

AUTOMATIC CONTROLLING FISH FEEDING MACHINE USING FEATURE EXTRACTION OF NUTRIMENT AND RIPPLE BEHAVIOR

HILMIL PRADANA AND KEIICHI HORIO

Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0196, Japan
pradana.muhamad-hilmil505@mail.kyutech.jp; horio@brain.kyutech.ac.jp

Received March 2021; revised July 2021

ABSTRACT. *Controlling fish feeding machine is a challenging problem because experienced fishermen can adequately control based on assumption. To build robust method for reasonable application, we propose automatic controlling fish feeding machine based on computer vision using combination of counting nutriment and estimating ripple behavior using regression and textural feature, respectively. To count number of nutriment, we apply object detection and tracking methods to acknowledging the nutriment moving to sea surface. Recently, object tracking is an active research and challenging problem in computer vision. Unfortunately, the robust tracking method for multiple small objects with dense and complex relationships is an unsolved problem in aquaculture field with more appearance creatures. Based on the number of nutriment and ripple behavior, we can control fish feeding machine which consistently performs well in real environment. The proposed method presents the agreement for automatic controlling fish feeding by the activation graphs and textural feature of ripple behavior. Our tracking method can precisely track the nutriment in next frame compared with other methods. Based on computational time, the proposed method reaches 3.86 fps while other methods spend lower than 1.93 fps. Quantitative evaluation can promise that the proposed method is valuable for aquaculture fish farm with wide application to real environment.*

Keywords: Fish feeding, Ripple behavior, Textural feature, Nutriment, Tracking, Multiple small objects

1. Introduction. Raising fish utilization to complete the demand of human and animal resourcing in worldwide drives researchers exploring and improving technology in aquaculture industry [1-3]. Based on global aquaculture production statistic database [4], the proportion of aquatic animals farmed is 55.1 million tonnes in 2009 and significantly increasing to 82.1 million tonnes in 2018. This trend indicates that aquaculture industry has to create a new technique to enlarge economic scale with reducing production cost and increasing production efficiency.

Optimizing fish feeding process is the most influential aspect because the process itself takes up to 40 percent of total production cost [5-7]. Increasing company profit with controlling adjustment of nutriment is a current problem of active aquaculture fish farm without wasting nutriment and dropping quality of fishes. Wasting nutriment create a big effect in water pollution because quality of water is highly related to ensure survival rate of fishes and to enlarge fish fertility rate. Adjustment for giving nutriment is critical component to manage aquaculture development related to quality of water. From economic aspect, controlling fish feeding gives more benefits and determines quality of management expenses for aquaculture industry.

Recently, there are numbers of researchers creating several techniques to control amount of nutriment given to fishes. Those techniques can be defined as mechanical device controls [8-18], and computer vision approaches [5,19-29]. Mechanical device control uses external sensor which has different functions for monitoring, identifying, and evaluating fish feeding behavior [8-11]. External sensors can be categorized by acceleration sensors attached inside fish body and sensors related to water quality parameters. Accelerative data loggers are used to identify and to classify different kinds of prey or nutriment types [9,12-14]. In conclusion, this sensor has higher accuracy and lower false detection rate. However, it can create injury to fish body and installation sensor is costly for each fish body. Another type of safe sensor, without harming fish body [15-18], is measurable sensors for detecting alteration of temperature and oxygen concentration. These sensors attach in underwater fish tank and calculate changing flow rate and temperature of the water caused by the activity of fish. Although these sensors create higher accuracy and prove save cost of fish feeding up to 29%, they can be easily interrupted by other parameters such as weather temperature.

On the other hand, computer vision approach has been commonly used in aquaculture industry because it performs in real-time controlling and low cost for maintainable equipment [27-29]. This approach can be useful to classify gender and species, age and size measurement, quantity and quality inspection, counting and monitoring fish behaviors [19,20]. Based on wavelength signal imaging, it can be categorized into infrared [19-23] and visible light [5,24-26]. The input video from infrared light is suitable to identify fish behavior and achieves better imaging effects under low visible light condition. However, it is suitable to identify fish behavior, analyzing of infrared imaging is remaining unsolved because object appearance from this imaging is unlike real condition. Besides that, application of detecting fish behavior using visible light gives several problems which only focuses on laboratory culture model [27-29]. It shows that the methods applied on laboratory culture model perform worst in real environment because several conditions cannot be anticipated such as flying bird, waste, appearance of fish tank boundary, and other undesirable objects. Automatic controlling fish feeding in real environment is still a challenging problem in aquaculture field because data appearance and weather condition can affect accuracy of detecting and tracking results. To analyze fish feeding behavior, two feature extractions such as tracking approach using regression and estimating ripple behavior using textural analysis are applied to controlling fish feeding machine which gives value for aquaculture fish farm with wide application to real environment datasets.

Object tracking is an active research in applications to computer vision [30-39]. Not only single-object-tracking but also multiple-object-tracking is an unsolved problem since multiple-object association requires correct collision in sequence frames. Multiple-object-tracking is a more reliable solution rather than applying many of single-object-tracking. The tracking problem can closely be determined to detection problem and even can be interpreted as an extension of object detection problem. Over the past decades, there are a number of researchers focused on developing tracking-by-detection to generate more accurate and faster determination of object tracking [32-39]. The tracking-by-detection method is the most popular tracking paradigm because object tracking is result of detection with the same label in sequential frame. Good quality of the detection algorithms determines tracking result and these algorithms mostly use convolutional neural networks (CNNs) for the detection step. Proving that object tracking is an extension of object detection techniques is that the region proposal network (RPN), one of faster R-CNN detectors [40], adopts SiamRPN tracker [41,42]. Multi-aspect-ratio anchors resolve bounding box estimation problem which has interference from previous trackers and significantly enhances the performance of siamese-network-based trackers. Unfortunately, the robust

tracking method for multiple small objects with dense and complex relationships is still a challenging problem in aquaculture field with more appearance creatures.

The idea of textural feature for understanding ripple behavior is applied from image quality assessment (IQA) index which can be represented as quantity of human perception referred to quality of image. Since texture is represented in photographic images, it is important to develop objective IQA index which has consistent value with perceptual similarity. In past decades, mean square error (MSE) and structural similarity (SSIM) had been the standard approach to evaluate the signal accuracy and quality although these approaches had the poor correlation result with human assumption and perception. Estimation of ripple behavior can also be defined by human assumption of which the size and number of ripple can be used to adjust the activity level of ripple.

In this paper, we present an automatic controlling fish feeding machine using combination of two feature extractions which estimate nutriment position and ripple behavior using regression and textural feature, respectively. The proposed method can ignore undesirable object by understanding behavior for each tracking object. Although the proposed method cannot reach higher accuracy comparing use of sensors [9,12-18], the advantage of our method can reduce equipment costs with requiring single camera placed above vessel with a highly distraction of ocean wave and innumerable small and dense nutriments. Our method is constructed by object detection of nutriment and ripple. Regression model is used to predict each detection of nutriment in next frame. Estimating ripple behavior is determined by textural feature with ripple detection area as input image to compute global variance of VGG network [43]. To summarize our work, we present several contributions.

- We propose a new novel automatic controlling fish feeding machine using feature extraction of nutriment and ripple behavior which can be useful to improve the production profit in fish farms by controlling the amount of nutriment in optimal rate and to optimize the use of fish feeding machine.
- We propose tracking approach using regression model which outperforms result compared with state-of-the-art multiple object tracking methods.
- We present the advantage of the proposed method which can perform well in real environment datasets.

The remainder of this paper is structured as follows. In Section 2, we briefly present regression and textural feature of ripple behavior technique to give robust solution performance and learning accuracy by combining regression as tracking approach and textural feature for estimating of ripple behavior. Experiment section containing the information of datasets, evaluation approach, and quantitative evaluation for regression and textural feature of ripple behavior is depicted in Section 3. Evaluation result and performance of the proposed method and comparison with state-of-the-art benchmark methods are presented in Section 4. In Section 5, we review advantages of the proposed method, limitations and the conclusions.

2. Proposed Method. Our formulation is based on tracking algorithm using regression where we estimate each nutriment detection in next frame and textural feature of ripple behavior as input image to compute global variance of VGG network. In order to provide some background and formally introduce our approach, we firstly provide diagram and algorithm of regression and textural feature of ripple behavior approaches. We then explain how the proposed method works to real environment. The proposed method contains three parts: detection, regression and textural feature of ripple behavior which are shown in Figure 1 where the detection result is represented as black color value and predicted result of regression is shown by red color value. Regression contains both transformation and inverse transformation, regression approach and counting nutriments while

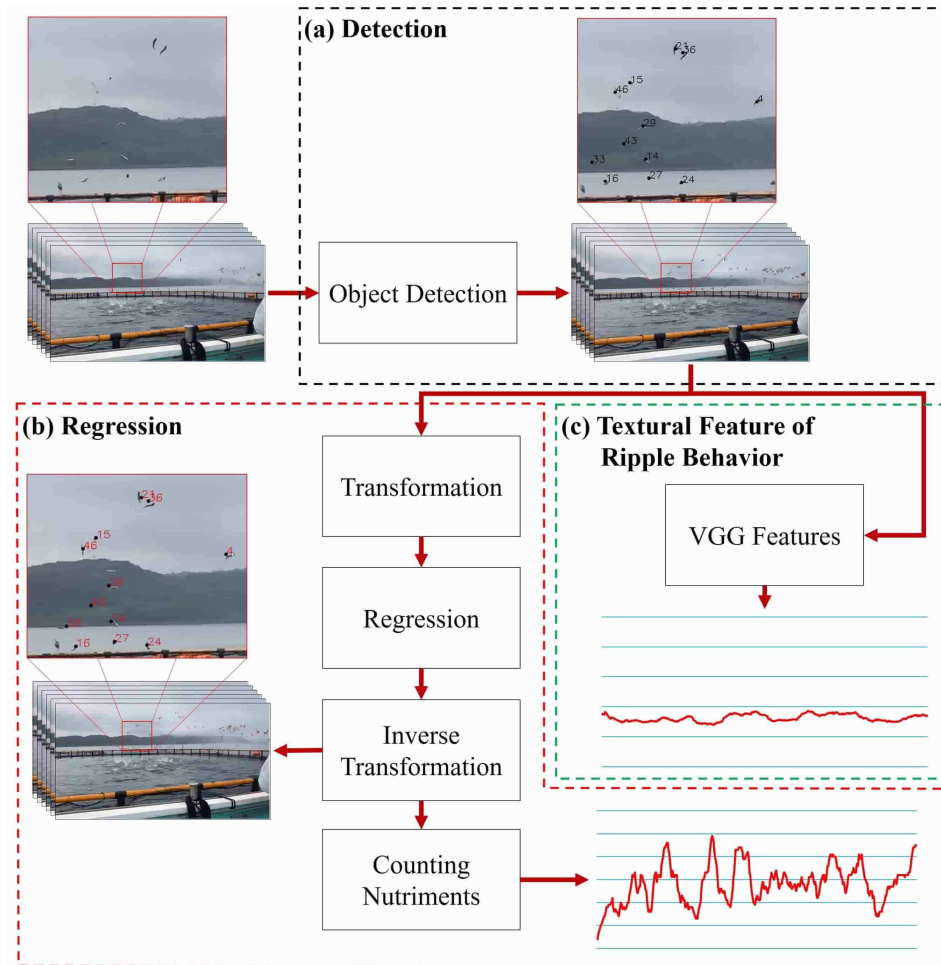


FIGURE 1. (color online) Diagram of the proposed method. (a) The input video is received and applied object detection using YOLOv4 [44]. (b) Creating model for predicted tuna nutriment in next frame using regression. Before and after regression process, we calculate transformation and inverse transformation to create robust result. (c) Textural feature of ripple behavior uses global variance of VGG model to represent estimation of ripple behavior for each frame on input video.

textural feature of ripple behavior uses VGG network as extracted features to compute global variance of VGG network. We also present the algorithm of the proposed method in Algorithm 1.

2.1. Object detection. The idea of object detection is to find bounding box in each nutriment and ripple area associated in regression and textural feature. In implementation of object detection, YOLOv4 [44] has different parts of object detectors: backbone, neck, and head. Backbone uses CSPDarknet53 as feature-extractor model and is more suitable for classification than for detection. Neck utilizes spatial pyramid pooling (SPP) which is a pooling layer removing fixed size constraint of network and path aggregation network (PAN) which aims to improve the information flow of segmentation network. The latter of neck architecture has different forms to the original PAN and it is a modified version with replacing the addition layer with a concat. On the other hand, the head of YOLOv4 [44] has a similar form with YOLOv3 [45] which computes training model with bounding box of object detection $B = (\hat{x}^b, \hat{y}^b, \hat{w}^b, \hat{h}^b)$ and bounding box $G = (x^g, y^g, w^g, h^g)$ of ground

Algorithm 1: Regression and textural feature of ripple behavior algorithm

Input : D : a fish feeding video data which contains f frames.
Auxiliary methods: Tuna nutriment detection, transformation, inverse transformation, regression, counting nutriments, VGG features.
Output : A graph of regression and textural feature of behavior.

- 1 input a fish feeding video data D .
- 2 **for** $\vartheta \leftarrow 1$ **to** f **do**
- 3 calculate bounding box of tuna nutriment detection $\varphi^t(\vartheta)$ and ripple area detection $\varphi^r(\vartheta)$.
- 4 create an area of top-left points $tl_{\vartheta 1,2} = (\hat{x}_{\vartheta}^{tl}, \hat{y}_{\vartheta}^{tl})$, top-right points $tr_{\vartheta 1,2} = (\hat{x}_{\vartheta}^{tr}, \hat{y}_{\vartheta}^{tr})$, bottom-left points $bl_{\vartheta 1,2} = (\hat{x}_{\vartheta}^{bl}, \hat{y}_{\vartheta}^{bl})$ and bottom-right points $br_{\vartheta 1,2} = (\hat{x}_{\vartheta}^{br}, \hat{y}_{\vartheta}^{br})$ of ripple detection.
- 5 compute normalized nutriments $\varphi^{\kappa}(\vartheta)$ from geometry transformation $\varphi^{\xi}(\vartheta)$ and rotated nutriments $\varphi^{\psi}(\vartheta)$.
- 6 predict position of normalized nutriments $\varphi^{\kappa}(\vartheta + 1)$ in next frame using regression model μ^x and μ^y .
- 7 transform the result of regression $\varphi^{\kappa}(\vartheta + 1)$ into original pixel position in real images $\varphi^t(\vartheta + 1)$ using rotated nutriments $\varphi^{\psi}(\vartheta + 1)$ and geometry transformation $\varphi^{\xi}(\vartheta + 1)$.
- 8 compute passed line $y^l(\vartheta) = \alpha_{\vartheta}x_{\vartheta}^l + \beta_{\vartheta}$ to count the nutriment.
- 9 calculate global variance of VGG network σ_{ϑ} using variance in each feature map in the i th VGG convolution layers $\sigma_{\vartheta}^{(i)}$ and variance of each feature map $\sigma_{\vartheta(j)}^{(i)}$.
- 10 **end**

truth data where x^g , y^g , w^g , and h^g are center x , center y , width, and height of bounding box in ground truth data, respectively. λ_x and λ_y represent the absolute top-left corner location of the current grid cell. w and h are width and height referenced to size of image. Bounding box prediction B can be defined as:

$$\begin{aligned}
 \hat{x}^b &= \zeta(x^g) + \lambda_x, \\
 \hat{y}^b &= \zeta(y^g) + \lambda_y, \\
 \hat{w}^b &= e^{w^g} * w, \\
 \hat{h}^b &= e^{h^g} * h,
 \end{aligned} \tag{1}$$

where ζ is a model referred to [44].

2.2. Regression approach. Given $\varphi^t(f) = \{T_{f1}, T_{f2}, \dots, T_{fa}\}$ is tuna nutriment detection a in frame f . For each detection, it is represented by $T_{fa} = (\hat{x}_{fa}^t, \hat{y}_{fa}^t, \hat{w}_{fa}^t, \hat{h}_{fa}^t)$. Each bounding box of tuna nutriment detection shows center of bounding box \hat{x}_{fa}^t , center of bounding box \hat{y}_{fa}^t , width \hat{w}_{fa}^t , and height \hat{h}_{fa}^t . To give robust solution, center of tuna nutriment \hat{x}_{fa}^t and \hat{y}_{fa}^t is transformed into different domains to reduce rotation and unknown position of camera effect. Regression is used to predict tuna nutriment \hat{x}_{fa}^t and \hat{y}_{fa}^t in next frame $f + 1$. The results of regression apply the inverse transformation method to moving back into original center point of predicted nutriments $\hat{x}_{(f+1)a}$ and $\hat{y}_{(f+1)a}$.

2.2.1. *Transformation.* We are given a pair of ripple area detection $\varphi^r(f) = \{R_{f1}, R_{f2}\}$ in frame f . Each ripple detection is represented by $R_{f1,2} = (\hat{x}_f^r, \hat{y}_f^r, \hat{w}_f^r, \hat{h}_f^r)$. We compute each component of ripple detection to be top-left points $tl_{f1,2} = (\hat{x}_f^{tl}, \hat{y}_f^{tl})$, top-right points $tr_{f1,2} = (\hat{x}_f^{tr}, \hat{y}_f^{tr})$, bottom-left points $bl_{f1,2} = (\hat{x}_f^{bl}, \hat{y}_f^{bl})$ and bottom-right points $br_{f1,2} = (\hat{x}_f^{br}, \hat{y}_f^{br})$ of ripple detection by the following:

$$\begin{aligned} \hat{x}_f^{tl} &= \frac{2\hat{x}_f^r - \hat{w}_f^r}{2}, \\ \hat{y}_f^{tl} &= \frac{2\hat{y}_f^r - \hat{h}_f^r}{2}, \\ \hat{x}_f^{br} &= \frac{2\hat{x}_f^r + \hat{w}_f^r}{2}, \\ \hat{y}_f^{br} &= \frac{2\hat{y}_f^r + \hat{h}_f^r}{2}. \end{aligned} \tag{2}$$

After calculating the corner of ripple detection, normalizing coordinate of nutriment is an important aspect to give robust solution. The idea behind the coordinate transformation of nutriments is that the video camera always moves around and is also that there is no distance information between camera and ripple location. The goal of normalizing coordinate is to transform the function $\varphi^t(f)$ of nutriments into $\varphi^\kappa(f) = \{\kappa_{f1}, \kappa_{f2}, \dots, \kappa_{fa}\}$ as normalization of nutriments. We assume that the ripple location can be defined as the parameter to acknowledge moving of camera and changing the distance between camera and ripple area. Bottom-left point lb_{f1} of ripple R_{f1} is the new center point of normalizing coordinate with moving the position of $(0, 0)$ pixel in image shown in Figure 2. To compute angle θ_f of both ripples, we use center of R_{f1} and R_{f2} which is defined as follows:

$$\theta_f = \arctan_2((\hat{y}_{f2}^r - \hat{y}_{f1}^r), (\hat{x}_{f2}^r - \hat{x}_{f1}^r)). \tag{3}$$

Angle θ_f is used to rotate center of tuna nutriment detection $(\hat{x}_{fa}^t, \hat{y}_{fa}^t)$ by bottom-left points $(\hat{x}_{f1}^{tl}, \hat{y}_{f1}^{br})$ of ripple R_{f1} which is presented by

$$\begin{aligned} \hat{x}_{fa}^\psi &= (\hat{x}_{fa}^t - \hat{x}_f^{tl}) \cos(\theta_f) - (\hat{y}_{fa}^t - \hat{y}_f^{br}) \sin(\theta_f) + \hat{x}_f^{tl}, \\ \hat{y}_{fa}^\psi &= (\hat{x}_{fa}^t - \hat{x}_f^{tl}) \sin(\theta_f) + (\hat{y}_{fa}^t - \hat{y}_f^{br}) \cos(\theta_f) + \hat{y}_f^{br}, \end{aligned} \tag{4}$$

where $\psi_{fa} = (\hat{x}_{fa}^\psi, \hat{y}_{fa}^\psi)$ is the rotating result from a rotated function of nutriments $\varphi^\psi(f) = \{\psi_{f1}, \psi_{f2}, \dots, \psi_{fa}\}$. After normalizing using rotation, translation is applied to changing the center point from the position of $(0, 0)$ pixel in image to bottom-left points bl_{f1} of ripple R_{f1} which is shown in Figure 2 defined as follows:

$$\begin{aligned} \hat{x}_{fa}^\xi &= \hat{x}_{fa}^\psi - \hat{x}_{f1}^{tl}, \\ \hat{y}_{fa}^\xi &= \hat{y}_{fa}^\psi - \hat{y}_{f1}^{br}, \end{aligned} \tag{5}$$

where $\xi_{fa} = (\hat{x}_{fa}^\xi, \hat{y}_{fa}^\xi)$ is geometry transformation result after rotation from a geometry transformation function of nutriments $\varphi^\xi(f) = \{\xi_{f1}, \xi_{f2}, \dots, \xi_{fa}\}$. Distance z is computed by top-left point tl_{f1} of ripple R_{f1} and bottom-right point br_{f2} of ripple R_{f2} shown by

$$z = \sqrt{(\hat{x}_{f1}^{tl} - \hat{x}_{f2}^{br})^2 + (\hat{y}_{f1}^{tl} - \hat{y}_{f2}^{br})^2}. \tag{6}$$

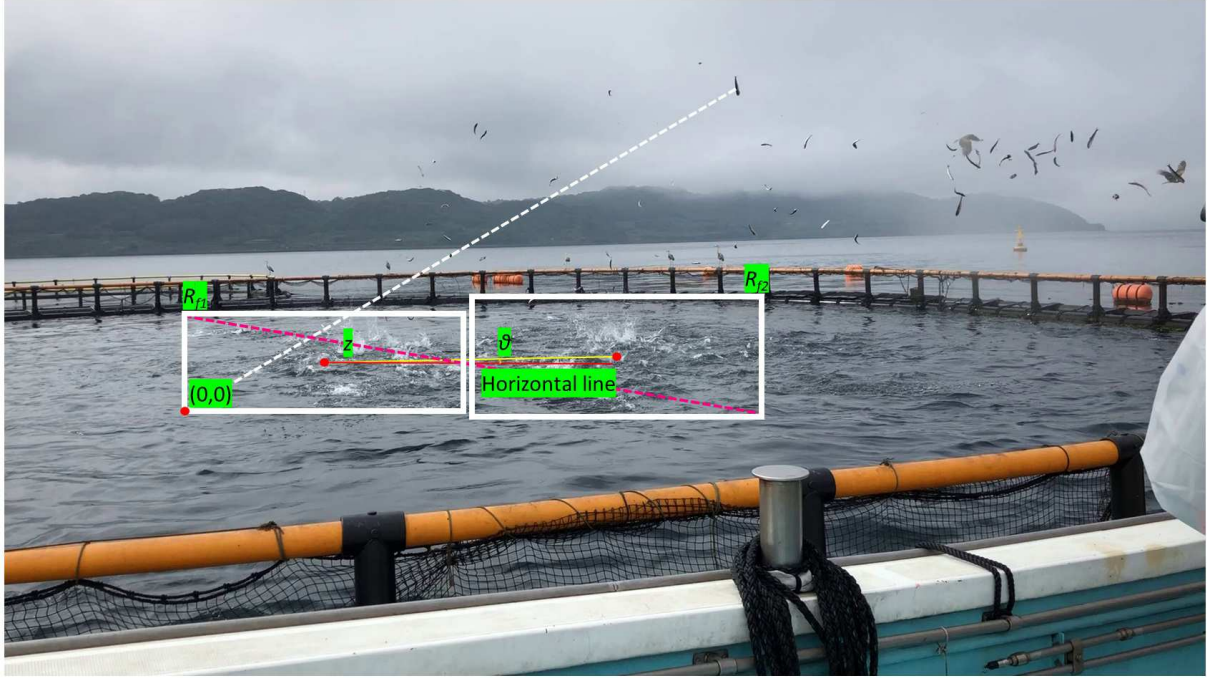


FIGURE 2. (color online) Visualization of transformation and inverse transformation approaches. White bounding box is represented as ripple detection. The center of coordinate position $(0, 0)$ after transformation is shown in bottom-left point lb_{f1} of ripple R_{f1} . The distance from the center of coordinate position $(0, 0)$ to nutrient is shown by white line. Angle θ is obtained from the difference between two center points of ripple and horizontal line. Pink line is computed by top-left points tl_{f1} of ripple R_{f1} and bottom-right points br_{f2} of ripple R_{f2} .

Normalization of nutriment $\kappa_{fa} = (\hat{x}_{fa}^\kappa, \hat{y}_{fa}^\kappa)$ is defined as fraction of geometry transformation ξ_{fa} with distance z which is explained by

$$\begin{aligned}\hat{x}_{fa}^\kappa &= \frac{\hat{x}_{fa}^\xi}{z}, \\ \hat{y}_{fa}^\kappa &= \frac{\hat{y}_{fa}^\xi}{z}.\end{aligned}\quad (7)$$

2.2.2. *Inverse transformation.* Inverse transformation is used to move back into original center points of nutriment \hat{x}_{fa} and \hat{y}_{fa} . This process is to calculate reverse transformation process from a function of normalized nutriment $\varphi^\kappa(f)$ to $\varphi^t(f)$. The inverse transformation is started from normalization of nutriment $\varphi^\kappa(f)$ to geometry transformation $\varphi^\xi(f)$ presented by

$$\begin{aligned}\hat{x}_{fa}^\xi &= \hat{x}_{fa}^\kappa * z, \\ \hat{y}_{fa}^\xi &= \hat{y}_{fa}^\kappa * z.\end{aligned}\quad (8)$$

After getting geometry transformation $\varphi^\xi(f)$, translation is applied to converting from a function of geometry transformation $\varphi^\xi(f)$ to rotated function $\varphi^\psi(f)$ which is represented by

$$\begin{aligned}\hat{x}_{fa}^\psi &= \hat{x}_{fa}^\xi + \hat{x}_{f1}^{tl}, \\ \hat{y}_{fa}^\psi &= \hat{y}_{f1}^{br} - \hat{y}_{fa}^\xi.\end{aligned}\quad (9)$$

Our goal is to transform a function of rotated nutriment $\varphi^\psi(f)$ into original pixel position in real images $\varphi^t(f)$ using (4) with negative of angle θ_f presented by

$$\theta_f = -\arctan_2 \left((\hat{y}_{f_2}^r - \hat{y}_{f_1}^r) \cdot (\hat{x}_{f_2}^r - \hat{x}_{f_1}^r) \right). \tag{10}$$

2.2.3. *Regression.* Small and dense nutriment can easily fail to be detected for several frames because current position and trajectory for each individual nutriment have different angles and speeds. To propose robust solution for undetected nutriments in several frames, regression can be a superior algorithm rather than tracking algorithm because performance of tracking depends on nutriment detection results in sequential frames. To improve regression result, the complexity of network by using multiple neuron and hidden layer is needed to precisely predict the moving nutriment with lower error direction. Model of regression μ^x and μ^y predicts the nutriments from current frame to next position of nutriment $(\hat{x}_{(f+1)a}^\kappa, \hat{y}_{(f+1)a}^\kappa)$ in the next frame which is defined as:

$$\begin{aligned} \hat{x}_{(f+1)a}^\kappa &= \mu^x(\hat{x}_{fa}^\kappa), \\ \hat{y}_{(f+1)a}^\kappa &= \mu^y(\hat{y}_{fa}^\kappa). \end{aligned} \tag{11}$$

The primary architecture of regression network is made in fully connected layers of 100, 80, 60, 40, 40, and 20 neurons with ReLU as activation function for individual \hat{x}_{fa}^κ and \hat{y}_{fa}^κ presented in Figure 3. MSE loss is used to predict position of nutriments in next frame. To predict $\hat{y}_{(f+1)a}^\kappa$, it requires two inputs \hat{x}_{fa}^κ and \hat{y}_{fa}^κ because in the object trajectory, y component has positive and negative values depending on turning point. To detect turning point, the position of component x is necessary to classify the movement of nutriments in correct direction.

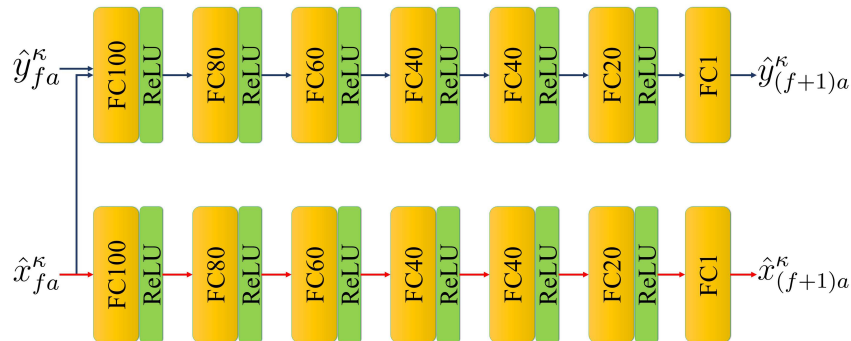


FIGURE 3. Network structure of regression model for predicting all nutriments detection in the next frame

2.2.4. *Counting nutriments.* Assuming bottom-left points lb_{f_1} as the center, it can be used to define passed line to count the nutriment. Given linear function $y^l(f) = \alpha_f x_f^l + \beta_f$ is created from two points $(\hat{x}_{f_1}^l, \hat{y}_{f_1}^l)$ and $(\hat{x}_{f_2}^l, \hat{y}_{f_2}^l)$. These points are produced from the corner and center of ripple which is defined as:

$$\begin{aligned} \hat{x}_{f_1}^l &= \hat{x}_{f_1}^{tl}, \\ \hat{y}_{f_1}^l &= \hat{y}_{f_1}^{br} - \frac{z}{\rho}, \\ \hat{x}_{f_2}^l &= \hat{x}_{f_1}^l + (\hat{x}_{f_2}^r - \hat{x}_{f_1}^r), \\ \hat{y}_{f_2}^l &= \hat{y}_{f_1}^l - (\hat{y}_{f_1}^r - \hat{y}_{f_2}^r), \end{aligned} \tag{12}$$

where $\rho = 3.6$ is a constant value. ρ is used to define the distance between passed line and ripple area. The gradient α_f and coefficient β_f of passed line $y^l(f) = \alpha_f x_f^l + \beta_f$ is calculated as:

$$\alpha_f = \frac{\hat{y}_{f2}^l - \hat{y}_{f1}^l}{\hat{x}_{f2}^l - \hat{x}_{f1}^l},$$

$$\beta_f = \hat{y}_{f1}^l - \alpha_f \hat{x}_{f1}^l.$$
(13)

2.3. Textural feature of ripple behavior. Ripple activity level is determined as human perception for the size and number of ripple appearing from the fish feeding process. Human perception cannot continuously be used to adjust fish feeding machine because its perception has no standard for validation or subjective methods. By following that, we propose robust method to estimate ripple behavior using VGG as base model for feature extraction which is presented in Figure 4. This idea is adopted from full reference image quality assessment (FR-IQA). FR-IQA is assessment of the quality of test images comparing with reference image for standard quality. The difference of FR-IQA and our approach is that FR-IQA requires reference and test image while our approach only needs single images to define ripple activity level.

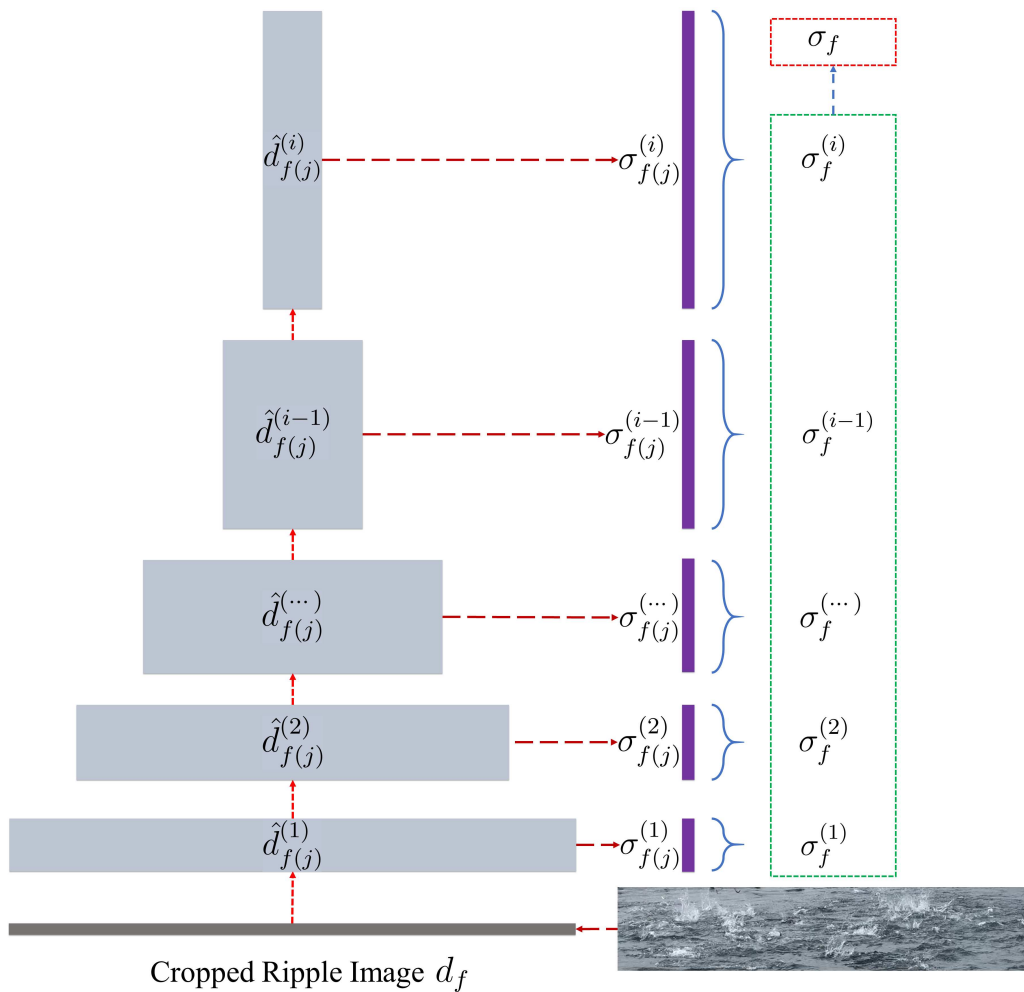


FIGURE 4. VGG architecture to calculate global variance σ_f for estimating level of ripple activity

2.3.1. *VGG feature.* Given cropped image d_f from point tl_{f1} to br_{f2} in original image as the input image of VGG network, VGG feature of convolutional layers is calculated by

$$f(d_f) = \left\{ \hat{d}_{f(j)}^{(i)}; i = 1, \dots, \omega; j = 1, \dots, \phi_i \right\}, \quad (14)$$

where $\omega = 5$ represents the number of VGG convolution layers. ϕ_i is the number of feature maps in the i th VGG convolution layers. Variance $\sigma_f^{(i)}$ of feature maps in the i th VGG convolution layers is presented as:

$$\sigma_f^{(i)} = \sqrt{\frac{\sum \left(\sigma_{f(j)}^{(i)} - \frac{\sum \sigma_{f(j)}^{(i)}}{\phi_i} \right)^2}{\phi_i}}, \quad (15)$$

where $\sigma_{f(j)}^{(i)}$ is variance of each feature map $\hat{d}_{f(j)}^{(i)}$ in VGG convolution layers. Global variance σ_f is computed by

$$\sigma_f = \sqrt{\frac{\sum \left(\sigma_f^{(i)} - \frac{\sum \sigma_f^{(i)}}{\omega} \right)^2}{\omega}}. \quad (16)$$

3. Experiment. We firstly explain the details of dataset used for this experiment. We then describe the evaluation approach for precisely presenting quantitative evaluation with various regression models.

3.1. Datasets. Our dataset contains a video of fish feeding process which has small and dense nutriment with 418 frames. Each dimension of video frame is 1920×1080 pixels. This sequential video is saved in MOV format with frame rate 30 frames/second. Range size of nutriment is starting from 9×6 to 13×36 pixels.

3.2. Evaluation approach. Evaluation approach is computed by measuring minimum Euclidean distance based on result for each regression model with ground truth μ_{gt} . Best regression model μ^* with minimum error distance to the ground truth μ_{gt} is defined as:

$$\mu^* = \arg \min_{\iota} \left(\left(\mu_{gt} - \sqrt{\left(\mu_{\iota}^x(\hat{x}_{fa}^{\kappa}) - \mu_{\iota}^y(\hat{y}_{fa}^{\kappa}) \right)^2} \right)^T \left(\mu_{gt} - \sqrt{\left(\mu_{\iota}^x(\hat{x}_{fa}^{\kappa}) - \mu_{\iota}^y(\hat{y}_{fa}^{\kappa}) \right)^2} \right) \right), \quad (17)$$

where $\iota \in [1, 6]$.

3.3. Quantitative evaluation. In this section, we aim to answer two main questions towards understanding our model. Firstly, we compare the performance of different regression models and observe performance of the proposed method. In Table 1, it shows the description of various regression models namely $R1$, $R2$, $R3$, $R4$, $R5$, and $R6$. We train each regression model with the same parameter with 10^6 iterations and 10^{-7} as learning rate. For conclusion, we present the result for each regression model in Table 2 and the visualization for each regression model is shown by Figure 5. We manually mark the correct position of nutriment in the next frame with $N = 50$. By following quantitative evaluation, regression model $R1$ has the lowest minimum error distance and we apply this model to being a base of regression model.

Secondly, we observe estimation of ripple behavior and use different images to represent performance of global variance of VGG network which is shown in Figure 6. For fair comparison, we use the image with the same dimension in which the size is 843×162 pixels. In conclusion, image without ripple in Figure 6(a) achieves 0.029 while image with

TABLE 1. Different regression models for evaluation process. Each model has different fully connected (fc) layers and different numbers of neurons.

Methods	fc1	fc2	fc3	fc4	fc5	fc6	fc7	fc8	fc9	fc10
<i>R1</i>	100	80	60	40	40	20				
<i>R2</i>	200	160	120	80	80	40				
<i>R3</i>	100	90	80	70	60	50	40	30	20	10
<i>R4</i>	200	180	160	140	120	100	80	60	40	20
<i>R5</i>	100	40								
<i>R6</i>	200	80								

TABLE 2. Statistical analysis of various regression models *R1*, *R2*, *R3*, *R4*, *R5*, and *R6* for quantitative evaluation using one-sample T-Test

Methods	<i>N</i>	Mean (pixels)	Std. Dev. (pixels)	Std. Err. (pixels)	95% Confidence Interval of the Diff.	
					Lower (pixels)	Upper (pixels)
<i>R1</i>	50	10.41	5.79	0.82	8.76	12.05
<i>R2</i>	50	10.84	5.77	0.82	9.20	12.48
<i>R3</i>	50	10.94	5.74	0.81	9.31	12.58
<i>R4</i>	50	10.89	5.81	0.82	9.24	12.54
<i>R5</i>	50	10.90	5.88	0.83	9.23	12.57
<i>R6</i>	50	10.99	5.82	0.82	9.34	12.64

ripple in Figure 6(b) reaches 0.3037. The value can represent the activity level for ripple area which means that ripple image in Figure 6(a) has lower activity than ripple image in Figure 6(b).

4. **Result.** Before we start to explain result of the proposed method, we firstly show evaluation result for better understanding the proposed method. We then compare the proposed method and state-of-the-art benchmark methods of tracking algorithm on our datasets and show the computational time to figure out the performance and advantage of the proposed method compared with state-of-the-art benchmark methods.

4.1. **Evaluation result.** To acknowledge the performance of YOLOv4 [44], mAP and loss are used as the training parameters. To train object detection using our dataset, we present the detail of parameter using 10k iterations with 512×288 pixels for image resizing from 1920×1080 pixels. Figure 7 displays training result and reaches 75% of maximum mAP and 0.9 as loss value after two days training using GeForce RTX 2080 Ti. Figure 8 shows the result of the proposed method in sequential frames of our video. By following these graphs, number of nutriments and estimation of ripple behaviors can be clearly used to identify the behavior of fish feeding activity. The number of nutriments is calculated by cumulative summation of nutriments in 20 frames while estimation of ripple behavior is computed by cumulative average of global variance in 20 frames. We also tested performance regression model of the proposed method with state-of-the-art tracking methods. Many of state-of-the-art tracking methods use multiple object tracking (MOT) [32-38]. These methods perform well in six public databases on pedestrian detection [46-48]. For evaluations, JDE method [32] has been chosen to represent MOT as benchmark method because its method has an accurate prediction based on re-implementation of faster object

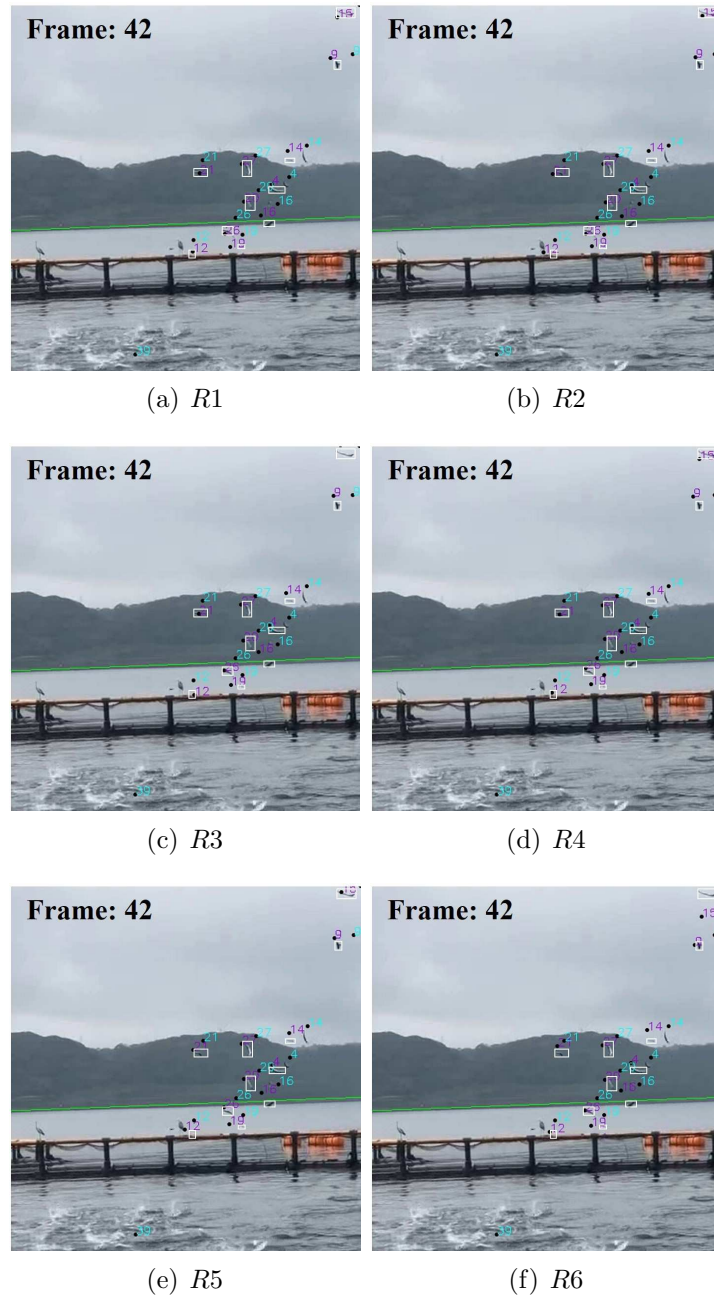


FIGURE 5. (color online) Visualization of various regression models $R1$, $R2$, $R3$, $R4$, $R5$, and $R6$. White box in each image represents ground truth of nutriment. Other two different colors show \hat{x}_{fa}^{κ} as blue light color and $\hat{x}_{(f+1)a}^{\kappa}$ as purple. We can see that white box and purple points in $R1$ more precisely track result compared with other regression models.

detection compared with [33-38]. We also use trajectory mapping [39] as a benchmark method because its method has a good performance in application to aquaculture fish tank. The last method to be a benchmark method is SORT [49]. We add the detection model of trajectory mapping to completely understand performance of tracking method.

In Figure 9(a), the proposed method is demonstrated to be able to track small nutriment well while trajectory mapping (TM), JDE and SORT with original YOLOv3 and trajectory mapping detection model perform poor (Figures 9(b), 9(c), 9(d), 9(e)) without



(a) Fish tank area without ripple behavior



(b) Fish tank area with ripple behavior

FIGURE 6. In this observation, we estimate ripple behavior using image with and without ripple area to compromise global variance of VGG network.

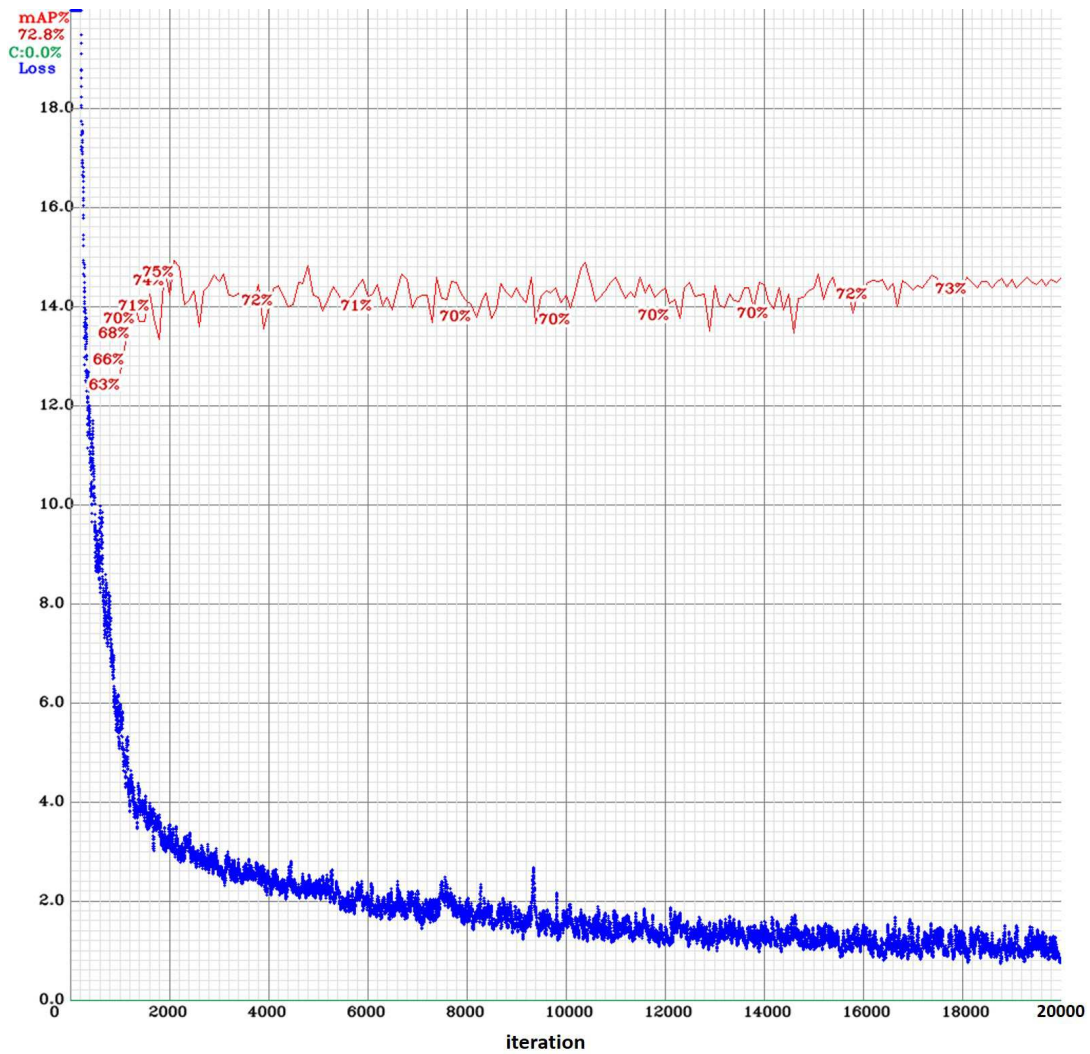
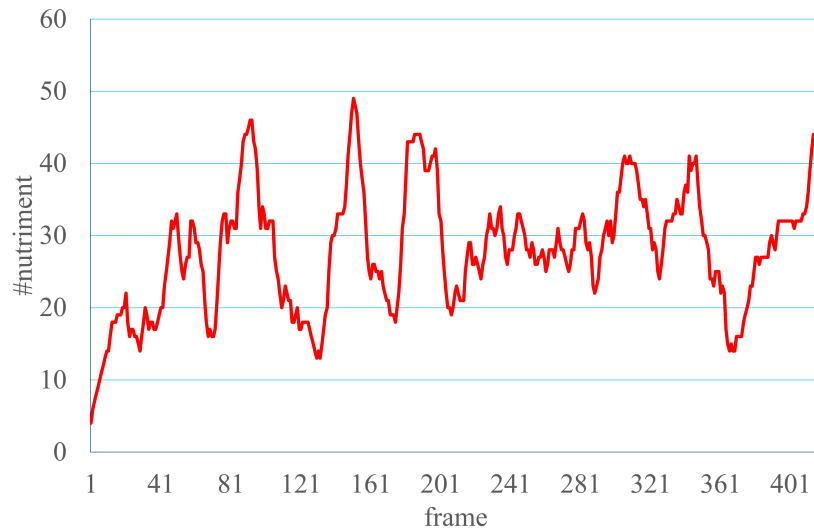


FIGURE 7. mAP and loss for all iterations during training model using YOLOv4 [44]. After 2k iterations, mAP curve of training model creates more stable result and loss value has slowly decreased for every iteration.



(a) Number of nutriments



(b) Estimation of ripple behaviors

FIGURE 8. Result of regression and estimate of ripple behavior. Graph (a) shows number of nutriment in each frame while graph (b) represents ripple behavior. Based on these graphs, the process of feeding activity is available.

tracking results of nutriment even trajectory mapping and SORT are able to detect some nutriment and ripple area. In this cases, we assume that the benchmark methods fail to run our datasets because the size of nutriment is too small (maximum size is 13×36 pixels) and the speed of nutriment is fast (average nutriment movement from fish feeding machine to ripple area is 23.8 frames).

4.2. Computational time. To describe the computational complexity and execution time of the proposed methodology, a computational time analysis is conducted using a video with 418 frames. The specification of hardware and software used to analyze the proposed method and state-of-the-art benchmark methods is shown in Table 3. Table 4 presents comparison of computational time (in fps) between the proposed method and benchmark methods (trajectory mapping, JDE and SORT with original YOLOv3 and trajectory mapping detection model). We reach 3.86 and 0.11 fps for average and standard

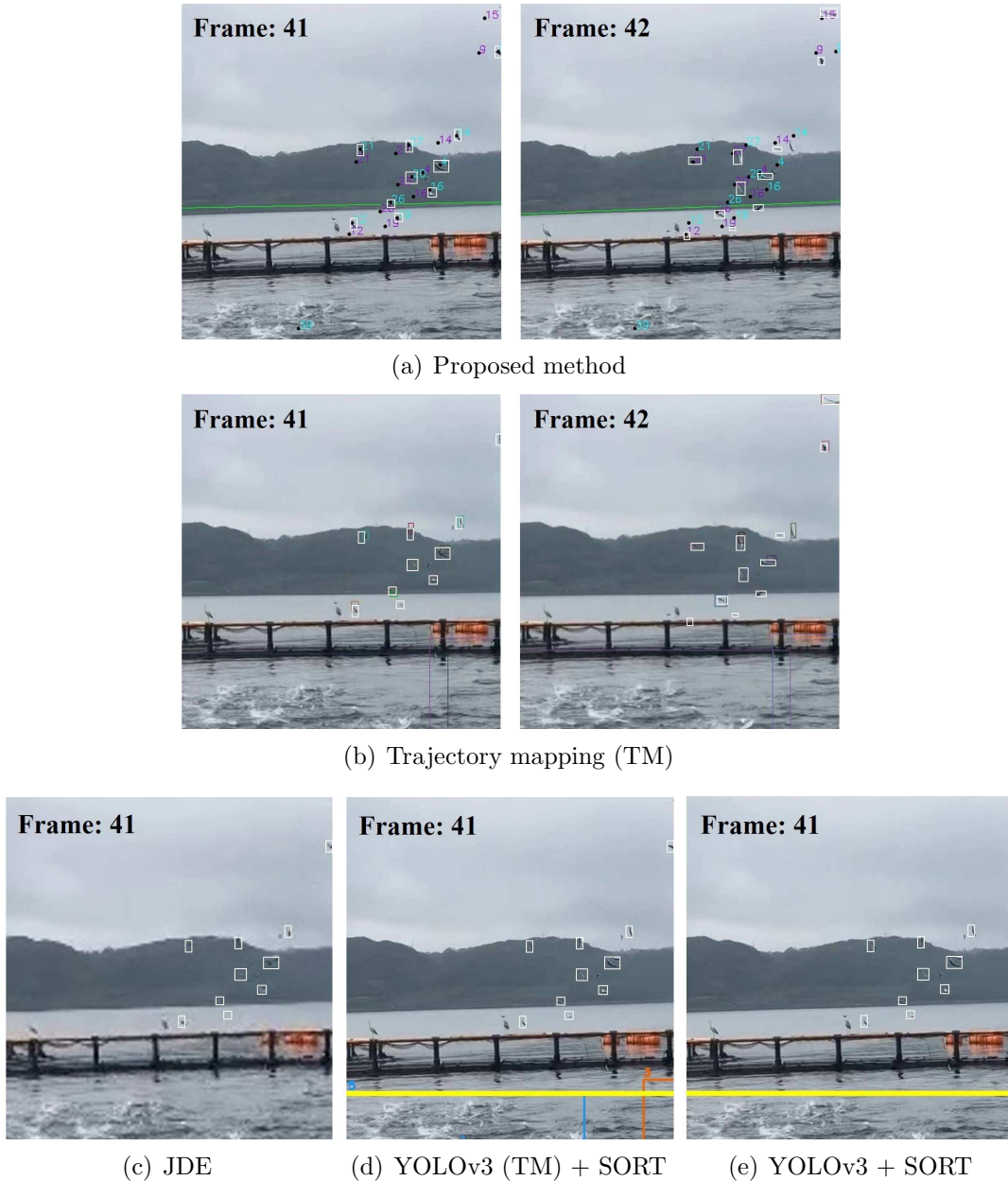


FIGURE 9. (color online) Observation of the proposed method shown in (a) and the results of benchmark methods shown in (b), (c), (d) and (e). White box in each image represents ground truth of nutriment. In the proposed method, there are two different colors to represent \hat{x}_{fa}^k as blue light color and $\hat{x}_{(f+1)a}^k$ as purple. We can see that white box and purple points in the proposed method show precisely tracked result and it proves that the proposed method predicts very well while benchmark methods perform poor without tracking results of nutriments even trajectory mapping and SORT are able to detect some nutriments.

deviation of computational time, respectively, while trajectory mapping needs 1.93 and 0.61 fps and JDE spends 1.87 and 0.07 fps in which these benchmark methods run twice slower than the proposed method. SORT provides average computational time without information of computational time for individual frame. Computational time for both detection model of YOLOv3 and trajectory mapping model with SORT performs worst

TABLE 3. Hardware and software environment for running the proposed method and state-of-the-art benchmark methods for comparison

Specification		
Hardware	CPU	Intel Core i7-9700 CPU @3.00GHz (8 CPUs)
	RAM	16 GB
	GPU	NVIDIA GeForce GTX 745
Software	OS	Windows 10 Pro 64-bit
	IDE	Microsoft Visual Studio Professional 2017 v.15.9.25
	Language	Python 3.6.6 64bit

TABLE 4. Comparison of computational time between the proposed method and benchmark methods

Methods	N	Mean (fps)	Std. Dev. (fps)	Std. Err. (fps)	95% Confidence Interval of the Diff.	
					Lower (fps)	Upper (fps)
Ours	418	3.86	0.11	0.01	3.85	3.87
Trajectory Mapping (TM)	418	1.93	0.61	0.03	1.87	1.99
JDE	418	1.87	0.07	0.00	1.86	1.87
YOLOv3 (TM) + SORT	418	0.47	—	—	—	—
YOLOv3 + SORT	418	0.45	—	—	—	—

compared with the proposed method and other benchmark methods in which it runs 0.45 and 0.47 fps, respectively. By analyzing computational complexity, the proposed method is the fastest model with twice for the different speeds compared with benchmark methods.

5. Conclusion and Discussion. Automatic controlling fish feeding machine using combination of two feature extractions is important to optimize the amount of nutriment giving to the fish. Feature extraction of nutriment and ripple behavior has presented an agreement to complement each other. Recent studies have shown that it is possible to track all detected objects in sequence frames on video. However, there is no agreement to track multiple small and dense nutriments and also to detect ripple activity area in the video. In this paper, automatic controlling fish feeding machine using feature extraction of nutriment and ripple behavior has been presented and demonstrated to be promising for optimizing fish feeding process to give an optimal rate of both costs and profits. We have demonstrated an area with and without ripple activity which has a big difference gap and the proposed method consistently performs well on the video containing small and dense nutriments. We expect the proposed method to open the door for future work and to go beyond for developing a large community-generated database and focus on integrating with the sensors to give more accurate and robust results.

REFERENCES

- [1] R. Subasinghe, D. Soto and J. Jia, Global aquaculture and its role in sustainable development, *Rev. Aquac.*, vol.1, no.1, pp.2-9, 2009.
- [2] I. Aliyu, K. J. Gana, A. A. Musa, J. Agajo, A. M. Orire, F. T. Abiodun and M. A. Adegboye, A proposed fish counting algorithm using digital image processing technique, *ATBU J. Sci., Technol. Educ.*, vol.5, no.1, pp.1-11, 2017.
- [3] J. Guillen, F. Natale, N. Carvalho et al., Global seafood consumption footprint, *Ambio*, vol.48, pp.111-122, 2019.

- [4] FAO, *Global Aquaculture Production Volume and Value Statistics Database Updated to 2018*, Tech. Rep., FAO Fisheries and Aquaculture Dept., 2020.
- [5] Y. Atoum et al., Automatic feeding control for dense aquaculture fish tanks, *IEEE Signal Process. Lett.*, vol.22, pp.1089-1093, 2015.
- [6] A. K. Sabari et al., Smart fish feeder, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol.2, pp.111-115, 2017.
- [7] P. C. Oostlander et al., Microalgae production cost in aquaculture hatcheries, *Aquaculture*, vol.525, DOI: 10.1016/j.aquaculture.2020.735310, 2020.
- [8] J. E. Andrew, C. Noble, S. Kadri, H. Jewell and F. A. Huntingford, The effect of demand feeding on swimming speed and feeding responses in Atlantic salmon *Salmo salar* L., gilthead sea bream *Sparus aurata* L. and European sea bass *Dicentrarchus labrax* L. in sea cages, *Aquac. Res.*, vol.33, pp.501-507, 2002.
- [9] T. Asaeda, T. K. Vu and J. Manatunge, Effects of flow velocity on feeding behavior and microhabitat selection of the stone Moroko Pseudorasbora parva: A trade-off between feeding and swimming costs, *Trans. Am. Fish. Soc.*, vol.134, pp.537-547, 2005.
- [10] C. Rillahan, M. Chambers, W. H. Howell and W. H. Watson, A self-contained system for observing and quantifying the behavior of Atlantic cod, *Gadus morhua*, in an offshore aquaculture cage, *Aquaculture*, vol.293, pp.49-56, 2009.
- [11] J. F. López-Olmeda, C. Noble and F. J. Sánchez-Vázquez, Does feeding time affect fish welfare?, *Fish Physiol. Biochem.*, vol.38, pp.143-152, 2012.
- [12] H. Tanoue, T. Komatsu, T. Tsujino, I. Suzuki, M. Watanabe, H. Goto and N. Miyazaki, Feeding events of Japanese lates *Lates japonicus* detected by a high-speed video camera and three-axis micro-acceleration data-logger, *Fish. Sci.*, vol.78, pp.533-538, 2012.
- [13] T. Noda, Y. Kawabata, N. Arai, H. Mitamura and S. Watanabe, Animal-mounted gyroscope/accelerometer/magnetometer: In situ measurement of the movement performance of fast-start behavior in fish, *J. Exp. Mar. Biol. Ecol.*, vol.451, pp.55-68, 2014.
- [14] J. Horie, H. Mitamura, Y. Ina, Y. Mashino, T. Noda, K. Moriya, N. Arai and T. Sasakura, Development of a method for classifying and transmitting high-resolution feeding behavior of fish using an acceleration pinger, *Anim. Biotelem.*, vol.5, 2017.
- [15] A. W. Stoner, M. L. Ottmar and T. P. Hurst, Temperature affects activity and feeding motivation in Pacific halibut: Implications for bait-dependent fishing, *Fish. Res.*, vol.81, pp.202-209, 2006.
- [16] G. M. Soto-Zarazúa, E. Rico-García, R. Ocampo, R. G. Guevara-González and G. Herrera-Ruiz, Fuzzy-logic-based feeder system for intensive tilapia production (*Oreochromis niloticus*), *Aquac. Int.*, vol.18, pp.379-391, 2010.
- [17] T. Wu, Y. Huang and J. Chen, Development of an adaptive neural-based fuzzy inference system for feeding decision-making assessment in silver perch (*Bidyanus bidyanus*) culture, *Aquac. Eng.*, vol.66, pp.41-51, 2015.
- [18] S. Zhao, W. Ding, S. Zhao and J. Gu, Adaptive neural fuzzy inference system for feeding decision-making of grass carp (*Ctenopharyngodon idellus*) in outdoor intensive culturing ponds, *Aquaculture*, vol.498, pp.28-36, 2019.
- [19] C. Zhou, K. Lin, D. Xu, L. Chen, Q. Guo, C. Sun and X. Yang, Near infrared computer vision and neuro-fuzzy model-based feeding decision system for fish in aquaculture, *Comput. Electron. Agric.*, vol.146, pp.114-124, 2018.
- [20] C. Zhou, D. Xu, K. Lin, C. Sun and X. Yang, Intelligent feeding control methods in aquaculture with an emphasis on fish: A review, *Rev. Aquac.*, vol.10, no.4, pp.975-993, 2018.
- [21] A. Pautsina, P. Čisáň, D. Štys, B. F. Terjesen and A. M. O. Espmark, Infrared reflection system for indoor 3D tracking of fish, *Aquac. Eng.*, vol.69, pp.7-17, 2015.
- [22] C. Hung, S. Tsao, K. Huang, J. Jang, H. Chang and F. C. Dobbs, A highly sensitive underwater video system for use in turbid aquaculture ponds, *Sci. Rep.*, vol.6, 2016.
- [23] C. Zhou, B. Zhang, K. Lin, D. Xu, C. Chen, X. Yang and C. Sun, Near-infrared imaging to quantify the feeding behavior of fish in aquaculture, *Comput. Electron. Agric.*, vol.135, pp.233-241, 2017.
- [24] Z. Liu, X. Li, L. Fan, H. Lu, L. Liu and Y. Liu, Measuring feeding activity of fish in RAS using computer vision, *Aquac. Eng.*, vol.60, pp.20-27, 2014.
- [25] Q. Guo, X. Yang, C. Zhou, K. Li, C. Sun and M. Chen, Fish feeding behavior detection method based on shape and texture features, *J. Shanghai Ocean Univ.*, vol.27, pp.181-189, 2018.
- [26] C. Zhou, D. Xu, L. Chen, S. Zhang, C. Sun, X. Yang and Y. Wang, Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision, *Aquaculture*, vol.507, pp.457-465, 2019.

- [27] B. Zion, The use of computer vision technologies in aquaculture – A review, *Comput. Electron. Agric.*, vol.88, pp.125-132, 2012.
- [28] S. G. Hassan, M. Hasan and D. Li, Information fusion in aquaculture: A state-of-the-art review, *Front. Agr. Sci. Eng.*, vol.3, p.206, 2016.
- [29] M. Saberioon, A. Gholizadeh, P. Cisar, A. Pautsina and J. Urban, Application of machine vision systems in aquaculture with emphasis on fish: State-of-the-art and key issues, *Rev. Aquac.*, vol.9, pp.369-387, 2017.
- [30] A. Soetedjo and I K. Somawirata, Improving traffic sign detection by combining MSER and Lucas Kanade Tracking, *International Journal of Innovative Computing, Information and Control*, vol.15, no.2, pp.653-665, 2019.
- [31] Y. Hashimoto, H. Hama and T. T. Zin, Robust tracking of cattle using super pixels and local graph cut for monitoring systems, *International Journal of Innovative Computing, Information and Control*, vol.16, no.4, pp.1469-1475, 2020.
- [32] Z. Wang et al., Towards real-time multi-object tracking, *CoRR*, vol.abs/1909.12605, 2019.
- [33] M. Babaei et al., A dual CNN-RNN for multiple people tracking, *Neurocomputing*, vol.368, pp.69-83, 2019.
- [34] F. Yu et al., POI: Multiple object tracking with high performance detection and appearance feature, in *Computer Vision – ECCV 2016 Workshops. ECCV 2016. Lecture Notes in Computer Science*, G. Hua and H. Jégou (eds.), Cham, Springer, 2016.
- [35] W. Lin et al., *Real-Time Multi-Object Tracking with Hyper-Plane Matching*, Technical Report, 2017.
- [36] S. Tang et al., Multiple people tracking by lifted multicut and person re-identification, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, pp.3701-3710, 2017.
- [37] Y. Zhang et al., A simple baseline for multi-object tracking, *CoRR*, vol.abs/2004.01888, 2020.
- [38] B. Pang et al., TubeTK: Adopting tubes to track multi-object in a one-step training model, *CoRR*, vol.abs/2006.05683, 2020.
- [39] H. Pradana and K. Horio, Tuna nutriment tracking using trajectory mapping in application to aquaculture fish tank, *2020 Digital Image Computing: Techniques and Applications*, pp.1-8, 2020.
- [40] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.39, no.6, pp.1137-1149, 2015.
- [41] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan and W. Hu, Distractor-aware siamese networks for visual object tracking, *ECCV*, pp.101-117, 2018.
- [42] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing and J. Yan, SiamRPN++: Evolution of siamese visual tracking with very deep networks, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4277-4286, 2019.
- [43] K. Simoyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv.org*, arXiv: 1409.1556, 2015.
- [44] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, YOLOv4: Optimal speed and accuracy of object detection, *CoRR*, vol.abs/2004.10934, 2020.
- [45] J. Redmon and A. Farhadi, YOLOv3: An incremental improvement, *CoRR*, vol.abs/1804.02767, 2018.
- [46] A. Ess et al., Depth and appearance for mobile scene analysis, *ICCV*, pp.1-8, 2007.
- [47] L. Leal-Taixe et al., MOTChallenge 2015: Towards a benchmark for multi-target tracking, *CoRR*, vol.abs/1504.01942, 2015.
- [48] A. Milan et al., MOT16: A benchmark for multi-object tracking, *CoRR*, vol.abs/1603.00831, 2016.
- [49] Bewley et al., Simple online and realtime tracking, *ICIP*, pp.3464-3468, 2016.