



*Research article*

## **Recognition of adherent polychaetes on oysters and scallops using Microsoft Azure Custom Vision**

**Dong-hyeon Kim<sup>1</sup>, Se-woon Choe<sup>2</sup> and Sung-Uk Zhang<sup>1,\*</sup>**

<sup>1</sup> Digital Twin Laboratory, Dong-Eui University, 176 Eomgwangro, Busanjin-gu, Busan 47340, Korea

<sup>2</sup> Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Gyeongsangbukdo 39253, Korea

\* **Correspondence:** Email: [zsunguk@deu.ac.kr](mailto:zsunguk@deu.ac.kr); Tel: +82518901645.

**Abstract:** Oyster and scallop cultures have high growth rates in the Korean aquaculture industry. However, their production is declining because of the manual selection of polychaete-adherent oysters and scallops. In this study, an artificial intelligence model for automatic selection of polychaetes was developed using Microsoft Azure Custom Vision to improve the productivity of oysters and scallops. A camera booth was built to capture images of oysters and scallops from various angles. Polychaetes in the images were tagged. Transfer learning available with Custom Vision was performed on the acquired images. By repeating the training and evaluation, the number of training images was increased by analyzing the precision, recall, and mean average precision using the Compact [S1] and General [A1] domains of Custom Vision. This paper presents the artificial intelligence model developed for the automatic selection of polychaete-adherent oysters and scallops as well as the optimal model development method using Microsoft Azure Custom Vision.

**Keywords:** object detection; transfer learning; aquaculture; custom vision; Azure

---

### **1. Introduction**

The aquaculture industry has been one of the fastest-growing food production sectors in recent decades. In particular, oyster farming has the fourth highest proportion of production in the South Korean aquaculture industry, accounting for approximately 8% of the total and approximately 73% of the

shellfish aquaculture industry. In addition, scallop farming has the highest growth rate in the aquaculture industry, and its production in 2021 increased by approximately 10% compared to that in 2020 [1,2]. This study was conducted on oysters, which account for the largest proportion of the shellfish farming industry in Korea, and on scallops, which present a clear growth rate.

Although shellfish farming has been showing an overall growth, the following factors can hinder it. First, shellfish farming, such as oyster and scallop farming, requires significant manual labor [3]. A typical example is the selection operation. The most harmful infestation in oyster and scallop farming is that of polychaetes, which attach to oysters and scallops and create perforations. Polychaetes damage the commercial value of oysters and scallops and cause their death. In addition, excreta from polychaetes are one of the main causes of contamination in aquaculture farms, causing numerous shellfish deaths [4–6]. Therefore, a screening process for oysters and scallops with polychaetes is essential, and it is typically performed by manual visual inspection. Consequently, quality selection is nonuniform, and the production volume varies depending on the number and skill level of the workers. The second factor is the decline in the fishing population. The fisherman population in South Korea was approximately 159,000 in 2020, a decrease of approximately 39% compared to that in 2011. Moreover, the proportion of the fisherman population aged 65 or older was approximately 36% total fisherman population in 2020, suggesting that aging is progressing. Consequently, the available manpower for the aquaculture industry has been decreasing.

The high dependence on manual labor and the reduction in labor force have been consequently decreasing the production in the oyster and scallop farming industries. Moreover, increase in individual labor costs and unit price may be caused by the decrease in the number of workers, which may lead to a further decrease in the demand. This poses a threat to the sustainability of these industries.

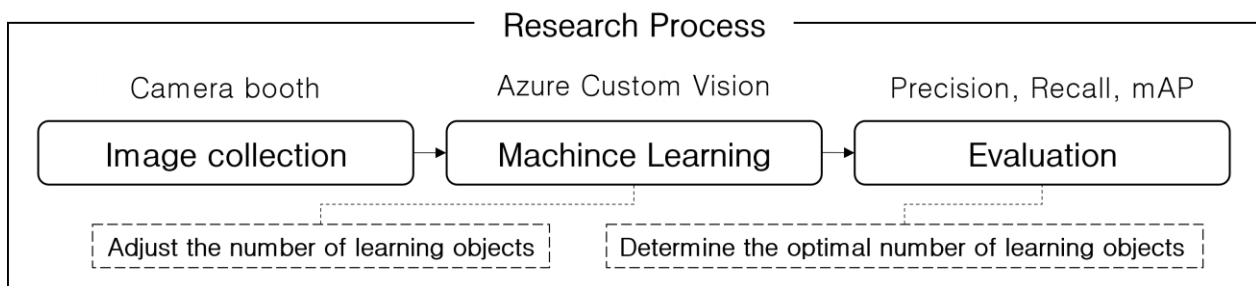
To solve this problem, simultaneously increasing productivity and sustainability is necessary by adopting an efficient production strategy [7]. As the aquaculture industry requires automation, the demand for artificial intelligence-based equipment is very high [8]. However, artificial intelligence technologies, such as machine learning, which can maximize industrial efficiency, are inappropriately utilized in the aquaculture industry [9]. Artificial intelligence and computer vision techniques are being studied for application to various aquaculture fields. Representatively, size classification for various fish and shellfish [10,11], seedling collection technique [12], gender classification [10,13], fish quality assessment [14], shell removal [15], behavioral pattern analysis [16,17], external feature extraction [10,18–21], and number of individuals [22] are being investigated. In this study, polychaetes, whose infestation is problematic in oyster and scallop farming in South Korea, were selected automatically using computer vision. For this purpose, an artificial intelligence model was developed utilizing the Microsoft Azure Custom Vision.

## 2. Methodology

This study aimed to develop an artificial intelligence model for the selection of oysters and scallops using Custom Vision from Microsoft Azure Cognitive Services. The research process is shown in Figure 1. A built camera booth is used to collect oyster and scallop object images, which are uploaded to Microsoft Azure Custom Vision. Machine learning is performed after labeling polychaetes in the uploaded images using the object recognition functionality of Custom Vision with a bounding box. The learned model determines the optimal hyperparameters by K-fold cross-validation to prevent overfitting, and a primary performance evaluation is performed using Custom Vision.

The trained model is published, and a secondary performance evaluation is performed based on the precision, recall, and mean average precision (mAP) on the images not used during training by loading these images in a prediction application. The prediction application was self-developed test software. Custom Vision is based on transfer learning, and accurate discrimination is possible even with a small amount of data; therefore, the model can be rapidly developed.

In this study, the number of images to obtain the optimal performance was determined. This was based on comparing the primary and secondary evaluation results according to the change in the number of training images, using Custom Vision to establish a model for polychaete recognition on oysters and scallops. The additionally constructed model was evaluated in terms of the performance index by domain change in Custom Vision.



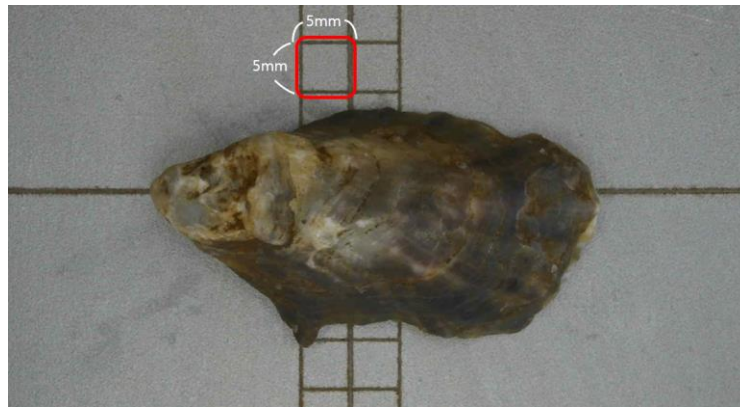
**Figure 1.** Research process.

### 2.1. Image collection of oyster and scallop objects using camera booth

Figure 2 shows an image of the interior of the camera booth. Three cameras are installed at angles of 180, 45, and 90°, respectively. Figure 3 shows an image of an oyster captured from the 90° position. The approximate size of this oyster is estimated using grids. Each grid has dimensions of  $5 \times 5 \text{ mm}^2$ . The grids act like a ruler to estimate the sizes of oysters and scallops with bare eyes.

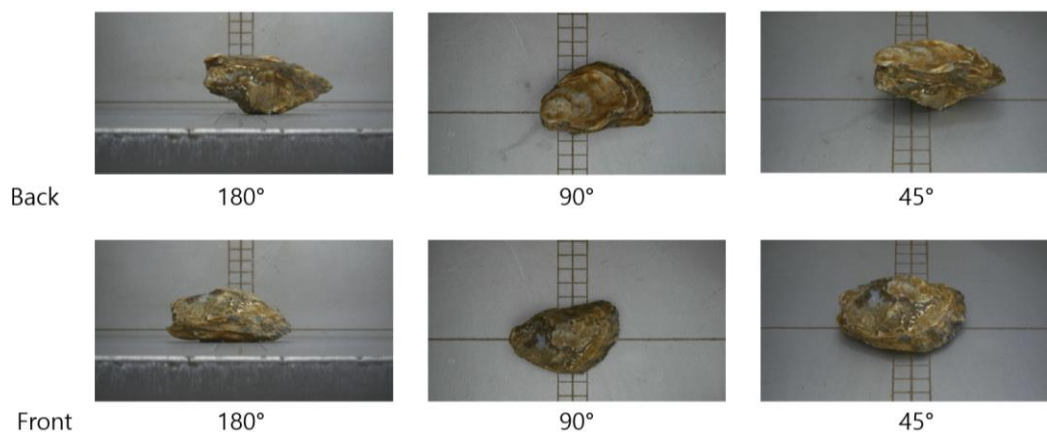


**Figure 2.** Inside of camera booth.

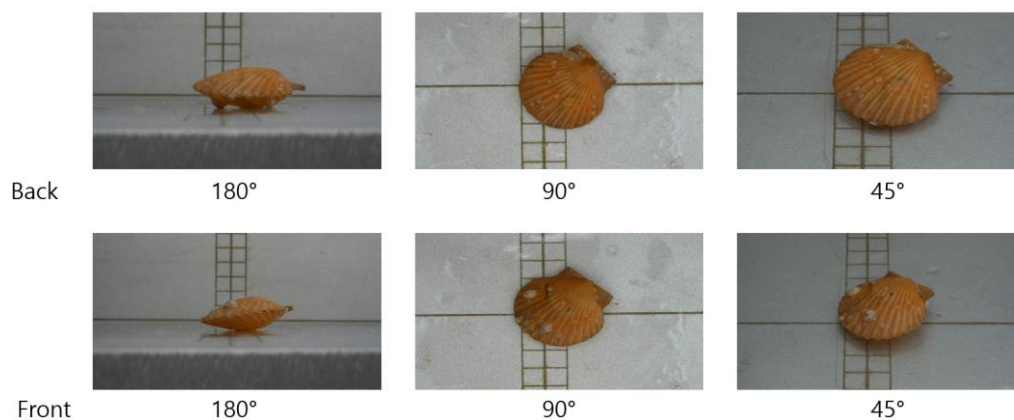


**Figure 3.** Image of oyster with grids.

For object recognition learning, images of oysters and scallops were captured using the camera booth, and a total of six images taken at 180, 45, and 90° from front and rear per oyster or scallop. Figures 4 and 5 show these images of an oyster and a scallop, respectively.



**Figure 4.** Six images of oyster with grids.



**Figure 5.** Six images of scallop.

Figures 6 and 7 show images of oysters and scallops, respectively, without and with attached polychaetes. The polychaetes in the images resemble white worms.



**Figure 6.** Images of oysters without and with attached polychaetes.



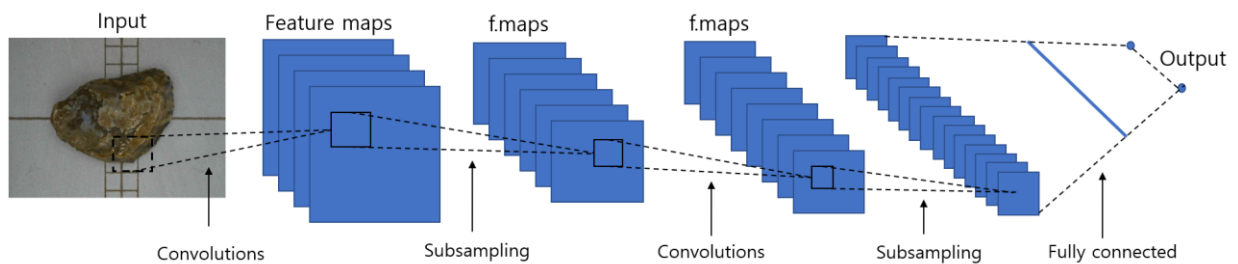
**Figure 7.** Images of scallop without and with attached polychaetes.

## 2.2. Machine learning with Microsoft Azure Custom Vision

### 2.2.1. Custom Vision

Custom Vision is an AI model development service among Microsoft Azure Cognitive Services, and it specializes in visual analysis. It categorizes stored images and defines tags within the images in conjunction with Microsoft Azure. It also uses transfer learning to recognize key differences between images and optimizes them to identify tagged objects rapidly. Transfer learning introduces a neural network structure that is trained in a specific field without modification, freezes the upper layer, and fine-tunes a part of the lower layer. In this process, the weights and biases derived from the upper layer are imported without modifications; however, they are readjusted to fit and apply to the new input data from the lower layer [23]. Therefore, fast and accurate learning results can be obtained by using a small amount of data during image recognition and classification. The layer located above the fine-tuned lower layer has a convolutional neural network (CNN) structure, which is used for labeling and learning [24]. Figure 8 shows an example CNN structure composed of three types of layers. Convolution layers, which are the first type, divide a two-dimensional picture into several small pictures and extract a feature map. The second type, i.e., subsampling layers extract the feature map to show the convolution feature map more simply and characteristically. The extracted feature values

are input into a multidimensional vector. The third type, i.e., fully connected layers, classify the features by matrix operations in space.

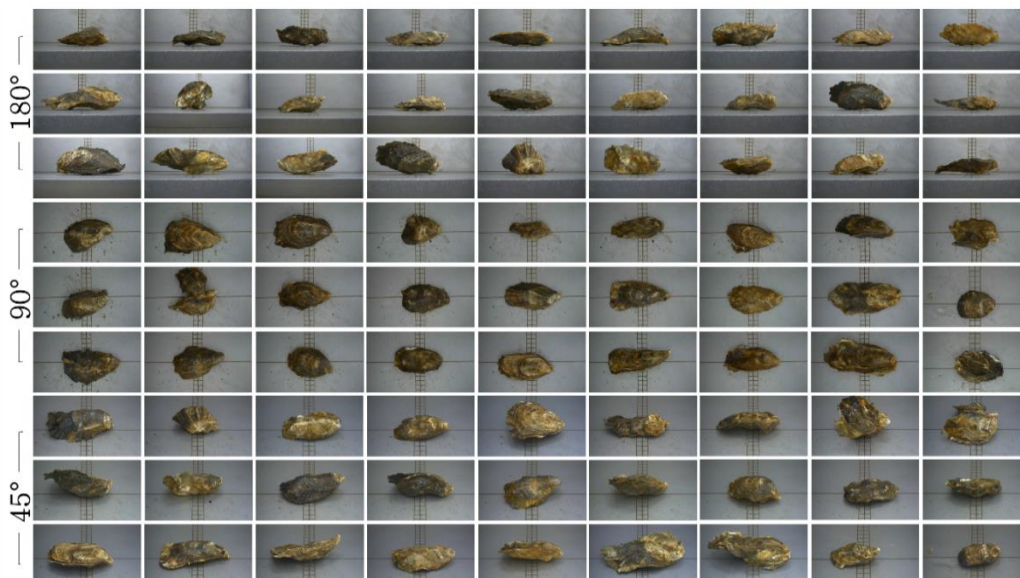


**Figure 8.** CNN structure for deep learning-based image recognition training.

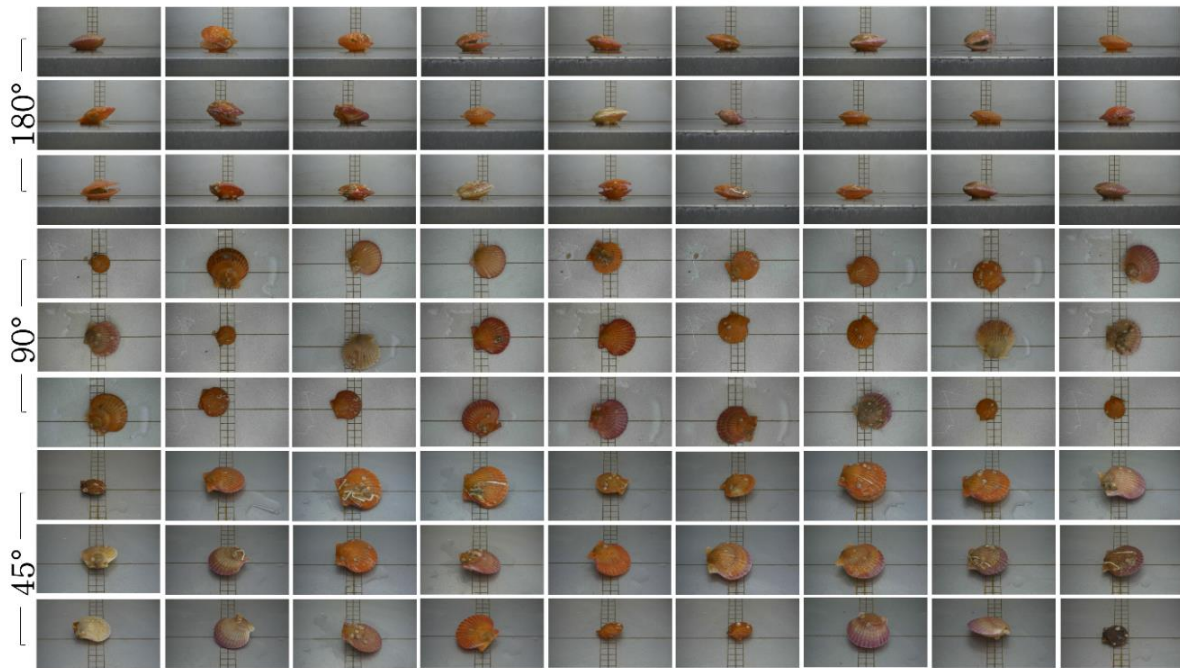
Custom Vision in Microsoft Azure Cognitive Services, which specializes in visual analysis, could simplify the development process required for training and model creation in this study.

### 2.2.2. Image tagging

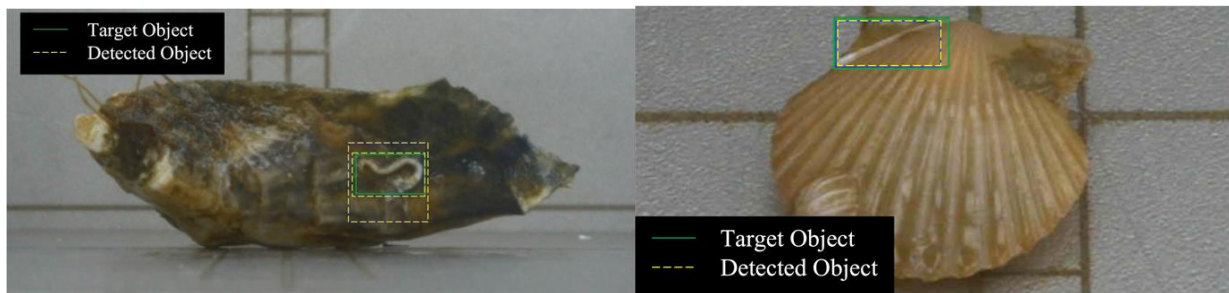
Figures 9 and 10 show examples of uploaded shellfish images. In Custom Vision, information is saved by tagging one or more objects per uploaded image, as shown in Figure 11. A bounding box is constructed around a tagged object, and the recognition probability is predicted by comparing this constructed bounding box with that of the detected object. Accordingly, it continuously learns and extracts features of a photo such that the features become invariant.



**Figure 9.** Image data of oysters.



**Figure 10.** Image data of scallops.



**Figure 11.** Oyster (left) and scallop (right) object detection.

### 2.2.3. Learning environment and conditions

Table 1 provides the details of the learning environments and conditions used in this study. Custom Vision is used as the learning environment, and it selects the learning type and domain to train the optimized model for a desired result. Project types in Custom Vision include classification and object detection, and in this study, the latter is used to identify polychaetes on oysters and scallops. The targets are polychaetes, and two types of learning are advanced and quick training. Advanced training is used when high accuracy is required, and because the computing time is long, it is used for final deployment in terms of time and cost. Quick training works well with many good samples and is a mode optimized for computing speed; therefore, it is mainly used when evaluating and improving models. In this study, to derive the optimal number of learning pictures, quick training is performed.

**Table 1.** Learning environment.

Platform	Custom Vision
Project Type	Object Detection
Target	Polychaeta (on oysters and scallops)
Training Type	Quick Training

Domain in Custom Vision is selectively used according to the learning object and the desired outcome. Table 2 summarizes the aims of different object recognition models as defined by Microsoft according to the domain. In this study, the Compact [S1] domain was selected assuming that it was mounted on an edge device, and the general General [A1] domain was used for performance comparison.

**Table 2.** Domain types for object recognition models.

Domain	Description
General	Use extensively when a suitable domain is unavailable or there is ambiguity in choosing the domain.
General [A1]	Similar to General, but requires a longer computing time, and the mAP fluctuates with an error of 1% on the same training data. More complex, used when accuracy is required.
Logo	Optimized for finding brand logos in images
Products on shelves	Optimized for detecting and sorting products on shelves
Compact domain	Optimized for constraints of real-time object detection on edge devices (requires a postprocessing logic)
Compact domain [S1]	Optimized for constraints of real-time object detection on edge devices (no postprocessing logic is required)

Table 3 defines the test types as Compact\_CV, Compact\_SW, General\_CV, and General\_SW according to the domain and the test method. Compact\_CV and General\_CV are evaluated by K-fold cross-validation, and Compact\_SW and General\_SW are evaluated by loading published models into self-developed software.

**Table 3.** Test types.

Test Type		Domain	
		Compact [S1]	General [A1]
Test Method	K-fold cross-validation	Compact_CV	General_CV
	Model published with software	Compact_SW	General_SW

Learning conditions are divided according to the number of learning images, which is a variable, and in this study, the number of learned images to obtain the optimal mAP according to the number of learning images was 50, which is the minimum number recommended by Microsoft. In the case of the Compact [S1]. domain, 41 conditions were configured and used to train up to 1050 images in increments of 25. In the case of the General [A1] domain for performance comparison, 50–300 images

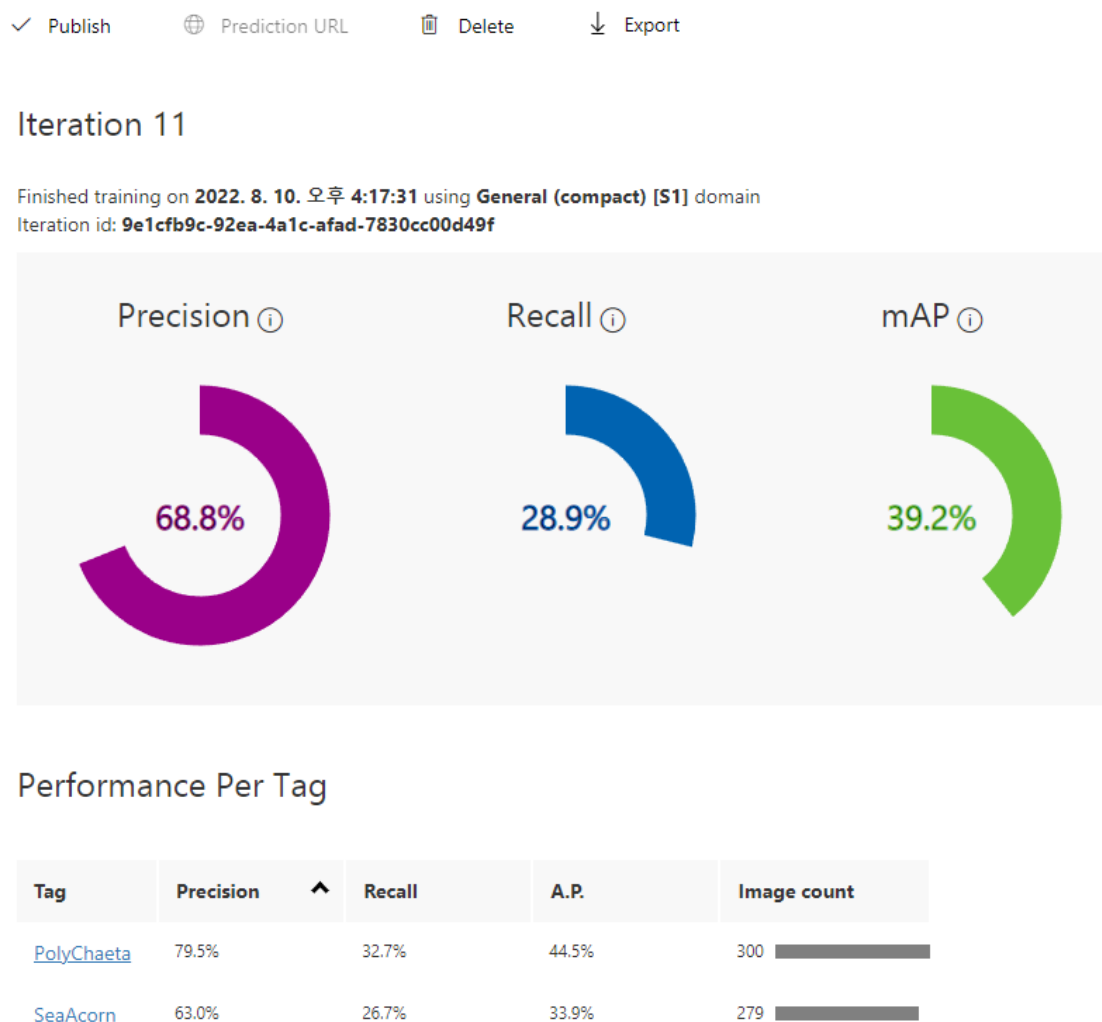


were trained in increments of 25 considering the learning time and amount of use, whereas 300–1000 images were trained in increments of 100. The evaluation condition was evaluated by designating 100 pictures that were not used during learning and loading them into the self-developed software. The evaluation condition was applied only to Compact\_SW and General\_SW.

### 2.3. Performance evaluation

The performance evaluation was performed twice, which are called primary and secondary performance evaluations. The primary performance evaluation involved applying K-fold cross-validation to Microsoft Azure Custom Vision, and it was defined as the first test. The secondary performance evaluation published the model trained in Custom Vision, called it from the prediction application, output the prediction performance, and defined it as the second test.

#### 2.3.1. Primary performance evaluation by K-fold cross-validation

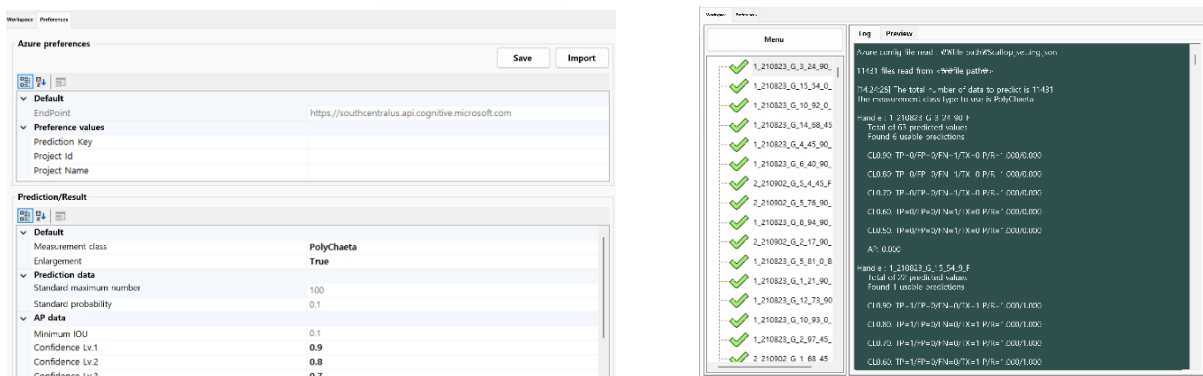


**Figure 12.** Example of first test output.

In the first test, K-fold cross-validation was used, which is a method for preventing overfitting and for self-performance evaluation in the learning process. The K-fold cross-validation results were obtained in the process of learning the model in Custom Vision. The number of groups is set according to the K value of K-fold cross-validation, and in Custom Vision, the K value was defined as 5. In detail, the uploaded data for learning were divided into five groups; K-1–4 were used for learning, and K-5 was used for hyperparameter tuning and performance evaluation. K-fold cross-validation prevents overfitting of the learned data by adjusting the thresholds for outliers and low-contribution values by a hyperparameter tuning process. Therefore, the object detection performance is similar for learned and new unlearned data, and the first test results output in terms of the precision, recall, and mAP are shown in Figure 12.

### 2.3.2. Secondary performance evaluation using published model

The second test called the published model from the prediction application and evaluated the model based on the prediction performance. In detail, the values as shown in Figure 13(a) are set, and the prediction key, prediction Id, prediction name, and endpoint of the model published in the prediction application are input.



(a) Prediction application setting

(b) Prediction results

**Figure 13.** Example of second test setup and output.

### 2.3.3. Performance evaluation indicators

In this study, precision, recall, and mAP were used as the model performance evaluation indicators. mAP is a useful evaluation indicator for measuring the accuracy of object recognition, and it can be obtained by drawing a precision–recall curve.

Based on the confusion matrix provided in Table 4, the precision and recall are as follows. Precision is the ratio of True\_Positive to those classified as True\_Positive and False\_Positive by the trained model. Recall is the percentage predicted as True\_Positive among those classified as True\_Positive and False\_Negative by the trained model. Type I error is a false positive in precision, whereas Type II error is a false negative in recall. Therefore, precision and recall can be expressed as in Eqs (1) and (2), respectively.

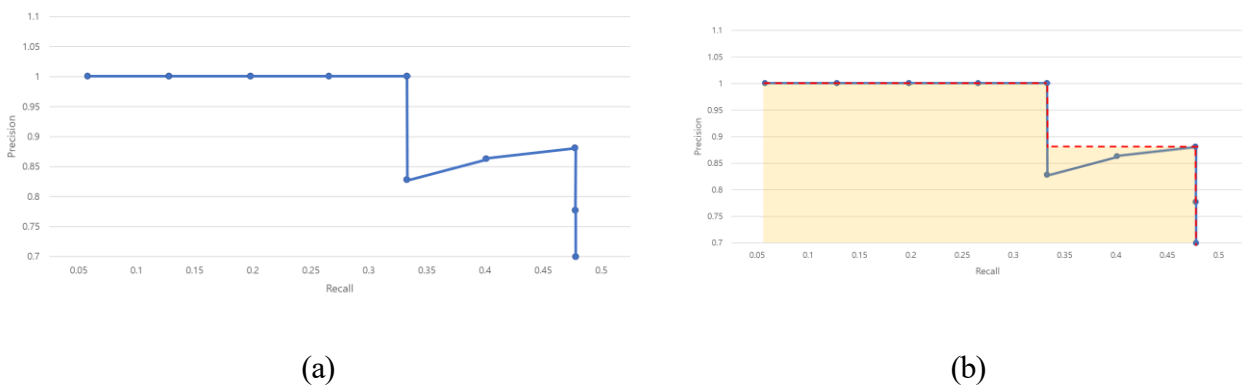
**Table 4.** Confusion matrix.

Confusion Matrix		Predicted	
		Positive (Detection)	Negative (Non-Detection)
Actual	Positive (Detection)	True_Positive	False_Negative (Type II error)
	Negative (Non-Detection)	False_Positive (Type I error)	True_Negative

$$Precision = \frac{True\_Positive}{True\_Positive + False\_Positive} \quad (1)$$

$$Recall = \frac{Ture\_Positive}{True\_Positive + False\_Negative} \quad (2)$$

A precision–recall curve can evaluate the detection performance based on the changes in the precision and recall values with the change in the threshold value. It is drawn as a two-dimensional graph, as shown in Figure 14(a), and its area is the average precision (AP), as shown in Figure14(b).



**Figure 14.** (a) Precision–recall curve and (b) Average precision from precision–recall curve.

The mAP is obtained by dividing the total number of APs of N object types by N, as expressed in Eq (3). By obtaining these two values, the performance of an object recognition algorithm can be quantitatively evaluated, and the mAP can be defined as the average of the detection target class.

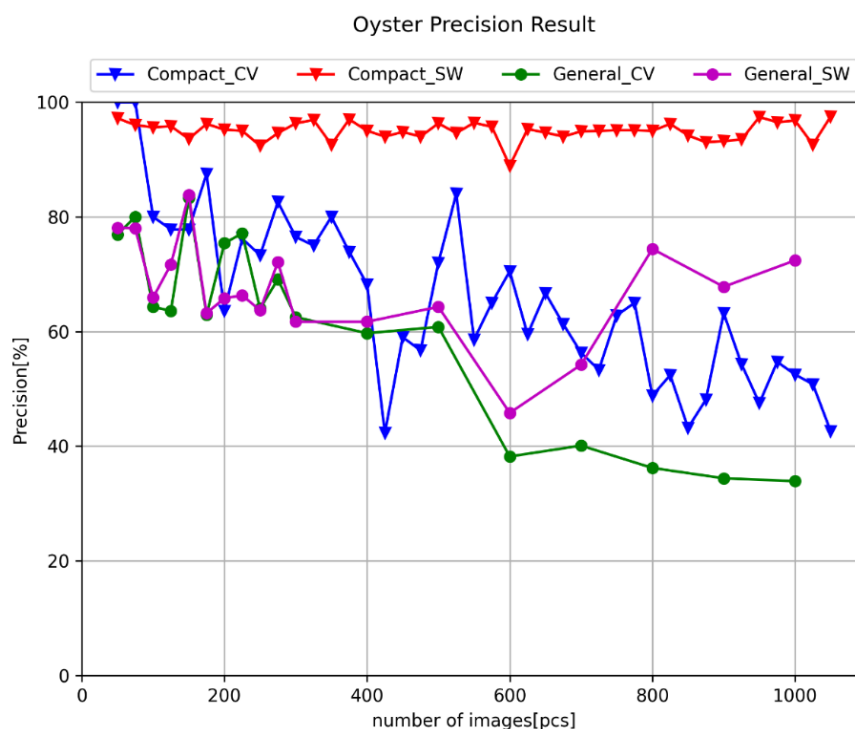
$$mAP = \frac{1}{N} \sum_{i=1}^n AP_i \quad (3)$$

### 3. Results and discussion

Images of individual oysters and scallops were used in the study. The minimum number of learning images in the Compact [S1] domain was 50 according to the recommended conditions of

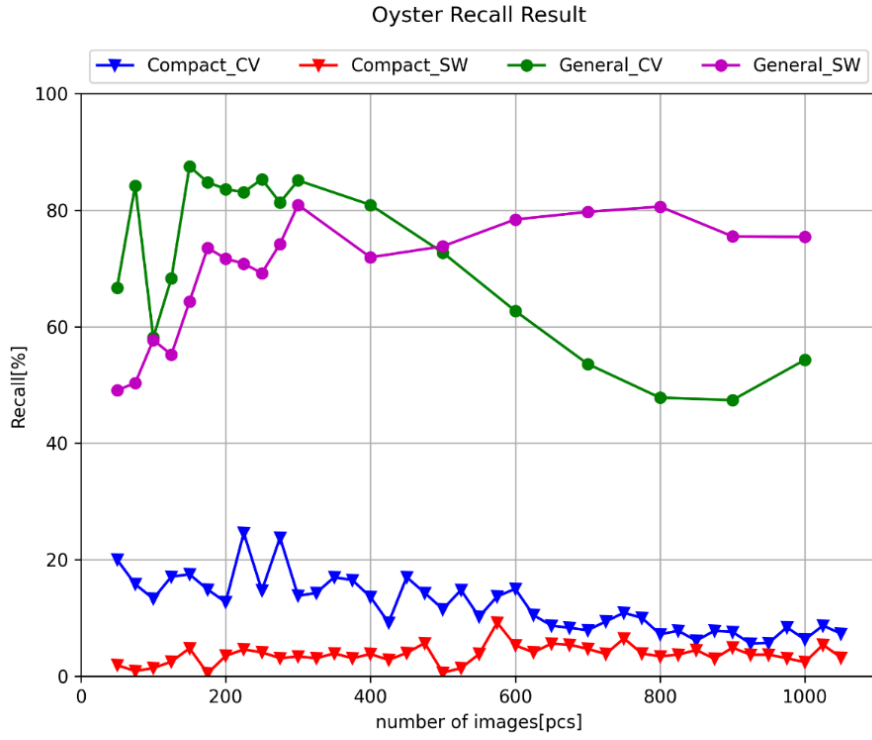
Custom Vision, which was increased to 1050 in increments of 25, and a total of 41 conditions were studied. In the General [S1] domain, the number of learning pictures was increased from 50 to 300 in increments of 25, and learning was performed by increasing the number of images from 300 to 1000 in increments of 100. The 100 test images that were not used for learning were only used in Compact\_SW and General\_SW. For performance evaluation, the Compact\_CV and Compact\_SW results were output as precision, recall, and mAP values under each condition in the Compact [S1] domain, and the General\_CV and General\_SW results were output as precision, recall, and mAP values for each condition in the General [S1] domain. For Compact\_CV and General\_CV, the K-fold cross-validation ( $K = 5$ ) provided by Custom Vision was used, and Compact\_SW and General\_SW were evaluated by inputting the model published in Custom Vision into the self-produced software.

Figure 15(a)–(c) shows the precision, recall, and mAP results for each condition of oysters according to the number of learned pictures. Several spikes are observed in the graphs. In the case of the compact domain, more spikes are detected than in the general domain. Since the compact domain was developed for real-time discrimination, the size of the learning model is relatively small. As a result, the indicators of the model change sensitively according to the amount of learning. And the fewer the number of images, the more spikes appear. It appears that the smaller the number of images, the more sensitively the quality of the learned image appears.

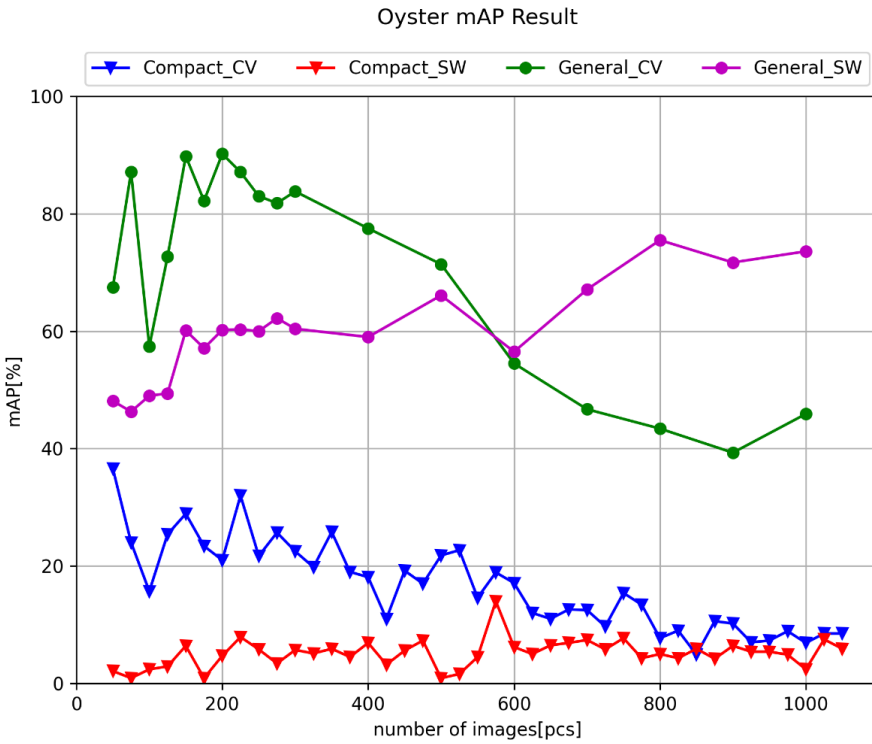


(a)

*Continued on next page*



(b)



(c)

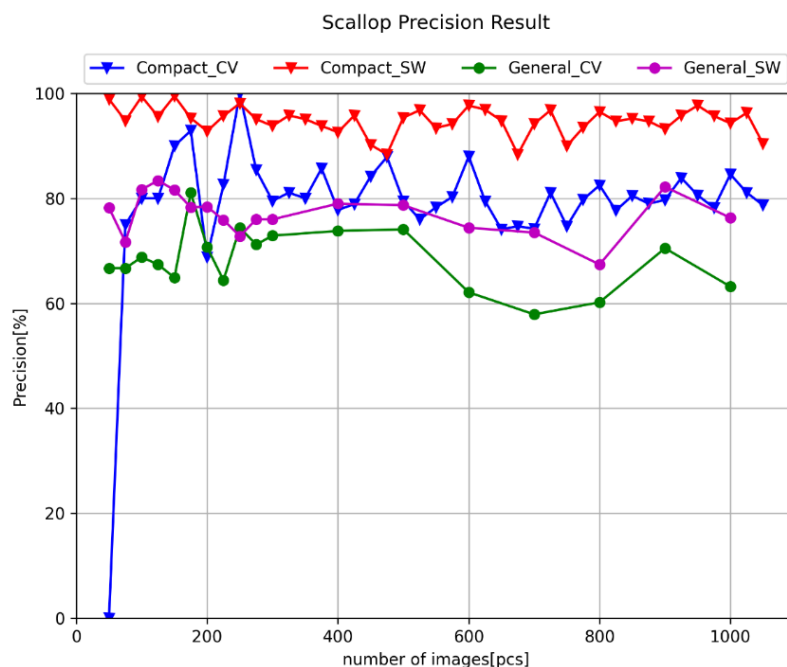
Figure 15. (a) Precision, (b) Recall, (c) and mAP vs. number of oyster images.

By oyster learning, in the precision results, Compact\_CV shows large fluctuations with an average of 65.5% and a standard deviation of 14.6%, and a continuously declining performance. In contrast, Compact\_SW maintains a consistently high performance with an average of 95.0% and a standard deviation of 1.7%. General\_CV shows an average of 60.1% and a standard deviation of 16.6%, and General\_SW achieves an average of 67.3% and a standard deviation of 9.0%.

In the recall results, Compact\_CV and Compact\_SW learning in the Compact [S1] domain present consistently low performances with averages of 12.2 and 3.7% and standard deviations of 4.7 and 1.6%, respectively. General\_CV and General\_SW learning in the General [S1] domain show averages of 71.5 and 69.6%, and standard deviations of 14.3, and 10.1%, respectively. Although the General [S1] domain has larger deviations than the Compact [S1] domain, it has a higher performance than the latter. Specifically, Type II error, which predicts that an individual with polychaetes does not have polychaetes, appears with a higher probability in Compact [S1] than in General [S1].

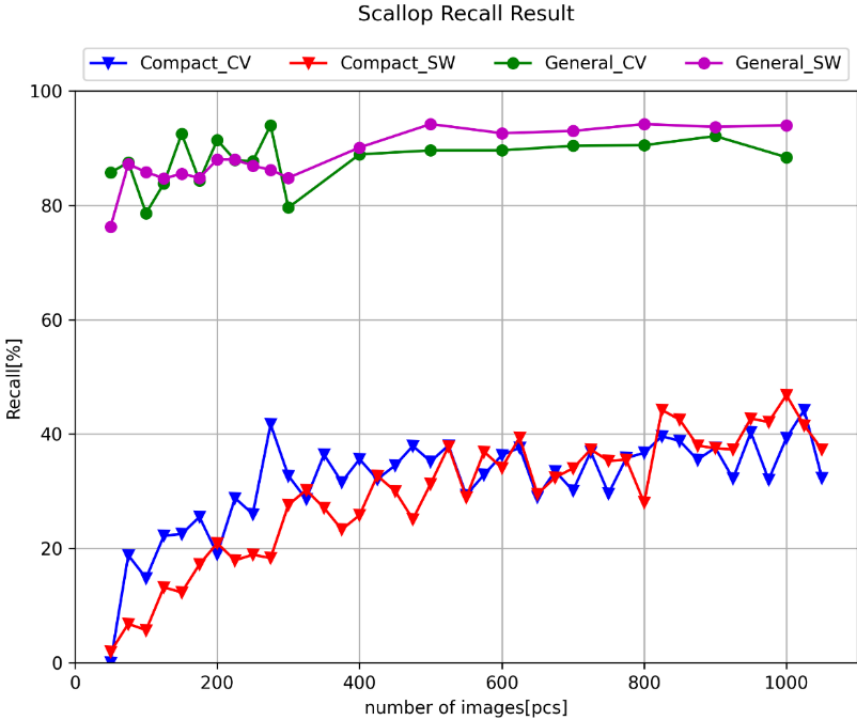
In the mAP results, Compact\_CV and Compact\_SW present averages of 16.5 and 5.1% and standard deviations of 7.6 and 2.4%, respectively. In contrast, General\_CV and General\_SW show averages of 70.1 and 60.1%, and standard deviations of 17.6 and 8.5%, respectively. Therefore, although General [S1] shows larger deviations than Compact [S1], the former has a higher accuracy. Characteristically, in both domains, the cross-validation performance significantly decreases significantly after learning of 500 images. In summary, as more images are learned, because of the complex shapes of oysters, the distinction between polychaetes and shell shapes becomes unclear, and the classification accuracy decreases.

Figure 16 shows the results of Compact\_CV, Compact\_SW, General\_CV, and General\_SW for scallops according to the number of learned pictures. Figure 16(a)–(c) presents the precision, recall, and mAP, respectively.

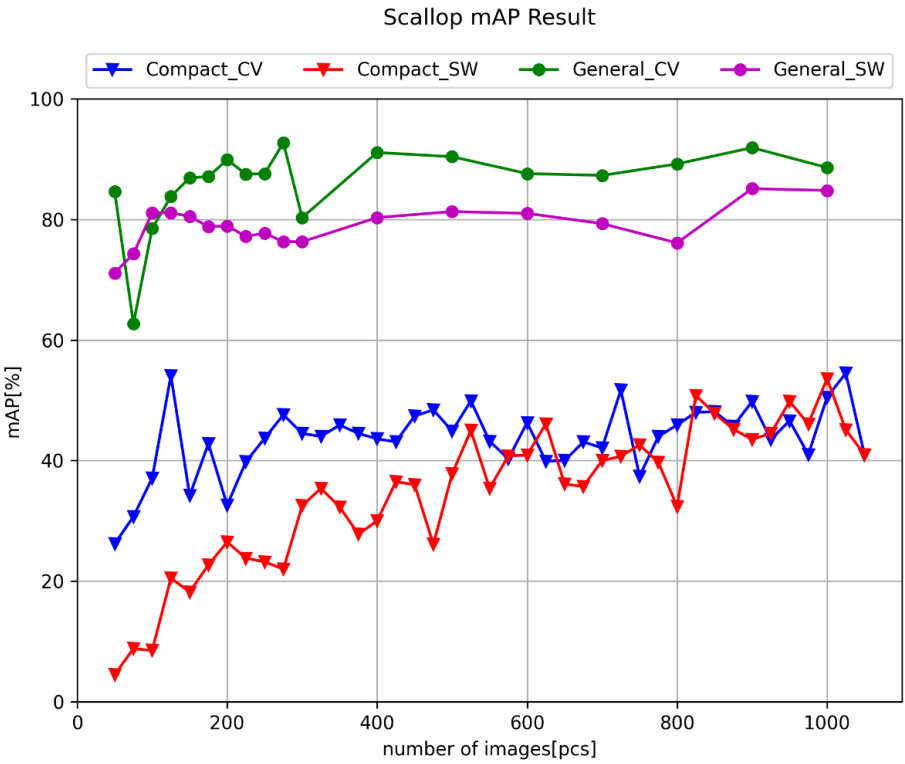


(a)

*Continued on next page*



(b)



(c)

Figure 16. (a) Precision, (b) Recall, (c) mAP vs. number of scallop images.

As a result of learning the scallop images, in the precision results, Compact\_CV shows an average of 78.9% and a standard deviation of 13.8%; however, its performance remains relatively constant after 300 images are learned. Compact\_SW maintains a consistently high performance, with an average of 94.8% and a standard deviation of 2.6%. General\_CV achieves a consistent performance with an average of 68.4% and standard deviation of 5.8%, and General\_SW shows an average of 77.0% and a standard deviation of 4.1%.

In the recall results, Compact\_CV and Compact\_SW on learning in the Compact [S1] domain show averages of 31.7 and 29.4% and standard deviations of 8.2 and 11.1%, respectively, and the accuracy of Compact [S1] gradually increases. General\_CV and General\_SW learning in the General [A1] domain achieve averages of 87.9 and 88.3% and standard deviation of 5.8 and 4.1% respectively. In the General [A1] domain, the deviations are smaller and the performances are higher than in the Compact[S1] domain. Thus, the Type II error is observed, similar to in the case of oysters, and Compact [S1] has a higher probability than General [A1]. However, differently, in the case of scallops, the accuracy of the Compact [S1] domain gradually increases.

In the mAP results, Compact\_CV and Compact\_SW show averages of 43.6 and 34.5% and standard deviations of 5.9 and 11.8%, respectively. In both domains, the increase in accuracy is relatively large before the number of training images reaches 300; however, after 300 images are learned, a constant performance is observed.

Thus, identification of polychaetes in the scallop images was clearer than in the oyster images; therefore, after learning 300 images, the increase in accuracy in case of scallops is not large, instead it gradually increases. In the case of oysters, where the identification of polychaetes in the images is unclear, the performance increases until 500 images are learned, following which the performance significantly decreases. Therefore, for images whose discrimination is unclear, if the appropriate number of learning images is exceeded, the classification boundary becomes vague; therefore, appropriately selecting this number is necessary. To appropriately select it, a suitable experimental method is required, which is related to the model development cost. In this study, the General [A1] domain, which used in the most general scenarios, and the Compact [S1] domain, which is a simplified domain, were used, and the learning time and cost were compared. When learning lesser than 1000 images, on average, approximately 15 min and 45 min per learning point were required for the Compact [S1] and General [A1] domains, respectively, under the same conditions. Converting these into fees, Compact [S1] required \$ 2.5 and General [A1] \$ 7.5 per learning point. Accordingly, to minimize the program development cost, it is appropriate to find an appropriate learning value using the Compact [S1] domain and make detailed adjustments using the General [A1] domain.

#### 4. Conclusions

This study was conducted to alleviate the high proportion of manual labor required in the selection of good products, which is a factors hindering the productivity of the oyster and scallop farming industries in South Korea. Using Microsoft Azure Custom Vision, an artificial intelligence model was built for the automatic selection of polychaetes on oysters and scallops. Learning and evaluation were repeatedly performed according to the increase in the number of training images for both the Compact [S1] and General [A1] domain. The results showed that the performance on oysters with complex shapes decreased and difficulties in identifying polychaetes increased as the number of training images increased. However, in case of scallops, which are relatively easy to discriminate and have simple



shapes, these domains maintained their performance after a certain number of performance improvements. To improve the detection performance using Custom Vision, delicate labeling is required for images of relatively complex shapes. Therefore, to minimize the development cost, finding a section where the performance change is not large by maximizing the Compact [S1] domain is necessary. the development cost is anticipated to be relatively minimized if the General [A1] domain is used for development in the section where the performance change is relatively constant for the Compact [S1] domain.

The polychaete identification model developed in this study for oysters and scallops can alleviate the manual dependence during quality selection. It is expected to contribute to the improvement of productivity in the oyster and scallop farming industries. This study is also expected serve as a reference for performance improvement and cost reduction using Microsoft Azure Custom Vision.

## Acknowledgments

We would like to thank Seabank Co., Ltd., for supporting our project. This research was supported by the Korea Basic Science Institute (National Research Facilities and Equipment Center) grant funded by the Ministry of Education (No. 2019R1A6C1010045).

## Conflict of interest

The authors declare there is no conflict of interest.

## References

1. Y. H. Park, M. S. Do, S. W. Rho, Development direction of individual oyster aquaculture industry in Korea, *J. Fish. Mar. Sci. Educ.*, **30** (2018), 913–922. <https://doi.org/10.13000/JFMSE.2018.06.30.3.913>
2. Y. D. Kim, C. Lee, G. S. Kim, M. Park, Y. C. Park, Y. S. Kim, et al., A study on argopecten irradians aquaculture in the north east sea regions, *Korean J. Malacol.*, **32** (2016), 279–287. <https://doi.org/10.9710/kjm.2016.32.4.279>
3. H. Hong, X. Yang, Z. You, F. Cheng, Visual quality detection of aquatic products using machine vision, *Aquac. Eng.*, **63** (2014), 62–71. <https://doi.org/10.1016/j.aquaeng.2014.10.003>
4. W. Sato-Okoshi, H. Abe, Morphological and molecular sequence analysis of the harmful shell boring species of polydora (Polychaeta: Spionidae) from Japan and Australia, *Aquaculture*, **368–369** (2012), 40–47. <https://doi.org/10.1016/j.aquaculture.2012.08.046>
5. W. Sato-Okoshi, K. Okoshi, B. S. Koh, Y. H. Kim, J. S. Hong, Polydorid species (Polychaeta: Spionidae) associated with commercially important mollusk shells in Korean waters, *Aquaculture*, **350–353** (2012), 82–90. <https://doi.org/10.1016/j.aquaculture.2012.04.013>
6. W. Sato-Okoshi, K. Okoshi, H. Abe, J. Y. Li, Polydorid species (Polychaeta, Spionidae) associated with commercially important mollusk shells from eastern China, *Aquaculture*, **406–407** (2013), 153–159 <https://doi.org/10.1016/j.aquaculture.2013.05.017>
7. A. L. T. Novaes, G. J. P. O. de Andrade, A. dos S. Alonço, A. R. M. Magalhães, Operational performance in aquaculture: A case study of the manual harvesting of cultivated mussels, *Aquac. Eng.*, **84** (2019), 67–79. <https://doi.org/10.1016/j.aquaeng.2018.12.006>

8. Y. Pyeon, Y. Kim, D. Kim, W. Oh, I. Han, K. Lee, Development of an automatic assembly machine for oyster farm lines, *J. Inst. Control. Robot. Syst.*, **24** (2018), 111–115. <https://doi.org/10.5302/J.ICROS.2018.17.0219>
9. C. A. Graham, H. Shamkhalichenar, V. E. Browning, V. J. Byrd, Y. Liu, M. T. Gutierrez-Wing, et al., A practical evaluation of machine learning for classification of ultrasound images of ovarian development in channel catfish (*Ictalurus punctatus*), *Aquaculture*, **552** (2022), 738039. <https://doi.org/10.1016/j.aquaculture.2022.738039>
10. C. Costa, F. Antonucci, C. Boglione, P. Menesatti, M. Vandeputte, B. Chatain, Automated sorting for size, sex and skeletal anomalies of cultured seabass using external shape analysis, *Aquac. Eng.*, **52** (2013), 58–64. <https://doi.org/10.1016/J.AQUAENG.2012.09.001>
11. A. Lapico, M. Sankupellay, L. Cianciullo, T. Myers, D. A. Konovalov, D. R. Jerry, et al., Using image processing to automatically measure pearl oyster size for selective breeding, in *2019 Digital Image Computing: Techniques and Applications (DICTA)*, 2019. <https://doi.org/10.1109/DICTA47822.2019.8945902>
12. S. Kakehi, T. Sekiuchi, H. Ito, S. Ueno, Y. Takeuchi, K. Suzuki, et al., Identification and counting of Pacific oyster *Crassostrea gigas* larvae by object detection using deep learning, *Aquac. Eng.*, **95** (2021), 102197. <https://doi.org/10.1016/J.AQUAENG.2021.102197>
13. B. Zion, V. Alchanatis, V. Ostrovsky, A. Barki, I. Karplus, Classification of guppies' (*Poecilia reticulata*) gender by computer vision, *Aquac. Eng.*, **38** (2008), 97–104. <https://doi.org/10.1016/J.AQUAENG.2008.01.002>
14. M. Dowlati, M. de la Guardia, M. Dowlati, S. S. Mohtasebi, Application of machine-vision techniques to fish-quality assessment, *TrAC Trends Analyt. Chem.*, **40** (2012), 168–179. <https://doi.org/10.1016/J.TRAC.2012.07.011>
15. N. E. Little, O. H. Smith, F. W. Wheaton, M. A. Little, Automated oyster shucking: Part II. Computer vision and control system for an automated oyster orienting device, *Aquac. Eng.*, **37** (2007), 35–43. <https://doi.org/10.1016/J.AQUAENG.2006.12.007>
16. D. Li, G. Wang, L. Du, Y. Zheng, Z. Wang, Recent advances in intelligent recognition methods for fish stress behavior, *Aquac. Eng.*, **96** (2022), 102222. <https://doi.org/10.1016/J.AQUAENG.2021.102222>
17. Z. Liu, X. Li, L. Fan, H. Lu, L. Liu, Y. Liu, Measuring feeding activity of fish in RAS using computer vision, *Aquac. Eng.*, **60** (2014) 20–27. <https://doi.org/10.1016/J.AQUAENG.2014.03.005>
18. H. M. Lalabadi, M. Sadeghi, S. A. Mireei, Fish freshness categorization from eyes and gills color features using multi-class artificial neural network and support vector machines, *Aquac. Eng.*, **90** (2020), 102076. <https://doi.org/10.1016/J.AQUAENG.2020.102076>
19. G. Xiong, D. J. Lee, K. R. Moon, R. M. Lane, Shape similarity measure using turn angle cross-correlation for oyster quality evaluation, *J. Food Eng.*, **100** (2010), 178–186. <https://doi.org/10.1016/J.JFOODENG.2010.03.043>
20. A. Banan, A. Nasiri, A. Taheri-Garavand, Deep learning-based appearance features extraction for automated carp species identification, *Aquac. Eng.*, **89** (2020), 102053. <https://doi.org/10.1016/J.AQUAENG.2020.102053>
21. S. S. Chen, F. W. Wheaton, Oyster hinge line detection using image processing, *Aquac. Eng.*, **8** (1989), 307–327. [https://doi.org/10.1016/0144-8609\(89\)90038-1](https://doi.org/10.1016/0144-8609(89)90038-1)

22. C. S. Costa, V. A. G. Zanoni, L. R. V. Curvo, M. de Araújo Carvalho, W. R. Boscolo, A. Signor, et al., Deep learning applied in fish reproduction for counting larvae in images captured by smartphone, *Aquac. Eng.*, **97** (2022), 102225. <https://doi.org/10.1016/J.AQUAENG.2022.102225>
23. C. Yang, C. Liu, C. Tan, F. Sun, T. Kong, W. Zhang, A survey on deep transfer learning, in *International Conference on Artificial Neural Networks*, 2018. [https://doi.org/10.1007/978-3-030-01424-7\\_27](https://doi.org/10.1007/978-3-030-01424-7_27)
24. M. Pejčinović, A review of custom vision service for facilitating an image classification, in *Proceedings of the Central European Conference on Information and Intelligent Systems*, (2019), 1–13. Available from: <https://www.proquest.com/openview/c1b73d7326a4d300905497cf6972c227/1?pq-origsite=gscholar&cbl=1986354>.



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)