IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 21, 2020, accepted December 7, 2020, date of publication December 10, 2020,
date of current version December 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043712

Anomalous Behaviors Detection for Underwater Fish Using AI Techniques

JUNG-HUA WANG[1,2], SHIH-KAI LEE[1], YI-CHUNG LAI[1], CHENG-CHUN LIN[1], TING-YUAN WANG[3], YING-REN LIN[1], TE-HUA HSU[4], CHANG-WEN HUANG[4,5], AND CHUNG-PING CHIANG[4]

[1]Department of Electrical Engineering, National Taiwan Ocean University, Keelung City 20224, Taiwan
[2]AI Research Center, National Taiwan Ocean University, Keelung City 20224, Taiwan
[3]Industrial Technology Research Institute (ITRI), Hsinchu 310401, Taiwan
[4]Department of Aquaculture, National Taiwan Ocean University, Keelung City 20224, Taiwan
[5]Center of Excellence for the Oceans, National Taiwan Ocean University, Keelung City 20224, Taiwan

Corresponding author: Jung-Hua Wang (jhwang@email.ntou.edu.tw)

This work was supported in part by the Center of Excellence for the Oceans (CEO), National Taiwan Ocean University, in part by the Center of Excellence for Ocean Engineering (CEOE), National Taiwan Ocean University, and in part by the Ministry of Science and Technology of Taiwan (MOST) AI Biomedical Research Center under Grant MOST 109-2634-F-019-001, Grant MOST 108-2634-F-019-001, and Grant MOST 107-2313-B-019-005.

ABSTRACT Anomalous events detection in real-world video scenes is a challenging problem owing to the complexity of anomaly and the untidy backgrounds and objects in the scenes. Although there are already many studies on dealing with this problem using deep neural networks, very little literature aims for real-time detection of the anomalous behavior of fish. This paper presents an underwater fish anomalous behavior detection method by combining deep learning object detection, DCG (Directed Cycle Graph), fish tracking, and DTW (Dynamic Time Warping). The method is useful for detecting the biological anomalous behavior of underwater fish in advance so that early countermeasures can be planned and executed. Also, through post-analysis it is possible to access the cause of diseases or death, so as to reduce unnecessary loss, facilitate precision breeding selection, and achieve ecological conservation education as well. A smart aquaculture system incorporating the proposed method and IoT sensors allows extensive data collection during the system operation in various farming fields, thus enabling to develop optimal culturing conditions, both are particularly useful for researchers and the aquaculture industry.

INDEX TERMS Anomalous behavior analysis, deep learning, object detection, tracking, dynamic time warping, directed cycle graph.

I. INTRODUCTION

Due to the global population growth, the average consumption of aquaculture products has been increasing rapidly for the last two decades. In response to the rising demand for aquaculture products, fishery foods supply in many countries has been dominated by farming products. However, the aquaculture industry worldwide is becoming more and more prone to huge losses owing to extreme climate change. To ensure a good harvest, it is essential to regularly monitor the health of farmed aquatic creatures such as fish or shrimps. For aquaculture operators, having the ability to detect or classify anomalous behaviors of farmed creatures is highly desirable. An anomalous behavior usually indicates a symptom of disease or sign of creatures being under stress, and deserves attention and analysis to find out possible causes. However, traditional methods often rely on manual inspections or personal subjective experiences to judge whether the farmed aquatic creatures are in a healthy state. Without automatic monitoring systems, one can only manually collect samples from sea cages or farms and check for abnormal symptoms based on their appearance, or passively wait for dead creatures to float on the water to determine the cause of death. Manual operations are often time-consuming and expensive, aside from the fact that aquatic creatures are prone to severe stress or sudden death. Also, toxins or pathogens from death fish could quickly contaminate the entire farm or sea cage. Unless otherwise stated, hereinafter we will use the term fish to represent underwater creatures, although our method is not just applicable to fish.

The associate editor coordinating the review of this manuscript and approving it for publication was Donghyun Kim.

With the advent of IoT and 5G technologies, the use of various sensors such as underwater cameras, dissolved oxygen meters, and temperature sensors allows a large amount of environmental data to be continuously and quickly collected. The combination of AI and IoT (AIoT) further accelerates the trend of smart aquaculture. Vast benefits of monitoring the health status of cultivated fish can be obtained from using AIoT, as aquaculture industry inevitably involves breeding and selection of brood fish. Their purpose is to effectively breed high-quality fish, so that the offspring can resist drastic changes in the environment. Fish with high-stress resistance are selected through a series of experiments such as salt tolerance, cold tolerance, heat tolerance, and disease resistance, etc. To avoid sudden or unexplainable death of precious or brood fish and to speed up the reproduction speed during reproduction, it is vital to automatically detect anomalous behaviors. Unfortunately, current breeding methods are expensive to conduct, because they still heavily rely on naked-eye observation of fish behaviors. Many control factors require careful attention and any failure would likely cause fish death without salient signs. From time to time, the breeder may be forced to terminate the experiment or adjust the experiment urgently. Therefore, there exists a great need for a real-time system that can automatically detect and classify anomalous fish behaviors.

Deep learning techniques can do end-to-end detection of instances of semantic objects such as fish without specifically defining features, and are typically built on convolutional neural networks (CNN) [1], [2]. Based on our previous work [3] employing deep learning Faster- rcnn [4] as an object detector to implement the tracking task for measuring the moving speed of fish, this paper presents a real-time solution for the problem of detecting anomalous behaviors for underwater fish. Real-time detection of anomalous behaviors can be applied to aquaculture farms and sea cages to help prevent diseases and sudden death, so as to reduce financial loss.

Our idea is rooted in the observation that no matter how the fish swims, some relative position relations between fish's body parts remain unchanged, e.g. the tail part is always located opposite to the head part, and the dorsal fin must be opposite to the fin. This property is referred to as antithesis invariance hereinafter. To classify fish behaviors, we employ deep learning object detector to detect the body parts of fish. Information (i.e. bounding boxes, coordinates, and class name thereof) provided by the object detector is exploited and fully utilized in all subsequent stages of the method. The proposed method mainly comprises the following technical elements: object detection, directed cycle graph (DCG) [5], fish tracking, and dynamic time warping (DTW) [6]. The rest of the paper is organized as follows. Section II highlights work done by others that are related to this paper. In particular, we review literature on fish detection, classification, and behavior analysis in various applications. Section III through Section VI elaborate our method stage by stage. Extensive empirical results are shown in Section VII, and finally concluding remarks and future work are given in Section VIII.

II. RELATED WORK

Although in the past two decades, methods concerning fish detection, classification, and recognition have been proposed, we have seen rather few training datasets related to fish healthy status, not to mention AI-based classification of anomalous behavior for aquatic creatures. As far as we know, studies on detection or recognition of fish anomalous behavior are very rare. [7]–[10] used image processing techniques to obtain fish movement trajectories for identifying anomalous behaviors. Semani [11] studied automatic fish recognition by focusing on the segmentation process, steps of feature extraction and classification. They also addressed the issue of characterizing moving fish using a robust mixture decomposition-based clustering algorithm [12]. Analogous to [11] and [12], the task of fish recognition in [13] is also based on background subtraction method. By identifying some critical situations met in video, [13] showed improvement over the classification made by [14]. In his doctoral thesis, Pinkiewicz [15] applied image segmentation and Kalman filtering to track fish in sea cages for short periods in order to calculate an average swimming speed and direction. Individuals were tracked on a prolonged basis for identifying agonistic behaviors and observing behaviors of small groups of fish in tanks. Spampinato [16] incorporated the Gaussian Mixture Model and Moving Average to detect fish, followed by tracking the fish using the Adaptive Mean Shift Algorithm. The fish trajectory was obtained using an unsupervised machine learning algorithm of clustering. They also analyzed fish behavior in typhoons using combinations of previous methods and event detection in [17].

Recently, deep neural networks have been used to achieve state-of-the-art results in various fish-related tasks. Since the main interest of this paper is directed to the detection of the anomalous behavior of fish using AI techniques, it is worth taking some time to review literature related to this aspect. Jalal [18] proposed a hybrid solution that combines optical flow and Gaussian mixture models with YOLO [19], aiming to act as a unified approach to classify fish in unconstrained underwater videos. In the context of multi-target fish tracking to provide a real-time response to long-lasting experiments using HQ video, [20] used the U-net architecture [21] to obtain the segmentation of fish in extreme conditions such as illumination changes combined with sudden background differences or other noise. To our best knowledge, the present work is the first attempt to provide real-time detection and classification of anomalous behaviors for multiple fish under high stress (e.g. high salinity, coldness). The proposed approach is characterized in that information provided by the deep learning-based object detectors is utilized throughout the subsequent steps of DCG, tracking, and DTW.

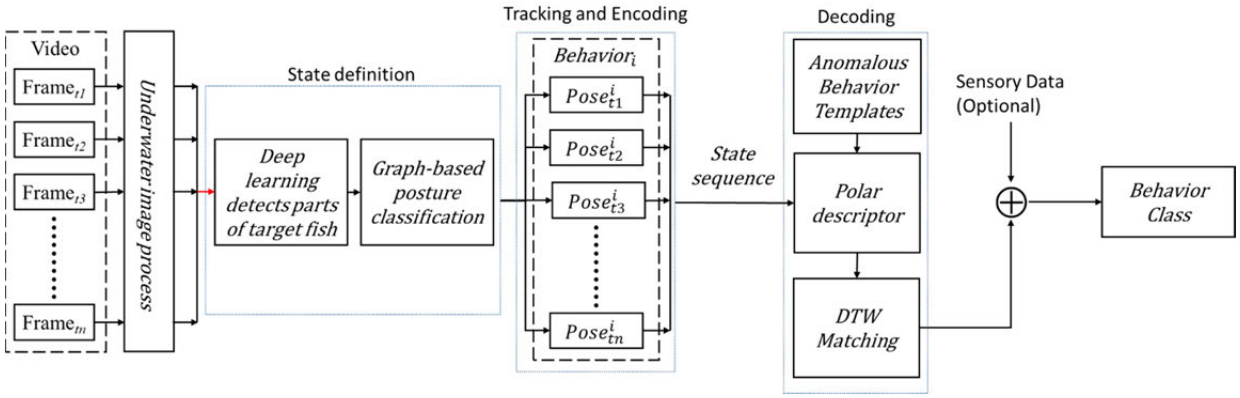


FIGURE 1. Flowchart of the propose method.

III. PROPOSED METHOD

As shown in Fig. 1, the proposed method consists of three main stages: State Definition, Tracking/Encoding, and Decoding by DTW. Their realizations involve the use of deep learning-based object detection, graph theory, tracking, DTW, and aquaculture domain knowledge. The stage of State Definition may use any deep learning inference models to perform object detection for multiple body parts of moving fish. Take tilapia as an example, eight key parts (main body, dorsal fin, pectoral fin, pelvic fin, head, eye, mouth, tail) are to be detected for the task of graph-based posture classification. Though not necessarily a mandate strategy, the stage of Tracking/Encoding can be conducted in such a way that whenever the posture of a target fish is successfully classified as anomalous, the first anomalous posture is used as a cue to start the tracking process. Finally, at the stage of Decoding, a state sequence consisted of a series of postures is to compare with 46 behavior templates suggested by aquaculture experts to make the final decision. With the streamlined framework in Fig.1, behavior detection results of different fishery species and environmental conditions (i.e., sensor data in Fig.1) can be integrally collected and subjected to bigdata analysis to provide optimal aquaculture conditioning, monitoring, and risk management to the aquaculture and breeding industry.

IV. STATE (POSTURE) DEFINITION

We use the positional relationship between key body parts of fish to define the fish posture. Owing to its fast identification of body parts and robustness to the untidy background, deep learning object detection is adopted in this work, rather than image processing approaches. After objection detection, a directed cycle graph (DCG) [6] is constructed. As will be seen, aside from its capability of exploiting the positional relationship between the body parts, DCG offers extra benefits.

A. FISH BODY PARTS DETECTION

Despite our method allows the use of any object detectors, hereinafter we will use Faster-rcnn [4] as an illustrative object detector. The Faster-rcnn was trained with 2986 images each containing multiple fish having their eight key body parts pre-labeled. Note that certain moving patterns may cause changes in the body shape, yet the positional relationship between the body parts remains unchanged. Thus, unless completely occluded, at least a certain number of body parts will be identified by a well-trained object detector. In light of this observation, one can therefore tackle spatio-temporal recognition problems such as the identification of anomalous behavior by classifying a state sequence containing a series of graph-based postures. Information needed to construct the graph and quantize the posture is all provided, directly or indirectly, by the object detector. Such a unified approach needs only a fairly small amount of labeled data for training the detection of fish body parts. In contrast, training Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM), and all their variants [22] inevitably require a large (if not infinite) amount of anomalous and normal behavior patterns. No question LSTM and RNN and derivatives are able to learn and remember a lot of longer-term information, e.g. sequences of 100 secs or more. However, one issue of them is that they are not hardware friendly, and it takes a lot of resources to train these networks fast. Also, it takes many resources to run these models in cloud that is not scalable, not to mention some field applications often require edge computing.

B. GRAPH-BASED POSTURE CLASSIFICATION

In this work, a DCG will be built for each fish if it is *sufficiently* detected. By "sufficiently" we mean at least two neighboring body parts, e.g. head and fin, are correctly detected by Faster-rcnn. The proof of which as well as the usefulness of DCG are elaborated as follows. A DCG is constructed as

$$G = (V, E) \tag{1}$$

$$E \subseteq \{\{x, y\} : (x, y) \in V^2, x \neq y\} \tag{2}$$

where V and E denote nodes and edges of a graph G, respectively. Whenever a fish is detected Faster-rcnn will draw a

bounding box to cover the entire fish body, meanwhile a separate node corresponding to the center of the bounding box drawn by Faster-rcnn will also be assigned to each of the correctly detected body parts. Among the eight key body parts (i.e. main body, dorsal fin, pectoral fin, pelvic fin, head, eye, mouth, tail), actually only four of them are needed for defining a posture (see Fig.2). In particular, pectoral fin, pelvic fin, eye, and mouth play an auxiliary role in building a DCG. If Faster-rcnn fails the detection of, say, head part, then eye or mouth can replace the head as a node if either one of them is detected. Likewise, as long as one of pelvic fin and pectoral fin is correctly detected, it is readily used as fin part. This strategy essentially increases the reliability of object detection, hence increasing the overall system performance.

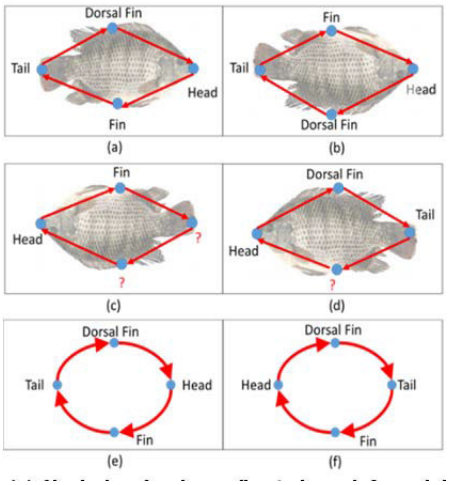


FIGURE 2. (a) Clockwise drawing a directed graph for a right-sided fish. (b) Clockwise drawing a directed graph for a left-sided fish. (c) Fish having the same graph as in (a). (d) Fish having the same graph as in (b). (e) DCG for right-sided posture. (f) DCG for left-sided posture.

Now consider head, dorsal fin, tail, and fin were all correctly detected. In that case there will be only two different DCGs if drawn clockwise. The first one is drawn for the fish in Fig.2a, which has a cyclic path of {head→fin→tail→dorsal fin→head} as shown in Fig.2e. The second one is drawn for the fish in Fig.2b, which has a cyclic path of {head→dorsal fin→tail→fin} as shown in Fig.2f. The difference in the sequence of the two paths comes from the fact that the fish in Fig.2a swims with its right body side facing the reader (called right-sided hereafter), whereas the fish in Fig.2b swims with its left side facing the reader (called left-sided hereafter). Namely, two fish having different DCGs must be flipping (vertically or horizontally) to each other. Conversely, if two fish share the same DCG such as the case in Fig.2a and Fig.2c, then if the fish in Fig.2a is right-sided, the fish in Fig.2c must be right-sided too. In fact, the posture of Fig.2c is just a result of rotating that of Fig.2a by 180°. Clearly, DCG is invariant to rotation.

The beauty of DCG is twofold. Firstly, from time to time Faster-rcnn could fail the detection of certain body parts, e.g. the symbols ‘‘?’’ in Fig.2c and Fig.2d. Yet, the only detected parts of the head and fin in Fig.2c are already sufficient to define a complete and valid DCG. To see this, we start with the antithesis invariance property, which requires the tail part must be located opposite to the head part, etc. Thus, knowing the position of the head (fin) is equivalent to knowing the position of the tail (dorsal fin). If the two detected parts are opposite to each other, the information provided by them is the same as if only one of them was detected. This dependency relation allows us to fill in the falsely detected parts and make the originally deficient cyclic path a valid DCG, namely having the complete path of {head→fin→tail→dorsal fin}. Secondly, by reading the path of a valid DCG, one can immediately judge which side the fish is facing toward the reader. As will be seen shortly, this property can help cut down the computation cost in the classification of postures by half!

We have seen that two fish having the same DCG may swim in opposite directions, e.g. Fig.2b vs. Fig.2d. To completely define the posture of fish, we present two ways to determine the swimming direction of fish. The first one utilizing the DCG scheme and capable of speeding up the posture classification is discussed below. After obtaining a DCG for the detected fish, A 3 × 3 grid is shown in Fig. 3a. The target fish is bounded by a yellow bounding box given by Faster-rcnn. The box is divided into 9 blocks having different sizes, with the center blocks being purposely set smaller than the rest. We then scan the nine blocks from top to bottom and left to right. In Fig.3a, where the head is found in the rightest block, then the corresponding element in a 3 × 3 matrix in Fig.3b will be filled with literal H. According to the path specified by DCG, the fin part will be found in the lower center block, and the corresponding element in a 3 × 3 pose matrix in Fig.3b is filled with literal F, and so on. We finally end up with a matrix in Fig.3b. For the fish swims toward the down right corner in Fig.3c, assume dorsal fin and head were undetected by Faster-rcnn. Now, with DCG and the successful detection of Tail and Fin, we can figure out that

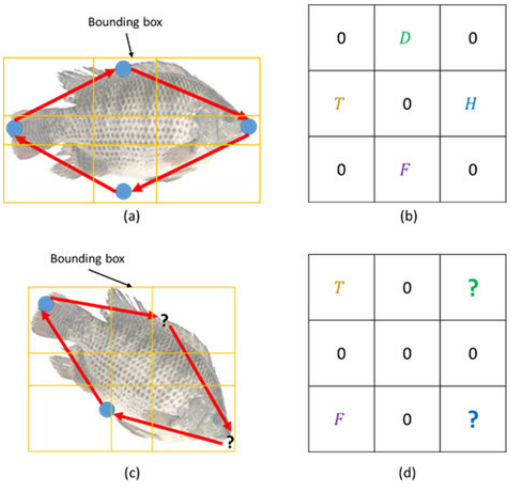


FIGURE 3. (a) code-31 posture. (b) corresponding 3 × 3 grid for (c). code-41 posture. (d) corresponding 3 × 3 grid for (c).

the missing 3rd and 9th parts in the pose matrix must be D and H, respectively. In this way, a full-filled pose matrix different from that in Fig.3b can still be established for the fish in Fig.3c, even though some parts were falsely undetected by Faster-rcnn.

We now ready to use the matrices obtained in Fig.3b and Fig.3d to determine the corresponding postures for the two fish in Fig.3a and Fig.3c, respectively. For the explanatory purpose, herein we only define sixteen different postures. Later we will see how to define more than 16 postures systematically. Among the sixteen template postures, eight of them being left-sided and the rest right-sided. Starting from the two upright postures in the first row of Fig.4, these sixteen template postures are obtained by rotating clockwise and counterclockwise at an interval of 45-degree. Also, the template postures were labeled by aquaculture experts into two major categories: normal (green) and anomalous (red) coded with two decimal digits, respectively. The rightmost digit is set to 1 representing the right-sided, and 0 the left-sided. Namely, the fish in Fig. 2(a) has its right side facing the reader. The leftmost digit encodes the swimming direction. Hence, {10, 20, 30, 40 . . . , 80} represent eight different swimming directions of a left-sided fish. Likewise, {11, 21, 31, 41 . . . ., 81} represent eight different swimming directions of a right-sided fish. Fig. 4 also lists sixteen separate matching matrices, acting as templates for classifying an input pose matrix. After obtaining the pose matrix, we can simply use Hamming distance or other similarity measures to identify which of 16 template postures the target fish belongs to.

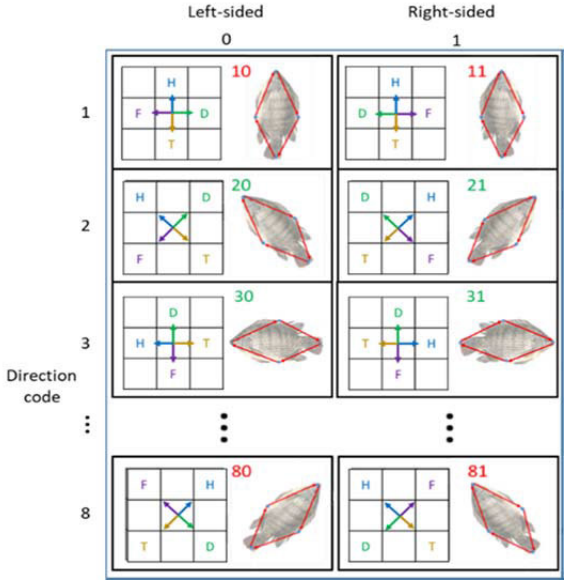


FIGURE 4. Sixteen matching matrices for posture classification.

To summarize, the posture of fish is identified by the following steps: (1) generating a valid DCG (examples as shown in Fig.2e and Fig.2f, (2) according to the valid DCG, the fish is determined either as right-sided or left-sided (3) if the fish is right-sided (left-sided), then the eight matrices listed in the right (left) column of Fig.4 is matched, one by one, with the pose matrix of Fig.3b. Use the fish of Fig.3a as an example, among all the 16 matching matrices in Fig.4, the matrix of code-31 in the 3rd row best matches with the input pose matrix of Fig.3b, hence the fish in Fig.3a will be assigned with the code of 31. In particular, because of DCG providing the information of which side the fish is facing, we need only perform 8 times of matching, instead of 16, so the computation time is cut half! We note that if a valid DCG is not possible for a target fish in the current image frame due to detection failure, then one can either use a trajectory prediction strategy before abandoning the tracking, or the posture code of the same fish in the last frame is used (i.e. patching).

V. TRACKING AND ENCODING

Our method allows us to track multiple fish simultaneously. Whenever Faster-rcnn detects a new fish in an input image frame, a tracking thread is set up for that fish. Each tracking thread keeps records of a fish under tracking, including the fish ID and 2-digit codes of previous postures classified. Each fish detected in an image frame will be tracked continuously until it is considered lost or a preset time-out has reached. Upon completion of a tracking thread, a state sequence containing several posture codes will be sent to DTW for determining if it is an anomalous behavior.

A. TRACKING

Fish tracking is difficult because the swimming direction of fish is not like a car, as fish may change its swimming direction abruptly. Even worse is that all Tilapia look alike. To solve these problems, we again rely on the coordinates of the bounding box. We calculate the overlapping area of box coordinates of the current frame and those of the next frame to determine whether two boxes contain the same fish. Denote F^t as the set of fish under tracking and f_i^t as an i^{th} fish at t^{th} frame. In (3), if $f_1^t = f_5^{t+1}$, it means the fish with ID:5 is the same as the one with ID:1 at the previous frame. Hence coordinates and posture code of f_5^{t+1} will be recorded in the thread of f_1^t. If no object matching the fish f_1^t was found at the next frame, then coordinates and posture code previously recorded will be used to predict [1, 2] the most likely path of the target fish for $(t+1)^{th}$ through $(t+v)^{th}$ frames, where v is heuristically set to 5. For example, in Fig. 5 we see the 2nd fish was lost in 7th frame, and no object in the next five frames matches the 2nd fish, thus the tracking thread terminated at 12nd frame (black dot in Fig.5). In contrast, assume the 3rd fish was also lost in 8th frame, but an object at 12th frames (green dot) that corresponds to the 3rd fish was found, then the tracking thread will be resumed at 12nd frame. Thus, Fig.5 verifies this scheme can predict the trajectory of fish and accordingly modifies the predicted trajectory over time, thus it can track a fast-moving target and can alleviate the problem

FIGURE 5. Tracking with prediction strategy for recovey of lost fish.

of sudden change in the swimming direction or sudden stop.

$$F^t = \left\{f_0^t, f_1^t, \cdots, f_n^t\right\}, \ldots, F^{t+v} = \left\{f_0^{t+v}, f_1^{t+v}, \cdots, f_m^{t+v}\right\} \tag{3}$$

B. ENCODING (BUILDING STATE SEQUENCES)

Input frames at different time points may contain a different number of fish, as shown in Eq.(3). A separate state sequence (thread) needs to be updated for each fish under tracking so that whenever a fish posture is classified, it will be assigned with a 2-digit code and put into the tracking record of that fish. Such an encoding procedure is applied to any fish under tracking, and it continues as input image frames are streamed into the system. In an arbitrary image frame, we may expect to see some new tracking threads are established, some updated, and some terminated. One key attribute of this work is that we consider the task of behavior classification as decoding a series of postures or states. Whenever a tracking thread is terminated, a state sequence can be converted to a time series and subjected to a DTW-based decoding process.

VI. DECODING BY DTW

In time series analysis, DTW has been applied to, in fact, any data that can be turned into a linear sequence, it is based on dynamic programming to calculate an optimal match between two temporal sequences, which may vary in speed or length. Due to the swimming speed and other motion factors, two state sequences with different lengths may belong to the same behavior. For example, a slower-moving fish and a faster swimming fish may act quite alike, yet they could be encoded into two different state sequences. Thus, in this work when a tracking thread is terminated, the optimal match between the resulting state sequence and the 46 behavior templates is conducted using the DTW technique. All the 46 behavior templates were consulted with aquaculture experts, 30 of them are of anomalous behaviors (see Table 1) and the rest 16 represent normal behaviors (not shown).

A. DYNAMIC TIME WARPING

Let t and r be two time series (in our case, state sequences) of lengths m and n, respectively. The goal of DTW is to find a mapping path $\{[p_1, q_1], [p_2, q_2] \ldots [p_k, q_k]\}$ such that the distance $\sum_{i=1}^{k} |t(p_i) - r(q_i)|$ on this path is minimized. To make DTW useful, three laws must be obeyed: Monotonicity, Continuity, and Boundary conditions, as prescribed in (4) and (5). The distance measures are calculated using (6) and the recursion formula in (7) with the initial condition $D(1,1) = |t(1) - r(1)|$. The resultant sequence after the termination of tracking may not have a length equal to those templates in Table 1. Also, the lengths of templates may be different from each other. Thus, if one can treat the state sequence as a time series, then DTW is an ideal choice for serving as a decoder for finding the best matching template.

$$w_k = (i,j), \quad w_{k+1} = \left(i', j'\right) \tag{4}$$

$$i \le i' \le i+1, \quad j \le j' \le j+1 \tag{5}$$

$$D(i,j) = |t(i) - r(j)| \quad or\ simply\ |\text{i-j}| \tag{6}$$

$$D(i,j) + \min\left[D(i-1,j), D(i,j-1), D(i-1,j-1)\right] \tag{7}$$

For the problem at hand, each posture is assigned with a 2-digit code. To facilitate meaningful DTW computation, herein we present a plausible and effective scheme for converting each coded posture into quantitative values for simulating physical signals at a specific time point. As a result, a sequence of postures turns into a linear sequence.

TABLE 1. Anomalous behavior templates.

15 anomalous templates of right-sided fish		15 anomalous templates of left-sided fish		Behavior class
1	{31,71,31}	2	{30,70,30}	Mild anomaly
3	{31,41,60,70,80,10,20,30}	4	{30,40,61,71,81,11,21,31}	Mild anomaly
5	{31,41,51,10,20,30}	6	{30,40,50,11,21,31}	Mild anomaly
⋮	⋮	⋮	⋮	⋮
27	{41,60,70,60,41}	28	{40,61,71,81,40}	Dying
29	{31,70}	30	{30,71}	Sink

B. POSTURE DESCRIPTOR FOR DTW COMPUTATION

To linearly quantize the posture of a target fish, we adopt the polar coordinate system. As before, the information provided by the object detector is utilized. First, the center of the main bounding box of fish is used as the origin, and the center of bounding boxes of head, dorsal fin, tail, and fin is used as the position of each part. Then, as shown in Fig.6. Cartesian coordinates are transformed to polar coordinates as follows: $(x_H = r_H cos\theta_H, y_H = r_H sin\theta_H)$, $(x_F = r_F cos\theta_F, y_F = r_F sin\theta_F)$, $(x_T = r_T cos\theta_T, y_T = r_T sin\theta_T)$, $(x_D = r_D cos\theta_D, y_D = r_D sin\theta_D)$, respectively. With the coordinates of x and y, the radii r_H, r_F, r_T, and r_D can be calculated according to the Pythagorean theorem. Also, the least positive coterminal angles of head, dorsal fin, tail, and fin are $\theta_H = cos^{-1}(x_H/r_H)$, $\theta_F = cos^{-1}(x_F/r_F)$, $\theta_T = cos^{-1}(x_T/r_T)$, and $\theta_D = cos^{-1}(x_D/r_D)$, respectively. For convenience, we will use a parameter γ to indicate which side the fish is facing the

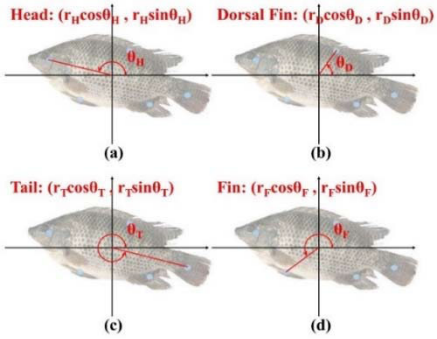


FIGURE 6. Positioning the four fish body parts using polar coordinate system. (a) θ_H for head. (b) θ_D for dorsal fin. (c) θ_T for tail. (d) θ_F for fin.

reader, if it is the right side, then γ is set to 1. Otherwise, γ is set to -1. Through geometry analysis, the relation between γ and the set of parameters of $\{\theta_H, \theta_F, \theta_T, \theta_D\}$ is prescribed by Lemma-1, which always holds regardless of the fish size and the total number of postures defined in Fig.4. As mentioned in Section IV, certain restrictions must be met to generate a valid DCG. Lemma-1 explicitly states these restrictions: (1) to generate a valid DCG, at least two neighboring body parts, e.g. head and fin, are correctly detected (2) the only criterion to determine the value of γ is the relative magnitude of angles defined by any two neighboring body parts that were correctly detected.

Lemma-1: $\gamma = -1$, iff $(\theta_H < \theta_F) \vee (\theta_T < \theta_D) \vee (\theta_H > \theta_D) \vee (\theta_T > \theta_F)$; $\gamma = 1$, iff $(\theta_H > \theta_F) \vee (\theta_T > \theta_D) \vee (\theta_H < \theta_D) \vee (\theta_T < \theta_F)$.

To proof Lemma-1, firstly, we first note that all the logical ORs are associated with two neighboring body parts. Secondly, there are at most 7 cases of detection failure, they are: {head}, {tail}, {head and tail}, {dorsal fin}, {fin}, {dorsal fin, fin}, and {main body}. Because if either one of these 7 cases occurs, then according to the antithesis invariance property, success detection of a single body part of the head (i.e. the first case) simply equals the second case, and vice versa. As such, the first three cases are equivalent to each other. Likewise, the last three cases are equivalent. Thus, there is no way to generate a valid DCG and hence the value of γ for these 7 cases. Interestingly, all these 7 cases have one thing in common: no two neighboring parts can be found.

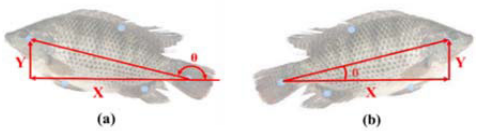


FIGURE 7. (a) Polar representation of left-sided fish. (b) Polar representation of left-sided fish.

After using Lemma-1 to determine the value of γ, we can further derive a scalar measurement for each posture in order to mimic a time signal for DTW computation. Fig. 7 shows a polar representation of a posture, using a vector going from tail to head. As the goal of DTW is to find out the best matching template for the input series of postures, we only concern with the key features that best represent a posture. Thus, the posture of a fish can be simply represented with the following polar descriptor given in (8).

$$\vec{P} = \begin{bmatrix} X \\ Y \\ \gamma \end{bmatrix} = \begin{bmatrix} cos\theta \\ sin\theta \\ \gamma \end{bmatrix}, \quad 0 \leqq \theta \leqq 2\pi \tag{8}$$

The nice thing about (8) is it is independent of the fish size. To generalize, a posture $\vec{P}_i$ at i^{th} time point after a rotation and a horizontal reflection operation can be obtained by

$$\begin{aligned} \overrightarrow{P_{i+1}} &= \begin{bmatrix} X_{i+1} \\ Y_{i+1} \\ \gamma_{i+1} \end{bmatrix} = \boldsymbol{RM}\vec{P}_i \\ \text{where } R &= \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 \\ sin\theta_i & cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ M &= \begin{bmatrix} -|\alpha|+1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -|\alpha|+1 \end{bmatrix}, \\ \theta_i &= \left(\frac{-2\pi}{k}\right)\gamma_{i+1}, \quad \alpha = (\gamma_{i+1} - \gamma_i), \\ k &= \text{control factor} \end{aligned} \tag{9}$$

Because both $\boldsymbol{R}$ and $\boldsymbol{M}$ are linear operators, the descriptor for a fish swimming in an arbitrary pose can be precisely defined by generalizing (9) to

$$\begin{aligned} \overrightarrow{P_h} &= \begin{bmatrix} X_n \\ Y_n \\ \gamma_m \end{bmatrix} = \boldsymbol{R^n M^m}\vec{P}_0, \\ h &= max\,(a, b)\,, a, b = 1, 2, .. \end{aligned} \tag{10}$$

By varying the value of k, any specific posture can be obtained by applying a times of rotations and b times of reflections to an initial posture $\vec{P}_0$. Compared to the 3×3 grid scheme, the polar descriptor conveys the posture information more neatly in terms of mathematical manipulations and physical interpretation. The greater the k value is, the finer difference between two separate postures can be defined. More importantly, all three physical values contained in the descriptor solely concerns the posture itself. Fig. 8 shows the corresponding descriptor for each of the sixteen postures. Upon the termination of a tracking tread, DTW is employed to match the final state sequence with each of the 46 behavior templates using (6) and (7). The template having the least distance is picked, and its class is output as the final decision.

Optionally, sensory data collected by various sensors can be incorporated into the final stage of Fig.1 to join the outcome of DTW for making the final decision and for any meaningful post-analysis. To name a few, when the temperature is too low, frostbite often comes with the fish's activity decreases and sinking to the bottom; hypoxia may easily lead to fish death in a salt-resistance experiment. If minor anomalous behaviors can be found soon enough and the fish

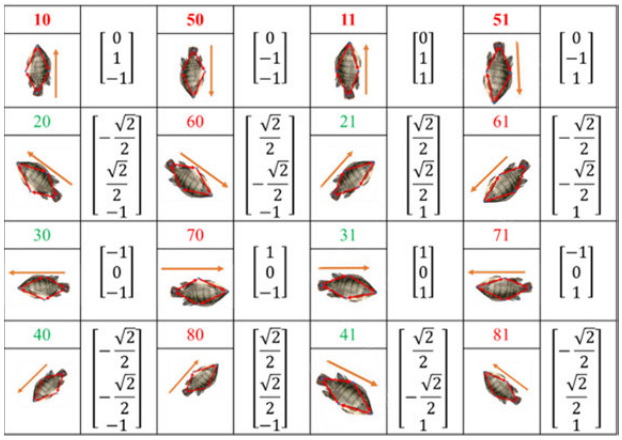


FIGURE 8. Sixteen posture codes and the corresponding polar descriptors, with green code representing a normal posture, red code representing an abnormal posture.

can be immediately treated, then fish might have a 70% chance of full recovery. On the other hand, if a dying anomalous behavior is spotted, it often means the fish might die within 24 hours. Experimental results given next are to show that anomalous behaviors can be detected at an earlier stage so that measures of drug administration and isolation facility can be deployed to improve the survival rate of underwater organisms.

VII. EXPERIMENTAL RESULTS

All the experiments (including training and test) of this work were based on the following configuration. Hardware: Intel i9-9900K CPU, Nvidia GeForce RTX 2080Ti x2 (SLI) GPU, 32G DDR4-3000 RAM. Software: Windows-10 OS, Python 3.7.9, Cuda 10.0, and Cudnn 7.6.5. Both Faster-rcnn [4] and Yolo-v4 [19], [23] use Tensorflow 1.15.0, EfficientDet-D1 [23] uses Pytorch 1.6.0. Excluding the configuration time, the computational complexity of our method can be shown as $O(n^2)$. To verify the effectiveness of our method, experiments of cold-resistance and salt-resistance were conducted, and anomalous behaviors were observed thorough out the drastic environmental changes. We recorded tilapia for 648 hours. Among all the images, anomalous behaviors account for 10% only, indicating they are relatively rare and hence costly in preparation of training data. Despite this, we prepared 33 video clips, 60 mins in total length, the massive amount of which should serve our purpose of performance test.

Fig.9(a) shows the object detection results for multiple fish. As can be seen, Faster-rcnn drew four main boxes covering the four fish respectively, amidst several other smaller boxes for body parts detected. In Fig.9(b), both fish-1 and fish-2 were detected with two body parts only. But because dorsal fin and head were correctly detected for fish-2, Lemma-1 is satisfied, so the falsely undetected parts of fin and tail can be filled in generating a valid DCG. For fish-1 in Fig.9(c), correct detection of head and tail does not meet any of the four conditions set forth in Lemma-1, thus a null descriptor [0, 0, 0] was assigned to fish-1 to indicate failure classification. The rest of the three fish were correctly coded and their corresponding polar descriptors are shown. Fig. 10 shows a resulting state sequence, where postures sequentially sampled at eight different frames were successfully coded as {31, 41, 51, 20, 20, 30, 30, 30}. The sequence was converted to a series of polar descriptor for DTW computation, and it best matches the template {31,41, 51, 10,20,30} with the least distance = 0.765.

We also compared the performances of our method when using three different state-of-the-art deep neural networks as object detectors: Faster-rcnn, EfficientDet-D1, and Yolo-v4. Using a test set of 192 state sequences (39 anomalous, 153 normal), the results are given in Table 2, 3, and 4, respectively. Faster-rcnn has the best performance with Accuracy ≈92.8%, F1-score =0.81, Precision ≈84%, Sensitivity ≈ 80%, and mAP (mean average precision) = 49.80%. In terms of F1-score, the performance ordered as follows: Faster-rcnn > Yolo-v4> EfficientDet-D1. Computation load breakdown into stages of Fig.1 is listed in Table 5. Faster-rcnn can reach 27 FPS (frames per sec), indicating our method is feasible for real-time underwater applications.

TABLE 2. Result of using Faster-rcnn as object detector.

		Prediction		mAP=49.80%
		Positive	Negative	
Ture Abnormal Behavior	Pos.	31 (TP)	8 (FN)	Sensitivity=79.5%
	Neg.	6 (FP)	147 (TN)	Specifity=96.1%
Accuracy= $\frac{TP+TN}{TP+TN+FP+FN}$ =92.8%		Precision=83.8%		F1-score=0.81
		FOR=5.16%		

TABLE 3. Result of using YOLO-v4 as object detector.

		Prediction		mAP=35.1%
		Positive	Negative	
Ture Abnormal Behavior	Pos.	22	17	Sensitivity=56.4%
	Neg.	6	147	Specifity =96.1%
Accuracy=88.0%		Precision=78.57%		F1-score=0.67
		FOR=10.37%		

TABLE 4. Result of using EfficientDet-D1 as object detector.

		Prediction		mAP=41.1%
		Positive	Negative	
Ture Abnormal Behavior	Pos.	24	15	Sensitivity=61.5%
	Neg.	20	133	Specifity =86.9%
Accuracy=81.8%		Precision=54.6%		F1-score=0.58
		FOR=10.14%		

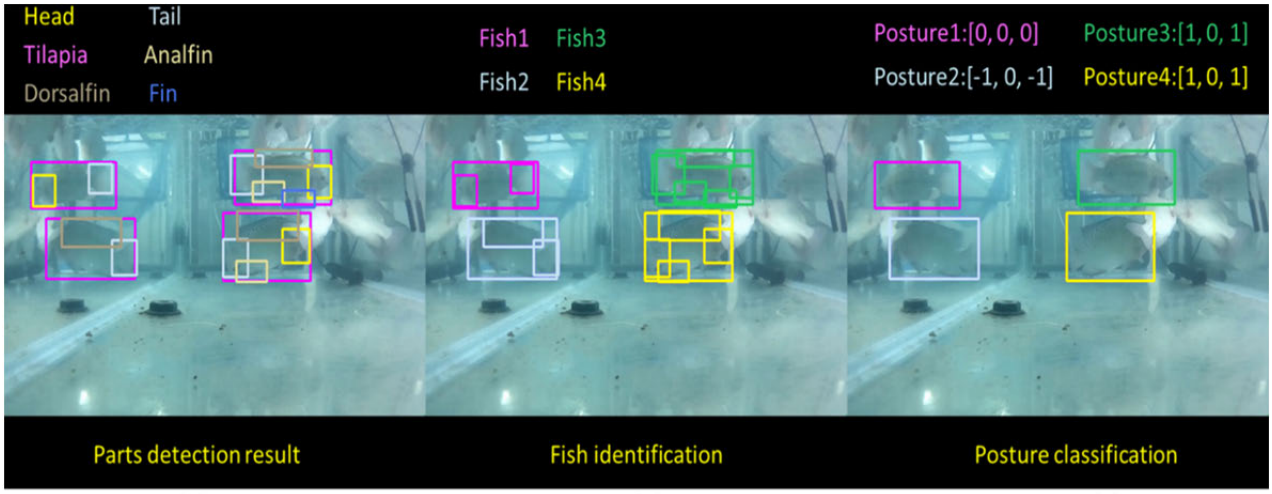


FIGURE 9. (a) Body parts detected. (b) Multiple number of fish identified and body parts thereof. (c) Three postures correctly coded, one failed.

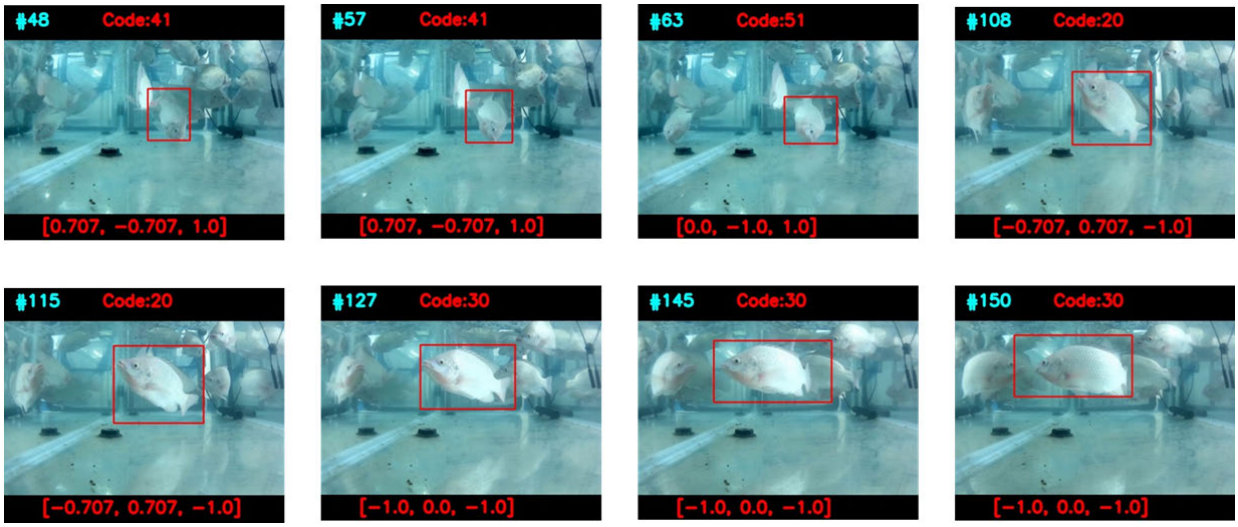


FIGURE 10. Illustration of a state sequence, which matches the template {31, 41, 51, 10, 20, 30} with Min(DTW score) = 0.765.

TABLE 5. Computation load profile.

in seconds	Faster-rcnn	YOLOv4	EfficientDet
Object detection	0.03463	0.05509	0.05690
Posture classification	0.00004	0.00005	0.00002
Tracking/encoding	0.00065	0.00108	0.00072
DTW	0.00996	0.01617	0.00939
Total time	0.04528	0.07239	0.06703
FPS	27.26	17.38	17.13

VIII. CONCLUDING REMARKS

We have presented a novel solution for the detection of anomalous events for underwater fish. The main contributions of this study are threefold: (1) based on information provided by object detectors to implement a graph-based classification scheme for encoding a fish posture, (2) realization of a real-time anomalous behavior detection/classification by treating the fish behavior as an encoded time series, each comprising a series of postures encoded in the course of a self-recovery tracking algorithm, (3) providing a framework for the establishment of AI training dataset and open dataset for fish behaviors. Compared to RNN and LSTM, our method is relatively economic as the training of deep learning-based object detection requires a much smaller amount of labeled data, and hence much less computing resource, to train the object detection of fish body parts.

Our method is characterized in that information such as coordinates of bounding boxes provided by the object detectors is fully utilized in the implementation of all subsequent processes of DCG, fish tracking, and polar descriptor calculations. With Lemma-1, a valid DCG can be generated despite some falsely undetected body parts This property is significant, as state-of-the-art object detectors can rarely deliver mAP exceeding 70%, such an inherently imperfection leads to occasional detection failure. A valid DCG used in conjunction with the bounding box of object detectors scribed into a 3×3 grid is effective for implementing a posture

classifier. We have verified the self-recovery tracking scheme can predict the trajectory of fish to alleviate the problem of sudden change in the swimming direction or sudden stop, implementation of which is also based on the object detection. In (9) and (10), we have seen that a unique polar descriptor can be defined for a specific posture based on bounding boxes. The descriptor assigned to a specific posture is independent of the fish size. Such scale invariance can only be achieved in RNN and LSTM at the expense of expensive hardware, computing time, and a huge amount of training data. More importantly, the polar descriptor is ideal for quantifying the coded posture so as to provide the signal values needed for DTW computation.

Experiments in Fig.9 and Fig.10 were conducted in a small glass tank to purposely illustrate the robustness of our method against untidy background and light reflections and interferences. The test results have also verified that despite the high performance of Faster-rcnn, imperfect instance segmentation could occur. However, as specified in Lemma-1, unless the detection result of the object detector fits in any of the six failure cases as discussed in Section VI, our method can handle most cases of failure detection such as fish-2 in Fig.9. Confusion matrix results given in Table 2,3, and 4 all verified that our method, when using different object detectors, always has stable performance in terms of classification accuracy and computation cost. In the future, more behavior templates or classes and environmental parameters of temperature and salinity will be used together to conduct a more extensive study on the relevance between behavior and other factors such as genome and stress/strain imposed on the fish. Last but not the least, we will also consider the deployment of edge computing in sea cages that involves the compressed or lite version of Faster-rcnn, EfficientDet-D1, and YOLO-v4. Extensive studies on the relevance between 0behavior and other factors such as genome and stress/strain imposed on the fish.

REFERENCES

[1] R.-J. Huang, C.-Y. Tsao, Y.-P. Kuo, Y.-C. Lai, C. Liu, Z.-W. Tu, J.-H. Wang, and C.-C. Chang, "Fast visual tracking based on convolutional networks," *Sensors*, vol. 18, no. 8, p. 2405, Jul. 2018.

[2] Y.-C. Lai, R.-J. Huang, Y.-P. Kuo, C.-Y. Tsao, J.-H. Wang, and C.-C. Chang, "Underwater target tracking via 3D convolutional networks," in *Proc. IEEE 6th Int. Conf. Ind. Eng. Appl. (ICIEA)*, Apr. 2019, pp. 485–490.

[3] C. C. Liu, "Incorporating object detection and stereo vision for real-time underwater fish range measurement," M.S. thesis, Nat. Taiwan Ocean Univ., Keelung, Taiwan, 2018.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[5] J. C. Fournier, *Théorie des Graphes et Applications: Avec Exercices et Problèmes*. Lavoisier, 2011.

[6] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, 1994, vol. 10, no. 16, pp. 359–370.

[7] M. Thida, H. L. Eng, and B. F. Chew, "Automatic analysis of fish behaviors and abnormality detection," in *Proc. MVA*, 2009, pp. 278–282.

[8] C. Beyan and R. B. Fisher, "A filtering mechanism for normal fish trajectories," in *Proc. 21st Int Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 2286–2289.

[9] C. Beyan and R. B. Fisher, "Detection of abnormal fish trajectories using a clustering based hierarchical classifier," Tech. Rep., 2013.

[10] C. Beyan and R. B. Fisher, "Detecting abnormal fish trajectories using clustered and labeled data," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1476–1480.

[11] D. Semani, T. Bouwmans, C. Frélicot, and P. Courtellemont, "Automatic fish recognition in interactive live videos," in *Proc. Int Workshop IVRCIA, 6th World Multi-Conf. Systemics, Cybern. Inform., Sci.*, 2002, pp. 14–18.

[12] D. Semani, C. Saint-Jean, C. Frélicot, T. Bouwmans, and P. Courtellemont, "Alive Fish species characterization for on line video-based recognition," *Proc. SPR*, vol. 2002, pp. 689–698, 2002.

[13] F. ElBaf, T. Bouwmans, and B. Vachon, "Comparison of background subtraction methods for a multimedia learning space," in *Proc. SIGMAP*, 2007, pp. 153–158.

[14] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1, Dec. 1999, pp. 255–261.

[15] T. Pinkiewicz, "Computational techniques for automated tracking and analysis of fish movement in controlled aquatic environments," Univ. Tasmania, Hobart, TAS, Australia, Tech. Rep., 2012.

[16] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H.-J. Chen-Burger, R. B. Fisher, and G. Nadarajan, "Automatic fish classification for underwater species behavior understanding," in *Proc. 1st ACM Int. Workshop Anal. Retr. Tracked Events Motion Imag. Streams (ARTEMIS)*, 2010, pp. 45–50.

[17] C. Spampinato, S. Palazzo, B. Boom, J. van Ossenbruggen, I. Kavasidis, R. Di Salvo, F.-P. Lin, D. Giordano, L. Hardman, and R. B. Fisher, "Understanding fish behavior during typhoon events in real-life underwater environments," *Multimedia Tools Appl.*, vol. 70, no. 1, pp. 199–236, May 2014.

[18] A. Jalal, A. Salman, A. Mian, M. Shortis, and F. Shafait, "Fish detection and species classification in underwater environments using deep learning with temporal information," *Ecol. Informat.*, vol. 57, May 2020, Art. no. 101088.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[20] R. Reig-Bolano, M. S. Iserra, and P. Marti-Puig, "Foreground detection in a multi-target fish tracking from video-recordings using U-Net based architecture," in *Artificial Intelligence Research and Development: Current Challenges, New Trends and Applications*, vol. 308. 2018, p. 381.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[22] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters," *Procedia Comput. Sci.*, vol. 125, pp. 676–682, 2018.

[23] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding, and S. Wen, "PP-YOLO: An effective and efficient implementation of object detector," 2020, *arXiv:2007.12099*. [Online]. Available: http://arxiv.org/abs/2007.12099

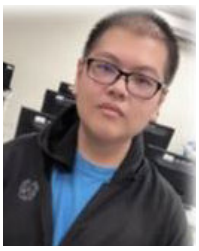
JUNG-HUA WANG was born in Keelung City, Taiwan, in 1959. He received the B.S. degree in electrical engineering from National Chen Kung University (NCKU), Taiwan, in 1983, and the M.Sc. and Ph.D. degrees in electrical engineering from Texas Tech University, USA, in 1990. Afterwards, he joined Texas Instrument as a Process Engineer. Before going to U.S. for academic degrees, he worked as a Test Engineer with the National Chung-Shan Institute of Science and Technology (CSIST), a primary research and development institution of Taiwan for two years. Since then, he joined the Department of Electrical Engineering, National Taiwan Ocean University. He is currently the Chairman of the Department of Electrical Engineering, National Taiwan Ocean University. He also serves as the Chairman of AI Research Center, National Taiwan Ocean University. In 2018, he visited the IBM Watson Center in New York and went on to SIO/UCSD for establishing international collaboration between NTOU and SIO. His research interests include image processing and neural networks. In recent years, he has focused on applying AI techniques underwater fish behavior analysis and applications. He works actively and closely with researchers in several other disciplines of aquaculture and genetic breeding. He holds nine patents and has presented and published over 90 technical articles in international journals and conferences.

SHIH-KAI LEE was born in Hsinchu, Taiwan, in 1994. He received the B.S. degree from the Department of Money and Banking, National Kaohsiung University of Science and Technology (NKUST), Taiwan, in 2016, and the M.Sc. degree in electrical engineering from National Taiwan Ocean University (NTOU), in 2020, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering. In 2019, he presented an article in IEEE AIVR International Symposium held in San Diego, CA, USA. He also visited SIO/UCSD to participate in an academic discussion with Dr. Jules. His research interests include image processing and neural networks, focusing on applications of AIoT to underwater fish behavior analysis.

YI-CHUNG LAI received the bachelor's degree in computer science from National Formosa University, in 2016, and the master's degree in electrical engineering from National Taiwan Ocean University (NTOU), in 2018. Since the end of 2017, he has been participating an AI research project financially supported by MOST of Taiwan government, aiming to improve the harvest of offshore aquaculture by combining techniques of AI, aquaculture, and other disciplines. During the past three years, his research interests include image processing, computer vision, and their applications using AI techniques. In 2018, he joined the group of AIBMRC of National Cheng-Kung University to visit IBM Watson Center in New York, and went on to SIO/UCSD for setting up international collaboration between SIO, NTOU, and AIBMRC. He has ample professional skills related to the AI and programming languages. He has published three international conference papers. His research interests also include image analysis and pose/behavioral analysis.

CHENG-CHUN LIN was born in Kaohsiung, Taiwan, in 1996. He received the B.S. degree in electrical engineering from National Taiwan Ocean University (NTOU), Taiwan, in 2018, where he is currently pursuing the master's degree with the Department of Electrical Engineering. His research interests include mathematics, physics, and programming languages. He is also a Team Member of the Machine Learning Laboratory, working on applying machine learning and deep learning techniques to diverse fields of aquaculture farming, oceanic engineering, and biomedicines as well.

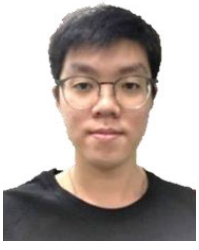
TING-YUAN WANG was born in Taipei City, Taiwan, in 1993. He received the B.S. degree in laboratory science and biotechnology from China Medical University, Taiwan, in 2016. He is currently pursuing the master's degree with the Department of Biomedical Engineering, National Yang-Ming University. As a substitute civilian serviceman, in 2018, he joined the Computational Intelligence Technology Center, Industrial Technology Research Institute (ITRI), there he works as a Research Assistance and is responsible for implementing Biomedical applications using machine learning and data science techniques. With skills in computer vision, image processing, and programming languages, he likes working actively and closely with researchers from other disciplines of bioscience and engineering fields. His hobbies include: playing violin, overseas sightseeing, and Japanese cuisine.



YING-REN LIN was born in Pingtung City, Taiwan, in 1998. He received the B.S. degree in information technology from Meiho University (MU), Taiwan, in 2020. He is currently pursuing the master's degree with the Department of Electrical Engineering, National Taiwan Ocean University (NTOU), Taiwan. His research interests include computer hardware, computer networking, and programming languages. He is also a Team Member of the Machine Learning Laboratory, working on applying deep learning and other machine learning techniques to diverse fields of aquaculture farming, oceanic engineering, and biomedicines as well.

TE-HUA HSU is currently with the Department of Aquaculture as an Assistant Professor and the Director of the Center for Aquatic Organisms and Conservation, National Taiwan Ocean University (NTOU). He has been actively participating in various research groups of genetic breeding team, AI breeding team, and genetic and fishery team in NTOU. His research interests include farming and breeding of mollusk, cuttlefish, marine crab, genetic management, environmental DNA, and so on. He has published 35 articles h, 18 of them are SCI indexed. He holds ten invention patents of Taiwan. He is a PI for over 12 research projects, including four from the Ministry of Science and Technology, six from the Council of Agriculture, and two other industry-academic projects. His research interests also include genetics, breeding, and stock enhancement of cuttlefish, marine crab, hard clam, and dog conch.

CHANG-WEN HUANG was born in Tainan City, Taiwan, in 1977. He received the B.S. degree in animal science from the National Chiayi Institute of Technology, Taiwan, in 1999, and the M.S. and Ph.D. degrees in animal science from National Chung Hsing University, Taiwan, in 2001 and 2007, respectively. Before going to National Taiwan Ocean University (NTOU), he had worked for five years as a Postdoctoral Fellowship with the Institute of Cellular and Organismic Biology (ICOB), Academia Sinica, and a Premier Research and Development Institution in Taiwan. Since then, he joined the Department of Aquaculture, NTOU. Before taking the position as the Associate Dean of the Department of Aquaculture, he used to be the Director of the Admission Division, NTOU. He also serves as the Deputy Director for the Operations Center of University-Industry Collaboration, NTOU. His research interests include aquatic genetics and molecular breeding. In recent years, he has focused on applying next-generation sequencing (NGS) techniques platform to the study of functional gene mechanisms, such as growth, stress resistance, and color development of commercial fish and ornamental aquatic species, and the industrial application of marker-assisted selection (MAS). He is also actively and closely cooperating with researchers in several other disciplines of electrical engineering, AI, and biological information. He has published more than 50 academic articles and reports in international journals, international and cross-strait exchange conferences.



CHUNG-PING CHIANG was born in New Taipei City, Taiwan, in 1996. He received the B.S. degree from the Department of Aquaculture, National Taiwan Ocean University. In 2019, he participated in Annual Forum for the Fisheries Society of Taiwan. He is currently involved in researches on marker-assisted selection and breeding of salt tolerance traits in Taiwanese Tilapia.