi: https://doi.org/10.1016/j.compag.2022.107580.

[15] R. Qi, H. Liu, and S. Liu, “Effects of Different Culture Densities on the Acoustic Characteristics of Micropterus salmoide Feeding,” *Fishes*, vol. 8, no. 3, pp. 126–126, Feb. 2023, doi: https://doi.org/10.3390/fishes8030126.

[16] D. Li, Z. Wang, S. Wu, Z. Miao, L. Du, and Y. Duan, “Automatic recognition methods of fish feeding behavior in aquaculture: A review,” Aquaculture, vol. 528, p. 735508, Nov. 2020, doi: https://doi.org/10.1016/j.aquaculture.2020.735508.

[17] M. A. Ferrer et al., “From operculum and body tail movements to different coupling of physical activity and respiratory frequency in farmed gilthead sea bream and European sea bass. Insights on aquaculture biosensing,” Computers and Electronics in Agriculture, vol. 175, p. 105531, Aug. 2020, doi: https://doi.org/10.1016/j.compag.2020.105531.

[18] D. Kim, J. H. Bong, and S. Jeong, “Enhancing Pose Estimation Using Multiple Graphical Markers with Spatial and Temporal Outlier Detection,” *Applied Sciences*, vol. 14, no. 22, p. 10225, Nov. 2024, doi: https://doi.org/10.3390/app142210225.

[19] Junardo Herdiansyah, S. Putra, and Dwi Septiyanto, “Implementation of Zhang’s Camera Calibration Algorithm on a Single Camera for Accurate Pose Estimation Using ArUco Markers,” *Journal of Fuzzy Systems and Control*, vol. 2, no. 3, pp. 176–188, Sep. 2024, doi: https://doi.org/10.59247/jfsc.v2i3.256.

[20] João Vitor Braun *et al.*, “A robot localization proposal for the RobotAtFactory 4.0: A novel robotics competition within the Industry 4.0 concept,” vol. 9, Nov. 2022, doi: https://doi.org/10.3389/frobt.2022.1023590.

[21] S. Gu *et al.*, “Navigation of biped wall-climbing robots using BIM and ArUco markers,” *Robotica*, pp. 1–21, Jan. 2025, doi: https://doi.org/10.1017/s0263574724002170.

[22] D. Avola, L. Cinque, G. L. Foresti, C. Mercuri, and D. Pannone, “A Practical Framework for the Development of Augmented Reality Applications by using ArUco Markers,” *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*, 2016, doi: https://doi.org/10.5220/0005755806450654.

[23] Z. Xu, J. Zhang, P.-W. Tsai, L. Lin, and C. Zhuo, “Spatiotemporal Mobility Based Trajectory Privacy-Preserving Algorithm in Location-Based Services,” *Sensors (Basel, Switzerland)*, vol. 21, no. 6, Mar. 2021, doi: https://doi.org/10.3390/s21062021.

[24] A. Khazetdinov, A. Zakiev, T. Tsoy, M. Svinin, and E. Magid, “Embedded ArUco: a novel approach for high precision UAV landing,” *IEEE Xplore*, May

01, 2021. https://ieeexplore.ieee.org/document/9438855 (accessed Jun. 07, 2023).

[25] Z. Qiu, D. Lin, R. Jin, J. Lv, and Z. Zheng, “A Global ArUco-Based Lidar Navigation System for UAV Navigation in GNSS-Denied Environments,” *Aerospace*, vol. 9, no. 8, p. 456, Aug. 2022, doi: https://doi.org/10.3390/aerospace9080456.

POSTER

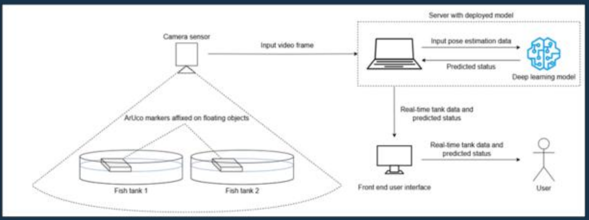
EXPLORING THE POTENTIAL OF USING ARUCO MARKERS TO MONITOR FISH FEEDING STATUS

By: Goh Ken How (21ACB02687)

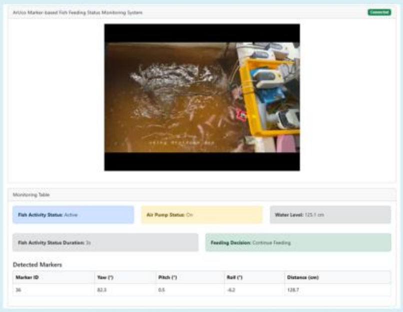
Introduction

Traditional fish feeding methods rely on manual observation, which can lead to overfeeding and inefficiency. This project introduces a low-cost, non-invasive monitoring system using ArUco markers and computer vision to track fish activity during feeding. By analyzing marker motion, an LSTM model predicts feeding status in real time and provides smart feeding recommendations. The system also detects air pump status and water level, and supports multi-tank monitoring with a single camera, offering a scalable solution for smarter aquaculture management.

Methodology


Server with deployed model
Camera sensor
Input video frame
Input pose estimation data
Predicted status
Deep learning model
ArUco markers affixed on floating objects
Real-time tank data and predicted status
Real-time tank data and predicted status
Fish tank 1
Fish tank 2
Front end user interface
User

- Video Capture: A smartphone camera records floating ArUco markers.
- Pose Estimation: Marker pose is extracted using OpenCV and converted to Euler angles and Z-distance.
- Data Processing: Movement data is grouped using a sliding window.
- Model Training: An LSTM model is trained to classify fish activity and air pump status.
- Deployment: Real-time predictions and feeding recommendations are displayed via a web-based interface.
- Multi-Tank Monitoring: Multiple markers are tracked in a single camera view.



ArUco Marker-based Fish Feeding Status Monitoring System

Connected

Monitoring Table

Fish Activity Status: Active

Air Pump Status: On

Water Level: 125.1 cm

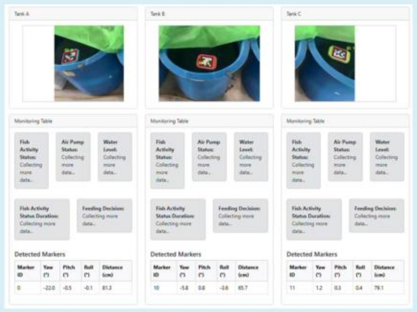
Fish Activity Status Duration: 3s

Feeding Decision: Continue Feeding

Detected Markers

Marker ID	Yaw (°)	Pitch (°)	Roll (°)	Distance (cm)
36	82.3	0.5	-6.2	128.7

Figure 1: User interface displaying real-time fish activity status, air pump status, water level, and feeding recommendations.



Tank A

Monitoring Table

Fish Activity Status: Collecting more data...

Air Pump Status: Collecting more data...

Water Level: Collecting more data...

Fish Activity Status Duration: Collecting more data...

Feeding Decision: Collecting more data...

Detected Markers

Marker ID	Yaw (°)	Pitch (°)	Roll (°)	Distance (cm)
0	-22.0	-0.5	-0.1	81.3

Tank B

Monitoring Table

Fish Activity Status: Collecting more data...

Air Pump Status: Collecting more data...

Water Level: Collecting more data...

Fish Activity Status Duration: Collecting more data...

Feeding Decision: Collecting more data...

Detected Markers

Marker ID	Yaw (°)	Pitch (°)	Roll (°)	Distance (cm)
10	-5.8	0.8	-3.6	65.7

Tank C

Monitoring Table

Fish Activity Status: Collecting more data...

Air Pump Status: Collecting more data...

Water Level: Collecting more data...

Fish Activity Status Duration: Collecting more data...

Feeding Decision: Collecting more data...

Detected Markers

Marker ID	Yaw (°)	Pitch (°)	Roll (°)	Distance (cm)
11	1.2	0.3	0.4	79.1

Figure 2: Multi-tank interface showing live marker data and status predictions for each tank.

Results

- The LSTM model achieved up to 92% accuracy in fish activity classification and 99% in air pump status detection.
- Feeding recommendations aligned well with observed fish behaviour during testing.
- The system supports real-time monitoring and multi-tank tracking using a single camera.

Conclusion

This project demonstrates a low-cost, scalable solution for monitoring fish feeding status using ArUco markers and deep learning. The system reduces reliance on manual observation, helps prevent overfeeding, and enhances decision-making in aquaculture operations. Its ability to support multi-tank setups makes it well-suited for commercial adoption.