# Meta-Learning with One-Shot Positive and Negative Bags of Examples for Feeding Splash Segmentation

*Chia-Cheng Sung (宋嘉誠), Chin-Chun Chang (張欽圳)†, Shyi-Chyi Cheng (鄭錫齊)*

Department of Computer Science and Engineering,
National Taiwan Ocean University, Keelung, Taiwan

**ABSTRACT**

In aquaculture, automatic feeding systems are often adopted to reduce operational costs. These systems must assess the appetite of fish because surplus feed is wasted and can pollute the water and cause fish diseases. The appetite of certain fish species can be indirectly analyzed from splashes created when they grab feed. The pattern and coverage of these feeding splashes are various and related to many factors, such as aquaculture environments and fish conditions. To adapt to various environments, a meta-learning approach is proposed to detect feeding splashes. The proposed detector is a multi-task semantic segmentation network with multiple output branches, each aimed at feeding splash segmentation for a specific kind of aquaculture environments. A task-clustering strategy is proposed to automatically determine the number of branches and the assignment of environments to branches. The proposed approach uses contrastive learning with one-shot positive and negative bags of examples to combine the output branches without human labeling. The proposed approach has been tested against data collected from several aquaculture environments, and experimental results have shown the feasibility of the proposed approach.

***Index Terms***— Feeding-Splash Detection; Semantic Segmentation; Meta-Learning; Multiple-Instance Learning; Contrastive Learning.

## 1. Introduction

In aquaculture, automatic feeding systems are often adopted to reduce the cost of human labor. However, due to their inability to assess the appetite of fish, most existing automatic feeding systems may overfeed [1]. The surplus feed is wasted and can pollute the water. Smart fish-feeding systems must be able to perceive the appetite of fish [2, 3].

Experienced aquaculture operators usually feed the fish in fish farms a certain amount of prepared feed through several times of feeding. After the first feeding, the fish usually starts to eat a few moments later and some fish species can also make splashes at the same time. Then, aquaculture operators continue feeding and observe the activity of the fish indirectly through feeding splashes. Once the strength and coverage of feeding splashes become weak and small, experienced aquaculture operators can be aware of satiety and health of the fish. Smart fish-feeding systems can assess fish feeding intensity indirectly through feeding splashes [4].

The visual pattern of feeding splashes is a type of spatio-temporal texture and is often identifiable if it is stronger than background water waves. The strength and pattern of background water waves can be various due to different weather conditions and the configurations and operations of farm facilities. Similarly, the strength and pattern of feeding splashes are various and related to a lot of factors, such as the fish species, fish quantity, fish's appetite, fish maturity, solar altitude, water color and turbidity, and the camera's view angle, etc. A variety of fish farms and patterns of feeding splashes are shown in Figures 1 and 2. Without increasing the burden on aquaculture operators, vision-based feeding splash detectors need to adapt to the environment automatically.

In this paper, because feeding splashes usually show specific patterns and irregular shapes, and because the coverage of feeding splashes is also an important factor in assessing feeding intensity, semantic segmentation networks are applied to detect feeding splashes. Since experienced annotators can usually identify feeding splashes in a single image, the input for the proposed detector is one single image. Thus, the proposed detector can be used with both stationary and non-stationary cameras. Additionally, images obtained before feeding usually contain no feeding splashes and can form a negative bag of examples. Images obtained a moment later after the first feeding can comprise a positive bag of examples because at least one of them usually contains feeding splashes if the appetite of the fish is not low. By leveraging this multiple-instance characteristic [5] of the initial feeding

†Corresponding author:Chin-Chun Chang E-mail:cvml@mail.ntou.edu.tw

**Fig. 1**: Examples of aquaculture environments, which are, from left to right, an aquaculture tank, two concrete tanks, an outdoor tank, and an offshore fish cage.


**Fig. 2**: Examples of feeding splashes, where feeding splashes are indicated by yellow circles and the other ripples are background water waves.

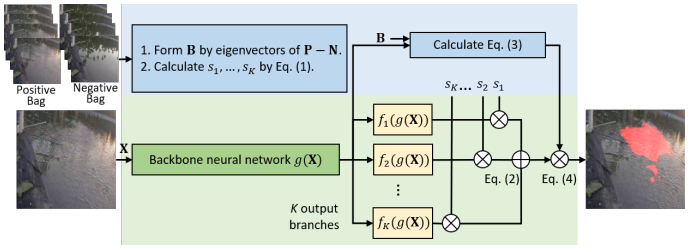



**Fig. 3**: The proposed feeding splash detector.

stage, the proposed detector adopts a meta-learning approach [6, 7] with one-shot positive and negative bags of examples to automatically adapt to different environments.

As Figure 3 shows, the proposed detector is a semantic segmentation neural network with multiple output branches, which are combined to form the network's final output. Each output branch is trained to detect feeding splashes in specific kinds of aquaculture environments. Because the pattern of feeding splashes and background water waves in different environments can vary and be incompatible, such a neural architecture can be trained more easily. In the inference phase, these output branches are combined in accordance with the analysis of the positive and negative bags of examples obtained from the initial feeding stage. Intuitively, each output branch is specific but less general. Combining these output branches can improve generalization.

The contributions of this paper are two-fold.

- First, the proposed neural architecture with the task clustering approach in mutli-task learning [8] facilitate the learning of feeding splash patterns in various and even incompatible kinds of aquaculture environments.
- Second, the proposed detector can adapt to an aquaculture environment by using positive and negative bags of examples without human annotations.

The remaining part of this paper is organized as follows. Related works are reviewed in Section 2. The proposed feeding splash detector is presented in Section 3. The experimental results are analyzed in Section 4. The concluding remarks are drawn in the last section.

## 2. Related Work

Computer vision systems provide noninvasive means of monitoring in aquaculture and are suitable for many aquaculture-related applications [9]. Deep neural networks have been applied to quantify the amount of excess feed pellets floating on the water and the feeding behavior of a fish school from the top view of the fish tank [10]. In outdoor fish farms, a classifier based on convolutional neural networks (CNNs) is adopted to determine the presence of feeding splashes on consecutive frames for analyzing the fish feeding state in an outdoor black-porgy pond [11]. In [12], CNN-based classifiers are adopted for determining both the feeding intensity and fish density. However, for analysis of the coverage of feeding splashes, semantic segmentation networks are more suitable than image classifiers.

Semantic segmentation networks have been applied to extract water bodies in remote sensing images [13] and to segment water and obstacle regions in maritime images for unmanned surface vehicles [14]. A variety of semantic segmentation networks have been proposed and can be categorized into CNN-based and transformer-based models. CNN-based models include FCN [15], DeepLab [16], P-Net [17], U-Net [18], SegNet [19], FgSeg-

Net [20], DANet [21], SegNeXt [22], etc. In [22], it shows that a strong backbone, multi-scale interaction, and spatial attention are three key properties of successful semantic segmentation networks.

On the other hand, due to the attention mechanism, transformer-based models, such as SETR [23] and SegFormer [24], are easier to capture the long-range dependency between patches in the image. For analyzing feeding splashes, models that are quickly adaptable to aquaculture environments without human intervention are needed.

Few-shot and zero-shot learning [25–29] provide means for a model to quickly adapt to a new environment with a few labeled data (a support set) or prompts. For this application, however, the support set comprises positive and negative bags of examples. The training set may include data from environments where feeding splashes are incompatible. These observations motivate the proposed approach. Such an application of semantic segmentation networks is rarely studied in the literature.

## 3. Methodology

In this section, the first part introduces the neural architecture. The second part presents the proposed step for determining the number of output branches, assigning aquaculture environments to the output branches, and simultaneously training the neural architecture. Finally, the proposed model with one-shot positive and negative bags of examples is presented.

### 3.1. The Neural Architecture for the Feeding-Splash Detector

The backbone of the feeding-splash detector is SegFormer because it can learn the long-range dependency between patches. Let $g(\mathbf{X})$ denote the output feature map of the backbone for the input image $\mathbf{X}$. The detector has several output branches. Each branch is aimed at detecting feeding splashes in specific training aquaculture environments. Let $f_i(g(\mathbf{X}))$ denote the $i$th branch for the input image $\mathbf{X}$:

$$\mathbf{D}_i = f_i(g(\mathbf{X})),$$

where the segmentation result $\mathbf{D}_i$ describes for each pixel the confidence level $\in [0, 1]$ of the presence of feeding splashes. This neural architecture is popular in multi-task learning because the backbone can learn effective feature representations from different aquaculture environments. Each branch can detect feeding splashes for specific kinds of aquaculture environments, thus resolving the incompatibility in feeding splashes that may exist between different aquaculture environments.

### 3.2. The Step of Training the Feeding-Splash Detector

Let $\{(\mathbf{X}_i, \mathbf{Y}_i, y_i, h_i)\}$ denote the training set for the detector, where $\mathbf{X}_i$ denotes the $i$th training image, $\mathbf{Y}_i$ denotes the annotation image for $\mathbf{X}_i$, $y_i \in \{1, \ldots, E\}$ denotes the aquaculture environment of $\mathbf{X}_i$, and $h_i \in \{1, \ldots, K\}$ denotes the output branch assigned to $\mathbf{X}_i$, where $E$ and $K$ denote the number of aquaculture environments and the number of branches, respectively. Generally, $K$ is not greater than $E$. The loss function for training the detector can be defined as follows:

$$\mathcal{L}(\{(\mathbf{X}_i, \mathbf{Y}_i, y_i, h_i)\})) = -\frac{1}{N}\sum_{i=1}^{N} \ell_{CE}(f_{h_i}(g(\mathbf{X}_i)), \mathbf{Y}_i)$$

where $N$ is the total number of training examples, and $\ell_{CE}(f_{h_i}(g(\mathbf{X}_i)), \mathbf{Y}_i)$ is the cross-entropy loss function for measuring the consistency between $f_{h_i}(g(\mathbf{X}_i))$ and $\mathbf{Y}_i$.

If the output branch for $\mathbf{X}_i$, $h_i$, is predefined and fixed, training the detector by minimizing the loss function is straightforward. However, when there are a number of aquaculture environments, determining the number of output branches and assigning the proper output branch to each training image are not always easy. Because each branch is aimed for specific kinds of aquaculture environments, training images from the same environment are assigned to the same branch; that is, $h_i = h_{i'}$ if $y_i = y_{i'}$. The proposed steps are described as follows.

Let $\mathcal{X}$ and $\mathcal{V}$ represent the training and validation sets, respectively. Let $IoU^*$ denote the best mean Intersection over Union (IoU) score obtained on the validation set. Let $K^*$ and $h_i^*$ denote the number of branches and the branch assignment for the $i$th environment associated with $IoU^*$, respectively. The steps for training and determining the number of branches and assignments are as follows:

- Step 1. $K \leftarrow 1; h_i \leftarrow 1, i = 1, \ldots, E$; $IoU^* \leftarrow 0$.
- Step 2. Repeat the following steps until $IoU^*$ is not improved for $\delta_s$ epochs.
  - Step 2.1. Assign the $E$ environments to the $K$ branches and obtain $h_1, \ldots, h_E$. (The steps will be described later.)
  - Step 2.2. Train the detector on $\mathcal{X}$ with respect to $h_1, \ldots, h_E$ for one epoch and then calculate the $IoU$ of the detector on $\mathcal{V}$ with respect to $h_1, \ldots, h_E$.
  - Step 2.3. If $IoU > IoU^*$, then $IoU^* \leftarrow IoU, K^* \leftarrow K, h_i^* \leftarrow h_i, i = 1, \ldots, E$, and save the model.
- Step 3. If $K < E$ and $K = K^*$, then add one branch and assign the $E$ environments to the $K + 1$ branches by the steps:

– Step 3.1. Assign the $E$ environments to the $K$ branches plus one extra branch and obtain $h_1, \ldots, h_E$ and $G'$, which is the group assigned to the extra branch. (The steps will be described later.)

– Step 3.2. Identify the branch $j$, where the environments in $G'$ have the least total training loss. Add a new branch and copy the weight of the $j$th branch to the new branch.

– Step 3.3. $K \leftarrow K + 1$; Go to Step 2.

- Step 4. Output $K^*$, $h_i^*$, $i = 1, \ldots, E$, and the model associated with $IoU^*$.

To reduce computational cost of Step 2, Step 2.1 is not executed if $h_1, \ldots, h_E$ do not change after several successive epochs.

Assigning the $E$ environments to the $K$ branches comprises two steps:

- Step 1. Partition the $E$ environments into $K$ groups: Let $\boldsymbol{\lambda}_i = [\lambda_{i1}, \ldots, \lambda_{iK}]^T$ be a vector for the $i$th environment, where $\lambda_{ij}$ denotes the average loss of the training example of the $i$th environment with respect to the $j$th branch. To partition the $E$ environments into $K$ groups, a distance measure $d(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_{i'})$ for the $i$th and $i'$th environments is defined as

$$d(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_{i'}) = \min_{j \in \{1,\ldots,K\}} \frac{\lambda_{ij} + \lambda_{i'j}}{2}.$$

  That is, the distance between two environments is small if the patterns of feeding splashes in the two environments can be well learned by an output branch. Based on this distance measure, the hierarchical clustering algorithm with the complete linkage is applied to partition the $E$ environments into $K$ groups, denoted as $G_1, \ldots, G_K$.

- Step 2. Assign the $K$ groups to the $K$ branches: Let $\mathbf{C} = [c_{ij}]$ denote a $K \times K$ cost matrix, where $c_{ij} = \frac{\sum_{u \in G_i} \lambda_{uj}}{|G_i|}$ denotes the average loss of environments in the $i$th group with respect to the $j$th branch. Apply the algorithm for the minimum sum assignment problem with the cost matrix $\mathbf{C}$ to assign the $K$ groups to the $K$ branches. If $G_i$ is assigned to the $j$th branch, then all environments in $G_i$ are assigned to the $j$th branch (i.e., $\forall i' \in G_i, h_{i'} = j$).

Assigning the $E$ environments to the $K$ branches plus one extra branch also involves two steps:

- Step 1. Partition the $E$ environments into $K + 1$ groups, and calculate the $(K + 1) \times K$ cost matrix $\mathbf{C}$.

- Step 2. Apply the algorithm for the minimum sum assignment problem with the cost matrix $\mathbf{C}$ to assign the $K + 1$ groups to the $K$ branches. The group, denoted as $G'$, that is not assigned to any of the $K$ branches is assigned to the extra branch (the $(K + 1)$th branch). The branch assignment for the environment can then be carried out as usual.

### 3.3. One-Shot Learning on Positive and Negative Bags of Examples

If the detector has only one output branch or the target environment is similar to one of the training environment, the appropriate output branch can be selected accordingly. Otherwise, the detector adapts to the target environment using one-shot learning.

Let $\mathbf{X}_1, ..., \mathbf{X}_{t_0}, ..., \mathbf{X}_{t_1}$ denote the image frames covering two important initial events of the feeding process: (1) the first feeding is finished at frame $\mathbf{X}_{t_0}$; and (2) feeding splashes must arise before frame $\mathbf{X}_{t_1}$. Ideally, weak or no feeding splashes exist in frames $\mathbf{X}_1, ..., \mathbf{X}_{t_0}$ and feeding splashes appear in at least one of the frames $\mathbf{X}_{t_0+1}, ..., \mathbf{X}_{t_1}$. Based on the setting of multiple-instance learning, $\mathbf{X}_1, ..., \mathbf{X}_{t_0}$ comprise a bag of negative examples, and $\mathbf{X}_{t_0+1}, ..., \mathbf{X}_{t_1}$ comprise a positive bag, which includes at least one positive example. Notice that in a positive example, the position where feeding splashes are present is unknown.

The response of the $i$th branch to feeding splashes in the image frame $\mathbf{X}_t$, denoted as $r_{it}$, can be defined as

$$r_{it} = \sum_{x,y} (f_i(g(\mathbf{X}_t)) \otimes \mathbb{I}(f_i(g(\mathbf{X}_t)) \geq 0.5))$$

where $\otimes$ denotes the element-wise multiplication operator and $\mathbb{I}(f_i(g(\mathbf{X}_t)) \geq 0.5)$ is the indicator function, which is 1 if the condition is true and 0 otherwise. The responses of the $K$ output branches, denoted as $p_i$ and $n_i$, for $i = 1, \ldots, K$, to feeding splashes in the examples of positive and negative bags can be analyzed by $p_i = \max_{t_0+1 \leq t \leq t_1} \{r_{it}\}$ and $n_i = \frac{1}{t_0} \sum_{t=1}^{t_0} r_{it}$, respectively. If the $i$th branch is appropriate for the target environment, its maximum response to feeding splashes in the positive bag example, $p_i$, should be strong, and its average response to feeding splashes in the negative bag example, $n_i$, should be weak. Accordingly, the scores for the $K$ branches to the target environment can be defined as

$$s_i = \frac{\exp(\alpha w_i)}{\sum_{i'=1}^{K} \exp(\alpha w_{i'})} \tag{1}$$

where $\alpha$ (set to 4 in this paper) is a hyper-parameter and $w_i = \frac{p_i}{n_i+1} \times \rho_i$ with $\rho_i = \frac{(n_i+1)^{-1}}{\sum_{i'=1}^{K}(n_{i'}+1)^{-1}}$. Thus, the output of the detector can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^{K} s_i f_i(g(\mathbf{X})). \tag{2}$$

Additionally, from positive bag examples and negative bag examples, it is possible to find the deep feature subspace covering the deep feature vectors of the feeding splashes in the target environment using contrastive learning. Let the matrices $\mathbf{P}$ and $\mathbf{N}$ represent the autocorrelation matrices of deep feature vectors of feeding splashes in positive bag examples and negative bag examples, respectively. Let $\mathbf{P}_i$ and $\mathbf{N}_i$, defined as follows, be the corresponding autocorrelation matrices for the $i$th branch:

$$\mathbf{P}_i = \sum_{t=t_0+1}^{t_1} \frac{r_{it}}{\sum_{t=t_0+1}^{t_1} r_{it}} \mathbf{F}_{it}\mathbf{W}_{it}\mathbf{F}_{it}^T,$$

$$\mathbf{N}_i = \frac{1}{t_0}\sum_{t=1}^{t_0} \mathbf{F}_{it}\mathbf{W}_{it}\mathbf{F}_{it}^T,$$

where the column vector of the matrix $\mathbf{F}_{it}$ comprises the deep feature vector in $g(\mathbf{X}_t)$ with the corresponding value of $f_i(g(\mathbf{X}_t)) \geq 0.5$, and this value divided by $r_{it}$ is the corresponding diagonal element of the diagonal matrix $\mathbf{W}_{it}$ for weighting. The autocorrelation matrices $\mathbf{P}$ and $\mathbf{N}$ can be estimated by $\mathbf{P} = \sum_{i=1}^{K} s_i\mathbf{P}_i$ and $\mathbf{N} = \sum_{i=1}^{K} s_i\mathbf{N}_i$.

The eigenvectors of $\mathbf{P} - \mathbf{N}$ associated with the eigenvalues greater than a threshold ratio of the largest eigenvalue can span the deep feature subspace for feeding splashes in the target environment. Let these eigenvectors comprise the column vectors of the matrix $\mathbf{B}$. Define the matrix $\mathbf{M} = \frac{\text{norm}(\text{proj}_{\mathbf{B}}(g(\mathbf{X}))}{\text{norm}(g(\mathbf{X}))}$, where the function $\text{proj}_{\mathbf{B}}(g(\mathbf{X}))$ maps each deep feature vector in $g(\mathbf{X})$ onto the feeding splash feature subspace, and the function $\text{norm}(g(\mathbf{X}))$ maps each deep feature vector in $g(\mathbf{X})$ to its $\ell_2$-norm. The matrix $\mathbf{M}$ represents the norm ratio of the projection of each deep feature vector in $g(\mathbf{X})$ onto the feeding splash feature subspace. Thus, the function $f_m(g(\mathbf{X})|\mathbf{B})$ defined as follows:

$$f_m(g(\mathbf{X})|\mathbf{B}) = \frac{\mathbf{M}}{\max(\mathbf{M})}, \tag{3}$$

can be used to suppress the output semantic map where the pattern of feeding splashes is unlikely to appear in the target environment. At last, another version of the detector can be defined as

$$f(\mathbf{x}) = f_m(g(\mathbf{X})|\mathbf{B}) \otimes \sum_{i=1}^{K} s_i f_i(g(\mathbf{X})). \tag{4}$$

## 4. Experimental Results

Eight semantic segmentation networks including FCN-8s [15], U-Net [18], SegNet [19], DeepLabv3 Plus [16], P-Net [17], FgSegNet [20], DANet [21], and SegFormer [24] were selected as baseline networks for performance comparisons. Table 1 shows the model size and complexity of those baseline models. Figure 2 shows five aquaculture environments for this study. Two hundred and twenty training examples and forty test examples were collected for each environment for this experiment. Training and test examples were obtained from different dates or feeding sessions.

**Table 1**: Comparisons of model size and complexity.

| Model Name | GFLOPS | Parameters |
|---|---|---|
| FCN-8s | 1,620 | 134,290,892 |
| SegNet | 2,490 | 29,469,058 |
| U-Net | 3,250 | 40,349,954 |
| P-Net | 469 | 122,306 |
| DeepLab | 588 | 11,852,610 |
| FgSegNet | 2,690 | 9,225,226 |
| DANet | 676 | 66,557,411 |
| SegFormer-MiT-B0 | 16 | 3,714,658 |
| SegFormer-MiT-B1 | 32 | 13,677,762 |
| SegFormer-MiT-B2 | 138 | 27,348,162 |

Figures 4 and 5 show the result of five-fold cross validation, where the backbone of SegFormer is MiT-B2. The following are observations on Figure 4.

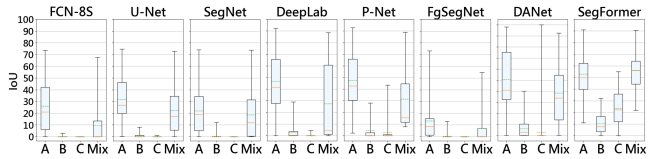
- From Figure 4, the detector dedicated to an environment is usually more accurate than the counterpart detector trained on examples of Envs. A, B, and C. However, the generalization of the detector dedicated to an environment is poor.
- The detectors trained on examples of Env. A, B, and C usually have larger variations than the counterpart detector dedicated to an environment.
- SegFormer-MiT-B2 trained on examples of Envs. A, B, and C is usually as accurate as the counterpart detector dedicated to Envs. A, B, or C.

From Figure 4, it is evident that as the number of different training aquaculture environments increases, the variation of the detector on test examples of these environments usually increases. The following are observations on Figure 5.

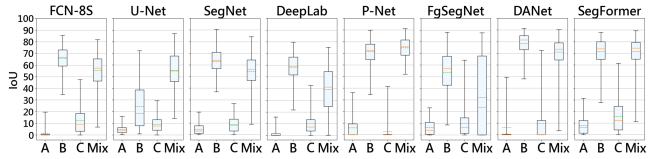
- Applying the detector trained on examples of Envs. A, B, and C to another Envs. D and E gives inaccurate detection results.

From Figure 5, including various training aquaculture environments is essential for developing a general feeding-splash detector.

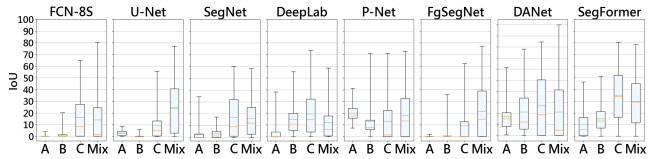
Figure 6 compares the experimental results of U-Net, DANet, SegFormer-B2, and the multi-branch SegFormer trained using the proposed task-clustering strategy, which were all trained on examples of Envs A, B, C, D, and E. As can be seen, the proposed detector is more accurate.




(a) Tested on Env. A




(b) Tested on Env. B




(c) Tested on Env. C

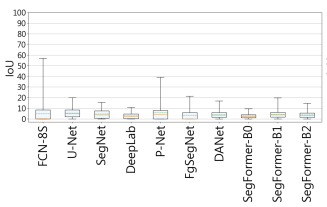
**Fig. 4**: Experimental results of models trained on Envs. A, B, and C and tested on Envs. A, B, and C, where A, B, and C denote detectors trained on the examples of Env. A, B, and C, respective, and Mix denotes the detector trained on the examples of Envs. A, B, and C.

Figure 7 shows the experimental results of the proposed detector with one-shot positive and negative bags of examples (Eq. (4)). The positive and negative bags include 15 images from a feeding session in the five environments, and the test set for each environment contains 40 images. Figure 8 shows examples of segmentation results. Experimental results show that the detector using Eq. (4), which utilizes one-shot positive and negative bags of examples to automatically combine multiple output branches, is feasible.

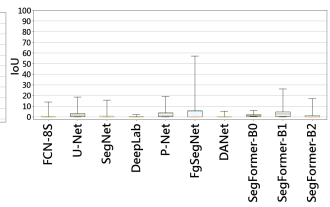
Finally, Figure 9 shows the frequency distribution of the branch number with respect to SegFormer with three backbones, and weaker backbones often allocate more output branches. Figure 10 shows the frequency of the five environments grouped together. For the backbones MiT-B0 and Mit-B2, Env. C, which is an aquaculture tank, is often assigned to a branch alone. For the MiT-B1 backbone, Env. C, along with either Env. B or Env. E, may be assigned to a branch, and Env. D is often assigned to a branch alone.

## 5. Conclusion

In this paper, a meta-learning approach to detecting feeding splashes in aquaculture environments has been proposed. The proposed detector is a semantic segmentation network with multiple output branches. Each




(a) Tested on Env. D




(b) Tested on Env. E

**Fig. 5**: Experimental results of models trained on examples of Envs. A, B, and C and tested on examples of Envs. D and E.

branch is aimed at specific kinds of aquaculture environments. Those branches can be combined for the target environment by one-shot positive and negative bags of examples without human annotations. Because the patterns of feeding splashes and background water waves in different aquaculture environments can be incompatible, the neural architecture with multiple output branches and the proposed training procedure with a task-clustering strategy, which can determine the number of branches and dispatch the training examples to appropriate branches, can facilitate the training of the detector. Experimental results have shown the feasibility of the proposed detector. In the future, the proposed detector will be enhanced to learn from more complex positive and negative bags of examples.

## Acknowledgement

The authors would like to thank National Science and Technology Council of Taiwan for financially supporting this research under Contract No. MOST 110-2221-E-019-049-MY3. The authors would also like to thank Fisheries Agency, Council of Agriculture of Taiwan for supporting this research and providing experiment environments under Contract No. 113FI-17.1.4-FA-02 and No. 113AS-6.1.1-FA-03.

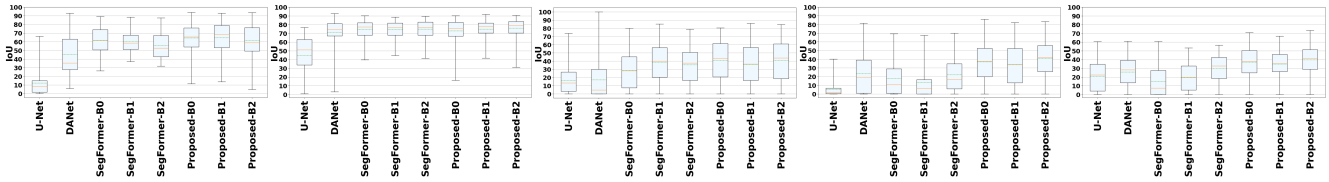
## REFERENCES

[1] M. A. M. Razman, A. P. P. A. Majeed, R. M. Musa, Z. Taha, G.-A. Susto, and Y. Mukai, *Machine Learning in Aquaculture: Hunger Classification of Lates calcarifer*. Springer Singapore, 2020.

[2] C. Zhou, D. Xu, K. Lin, C. Sun, and X. Yang, "Intelligent feeding control methods in aquaculture with an emphasis on fish: a review," *Reviews in Aquaculture*, 2017.

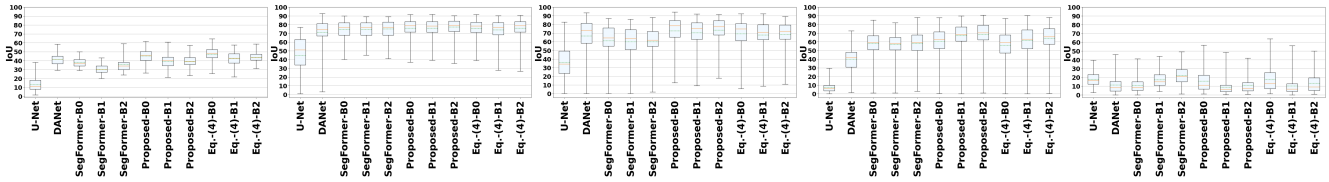
[3] L. Parra, L. García, S. Sendra, and J. Lloret, "The use of sensors for monitoring the feeding process and adjusting the feed supply velocity in fish farms," *Journal of Sensors*, pp. 1–14, 2018.

[4] Y. Wu, X. Wang, Y. Shi, Y. Wang, D. Qian, and Y. Jiang, "Fish feeding intensity assessment method us-




(a) Tested on Env. A (b) Tested on Env. B (c) Tested on Env. C (d) Tested on Env. D (e) Tested on Env. E

**Fig. 6**: Experimental results of models trained and tested on examples of Envs. A, B, C, D, and E, where Proposed-B0, -B1, and -B2 denote the multiple branches.




(a) Tested on Env. A (b) Tested on Env. B (c) Tested on Env. C (d) Tested on Env. D (e) Tested on Env. E

**Fig. 7**: Comparisons of the proposed method with one-shot positive and negative bags of examples (Eq. (4)).

ing deep learning-based analysis of feeding splashes," *Computers and Electronics in Agriculture*, vol. 221, p. 108995, 2024.

[5] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The Knowledge Engineering Review*, vol. 25, no. 1, pp. 1–25, 2010.

[6] C. Lemke, M. Budka, and B. Gabrys, "Metalearning: a survey of trends and technologies," *Artificial intelligence review*, vol. 44, no. 1, pp. 117–130, 2015.

[7] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5149–5169, 2022.

[8] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2022.

[9] M. Cui, X. Liu, H. Liu, J. Zhao, D. Li, and W. Wang, "Fish tracking, counting, and behaviour analysis in digital aquaculture: A comprehensive review," 2024. [Online]. Available: https://arxiv.org/abs/2406.17800

[10] Y. Zhang, C. Xu, R. Du, Q. Kong, D. Li, and C. Liu, "MSIF-MobileNetV3: An improved MobileNetV3 based on multi-scale information fusion for fish feeding behavior analysis," *Aquacultural Engineering*, vol. 102, AUG 2023.

[11] W.-C. Hu, L.-B. Chen, B.-K. Huang, and H.-M. Lin, "A computer vision-based intelligent fish feeding system using deep learning techniques for aquaculture," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 7185–7194, 2022.

[12] L. Zhang, Z. Liu, Y. Zheng, and B. Li, "Feeding intensity identification method for pond fish school using dual-label and MobileViT-SENet," *Biosystems Engineering*, vol. 241, pp. 113–128, MAY 2024.

[13] M. Lu, L. Fang, M. Li, B. Zhang, Y. Zhang, and P. Ghamisi, "NFANet: A novel method for weakly supervised water extraction from high-resolution remote-sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[14] B. Bovcon and M. Kristan, "WaSR–A water segmentation and refinement maritime obstacle detection network," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[15] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, 2018.

[17] G. Wang, M. Zuluaga, W. Li, R. Pratt, P. Patel, M. Aertsen, T. Doel, A. David, J. Deprest, S. Ourselin, and T. Vercauteren, "DeepIGeoS: A deep interactive geodesic framework for medical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1559–1573, 2019.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234–241.

[19] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

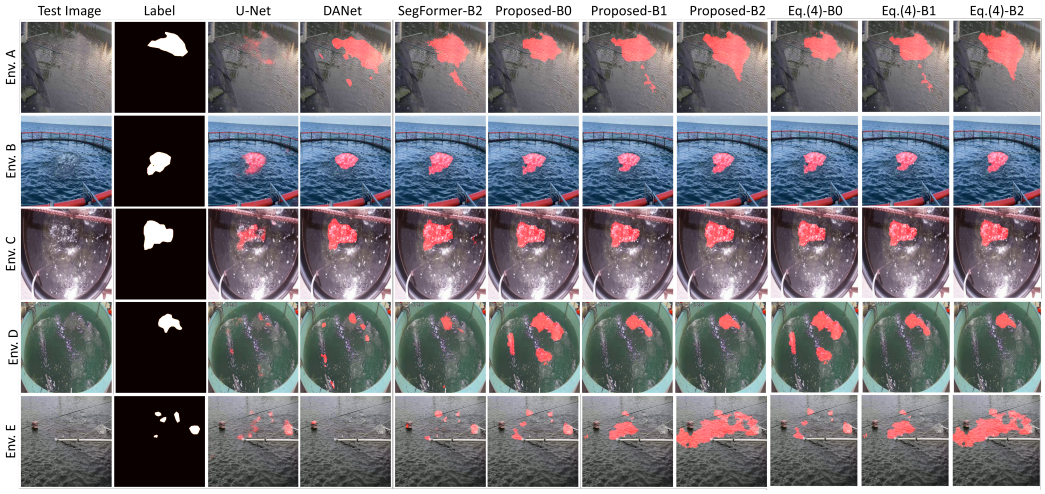



**Fig. 8**: Examples of segmentation results.

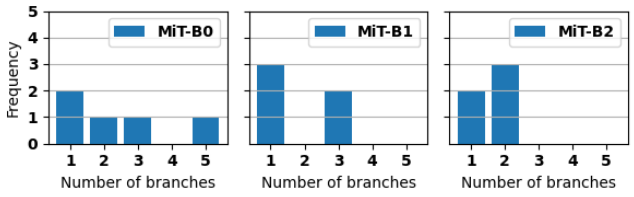



**Fig. 9**: Frequency distributions of the branch number with respect to the three backbones.

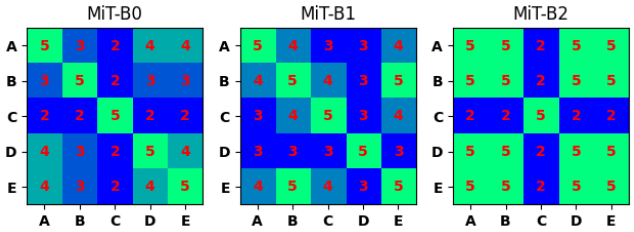



**Fig. 10**: Frequencies of Envs. A, B, C, D, and E in the same group with respect to the three backbones.


[20] L. A. Lim and H. Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, 2018.

[21] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3141–3149.

[22] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-m. Hu, "SegNeXt: Rethinking convolutional attention design for semantic segmentation," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 1140–1156.

[23] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6877–6886.

[24] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 12 077–12 090.

[25] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 1050–1065, 2022.

[26] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," *ACM Comput. Surv.*, vol. 55, no. 13s, jul 2023.

[27] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. B. Girshick, "Segment anything," *ArXiv*, vol. abs/2304.02643, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257952310

[28] W. Ren, Y. Tang, Q. Sun, C. Zhao, and Q.-L. Han, "Visual semantic segmentation based on few/zero-shot learning: An overview," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 5, pp. 1106–1126, 2024.

[29] J. Wu, X. Li, S. Xu, H. Yuan, H. Ding, Y. Yang, X. Li, J. Zhang, Y. Tong, X. Jiang, B. Ghanem, and D. Tao, "Towards open vocabulary learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 7, pp. 5092–5113, 2024.