Aquaculture 557 (2022) 738334

Contents lists available at ScienceDirect

Aquaculture

journal homepage: www.elsevier.com/locate/aquaculture

Using a convolutional neural network for fingerling counting: A multi-task learning approach

Diogo Nunes Gonçalves[a,b], Plabiany Rodrigo Acosta[a], Ana Paula Marques Ramos[c,d,*], Lucas Prado Osco[e], Danielle Elis Garcia Furuya[c], Michelle Taís Garcia Furuya[c], Jonathan Li[g], José Marcato Junior[f], Hemerson Pistori[a,b], Wesley Nunes Gonçalves[a,f]

[a] *Faculty of Computer Science, Federal University of Mato Grosso do Sul, Av. Costa e Silva, Campo Grande 79070-900, MS, Brazil*
[b] *INOVISAO, Dom Bosco Catholic University, Avenida Tamandaré, 6000, Campo Grande 79117-900, MS, Brazil*
[c] *Program of Environment and Regional Developement, University of Western São Paulo, Raposo Tavares, km 572, Presidente Prudente 19067-175, SP, Brazil*
[d] *Program of Agronomy, University of Western São Paulo, Raposo Tavares, km 572, Presidente Prudente 19067-175, SP, Brazil*
[e] *Faculty of Engineering and Architecture and Urbanism, University of Western São Paulo, Raposo Tavares, km 572, Presidente Prudente 19067-175, SP, Brazil*
[f] *Faculty of Engineering, Architecture, and Urbanism and Geography, Federal University of Mato Grosso do Sul, Av. Costa e Silva, Campo Grande 79070-900, MS, Brazil*
[g] *Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada*

ARTICLE INFO

Keywords:
Fingerlings count
Multi-task learning
Convolutional neural network

ABSTRACT

Fingerling counting is an important task for decision-making in the aquaculture context. The counting is usually performed by a human, which is time-consuming and prone to errors. Artificial intelligence methods applied to image interpretation can be a great strategy for solving this task automatically. However, applying machine learning to attend to aquaculture issues is an underexplored field that requires novel investigations, especially of methods that explore temporal information in videos. In this study, we propose a new method to locate and count fingerlings in a sequence of images using convolutional neural networks. The proposed method estimates three tasks in a multi-task approach. The first task consists of predicting the probability of a fingerling occurring in each pixel of the frame, while the second and third tasks estimate the movement performed by the fingerlings. Motion prediction is used as a complement to fingerling detection, including relevant information especially when two or more fingerlings are in contact. Experimental results indicated that the use of temporal information considerably increases the results, reaching F1 of 97.89. The proposed method was evaluated in frames with different numbers of fingerlings (from 0 to 10) and all obtained relevant results, with an F1 of 95.42 or higher. The study also showed that, in most cases, the proposed method can detect the contact of two or more fingerlings, which is considered the main challenge of the detection and counting of fingerlings.

1. Introduction

Fingerling counting is the task of estimating the number of animals in a given area for decision-making. This data is important to calculate the production potential, the necessary amount of feed, and the sale of a specific amount of animals. Counting is usually performed visually by a human, although it is time-consuming and error-prone. Currently, a translocation between tanks is performed by estimating the weight of animals in a sieve (e.g., one kilo is equivalent to N fingerlings on average) whereas, in sales, excessive time is required to count the exact number of animals (Zhang et al., 2020a).

To reduce errors and speed up the process, automatic systems using images have been proposed (França Albuquerque et al., 2019; Garcia et al., 2020). These systems collect images and the counting is performed by analyzing the images, making counting faster and less costly. Counting occurs in two ways: i) detecting each fingerling (detection-based methods) or ii) regressing a number that correlates the entire image or parts of it with the number of fingerlings (regression-based methods). In addition, these methods can include temporal information from a sequence of images.

Detection-based methods locate each fingerling in the image. Garcia et al. (2020) and França Albuquerque et al. (2019) presented a system

[*] Corresponding author at: Environment and Regional Development programs, and Agronomy, University of Western São Paulo, Raposo Tavares, km 572, Presidente Prudente 19067-175, SP, Brazil.

E-mail address: anaramos@unoeste.br (A.P.M. Ramos).

https://doi.org/10.1016/j.aquaculture.2022.738334
Received 11 June 2021; Received in revised form 19 April 2022; Accepted 7 May 2022
Available online 13 May 2022

for counting fingerlings using background subtraction, blob detection and Kalman filter. Although relevant results have been achieved, the system is susceptible to a large amount of parameterization (e.g., average fingerling size, average distance, etc.), which makes it difficult to use on a large scale. Some studies already used neural networks in aquaculture cases (Sveen et al., 2021; Zhao et al., 2018; Zhou et al., 2019). Recently, counting methods have been proposed using convolutional neural networks (CNNs), such as R-CNN used by Salman et al. (2019), Faster R-CNN Ren et al. (2015), and FCOS Tian et al. (2019). These methods consider a bounding box for each object and can provide both the position (center of the bounding box) and the count (number of bounding boxes). To analyze the effectiveness of the Convolutional Neural Network (CNN) in detecting and counting fingerlings, Lainez and Gonzales (2019) tested the method based on image processing on four sizes of tilapia fingerlings, achieving an average accuracy greater than 0.99.

Fishes species recognition can also be useful in counting tasks (Dos Santos and Gonçalves, 2019). Li et al. (2015) apply Fast R-CNN for fish detection and recognition in complex underwater environments and Villon et al. (2018) compare the performance of a CNN with the human ability to identify fish species. Despite achieving satisfactory results, such works do not focus on fish counting.

On the other hand, regression-based methods directly estimate the number of fingerlings establishing a correlation between the features extracted from the image and the target number. Zhang et al. (2020a) proposed a method that divides the image into sub-images containing one or more fish using segmentation. For each subimage, regression is applied to estimate the number of local fish and contributes to the total image count. Fan and Liu (2013) proposed a method that estimates the number of fingerlings based on geometric features (e.g., area, perimeter). The features are inputs for the least squares support vector machine (LS-SVM) that performs the regression. CNNs have also been used for regression and counting fish. Zhang et al. (2020b) proposed a hybrid neural network model to estimate a density map and the total number of fish in the image. The hybrid model based on a multi-column CNN and a dilated CNN obtained an accuracy above 0.95 and a Pearson correlation coefficient, referring to the ground truth and the estimation, of 0.99.

Although the regression-based methods have good results, they are not able to estimate the position of each fingerling in the image. In addition, the fingerlings are distributed inhomogeneously in the images containing, in general, more examples with low density. Thus, these methods present a long tail distribution of counts, providing underestimations in high density regions and overestimations in low density regions (Liu et al., 2020).

Although recent methods have obtained promising results, the high density with overlapping fingerlings is a challenge for counting. In general, object detection methods are not suitable for dense object scenarios (Goldman et al., 2019). In this case, the overlapping of the bounding boxes due to occlusion makes detection and counting difficult. To assist in counting in occlusion scenarios, counting in a sequence of images can be important. Analysis of the movement that objects perform in frames can provide valuable information that is not always taken into account when counting objects in an image. In this context, studies show that the movement of objects can assist in the detection and counting, as well as distinguishing them from the background (Ma et al., 2015; Nam and Han, 2016; Danelljan et al., 2015; Wang et al., 2019; Hou et al., 2019; Gonçalves et al., 2020).

In this regard, we proposed a detection-based method for analyzing fingerlings in a video stream. Locating or detecting fingerlings consists of identifying the positions (x, y) in the frame. In this way, the count can be obtained by the number of fingerlings detected. Our approach fits into the detection-based category as it performs the fingerling detection through the confidence map, different from regression-based methods that estimate the quantity directly from the image. It refers to an original approach for locating and counting fingerlings in a video stream using convolutional neural networks. We hypothesize that there is an improvement in the detection when the past frame information is used to count and locate fingerlings in a current frame. For example, knowing the movement of fingerlings from the previous frame can benefit detection of the current frame. Up to the writing moment, this refers to the first attempt to combine the movement of the fingerlings from one frame to the other for estimating a movement direction vector which improves robustness in fingerling detection. Our dataset is composed of videos with up to 10 fingerlings and with contact between them, which makes detection difficult. Despite this, the results showed the promising results with an F1 of 97.89.

2. Materials and methods

2.1. Image capture and dataset

The videos used in this work was collected by (França Albuquerque et al., 2019; Garcia et al., 2020). To capture the images, a closed structure with a ramp inclined at approximately 12 degrees was used. The inclined ramp helps the fingerlings slide with water, which flows continuously. On top of the structure, a Logitech C920 camera was placed to capture images at 30 frames per second with a resolution of 640 × 480 pixels. To improve image quality, a light source was placed for indoor lighting.

To build the dataset, 20 videos were captured in a company located in Terenos, Mato Grosso do Sul, Brazil. The fingerlings used in the dataset are of the Pintado real species due to their importance in production. In the experiments, the frames were scaled to 512 × 512 pixels. Table 1 shows the number of frames and the total number of fingerlings for each of the training, validation, and test sets. The number of fingerlings per frame is shown in Table 2. Most frames have up to two fingerlings, although challenging scenarios with up to 10 fingerlings are present in the dataset. We also counted the number of times two or more fingerlings were in contact. In the test frames, there were 111 adhesions between fingerlings, which poses a greater challenge in detection. Adhesion or contact between fingerlings usually occurs in frames with 6–10 fingerlings, and of the 69 frames, 41 of them have contact with one or more fingerlings.

Each frame was manually annotated with the center of mass of each fingerling. Given an image, an expert annotated a point at the approximate center of mass. In addition, the center of mass of each fingerling in the previous frame is available to assist in the inclusion of temporal information, as used by the proposed method.

2.2. Proposed approach

This section describes a method for detecting and counting fingerlings in a video. Fig. 1 presents an overview of our method. Initially, two frames are concatenated (Fig. 1(a)) and a feature map is extracted using a CNN (Fig. 1(b)). This feature map is given as an input to the multi-task learning (Fig. 1(c)) that estimates i) the probability of a pixel being part of a fingerling, ii) the probability of the pixels belonging to the movement of a fingerling from a previous frame to the current one, and iii) a movement direction vector for each pixel. From the estimation of the first task, the proposed method detects the fingerlings forming a complete bipartite graph (Fig. 1(d)). This graph is composed of two groups of vertices representing the fingerlings in the previous frame (circled in blue) and the current frame (circled in orange). Each vertex of a group is

Table 1
Description of the dataset in relation to the number of frames and fingerlings.

Set	N. of frames	N. of fingerlings
Train	2730	4079
Validation	210	461
Test	1080	2102
Total	4020	6642

Table 2
Number of fingerlings per frame.

N. of Fingerlings per frame	N. of frames		
	Train	Val	Test
0–2	2192	158	749
3–5	466	29	262
6–10	72	23	69

connected to all vertices of the other group by edges. Edge weights are calculated using the movement information estimated in the multi-task learning. Finally, the fingerlings from the previous frame are matched with those detected in the current frame, discarding those without a match. In addition, fingerlings from the current frame without a match but with a high probability of occurrence are kept, as they are probably new fingerlings entering the scene. The subsections below describe each step in detail.

2.2.1. *Feature map extraction*

Given two consecutive RGB frames I_t and I_{t-1} with $w \times h \times 3$, they are concatenated to form an input $I = [I_{t-1}, I_t]$ with dimension $w \times h \times 6$. In this way, the previous frame I_{t-1} can add relevant information for the detection and counting of fingerlings in frame I_t. Input I is passed through a CNN to extract a feature map. The backbone is composed of convolutional layers similar to the VGG16 architecture (Simonyan and Zisserman, 2015) (see Fig. 2).

2.2.2. *Multi-task learning*

For fingerling count, three tasks are learned to take advantage of the temporal information in a video. The first task is to estimate a confidence map for the position of the fingerlings in frame t only (see Fig. 3 (c)). Thus, the first task predicts the probability of any individual image pixel being part of a fingerling. The ideal scenario would be to use fingerling segmentation, however, only the centroid of each fingerling was labeled and made available in the dataset. Therefore, we use a confidence map in the first task instead of predicting the centroids directly.

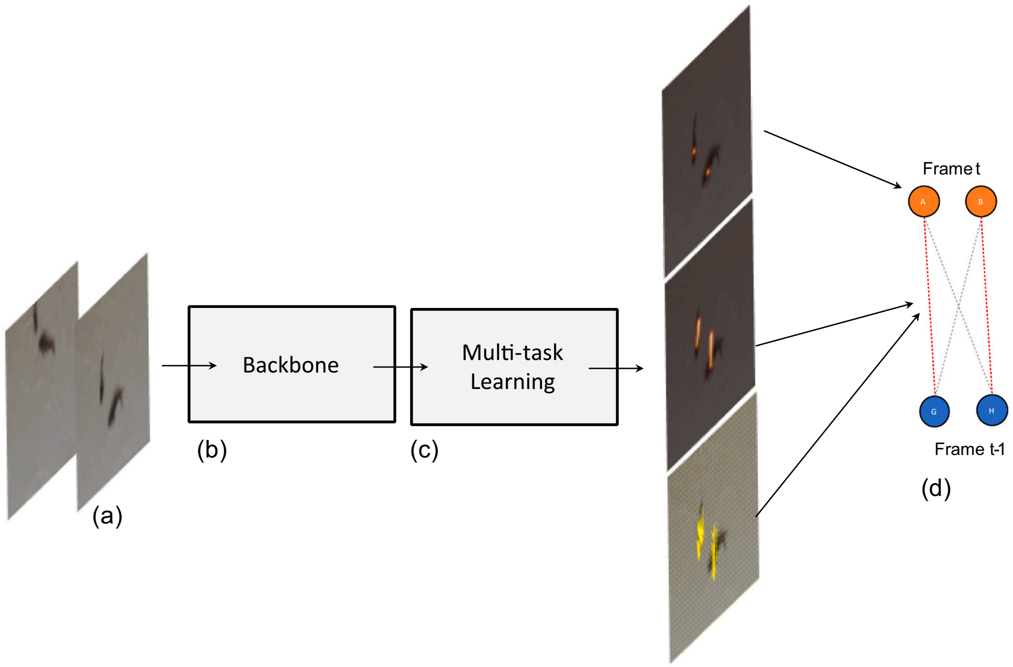


Fig. 1. General illustration of the proposed method. Initially (a) two consecutive frames are used to extract a (b) feature map via a backbone. This map is used to estimate (c) three tasks, one related to the position of the fingerlings and two related to the movement from one frame to another. Finally, these tasks are used to compose a (d) bipartite graph to detect fingerlings in the current frame.

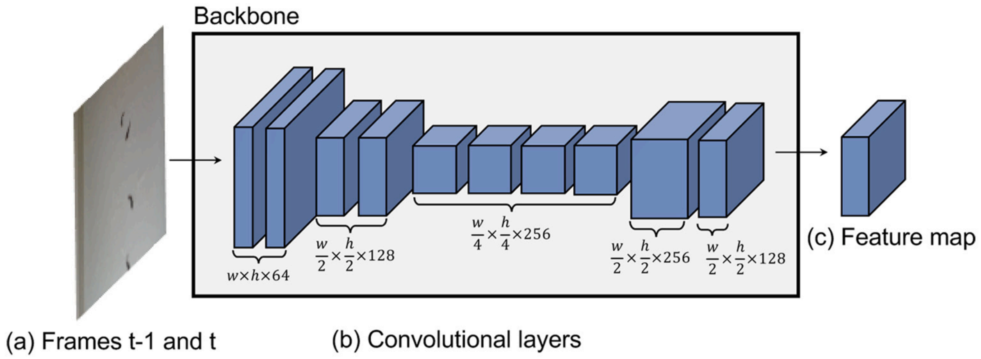

Backbone
$w \times h \times 64$
$\frac{w}{2} \times \frac{h}{2} \times 128$
$\frac{w}{4} \times \frac{h}{4} \times 256$
$\frac{w}{2} \times \frac{h}{2} \times 256$
$\frac{w}{2} \times \frac{h}{2} \times 128$
(c) Feature map
(a) Frames t-1 and t
(b) Convolutional layers

Fig. 2. Extraction of the feature map from two frames using a backbone based on the VGG architecture. The first two convolution layers have 64 filters of size 3 × 3, followed by a maxpooling layer with window 2 × 2 and stride 2. Then, two layers with 128 filters, one of maxpooling and another four convolution layers with 256 filters are used. As the estimate of the fingerling positions is a dense map, an upsampling layer is applied followed by two convolution layers with 256 and 128 3 × 3 filters, respectively. The last convolutional layer provides the feature map with resolution $\frac{w}{2} \times \frac{h}{2}$. Reducing the feature map by half is important to extract local features while decreasing the computational cost of the backbone.

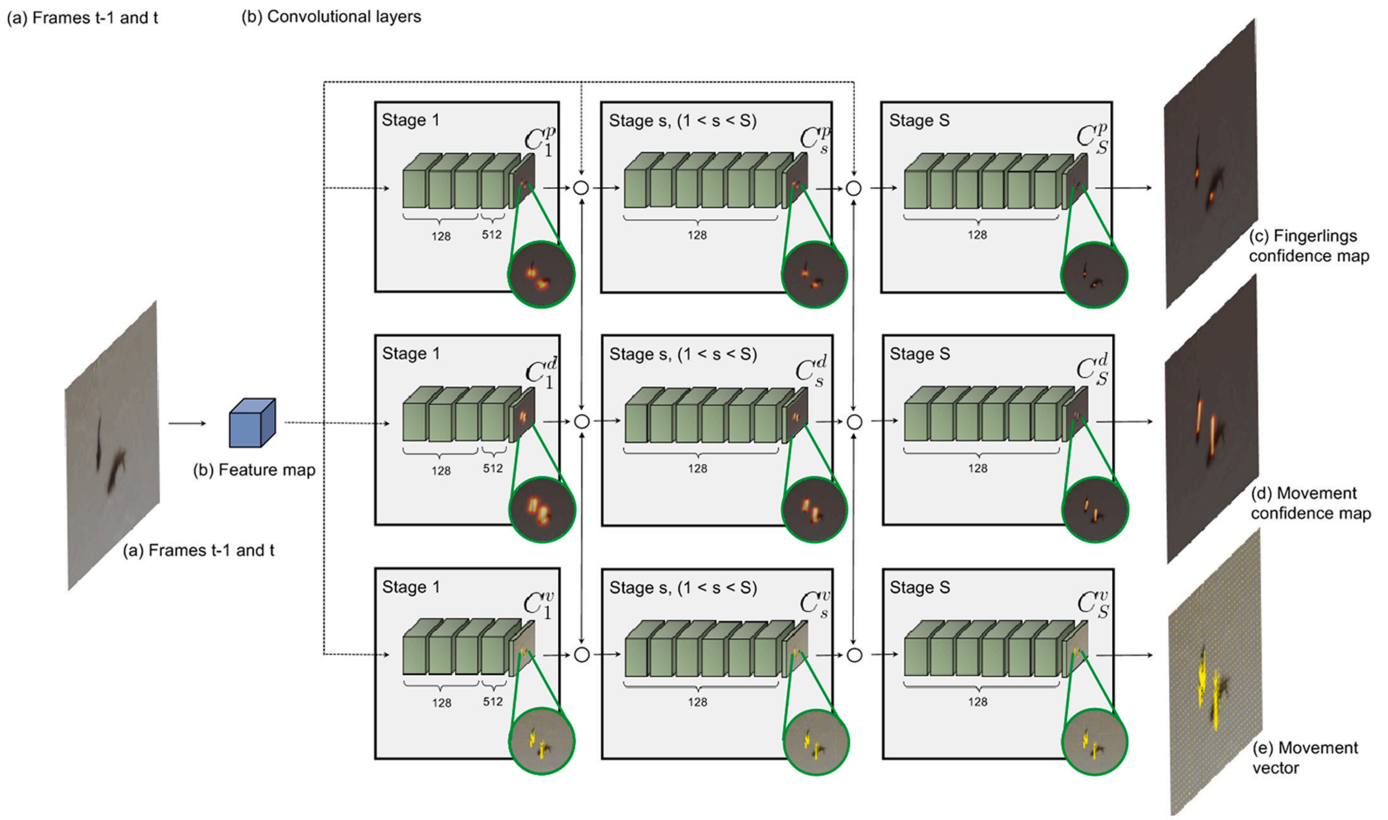


Fig. 3. Multi-task learning that estimates the position of fingerlings, vector and confidence map of the movement.

The second and third tasks estimate temporal information to assist in detection and counting. For this, consider that a fingerling is in a certain position (x_{t-1},y_{t-1}) in frame $t-1$ and that, in the next frame t, it has moved to a position (x_t,y_t). The second task (Fig. 3(d)) estimates a confidence map that represents the probability that a pixel belongs to the movement performed by a fingerling. Therefore, this task produces a high probability for all pixels belonging to the line that connects (x_{t-1}, y_{t-1}) to (x_t,y_t). This information is equivalent to the estimation of the footprint left by the fingerling from one frame to another. The third task is similar to the second, but estimates for each pixel a vector that points in the direction of the movement performed by the fingerling (Fig. 3(e)). This third task is related to the dense optical flow, but here estimated by a branch of the proposed method.

Given the feature map extracted from two frames, each task is estimated on a branch consisting of S stages (Fig. 3). The first stage of each of the three branches receives the feature map F and performs a series of convolution layers. The first three convolution layers have 128 filters of size 3 × 3 followed by a layer with 512 filters of size 1 × 1, all with the ReLU activation function. The last layer of the first and second branches has only one filter to estimate a confidence map for the position of the fingerlings in frame t (C_1^p) and a confidence map corresponding to the temporal movement of the fingerlings (C_1^d). The last layer of the third branch has two filters to estimate a motion vector on the x and y-axis (C_1^v).

At the end of the first stage of each of the three branches, estimates C_1^p, C_1^d, and C_1^v could be used to detect fingerlings. However, we found that they can be refined by more convolutional layers as shown in the experiments. As the information from a previous stage is concatenated, the stages assist in collaborative learning between tasks. The task of detecting the position of the fingerlings in the current frame can be impacted by information related to their movement and direction. In general, the first stage provides a rough prediction that is further refined by the other stages with the exchange of information between tasks.

For refinement, the later stage s concatenates the estimates from the previous stage C_{s-1}^p, C_{s-1}^d, C_{s-1}^v and the feature map F to estimate the refined information C_s^p, C_s^d and C_s^v. The final $S-1$ stages are composed of seven convolutional layers, five layers with 128 7 × 7 filters, one layer with 128 1 × 1 filters, and a final layer with the number of filters according to the first stage.

2.2.3. *Modeling fingerlings movement*

The position of the fingerlings in frame t is obtained by the peaks in the confidence map of the last stage C_S^p. A position (x,y) is a peak if its probability in C_S^p is greater than its 8 neighbors. To prevent low probability peaks from being detected as fingerlings, a position (x,y) is considered only if its probability is greater than a threshold τ.

The positions detected as fingerlings in the current frame and a previous frame are modeled with a complete bipartite graph. The vertices correspond to the detected fingerlings, being a set of vertices composed of fingerlings from the current frame t and the other set from the previous frame $t-1$. Fig. 4(a) presents an example of the complete bipartite graph with fingerlings detected in a previous frame (vertices in blue) and the current one (vertices in green). The vertices i of the previous frame are connected by edges e_{ij} with all the vertices j of the current frame as illustrated by the red edges in Fig. 4(a).

To include temporal information, the weight of an edge is calculated using estimates from motion vector C_S^v and motion confidence map C_S^d as shown in Figs. 4(a) and 4(b). Given an edge e_{ij}, equidistant points are sampled from the line segment between (x_i,y_i) and (x_j,y_j). For example, consider the edge connecting vertices A and C in Fig. 4(a). This edge can be seen as a line and equidistant points belonging to it can be sampled.

For each sampled point (x_l,y_l), we calculate the alignment between the line segment $\overline{(x_i,y_i)\left(x_j,y_j\right)}$ and the motion vector estimated in C_S^v(x_l, y_l). The alignment between two vectors can be calculated using the dot product according to Eq. 1 (Cao et al., 2017). Finally, the weight of the

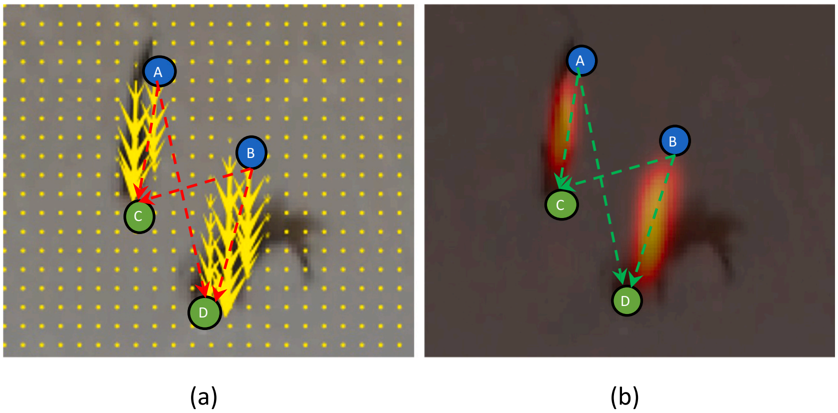


Fig. 4. Process for calculating the two weights of an edge based on the (a) motion vector and (b) motion confidence map. The vertices in blue and green correspond to the fingerlings detected in the previous and current frames. We can see that the edge that connects the vertices A and C is aligned with the motion vectors (Fig. 4(a)) and the motion confidence map (Fig. 4(b)), both predicted by the proposed method. Therefore, this edge has a greater weight than the edge that connects the vertices A and D or B and C, for example. A high weight is also associated with the edge that connects vertices B and D. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

edge considering the alignment e^{v}_{ij} is given by the sum of the alignment of all the sampled points, $e^{v}_{ij} = \sum_{l} e^{v}_{ij,\,l}$, where $e^{v}_{ij,\,l}$ is the alignment between the sampled point l and the edge (Eq. 1). In Fig. 4(a), the edge connecting A and C (red vector) is aligned with the motion vectors (yellow vectors) and therefore its weight is high. On the other hand, the weight of the edge connecting A and D is low, as the alignment between the edge and the vectors is different.

$$e^{v}_{ij,l} = C^{v}_{S}(x_l, y_l)\cdot\frac{(x_j, y_j) - (x_i, y_i)}{\| (x_j, y_j) - (x_i, y_i)\|_2}. \qquad (1)$$

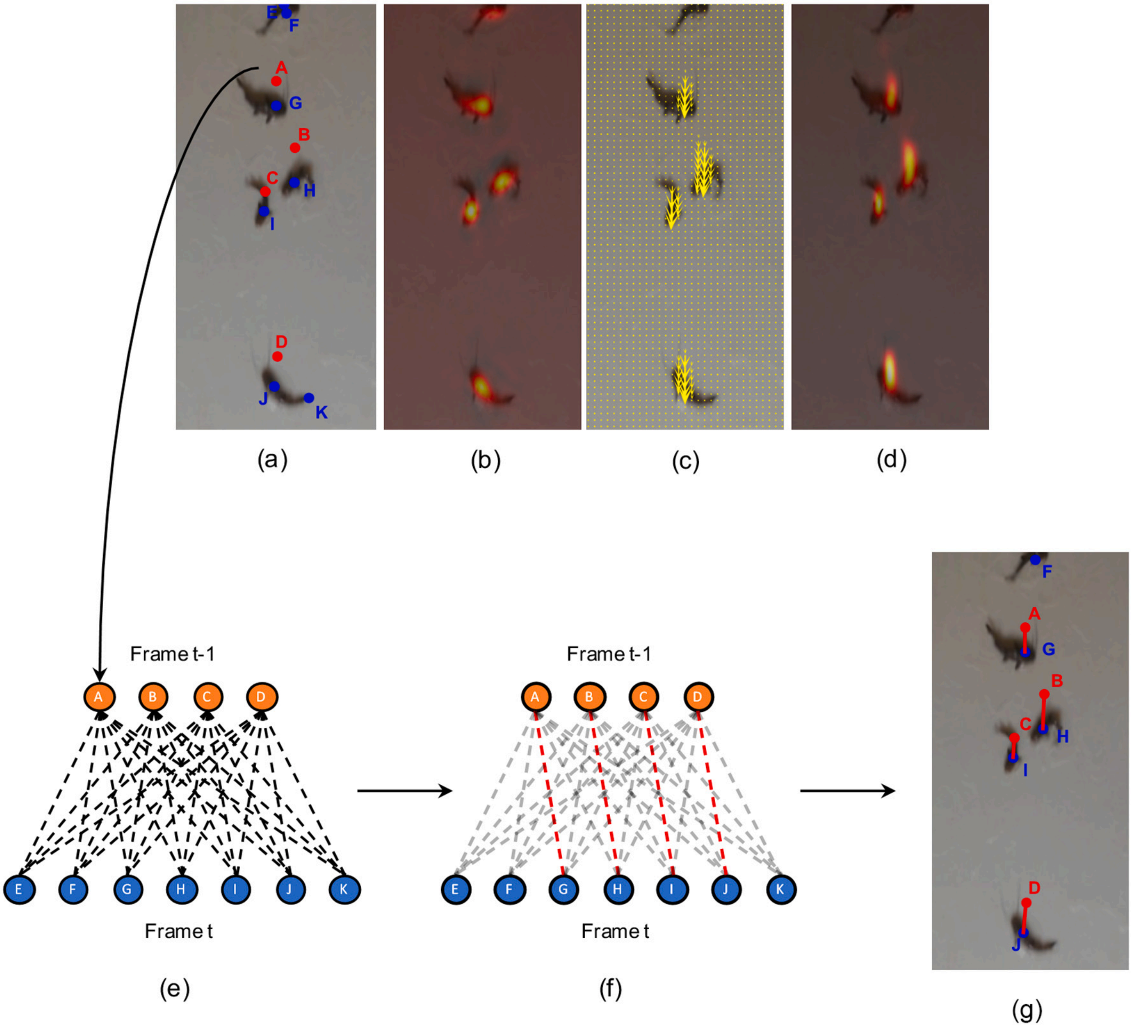


Fig. 5. Steps for detecting and counting fingerlings in a current frame. For a frame t, Fig. 5(a) shows the fingerlings detected in the previous frame $t-1$ (red dots) and in the current frame (blue dots). The confidence map of the fingerlings position is shown in Fig. 5(b). Fig. 5(c) represents the confidence map of the movement vectors and Fig. 5(d) represents the confidence movement map. The complete bipartite graph connecting the fingerlings from the previous and current frames is shown in Fig. 5(e). Figs. 5(f) and 5(g) shows the optimal match obtained using the Hungarian algoritm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In addition to the previous weight, we calculate a weight e_{ij}^{d} based on the probability that a pixel belongs to the movement performed by a fingerling. Similarly, we sample points from the line segment $\overline{(x_i,y_i)\left(x_j,y_j\right)}$ in the motion confidence map C_S^d. The weight is given by the sum of each sampled point l according to Eq. 2. In Fig. 4(b), we can see that the probability of movement at the points belonging to the edge between A and C is high. On the other hand, the edge weight between A and D is low, as the probabilities estimated by our method are also low.

$$e_{ij}^{d} = \sum_{l} C_S^d(x_l, y_l). \tag{2}$$

Finally, the weight of an edge e_{ij} is given by the sum of the two weights to include information from the two tasks (i.e., motion vector and motion confidence map):

$$e_{ij} = e_{ij}^{v} + e_{ij}^{d} \tag{3}$$

2.2.4. Fingerlings detection

To detect the fingerlings in the frame t to compose the bipartite graph, we use the confidence map prediction C_S^v searching for peaks even with a low threshold τ_{low}. Thus, the set of fingerlings detected in the frame t is generally greater than ideal and also greater than the number of fingerlings in the previous frame. The fingerlings in frame t are associated with the fingerlings in frame $t-1$, that is, we need to find a pair of fingerlings. The fingerlings that are not associated with any fingerling in the previous frame, but their peak is greater than τ_{high} are maintained as they are probably fingerlings entering the scene. After preliminary experiments, we used $\tau_{low} = 0.005$ and $\tau_{high} = 0.01$.

The optimal association between fingerlings in the complete bipartite graph is reduced to a maximum weight matching problem. Given a bipartite graph, a maximum matching is a subset of edges whose sum of their weights is maximized and that any two edges do not share a vertex. To find the optimal matching, we use the Hungarian algorithm (Kuhn, 1955).

Fig. 5 shows an example of the detection of fingerlings using the complete bipartite graph. The predictions for a frame t are shown in Figs. 5(b), 5(c) and 5(d), corresponding respectively to the confidence map of the fingerlings position, movement vectors and confidence movement map. Fig. 5(a) shows the fingerlings detected in a previous frame (red dots represented by the letters A to D) and the fingerlings detected in the current frame (blue dots with letters from E to K). We can see that the number of fingerlings detected in the frame t is overestimated due to the low threshold used in the confidence map (Fig. 5 (b)). Then, the complete bipartite graph is constructed (Fig. 5(e)) and the edges are weighted based on the confidence maps illustrated in Figs. 5(c) and 5(d). The optimal match is obtained using the Hungarian algorithm as shown in Fig. 5(f). Therefore, the fingerlings associated with a previous fingerling are maintained. In addition to these, the fingerlings not associated but with a high peak are also maintained (see the fingerling represented by the letter F in Fig. 5(g)).

2.3. Experimental setup

2.3.1. Proposed method training

The predictions made by the proposed method using a CNN were trained using stochastic gradient descent. Our loss function is applied at the end of each stage s according to Eqs. 4, 5 and 6 for the predictions of the fingerlings confidence map, movement confidence map and movement vector, respectively. Finally, the overall loss function is given by Eq. 7.

$$f_s^p = \sum_i \| \widehat{C}_s^p(i) - C_s^p(i)\|_2^2 \tag{4}$$

$$f_s^d = \sum_i \| \widehat{C}_s^d(i) - C_s^d(i)\|_2^2 \tag{5}$$

$$f_s^v = \sum_i \| \widehat{C}_s^v(i) - C_s^v(i)\|_2^2 \tag{6}$$

$$f = \sum_{s=1}^{S}\left(f_s^p + f_s^d + f_s^v\right) \tag{7}$$

where $\widehat{C}_s^p, \widehat{C}_s^d$ and $\widehat{C}_s^v$ are the ground truths for fingerling positions, movements and vectors, respectively.

Ground truths $\widehat{C}_s^p, \widehat{C}_s^d$ and $\widehat{C}_s^v$ are generated as follows. $\widehat{C}_s^p$ for a stage s is generated by calculating a Gaussian convolution across each pixel labeled as a fingerling position (Osco et al., 2021). To promote refinement during the stages, the Gaussian kernel of each stage has a standard deviation equally spaced between $[\sigma_{max}, \sigma_{min}]$. On the other hand, $\widehat{C}_s^d$ is generated from the movement of each fingerling. For this, a Gaussian kernel is positioned in each pixel of the line that connects the position of a fingerling in the previous and current frames that were previously labeled. The parameters of the Gaussian kernel of each stage follow the previous one. Finally, $\widehat{C}_s^v$ is constructed similarly to $\widehat{C}_s^d$, but using unit vectors. $\widehat{C}_s^v$ is a unit vector that points from the position of a fingerling in the previous frame to its position in the current frame.

Fig. 6 presents the ground truths for three stages using different values of σ. The RGB image is shown in Fig. 6(a) while the ground truths are shown in Figs. 6(b), 6(c) and 6(d). We can see that the first stage (first column of images) has more coarse ground truths while the ground truths of the later stages are more adjusted. This allows the proposed method to learn to refine its predictions in the later stages.

During training, the backbone was initialized with the pre-trained weights on ImageNet. We used the stochastic gradient descent optimizer with a learning rate of 0.01, a momentum of 0.9, and batch size of 2 during 100 epochs. These parameters were defined after preliminary experiments with the validation set.

Fig. 7 shows the loss function in the training and validation set for the three tasks separately, with the final loss corresponding to their sum. The blue, red and green curves correspond to the tasks of 1) fingerlings confidence map, 2) movement confidence map and 3) movement vector, respectively (Eqs. 4, 5 and 6). The loss function of the second and third tasks has higher values than the first one. This is because the movement of the fingerlings (second and third task) occupies a larger area of the image when compared to the center of the fingerlings (first task), and therefore a higher value is expected. Despite this, the loss function of the three tasks has similar values at the end of the training, i.e., 0.00026, 0.00067, and 0.00065. This indicates that multi-task learning was important and acted actively to solve the problem. It is also possible to observe that the loss in the training and validation set are close, indicating that there was no overfitting.

2.3.2. Metrics

To assess the detection of fingerlings, we use the Precision, Recall and F1 (F-measure) commonly applied in the literature. These metrics can be calculated according to Eqs. 8, 9, and 10.

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = 2\cdot\frac{P\cdot R}{P + R} \tag{10}$$

where TP, FP and FN stand for True Positive, False Positive, and False Negative, respectively. Since the labeling of each fingerling is only one point, a prediction is correctly assigned to a labeled fingerling if the distance between them is less than 20 pixels. This distance was empirically chosen to cover a fingerling in the image.

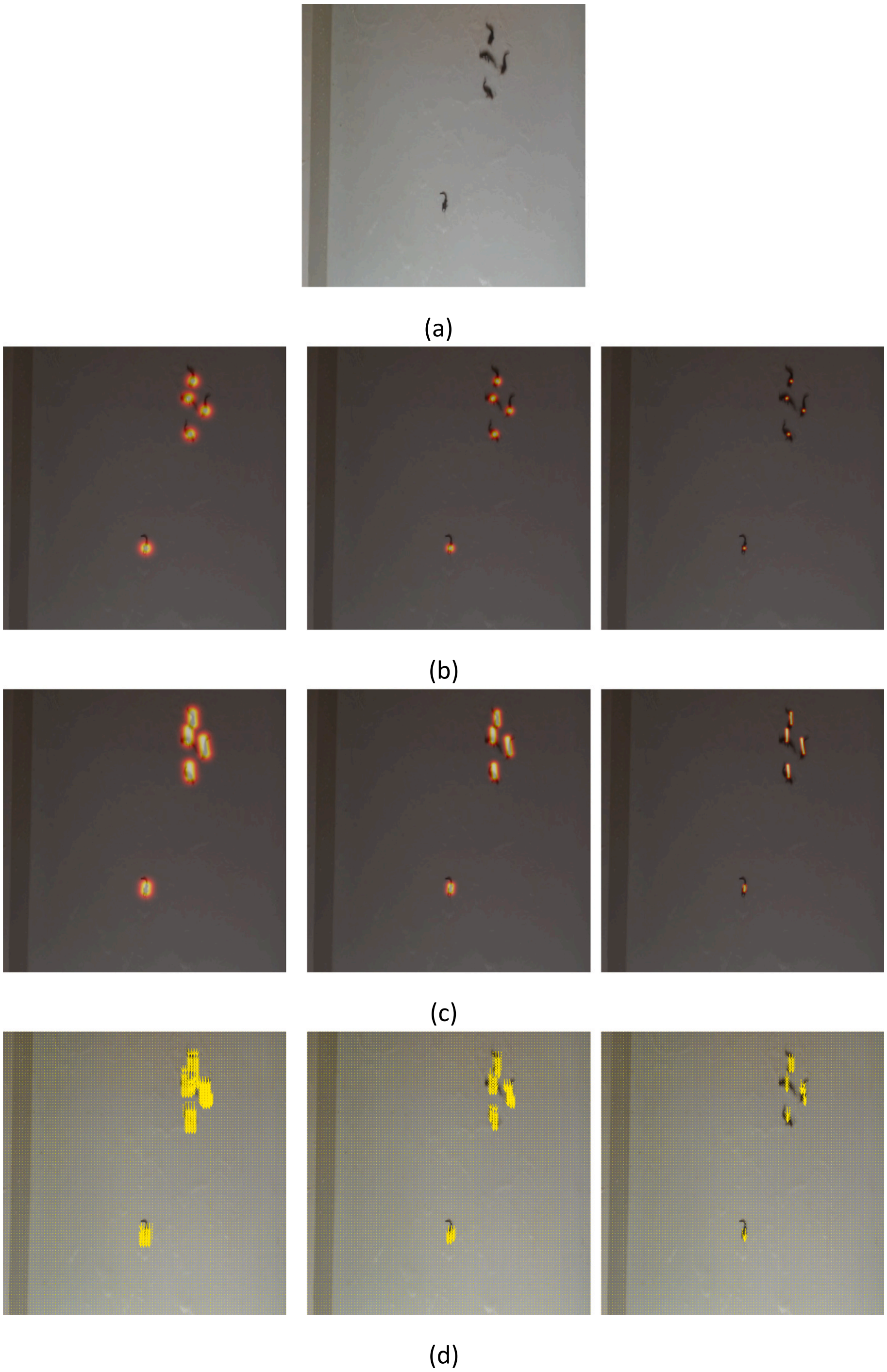


Fig. 6. Example of the ground truths generated for frame. Each column of images presents the ground truth for the stages.

3. Experiments and results

3.1. Assessment of multi-task learning

We assess the main parameters of the multi-task learning, including σ_{max} and σ_{min} (ground truth generation) and the number of stages S. The results for different values of σ_{max} are shown in Table 3 (we fix $S = 2$ and $\sigma_{min} = 1.0$). The best result was obtained with $\sigma_{max} = 4$, as it adequately covers the fingerling (see Fig. 8(a)).

We also assessed the influence of σ_{min} according to Table 4. σ_{max} was set to 4 due to previous results and we maintained two stages. A small value for σ_{min} gives superior results because the smaller spreading around the fingerling allows for precise refinement of its position, even when two fingerlings are nearby. For comparison, Fig. 8 shows the prediction of multi-tasks using $\sigma = 1$ and 4. It is possible to observe that the predictions using smaller values are more adjusted to the center of the fingerling.

Finally, we evaluated the number of stages as reported in Table 5. When using only one stage, the results are inferior to the others, which shows that refinement is an important part of the proposed method. Using two and three stages, the proposed method achieves its best results with F1 of 98.11 and 97.89, respectively. With more stages, the number of layers and consequently the number of weights to be learned increases, which can make training difficult. With these experiments,

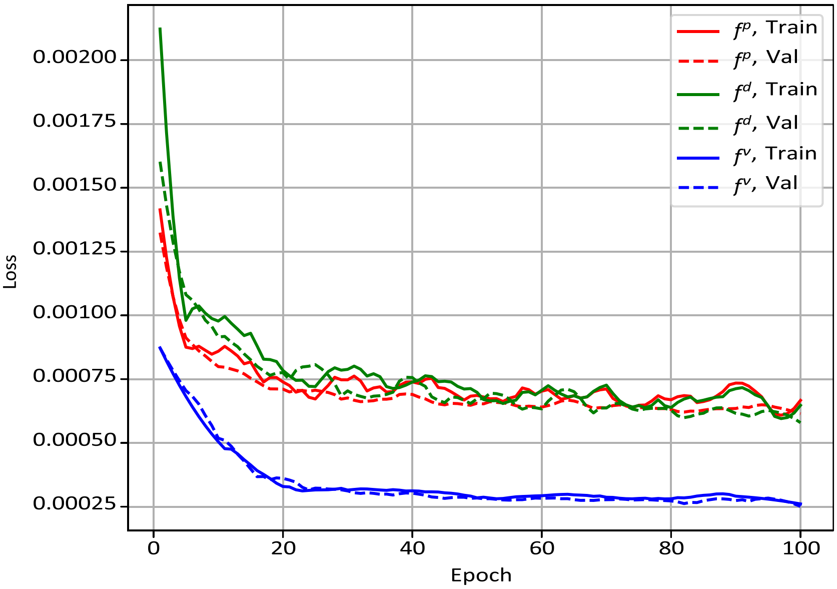


Fig. 7. Loss function of the three tasks in the training and testing set.

Table 3
Influence of σ_{max} on fingerling count using $\sigma_{min} = 1$ and number of stages $S = 2$.

σ_{max}	Precision	Recall	F1
1	90.03	98.20	92.90
2	94.09	98.15	95.37
3	96.96	99.02	97.61
4	97.51	99.37	98.12
5	96.07	99.20	97.17

the best results of the proposed method were obtained using $\sigma_{max} = 4$, $\sigma_{min} = 1$ and number of stages $S = 2$ or 3.

3.2. Temporal analysis in the detection of fingerlings

The first and second lines of Table 6 presents the results by performing the detection directly on the confidence map of the fingerlings' positions. The first line presents the results when one frame is used as input while the second line when two frames are considered as input. These experiments show the results without temporal information from the other tasks, although the prediction may contain indirect temporal information (e.g., two-frame input). We can see that the input with one or two frames presents similar results, with F1 of 93.96 and 93.25, respectively.

In the proposed method, the detection of fingerlings in the current frame occurs after the association in the complete bipartite graph, where the edge weight is calculated based on the fingerlings movement. The other lines of Table 6 show the results when temporal information is used explicitly. The second and third lines present the results considering separately the two motion predictions in edge weight. Finally, the last line of the table presents the results of the proposed method, in which the two predictions are used. The results show that not using temporal information directly results in low accuracy, as information on the amount of previous fingerlings is relevant to detect fingerlings in the current frame. Despite a small improvement, using temporal information in isolation (second and third rows of the table) also leads to an overestimation of fingerlings (false positives). On the other hand, the proposed method decreases the detection of false fingerlings improving the precision without decreasing the recall.

Detection without the use of temporal information is not adequate, especially when the fingerlings are in overlap close, forming a composite representation. The use of temporal information increases results considerably (e.g., from 93.25 to 96.94 and 96.82). The combination of the two predictions further increases the results, reaching F1 of 97.89.

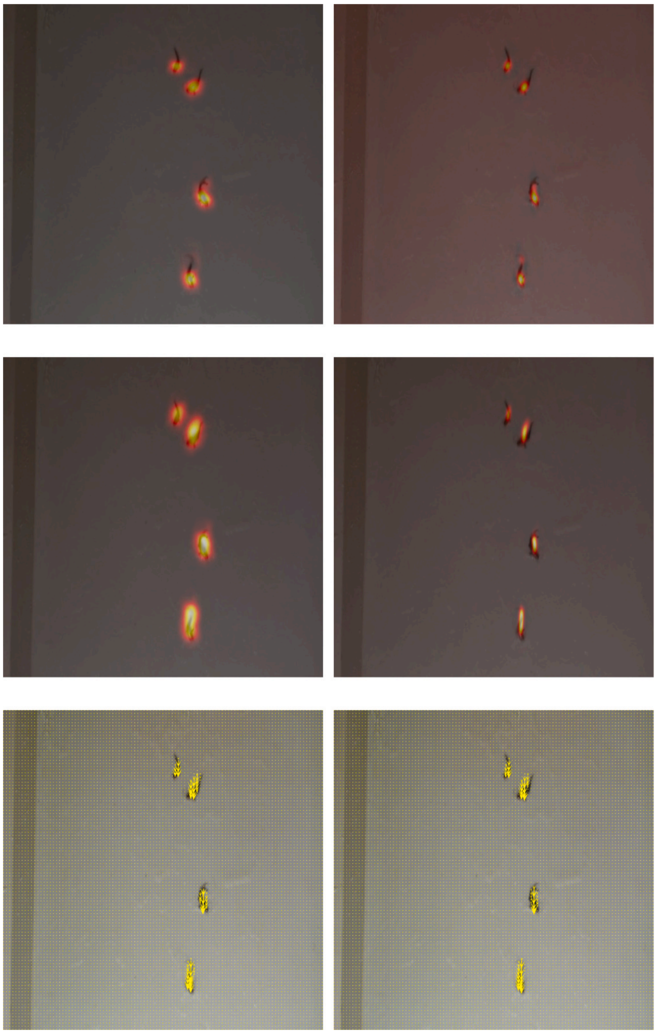
3.3. Density analysis

Table 7 presents the results considering the detection in frames with low (0 to 2 fingerlings), medium (3 to 5) and high (6 onwards) presence of fingerlings. With up to two fingerlings per frame, the proposed method reached F1 of 98.61 while from three to five fingerlings, F1 of 97 was obtained. Fig. 9 shows examples of detection of fingerlings in the 0–2 and 3–5 ranges. Red and blue dots indicate the position of fingerlings detected in the previous and current frames, respectively. The connections show the result of the association of the bipartite graph. (See Fig. 1.)

Relevant results were also obtained in frames with a large number of fingerlings (6–10) with an F1 of 95.42. It is interesting to note that precision was slightly lower in frames with 3–5 fingerlings compared to frames containing 6–10 fingerlings. However, this difference is small and we believe it is due to the greater adhesion between fingerlings of these frames. In any case, F1 that combines precision and recall proved to be consistent with the challenge posed by the numbers of fingerlings. Examples of detection with high density of fingerlings (6–10) are shown in Fig. 10. The proposed method was able to detect six, seven and ten fingerlings even when they are close and moving due to the use of multi-tasks approach.

The main challenge in the detection and counting of fingerlings is the overlap of two fingerlings visually forming a composite representation. Despite the challenge, the proposed method is able to detect the two fingerlings in most cases, as shown in Fig. 11. This is possible due to the association of a fingerling detected with low probability in the current frame with a fingerling in the previous frame. Without this association and the use of multi-task, one of the fingerlings would be discarded due to its low probability.

On the other hand, the errors of the proposed method occur mostly when two or more fingerlings enter the scene connected. The sequence of frames in Fig. 12 illustrates this situation. Although it is not possible to visually observe, three fingerlings enter the scene, but only one fingerling is detected initially. In the following frame, the proposed method detects two fingerlings while the third fingerling is only detected in the seventh frame of that sequence.



(a) $\sigma = 4$ (b) $\sigma = 1$

Fig. 8. Example of predictions made in (a) the first and (b) last stages. Each row of images represents the predictions for the position of the fingerlings, the confidence map of the movement and the movement vectors.

Table 4
Influence of σ_{min} on fingerling count using $\sigma_{max} = 4$ and number of stages $S = 2$.

σ_{min}	Precision	Recall	F1
1	97.51	99.37	98.12
2	95.56	99.35	96.87
3	94.52	98.48	95.90

Table 5
Influence of the number of stages on fingerling count using $\sigma_{min} = 1$ and $\sigma_{max} = 4$.

Stages (S)	Precision	Recall	F1
1	83.73	97.80	88.40
2	97.51	99.37	98.11
3	97.45	98.99	97.89
4	94.81	98.96	96.26

3.4. Comparison with object detection methods

This section compares the results of the proposed method with two object detection methods, Faster R-CNN Ren et al. (2015) and FCOS Tian et al. (2019). Faster R-CNN is a widely used object detection method and the basis for many works. FCOS is an object detection method that is similar to our proposal, as it performs detection on a confidence map. Both methods were trained and applied in isolated frames in order to validate the hypothesis that the proposed method benefits from temporal information.

Table 6
Comparative results using temporal information on edge weight.

Temporal Information	Precision	Recall	F1
Confidence map (one input frame)	92.09	97.15	93.96
Confidence map (two input frames)	90.27	98.74	93.25
C_S^d	95.97	98.82	96.94
C_S^v	95.87	98.72	96.82
Both	97.45	98.99	97.89

Table 7
Results of detection and counting in frames with different amounts of fingerlings.

N. of Fingerlings per frame	Precision	Recall	F1
0–2	98.01	99.81	98.61
3–5	95.94	98.87	97.00
6–10	96.81	94.41	95.42

Table 8 presents the comparative results using F1 (harmonic mean between precision and recall). We can see that FCOS and the proposed method outperform Faster R-CNN when there are few fingerlings in the frames, i.e., from 0 to 2 fingerlings. In this case, information from isolated frames is sufficient to detect and count the fingerlings, as there is little or no contact between them. On the other hand, when the number of fingerlings increases and consequently the contact between them, the proposed method outperforms the others. This can be observed for frames with 3–5 fingerlings, in which the proposed method reached F1 of 97 against 94.15 and 92.23 of FCOS and Faster R-CNN, respectively. For frames with 6–10 fingerlings, the proposed method provided F1 of 95.42 while FCOS and Faster R-CNN provided 91.86 and 93.84, respectively. The results corroborate the importance of using temporal information when there is contact between fingerlings.

4. Discussion

We propose an approach that allows us to automatically locate and count fingerlings in RGB images based on a multi-task convolutional neural network. This method can support aquaculturists in many important tasks, such as translocation of animals between breeding tanks and sale issues. Although the experiments were developed with a type of commercial fish, the Pintado, due to their importance in production in Brazil, the proposed method is capable to deal with fingerlings fishes in general, implying in its generability application. Our results demonstrated, with high accuracy (F1 of 95.42 or higher), that the CNN is capable to deal with different numbers of fingerlings (with up to 10) in the images. Identifying the contact of two or more fingerlings up to now represented a challenge (França Albuquerque et al., 2019; Garcia et al., 2020; Fouad et al., 2013; Rauf et al., 2019; Salman et al., 2019), but our method was able to solve this issue, which is an advantage for aquaculture tasks related to detecting and counting of fingerlings.

To evaluate the computational cost of the proposed method, we calculated the average time for prediction on a computer with 16 GB memory and an NVIDIA RTX 2080 card. The results showed that a frame runs in 0.26 s on average, which makes it feasible to use in real applications.

The main characteristic of the proposed method is related to its strategy of taking the advantage of the temporal information in the video (i.e., sequence of RGB imagery) to develop the task of interest. For

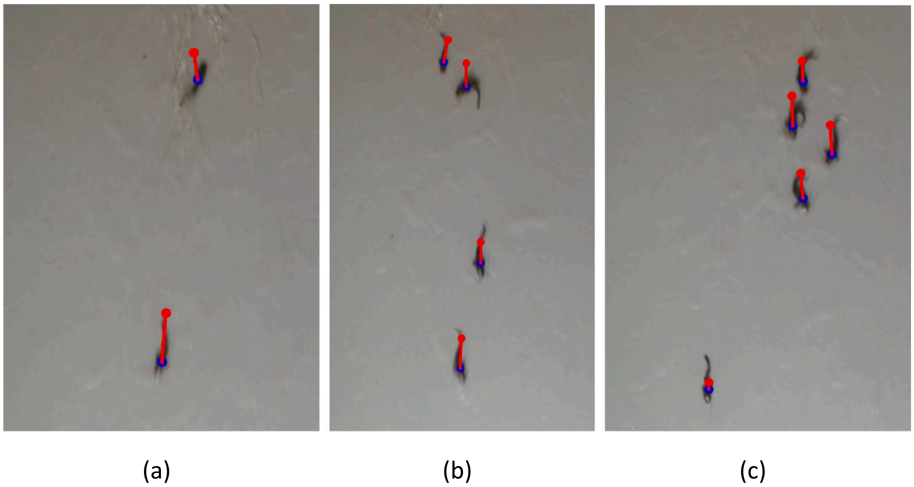


Fig. 9. Examples of detection of (a) two, (b) four and (c) five fingerlings.

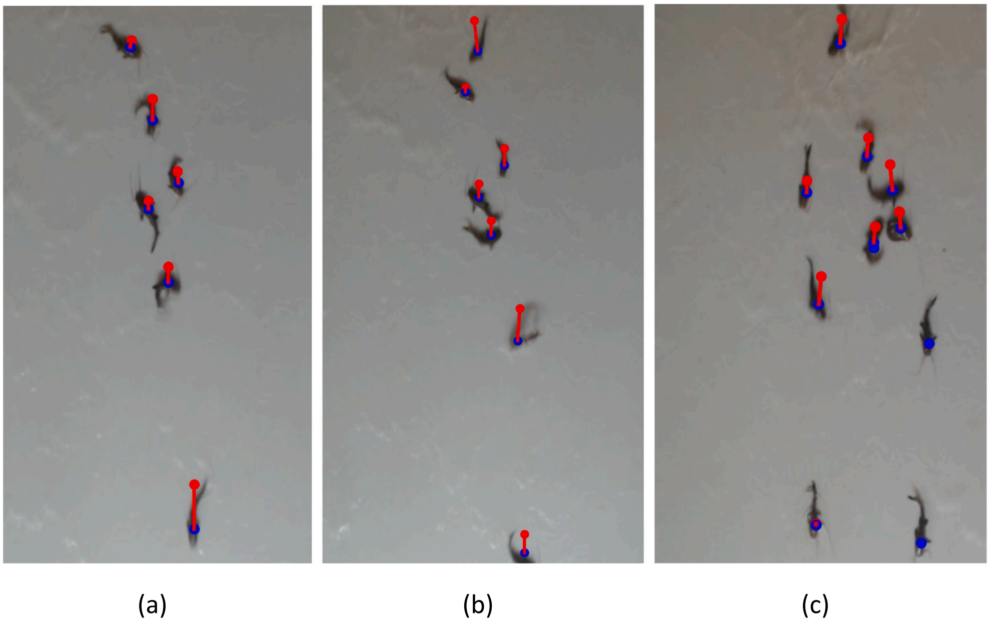


Fig. 10. Examples of detection of (a) six, (b) seven and (c) ten fingerlings.

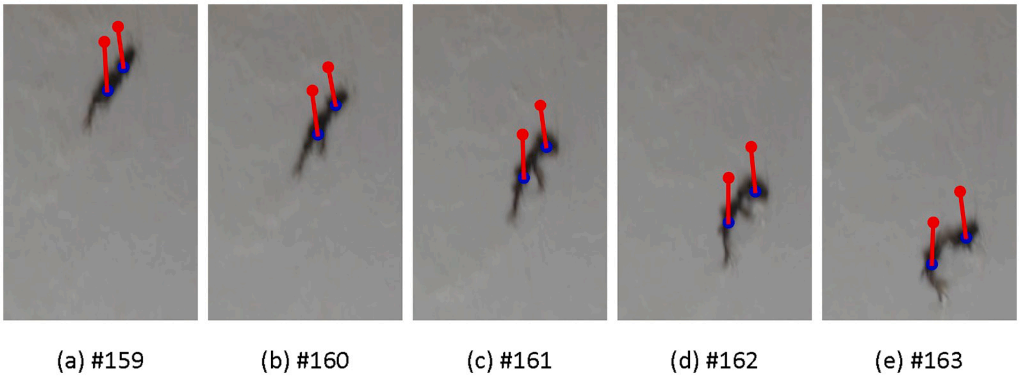


Fig. 11. Example of counting and detecting fingerlings in contact.

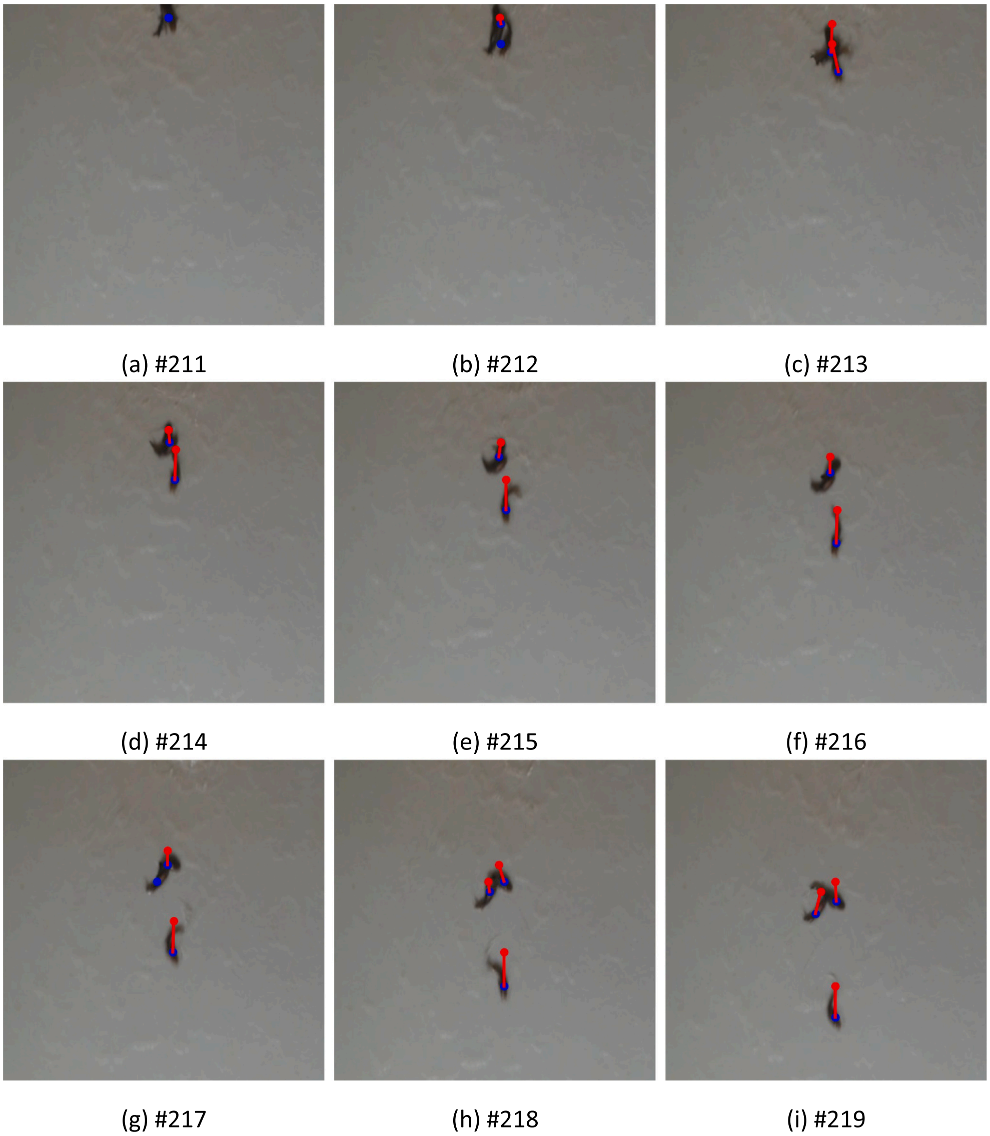


Fig. 12. Sequence of frames illustrating one of the main challenges for the proposed method.

Table 8
Comparative results between the proposed method and object detection methods using F1.

N. of Fingerlings per frame	Faster R-CNN	FCOS	Proposed Method
0–2	96.35	98.67	98.61
3–5	92.23	94.15	97.00
6–10	93.84	91.86	95.42

that, three tasks are learned: (i) to estimate the position of the fingerlings in a frame using the probability information that any individual image pixel is part of a fingerling; (ii) to determine the probability that a pixel belongs to the movement performed by a fingerling; and, (iii) to estimate, for each pixel, a vector that points in the direction of the movement performed by the fingerling. Consequently, the positions detected as fingerlings in the current frame and a previous frame are modeled with a complete bipartite graph. When evaluating the influence of parameters like σ_{max}, σ_{min}, and the number of stages of the multi-task learning, we found out that the results are affected, being the $\sigma_{max} = 4$, $\sigma_{min} = 1$, and the number of stages $S = 2$, the best configuration to obtain the higher F1 value in the proposed task. This configuration is recommended to be adopted in future experiments with other fingerlings datasets. We also noted that the results are substantially improved (the F1 value achieved 98% as can see Table 6, and it is the harmonic mean between precision and recall measurements) when the detection of fingerlings occurs using temporal information. This fact probably is because the complete bipartite graph is adopted, in which the edge weight is calculated using the fingerlings movement. These findings point out that temporal information usage is essential when the fingerlings are particularly in overlap close.

The evaluation of the proposed method for detecting and counting different amounts of fingerlings in frames is an important issue especially to prove its generalization capacity. Although the literature presents several studies (França Albuquerque et al., 2019; Garcia et al., 2020; Fouad et al., 2013; Rauf et al., 2019; Salman et al., 2019), related to the count of fingerlings, the main challenge in the detection and counting of these targets is the overlap of them, which visually forms a composite representation. Our results demonstrated the capacity of our approach of solving these situations like the number of fingerlings varying 6 to 10 elements. This was possible because our model associates the information of fingerling between frames (current and previous frame). The not usage of this information probably would result in one of the fingerlings discard which impacts the counting task accuracy. The main challenge faced by the proposed approach occurred mostly when two or more fingerlings enter the scene connected, however even in this situation our approach delivered high performance as demonstrated in Fig. 12.

5. Conclusions and future works

In this paper we proposed a detection-based method for counting fingerlings in an image sequence. Our method fits into the detection-based category, as it performs the fingerling detection through the confidence map, different from regression-based methods that estimate the quantity directly from the image. The proposed method showed satisfactory results to locate and count fingerlings using convolutional neural networks. The experimental results indicated that the use of temporal information increases results considerably, reaching F1 of 97.89. The proposed method was evaluated in frames with different numbers of fingerlings. The results showed that with up to two fingerlings per frame, the proposed method reached F1 of 98.61, from three to five fingerlings, F1 of 97, but also obtained relevant results in frames with a large number of fingerlings (6–10) with an F1 of 95.42. Due to the use of the multi-task approach, the proposed method was able to detect six, seven, and ten fingerlings even when they are close. Another advantage of this study is that, in most cases, the proposed method can detect the overlap of two or more fingerlings, which is considered the main challenge in the detection and counting of fingerlings. For future works, we suggest applying the proposed approach in images with an even more dense number of fingerlings. Moreover, we suggest testing the developed method using images capture by a camera with less resolution to verify its generalization ability to detect and count the fingerlings. Fingerling tracking is also a future work to assist the individual fingerling counts.

Author contributions

Conceptualization: José Marcato Junior; Marina de Nadai Bonin Gomes. Methodology: Maximilian Jaderson de Melo; Wesley Nunes Gonçalves. Software: Jonathan de Andrade Silva; Diogo Nunes Gonçalves. Validation: Marina de Nadai Bonin Gomes; Wesley Nunes Gonçalves. Formal analysis: Maximilian Jaderson de Melo; Lucas Prado Osco. Writing - Original Draft:Maximilian Jaderson de Melo; Ana Paula Marques Ramos; José Marcato Junior; Michelle Taís Garcia Furuya. Writing - Review & Editing: Lucas Prado Osco; Diogo Nunes Gonçalves; Wesley Nunes Gonçalves. Supervision: José Marcato Junior; Wesley Nunes Gonçalves; Marina de Nadai Bonin Gomes. Funding acquisition: José Marcato Junior; Wesley Nunes Gonçalves

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was supported by the FUNDECT - State of Mato Grosso do Sul Foundation to Support Education, Science and Technology, CAPES - Brazilian Federal Agency for Support and Evaluation of Graduate Education, and CNPq - National Council for Scientific and Technological Development. The Titan V and XP used for this research was donated by the NVIDIA Corporation. Additionally, the Pacu Aquaculture Ltd.

References

Cao, Z., Simon, T., Wei, S., Sheikh, Y., 2017. Realtime multi-person 2d pose estimation using part affinity fields. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1302–1310. https://doi.org/10.1109/CVPR.2017.143.

Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M., 2015. Convolutional features for correlation filter based visual tracking. In: The IEEE International Conference on Computer Vision (ICCV) Workshops.

Dos Santos, A.A., Gonçalves, W.N., 2019. Improving pantanal fish species recognition through taxonomic ranks in convolutional neural networks. Ecol. Inform. 53, 100977 https://doi.org/10.1016/j.ecoinf.2019.100977.

Fan, L., Liu, Y., 2013. Automate fry counting using computer vision and multi-class least squares support vector machine. Aquaculture *380-383*, 91–98. https://doi.org/10.1016/j.aquaculture.2012.10.016.

Fouad, M.M.M., Zawbaa, H.M., El-Bendary, N., Hassanien, A.E., 2013. Automatic nile tilapia fish classification approach using machine learning techniques. In: 13th International Conference on Hybrid Intelligent Systems (HIS 2013), pp. 173–178. https://doi.org/10.1109/HIS.2013.6920477.

França Albuquerque, P.L., Garcia, V., da Silva Oliveira, A., Lewandowski, T., Detweiler, C., Gonçalves, A.B., Pistori, H., 2019. Automatic live fingerlings counting using computer vision. Comput. Electron. Agric. *167*, 105015 https://doi.org/10.1016/j.compag.2019.105015.

Garcia, V., Sant'Ana, D.A., Garcia Zanoni, V.A., Brito Pache, M.C., Naka, M.H., França Albuquerque, P.L., Pistori, H., 2020. A new image dataset for the evaluation of automatic fingerlings counting. Aquac. Eng. *89*, 102064 https://doi.org/10.1016/j.aquaeng.2020.102064.

Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., Hassner, T., 2019. Precise detection in densely packed scenes. In: IEEE Conf. On Computer Vision and Pattern Recognition, pp. 5227–5236. arXiv:1904.00853.

Gonçalves, P., Lourenço, B., Santos, S., Barlogis, R., Misson, A., 2020. Computer vision intelligent approaches to extract human pose and its activity from image sequences. Electronics 9, 159. https://doi.org/10.3390/electronics9010159.

Hou, B., Li, J., Zhang, X., Wang, S., Jiao, L., 2019. Object detection and tracking based on convolutional neural networks for high-resolution optical remote sensing video. In: IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 5433–5436. https://doi.org/10.1109/IGARSS.2019.8898173.

Kuhn, H.W., 1955. The hungarian method for the assignment problem. Naval Res. Logist. Q. 2, 83–97. https://doi.org/10.1002/nav.3800020109 arXiv: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109.

Lainez, S.M.D., Gonzales, D.B., 2019. Automated fingerlings counting using convolutional neural network. In: 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), pp. 67–72. https://doi.org/10.1109/CCOMS.2019.8821746.

Li, X., Shang, M., Qin, H., Chen, L., 2015. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In: OCEANS 2015 - MTS/IEEE Washington, pp. 1–5. https://doi.org/10.23919/OCEANS.2015.7404464.

Liu, L., Lu, H., Xiong, H., Xian, K., Cao, Z., Shen, C., 2020. Counting objects by blockwise classification. IEEE Trans. Circ. Syst. Video Tech. 30, 3513–3527. https://doi.org/10.1109/TCSVT.2019.2942970.

Ma, C., Huang, J.-B., Yang, X., Yang, M.-H., 2015. Hierarchical convolutional features for visual tracking. In: The IEEE International Conference on Computer Vision (ICCV).

Nam, H., Han, B., 2016. Learning multi-domain convolutional neural networks for visual tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Osco, L.P., dos Santos de Arruda, M., Gonçalves, D.N., Dias, A., Batistoti, J., de Souza, M., Gonçalves, W.N., 2021. A cnn approach to simultaneously count plants and detect plantation-rows from uav imagery. arXiv:2012.15827.

Rauf, H.T., Lali, M.I.U., Zahoor, S., Shah, S.Z.H., Rehman, A.U., Bukhari, S.A.C., 2019. Visual features based automated identification of fish species using deep convolutional neural networks. Comput. Electron. Agric. 167, 105075 https://doi.org/10.1016/j.compag.2019.105075.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc, pp. 91–99 volume 28.

Salman, A., Siddiqui, S.A., Shafait, F., Mian, A., Shortis, M.R., Khurshid, K., Schwanecke, U., 2019. Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES J. Mar. Sci. 77, 1295–1307. https://doi.org/10.1093/icesjms/fsz025.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations, p. 14.

Sveen, L., Timmerhaus, G., Johansen, L.-H., Ytteborg, E., 2021. Deep neural network analysis - a paradigm shift for histological examination of health and welfare of farmed fish. Aquaculture 532, 736024. https://doi.org/10.1016/j.aquaculture.2020.736024.

Tian, Z., Shen, C., Chen, H., He, T., 2019. FCOS: Fully convolutional one-stage object detection. In: IEEE International Conference on Computer Vision, pp. 9627–9636.

Villon, S., Mouillot, D., Chaumont, M., Darling, E.S., Subsol, G., Claverie, T., Villéger, S., 2018. A deep learning method for accurate and fast identification of coral reef fishes in underwater images. Ecol. Inform. 48, 238–244. https://doi.org/10.1016/j.ecoinf.2018.09.007.

Wang, Y., Luo, X., Ding, L., Fu, S., Wei, X., 2019. Detection based visual tracking with convolutional neural network. Knowl.-Based Syst. 175, 62–71. https://doi.org/10.1016/j.knosys.2019.03.012.

Zhang, L., Li, W., Liu, C., Zhou, X., Duan, Q., 2020a. Automatic fish counting method using image density grading and local regression. Comput. Electron. Agric. 179, 105844 https://doi.org/10.1016/j.compag.2020.105844.

Zhang, S., Yang, X., Wang, Y., Zhao, Z., Liu, J., Liu, Y., Zhou, C., 2020b. Automatic fish population counting by machine vision and a hybrid deep neural network model. Animals 10. https://doi.org/10.3390/ani10020364.

Zhao, J., Bao, W., Zhang, F., Zhu, S., Liu, Y., Lu, H., Ye, Z., 2018. Modified motion influence map and recurrent neural network-based monitoring of the local unusual behaviors for fish school in intensive aquaculture. Aquaculture 493, 165–175. https://doi.org/10.1016/j.aquaculture.2018.04.064.

Zhou, C., Xu, D., Chen, L., Zhang, S., Sun, C., Yang, X., Wang, Y., 2019. Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision. Aquaculture 507, 457–465. https://doi.org/10.1016/j.aquaculture.2019.04.056.