Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **9**, Number **4**, November **2024**

Application of Convolutional Block Attention Module in Marine Biodiversity Research Using Convolutional Blocks: A Deep Learning Method for Swimming Crab Recognition

Yi-Xian Gu, Yi-Jui Chiu*

School of Mechanical and Automotive Engineering
Xiamen University of Technology, Xiamen 361024, China
2221011010@s.xmut.edu.cn, chiuyijui@xmut.edu.cn

Yi-Jia Shih

The Center for Marine Policy Studies
National Sun Yat-sen University, Kaohsiung 80424, Taiwan, China
eja0313@gmail.com

*Corresponding author: Yi-Jui Chiu

Received March 4, 2024, revised June 30, 2024, accepted September 12, 2024.

Abstract. *Accurate identification and classification of marine organisms is of particular importance. To investigate marine ecosystems and elucidate the biodiversity within the marine crab family, this research primarily focuses on Portunus pelagicus, while also encompassing Portunus sanguinolentus, Portunus sayi, and Portunus trituberculatus. Portunus pelagicus is a commercially important marine organism widely distributed in Asia and the Pacific. This study uses the You Only Look Once (YOLO) object detection model but improves its robustness and generalization ability by introducing certain modifications. Namely, the Mixup data enhancement method is used in the dataset preprocessing phase, which improves the mean average precision (mAP) of the model from 88.88 to 94.67 % for Portunus pelagicus, while the F1 score is improved from 0.79 to 0.89. Then, the convolutional block attention module (CBAM) attention mechanism is added to the YOLOv5 model. The addition of the spatial and channel attention module improves the mAP of the model for Portunus pelagicus to 99.2 %, while the overall mAP value of the model reaches 87.30 %. Further applying the YOLOv8 model with a CBAM module increases the model's mAP for Portunus pelagicus to 99.9 %, while the model's overall mAP value reaches 92.89 %. The overall F1 score is 0.86 at a confidence threshold of 0.5, and the model's mAP value after 75 epochs reaches 93.2 %. Finally, the proposed method is compared with the SSD, Efficientdet, and Faster-Rcnn algorithms. The comparison results indicate that the proposed optimized YOLOv8(CBAM) model performs well in recognition and classification tasks on small datasets and outperforms other models. The results presented in this study provide a useful reference for future applications of deep learning techniques in similar environments.*
Keywords: YOLOv5, CBAM, YOLOv8, *Portunus pelagicus*, Mixup

1. **Introduction.** Deep learning is gradually being integrated into various scientific fields. Ma et al. [1] proposed a novel point-by-point filtered CNN branch for automatically integrating and passing features to a learning architecture for processing medical images. Wu et al. [2] proposed a spectral convolutional neural network model based on Adaptive

Fick's Law Algorithm aimed at solving the remote sensing image classification problem. Zhang et al. [3] applied motion classification algorithms with linear decision making and support vector machine to human motion recognition. In this paper, deep learning and marine life image recognition are combined.

Recent advances in the field of deep learning have revolutionized the detection and classification methods of marine organisms. Knausgård et al. [4] used the You Only Look Once (YOLO) algorithm for object detection in temperate fish. A convolutional neural network (CNN) and a squeeze and excitation architecture were used to classify the identified temperate fish. Guénard et al. [5] employed a deep feed-forward artificial neural network (ANN) approach. The ANN was used to model the distribution of lake sturgeon and white bass in a changing estuarine environment. The application of multiple descriptors provided a classification accuracy of 94%. Han et al. [6] successfully identified and localized targets in underwater environments with the help of area proposal networks. This model allowed underwater robots to achieve effective marine product collection. Jose et al. [7] used a cubic support vector machine (SVM) classifier along with the complex wavelet transform for feature extraction and proposed an automatic tuna classification system, which could achieve an accuracy of approximately 95%. Mana et al. [8] proposed an intelligent deep learning network based on marine fish species classification technology. A water wave optimization technique was used based on classification with an optimal deep kernel limit learning machine. Tan et al. [9] explored the application of data enhancement in the field of marine image classification and found that traditional enhancement methods performed poorly in this task. They proposed an innovative enhancement strategy that can outperform the AutoAugment method in marine image classification. Purcell et al. [10] trained neural networks based on the ResNet-50 and MobileNet V1 architectures and achieved the detection and recognition accuracy of approximately 80% for 10 types of marine objects.

However, the identification of challenging organisms in complex environments is a highly demanding task. The related studies used the MobileNet, Mask Rcnn, capsule networks, and wavelet kernel extreme learning machine approaches to address this problem. Zhang et al. [11] used the MobileNetv1 as a backbone of the SSD; they investigated the enhancement of small target features and suppression of irrelevant features by using the feature receptive domain block and attention mechanism. The average accuracy was improved by 5.1%, and better robustness was achieved. Al Duhayyim et al. [12] proposed the IDLAFD-UWSN model, which combines the Mask RCNN, capsule network, and wavelet kernel extreme learning machine. Cao et al. [13] proposed Faster MSSDLite, which is a real-time robust underwater live crab detector based on deep learning; this model combines MobileNetV2, deep separable convolutional, and feature pyramid networks. In addition, a uniformly quantized CNN was employed for error correction. Ridge et al. [14] developed the OysterNet model, which helps unmanned aircraft systems determine the extent of oyster reefs. Piazza et al. [15] applied deep learning to scanning electron microscope images and developed a CNN-based classification model using a dummy classifier.

Information feature recognition with invisible features (e.g., age of marine organisms and voice) has been a hot topic in current research. The related studies have experimented with neural network structures that are more biased toward data processing, such as DNCNN and Mask_LaC R-CNN. Martinsen et al. [16] used an automated process based on CNNs to estimate the age of fish. The resulting CV values averaged about 10%, and the model's decision-making process was resolved. Vickers et al. [17] developed a research method based on denoising CNNs (DNCNNs) and denoising autoencoders (DAEs) and achieved a significant improvement in the sound classification accuracy of North Atlantic right whales. Bermant et al. [18] trained recurrent neural networks to classify whale tail

segment types and vocal clades; the overall accuracy was over 93%, while a 99.4% recognition accuracy was achieved for specific individual whales. Zhong et al. [19] developed a deep learning-based model to categorize sounds automatically. The main goal was to reduce the labor and time of manually identifying beluga whale sounds. They achieved an accuracy rate of 96.57% and a recall rate of 92.26%. Han et al. [20] designed a non-contact fish morphological parameter measurement system. In their work, the real scene was simulated by various data extension techniques, and the loss function and network structure of the Mask R-CNN were improved. Chang et al. [21] proposed a preprocessing CNN, using semi-supervised learning training to process continuous sonar images and standardized feature mapping to solve the fish segmentation problem of the Mask R-CNN in different shallow-water fish farms.

The monitoring of mass migration of marine organisms is crucial in the context of marine ecological protection. In related research, different advanced techniques, such as Kalman filter and transfer learning, have been widely used for tracking marine fish communities. Kandimalla et al. [22] developed an unlabeled fish monitoring platform and constructed a deep learning framework by incorporating a Kalman filter. Public sonar and optical data were employed to detect, categorize, and track multiple fish species. Lumini et al. [23] proposed an automatic plankton identification system that incorporates different deep learning-based methods. Their system can enhance the diversity of classifiers using the fine-tuning of deep learning-based models and migration learning. Conrady et al. [24] used the Mask R-CNN target detection framework for automatic localization, classification, counting, and tracking of fish in underwater environments. Alshdaifat et al. [25] proposed an improved framework for the segmentation of fish instances in underwater videos. This framework uses enhanced detection and dynamic instance segmentation methods based on the area proposal networks. Xu et al. [26] developed an innovative classification method for PSP and non-PSP microalgae by combining three-dimensional fluorescence, machine learning, and deep learning. This method achieved an accuracy above 94% in classifying 12 microalgae, and the identification accuracy of PSP microalgae was even higher, 96.25%. This technique could help to identify toxic microalgae accurately in real-time. Baek et al. [27] simulated an ocean model of Chained Alexandrium bloom by using the CNN models. The classification CNN determined the bloom onset, and the regression CNN estimated the bloom density. The accuracy and root mean square error reached 96.8% and 1.20 [log(cell L^{-1})], respectively. Martin-Abadal et al. [28] proposed an automatic jellyfish detection and quantification model based on deep target detection neural networks. The proposed model achieved an accuracy of 93.8% in the jellyfish detection and quantification task for real-time processed video sequences. This study provided an effective monitoring method for developing a jellyfish early warning system.

Recognizing intricate organisms in complex environments is an ongoing challenge. As both MobileNet and Mask RCNN exhibit limitations in robustness, feature detection, and real-time performance, an integrated approach is needed. This study considers the diversity of marine organisms and their significant impact on global ecosystems. This study examines *Portunus pelagicus*, *Portunus sanguinolentus*, *Portunus sayi*, and *Portunus trituberculatus*, but the main focus is on *Portunus pelagicus*, which is a commercially important marine organism with a wide distribution in Asia and the Pacific. This study uses deep learning-based techniques to accurately recognize and classify the aforementioned creatures. The YOLO object detection model is selected. In the model training phase, the Mixup data enhancement method is employed to enhance the robustness and generalization ability of the model. The convolutional block attention module (CBAM) attention mechanism is also introduced to enhance the recognition accuracy of the model further.

2. **Detection target: *Portunus pelagicus*.** Brief descriptions of *P. sanguinolentus*, *P. trituberculatus*, and *P. sayi* are provided below, as well as a more detailed introduction to *P. pelagicus*, which is the focus of this study. Photographs of the four crab species in the genus *Portunus* are shown in Figure 1.

FIGURE 1. Four species of genus *Portunus*

Portunus pelagicus is a tropical marine crustacean belonging to the family Portunidae. Its cephalothoracic armor is transversely ovate, roughly muscled, and the granular surface is clearly visible. The chelicerae of the distant water pike crabs are relatively long and asymmetrical, and the same granular texture is visible on them. Also, there are two distinct ridges on the dorsal side of the chelicerae. Several images from the dataset used for target detection for distant pike crabs are shown in Figure 2.

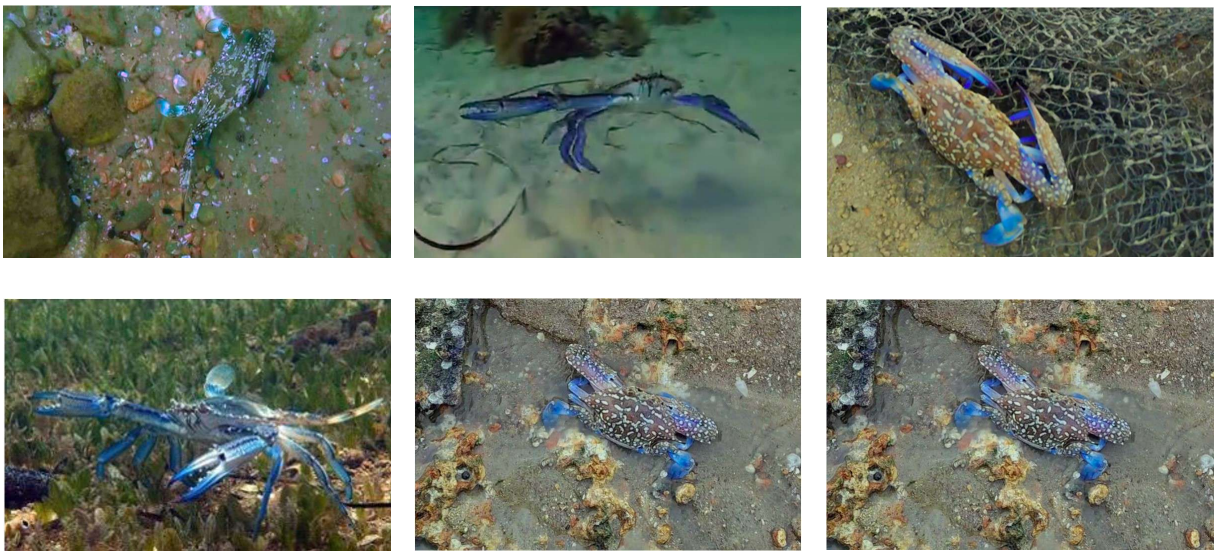

FIGURE 2. Images used for the detection of *Portunus pelagicus* objects

Portunus sanguinolentus is an invertebrate marine organism belonging to the family Portunidae. Its cephalothoracic armor has a coxal shape, which is slightly elevated in the middle, with three conspicuous wart-like projections on the surface. The chelicerae are well-developed, with long segments that are prismatic and equipped with blunt teeth on the inner side. The underside of its shell has three eye spots, and the shell color is grayish-green.

Specimens of *Portunus trituberculatus* have a considerable size, with individuals reaching a weight of up to 1,000 g and a carapace width of up to 200 mm. The structure of the *P. trituberculatus* consists of a head, a thorax, an abdomen, and appendages. The species is similar to *P. pelagicus* in general appearance, but *P. trituberculatus* can be easily distinguished from *P. pelagicus* because it has three frontal teeth (*P. pelagicus* has four frontal teeth) and four spines in the merus of chelipeds (*P. pelagicus* has three spines).

Portunus sayi is a special type of crab from the family Portunidae. The thoracic armor portion has many unique features, including long, straight dorsal spines and full lateral spines. Its head has long mouth spines that are slightly longer than the base of the antennae, and its antennae have a single-branched form, fuller at the base, equipped with 17 or 18 sharp spines. These traits denote a powerful tool for defending from predators or competing conspecifics.

3. Detection algorithm: the YOLO Series.

3.1. **Algorithmic structure.** This study employs the YOLOv5 algorithm with an added Convolutional Block Attention Module (CBAM) for the image recognition of *P. pelagicus*. The main framework of the YOLOv5 algorithm is primarily divided into three components.

First, feature extraction of *P. pelagicus* is performed by the backbone network CSP-Darknet. Then, feature enhancement is realized on the recognized target using a feature pyramid network. Finally, the object corresponding to the feature point is predicted by the Yolo head. The specific process is illustrated in Figure 3.

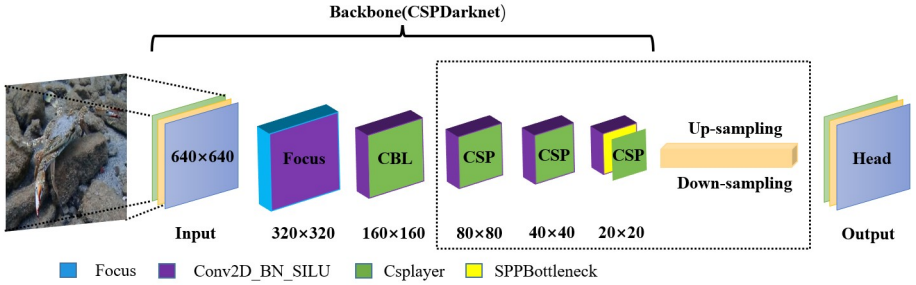

FIGURE 3. Integrated framework of the YOLOv5 model for the detection of pike crab

CBAM includes a spatial attention module and a channel attention module. The spatial attention module facilitates the determination of the extent to which specific regions or parts of an image, such as the crab shell and crab legs, affect the recognition or behavioral analysis result of *P. pelagicus*. The channel attention mechanism helps in ascertaining the impact of channels (e.g., color) on the recognition or classification result when processing the stone crab images. In summary, image data of *P. pelagicus* can be analyzed with high precision and efficiency by incorporating the CBAM into the detection model. This aids in a deeper understanding of its biological characteristics and behaviors. The CBAM module is presented in Figure 4.

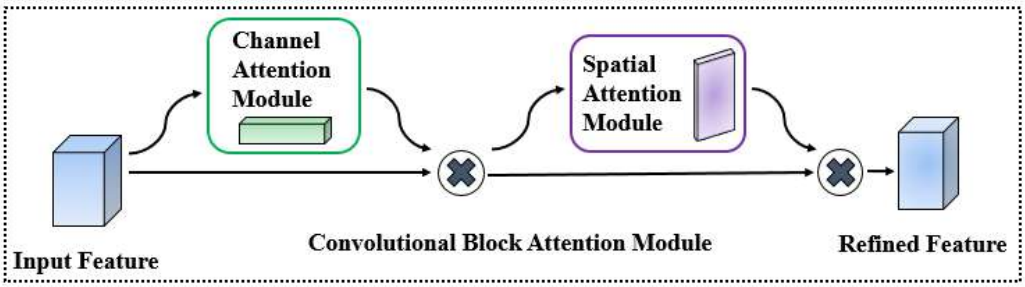

FIGURE 4. An overview of the CBAM

Moreover, this study employs the YOLOv8 algorithm with an added CBAM to optimize the recognition results. The network framework of the YOLOv8 algorithm mirrors that

of the YOLOv5 algorithm. YOLOv5 utilizes the focus structure for feature extraction, whereas YOLOv8 employs a standard 3×3 convolution with a step size of 2. This approach enhances recognition speed at the expense of some perceptual nuance. The preprocessing steps of the CSPlayer module are also reduced from three to two. Finally, the Dense Label Fusion module is introduced to calculate the regression values. The specific process is illustrated in Figure 5.

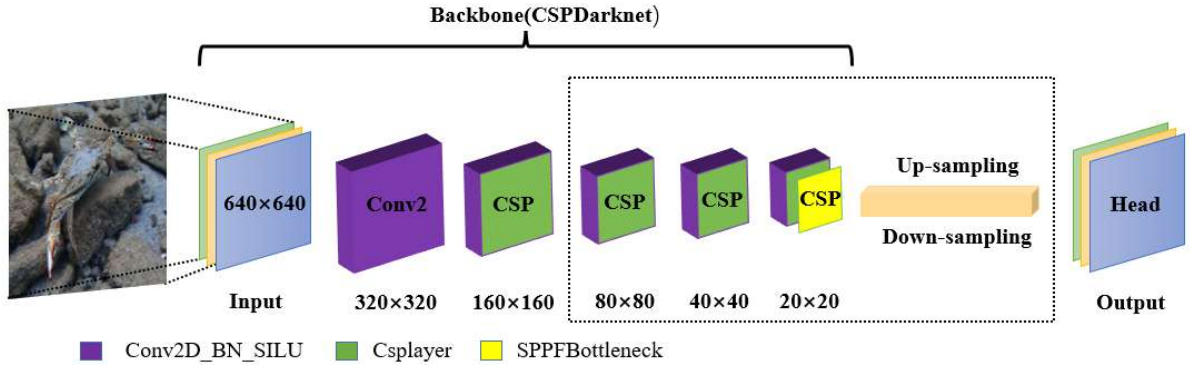

FIGURE 5. Integrated framework of the YOLOv8 model for the detection of pike crab

3.2. **Mathematical derivation of the algorithm.** In the research on *P. pelagicus*, there have been many core techniques and methods that require in-depth understanding. In some cases, multiple detection boxes can overlap at certain locations in an image, which can cause multiple detections of the same target. The non-maximum suppression (NMS) has been introduced to eliminate redundant detection boxes, ensuring that only the most representative prediction boxes are retained.

Anchor boxes denote a set of predefined rectangular boxes with different aspect ratios and scales. The purpose of anchor boxes is to adapt to various shapes of target objects. Each grid in the YOLOv8 model with the added attention mechanism is equipped with three types of anchor frames. Considering different feature map scales, there are three specific anchor boxes for each scale, which results in a total of nine anchor boxes. The IOU definition and the relationships between various boxes are presented in Figure 6.

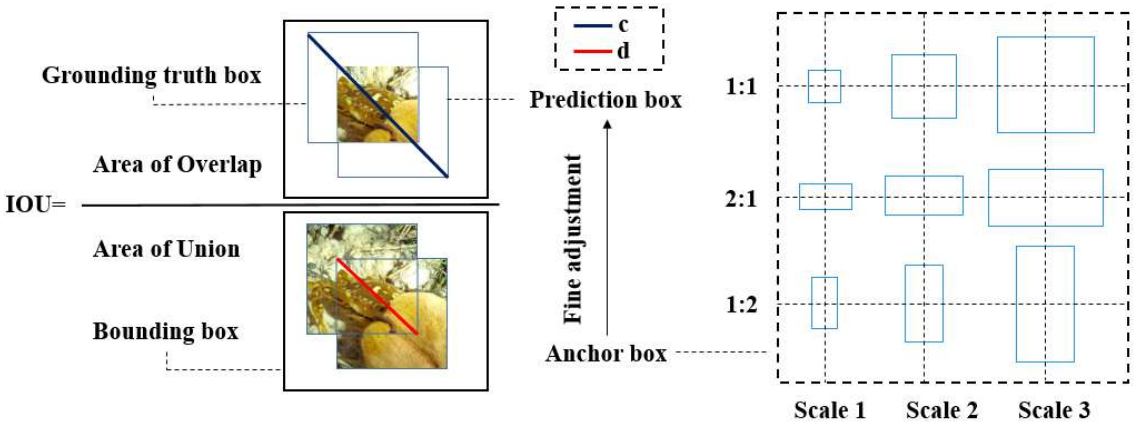

FIGURE 6. The definition of intersection over union and the relationships between various boxes

In this study, two evaluation metrics are used, precision and recall, which are defined by Equations (1) and (2), respectively. Precision is calculated as the number of true positives

divided by the sum of true positives and false positives, whereas recall is calculated as the number of true positives divided by the sum of true positives and false negatives.

$$\text{Precision } = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall } = \frac{TP}{TP + FN} \tag{2}$$

Based on the IOU, this study proposes the concepts of generalized intersection over union (GIOU), dense intersection over union (DIOU), and complete intersection over union (CIOU). The proposed algorithm uses the CIOU as a loss for bounding box regression.

The geometric relationships between the ground truth box and the predicted box in images are illustrated in Figure 6. The CIOU combines the IOU, the distance from the center of the frame, and the aspect ratio, and its penalty items are defined as follows:

$$\mathcal{R}_{\text{CIOU}} = \frac{\rho^2\left(\mathbf{b}, \mathbf{b}^{gt}\right)}{c^2} + \alpha v \tag{3}$$

where b^{gt} represents the ground truth; $\rho(\mathrm{b}, \mathrm{b}^{gt})$ is defined as the distance between the centroid of the prediction box and the ground truth box; c represents the length of the diagonal of the large rectangular box formed by the union of the predicted box and the ground truth box; v is defined as the fit of the aspect ratio, and α is the trade-off parameter, and they are respectively defined as follows:

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{4}$$

$$\alpha = \frac{v}{(1 - \text{IOU}) + v} \tag{5}$$

The IOU definition is presented in Figure 4, where w^{gt} and h^{gt} represent the width and height of the ground truth box, respectively; meanwhile, w and h represent the width and height of the predicted box, respectively.

Finally, the loss function of the CIOU is defined as follows:

$$\mathcal{L}_{\text{CIOU}} = 1 - \text{IOU} + \frac{\rho^2\left(\mathbf{b}, \mathbf{b}^{gt}\right)}{c^2} + \alpha v \tag{6}$$

The area factor is given a higher priority in the bounding box regression when the CIOU is used as an assessment metric. This means that the CIOU puts more emphasis on the actual area overlapping of the two bounding boxes than on other similarity metrics. Therefore, the evaluation of bounding boxes is more comprehensive and accurate.

4. **Algorithm recognition procedure.**

4.1. **Parameter settings.** The parameter settings of the YOLOv8 algorithm are shown in Table 1. The batch size is set to 32, and the confidence threshold is set to 0.001 to filter out unreliable detection results rigorously. A larger number of training epochs can allow the model to fit the data better, but an excessive number of training epochs can also lead to model overfitting. Therefore, the number of epochs is set to 100. Weight Decay is a regularization technique that can help control model complexity and prevent model overfitting. The weight attenuation factor, also known as λ, was set to 5×10^{-4}. Namely, selecting appropriate weight decay can improve the generalization ability of the model. Further, momentum is a parameter of an optimization algorithm used to accelerate the convergence process. In this study, the momentum value is set to 0.937, which means that the effects of the previous update steps are considered in each update. Finally, the image size is set to 640, meaning that an image has 640×640 pixels.

TABLE 1. Model parameter settings

Hyperparameter	Numerical value
Batch size	128
Confidence threshold	0.001
Epoch number	100
Weight attenuation factor	$5e-4$
Momentum	0.937
Image size	640

4.2. **Data processing.** Image processing was initially conducted using the HSV color model in the training phase. Certain changes were made to the training images of the dataset in terms of hue, saturation, and value. In addition, image transformations, such as rotation, translation, scaling, and shearing, were employed.

The image rotation angle ranged from $-45°$ to $45°$. The horizontal (left–right) and vertical (up–down) pans encompass approximately 90% of the image's width and height, respectively. The image scaling range was 90%. The image shear was tilted horizontally and vertically in the range of $(-10, 10)$. Also, the translation and image perspective transformations were implemented. The image perspective transformation is a technique that simulates a three-dimensional viewpoint, making an image appear as if it is viewed from a different angle. Finally, the two methods, mosaic and Mixup, were used to merge the four images together. Image blending denotes the process of mixing the pixel values of two images according to a certain ratio. There were 1,034 images before image enhancement and 1,212 images after image enhancement.

The data distributions for the four identified targets are shown in Figure 7(a). The sample sizes for the marine organisms were as follows: 865 specimens were *P. pelagicus*, 129 were *P. trituberculatus*, 168 were *P. sanguinolentus*, and 50 were *P. sayi*. Figure 7(c) illustrates the distribution of the location of the center of the prediction box. Figure 7(d) shows the height and width information of the prediction box. The majority of the samples are predominantly dispersed within a region demarcated by relative coordinates ranging from (0 to 0.4, 0 to 0.4).

5. Recognition Results.

5.1. **Preliminary preparation.** The YOLOv5 algorithm was executed using both its standard configuration (YOLOv5) and the configuration with data augmentation used in the preprocessing stage (YOLOv5 (Mixup)) as separate runs. Figure 8 presents a visual representation of the mAP values for various categories using the two methods. The results in Figure 8 show that *P. pelagicus* had the highest accuracy, which was much higher than that of the other species. The mAP values of the two methods for *P. sanguinolentus*, *P. sayi*, and *P. trituberculatus* were 17% and 20%, 8% and 17%, and 19% and 45%, respectively.

Figure 9 shows the comparison results of the F1 values between the two algorithms. The YOLOv5 algorithm achieved the highest F1 value for *P. pelagicus*, having a value of 0.79. The F1 value of the YOLOv5 (Mixup) algorithm for *P. pelagicus* was 0.89. Thus, using the data augmentation method during the preprocessing stage resulted in a 10% improvement in the F1 score in target recognition.

The PR plots of the two algorithms for *P. pelagicus* are shown in Figure 10, showcasing mAP values of 88.88% for the YOLOv5 algorithm and 94.67% for the YOLOv5 (Mixup) algorithm.

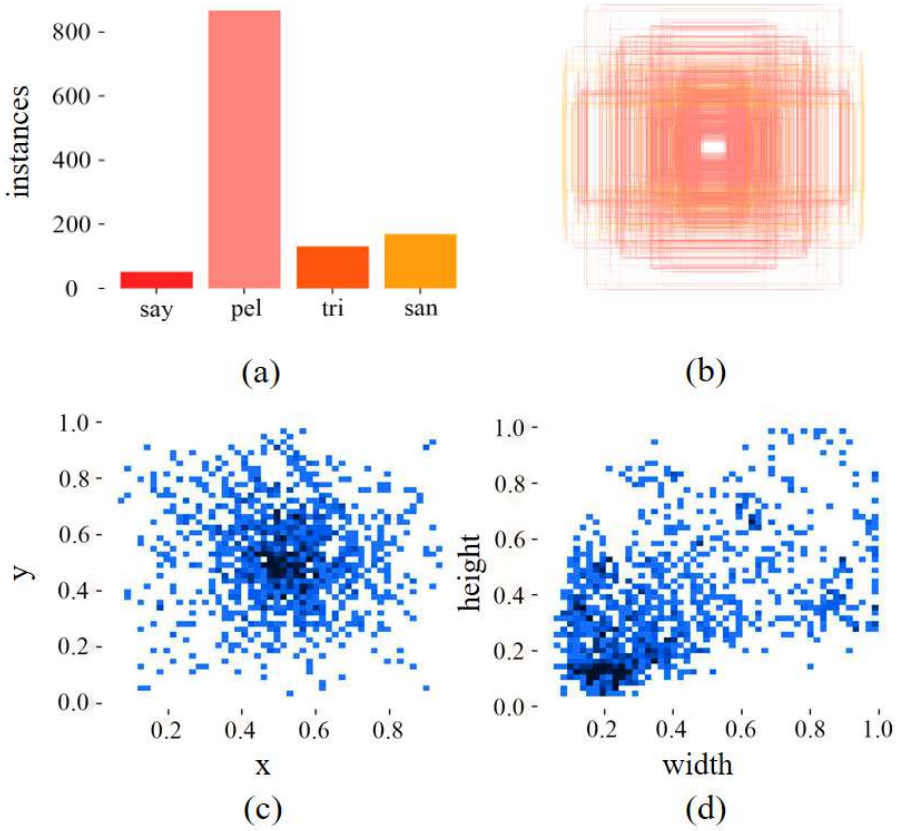

FIGURE 7. Distribution of data within the training set and the distribution of annotated frames during the training process

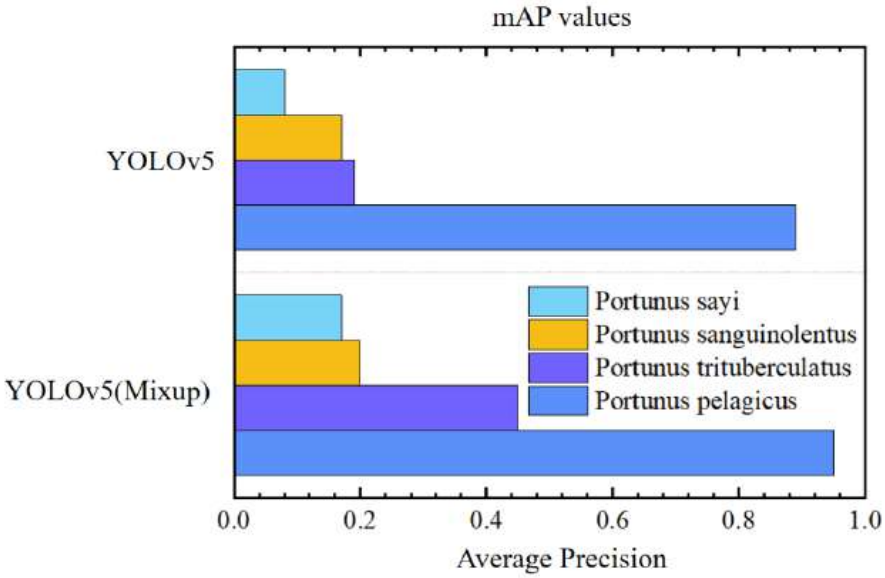

FIGURE 8. Comparison of mAP values between the two algorithms

Based on the results, the algorithm without data preprocessing performed slightly worse compared to the post-processed algorithm. Thus, this processing can result in performance improvements. However, excessive data enhancement in the data preprocessing stage can lead to model overfitting and even have a negative effect on the model performance.

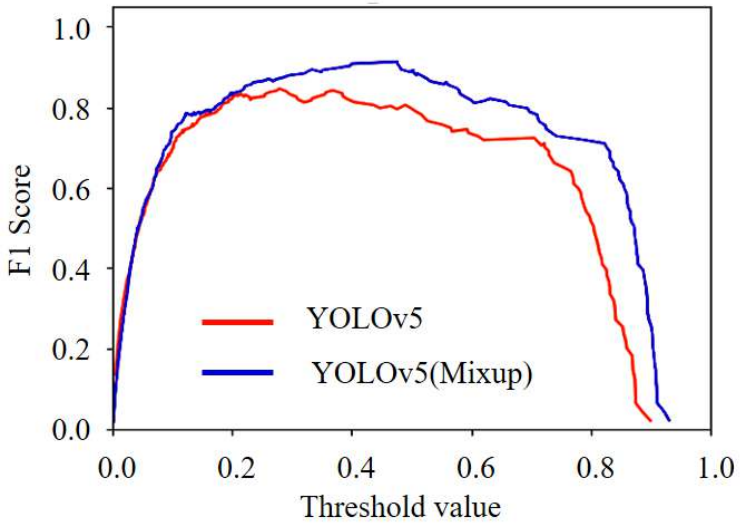

FIGURE 9. Comparison of F1 values for Portunus pelagicus using the two algorithms

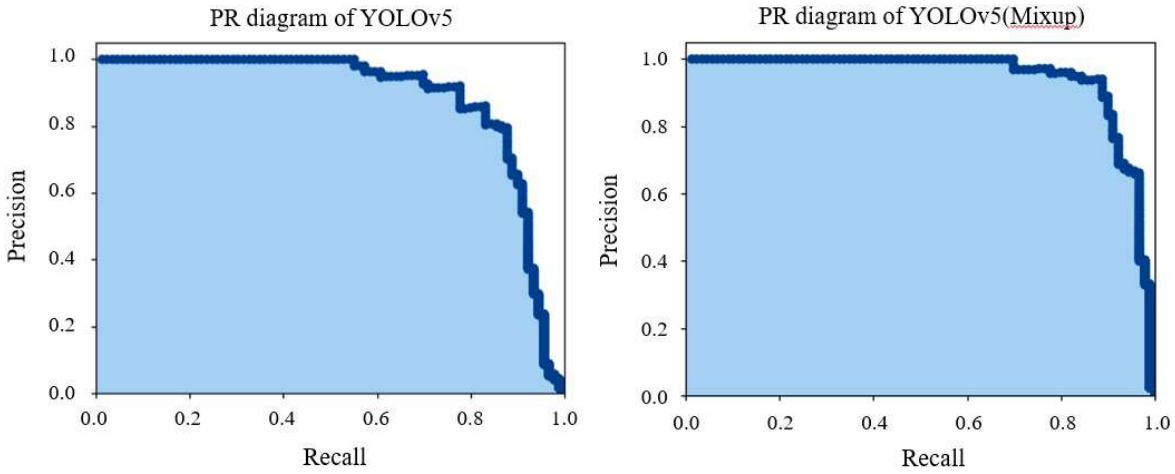

FIGURE 10. Comparison of PR maps for Portunus pelagicus between the two algorithms

5.2. **Subsequent Identification results.** The bounding box loss results on the training data are presented in Figure 11(a). The CIOU approach was employed to calculate the difference between the measured bounding box and the actual bounding box. In epochs 0–100, the loss values were gradually reduced from 0.1 to close to zero, suggesting that the accuracy of predictions for bounding boxes improves rapidly during the learning process.

Figure 11(b) represents the target loss results on the training set. This loss function ensures that the model can correctly predict the presence or absence of a target. In the dataset, there were some unpredictable samples, so regardless of how much the model learned from the data, it was difficult to predict the existence of the target with high accuracy for all samples. Therefore, the loss value gradually reduced from 0.032 to 0.015 instead of reducing to close to zero.

Figure 11(c) shows the classification loss results on the training dataset. This loss value indicated the difference between the categories predicted by the model and the true categories when the model needed to predict multiple categories.

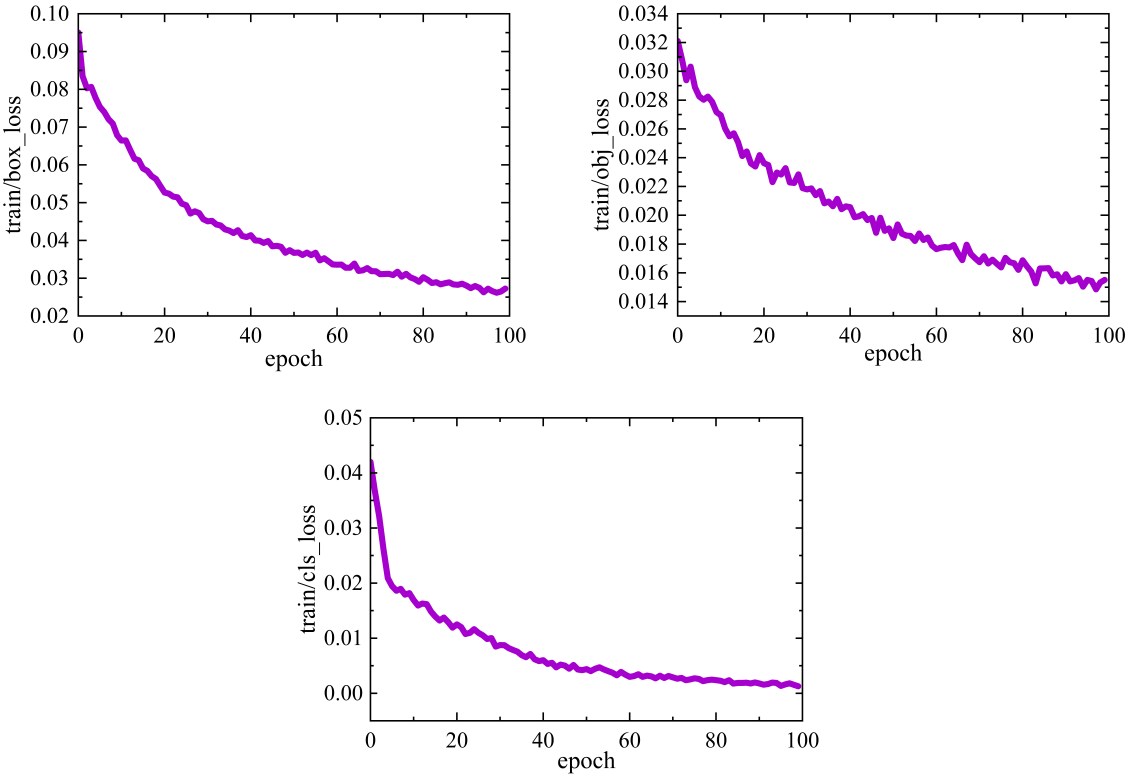

FIGURE 11. Loss results on the training dataset during training

Figure 12(a) illustrates the boundary loss results on the validation dataset. The bounding box loss (BBL) value commenced at 0.074 at the beginning of training. Subsequently, the bounding box loss gradually diminished with the progression of model training and stabilized at 0.016 at the training process's conclusion. This fluctuating downward trend indicated that the model gradually learned to localize the target accurately during training. The overall trend suggested a consistent increase in the model's bounding box predictive power, despite certain fluctuations in the loss decline.

Figure 12(b) presents the target loss results on the validation dataset. The initial value of the target loss was 0.028. Mirroring the bounding box loss, this loss also exhibited a fluctuating descending trend and stabilized at approximately 0.01 at the end of the training process. The decrement in the target loss indicated a gradual enhancement in the model's capability to distinguish between the foreground targets and the background.

Figure 12(c) displays the classification loss results on the validation dataset. The classification loss diminished gradually from the initial value of 0.036 and ultimately converged to zero. This was a highly positive indication, signifying that the model had learned to categorize detected targets with high accuracy. An approaching-zero classification loss denoted a very low classification error rate of the model on the validation dataset, thereby further reinforcing the model's robustness and reliability.

Figure 13(a) illustrates the accuracy results on the validation dataset. The results demonstrated that the model could learn very rapidly in the initial training phase, with almost all samples with positive predictions being truly positive. During the initial phase of model training, in epochs 0–25, the accuracy escalated swiftly to close to one, where one denotes the perfect accuracy of 100%. However, a subsequent rapid decline to approximately 0.3 highlighted that indistinguishable samples or noisy data precipitated model

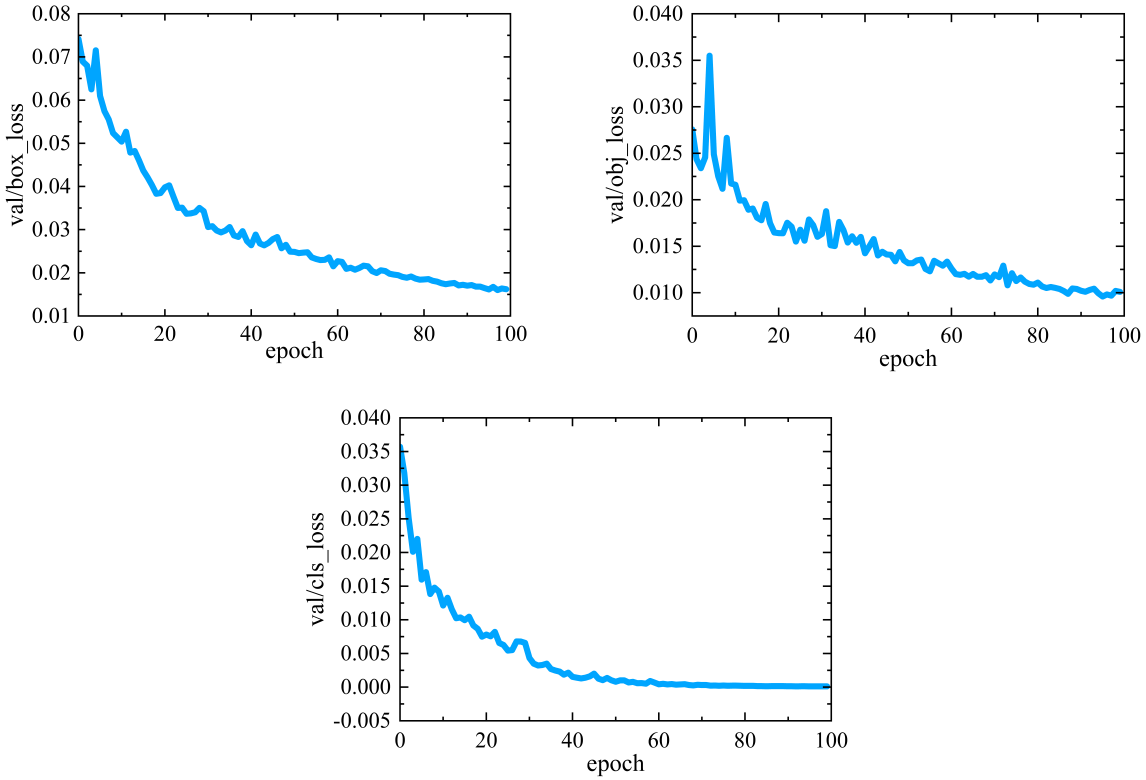

FIGURE 12. Loss results on the validation dataset during training

overfitting, leading to inaccuracies in the predictions in the short term. The gradual fluctuation up to 0.8 in epochs 30–100 indicated that the model was gradually adapting to the data and finding a better equilibrium to enhance its prediction accuracy.

Figure 13(b) depicts the change in the recall value during the model training process. The results revealed a rapid increase in the recall value to 0.2 in the early training phase, signifying the model's proficiency in correctly identifying positive samples in the dataset. However, thereafter, the recall value plummeted rapidly and converged to zero. The appearance of a peak value indicated that the model abruptly regained its predictive capability in a particular iteration; however, this capability was rapidly diluted by the stable training process.

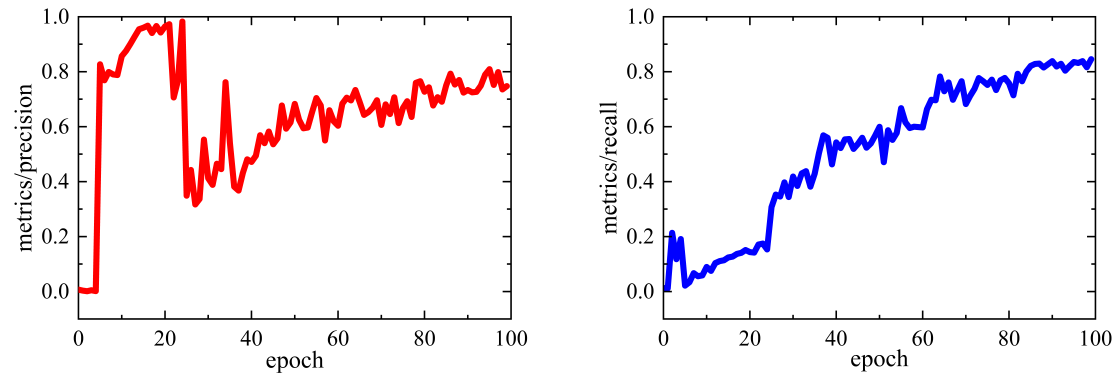

FIGURE 13. The loss results on the validation dataset during training

Figure 14(a) displays the average accuracy results obtained at the IOU threshold of 0.5. The results indicate that the accuracy gradually increased from zero during training and finally stabilized at 90%. This indicated that the model achieved predictions with a high degree of overlap with the true labeling in most cases.

Figure 14(b) shows the average accuracy results obtained as the IOU threshold changes from 0.5 to 0.95. The result was obtained by summing the mAP values at each IOU threshold value (10 threshold values from 0.5 to 0.95, with a 0.05 increment) and dividing the sum by 10. In this way, the performance of the model could be evaluated more comprehensively under different IOU threshold values. The description of the loss function of the YOLOv8(CBAM) algorithm is described above.

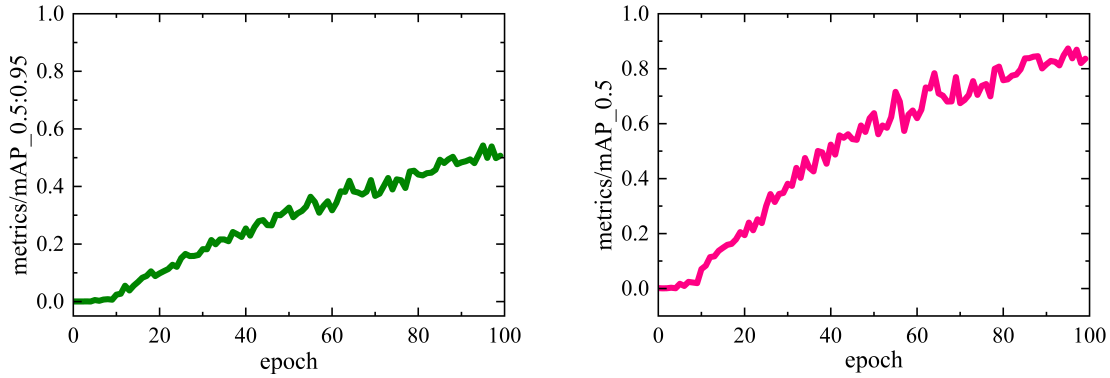

FIGURE 14. Average accuracy under different IOU threshold values

The variations in the accuracy rate of the YOLOv5(CBAM) algorithm under different confidence threshold values are illustrated in Figure 15. At the confidence threshold of 0.947, the accuracy rate reached 1.00 for all categories. The findings demonstrate that the precision of the detection results for the *P. sanguinolentus* category was higher than that for *P. sayi* in some medium-confidence cases. The recall-confidence plot was employed to assess the algorithm's efficacy in target detection. The correlation between the confidence and recall values is depicted in Figure 16. Remarkably, the recall value for all target categories reached 0.99 when the confidence threshold was adjusted to zero. The YOLOv5(CBAM) algorithm could accurately detect 99% of the actual positive samples. Furthermore, the algorithm managed to identify almost all actual positive samples with a high recall value at significantly low confidence threshold values. However, this scenario potentially increased the number of incorrect positive sample predictions due to the low confidence threshold, making the algorithm more susceptible to including low-confidence detections as positive samples. Notably, the recall value of the algorithm for *P. trituberculatus* surpassed that for *P. sayi* when the confidence threshold level was above 0.5. Variations in the accuracy rate of the YOLOv5(CBAM) algorithm at distinct recall values are illustrated in Figure 17. The differential performance of the algorithm across categories, determined by analyzing under varying threshold conditions, is presented. The PR curve exhibits a gradual downward stepwise trend. Among the PR curves, the curves for the *P. pelagicus* category had the largest area, indicating superior model precision for this category. At a recall of 0.3, the model's precision for the *P. sanguinolentus* category was slightly higher than for the *P. sayi* category. The PR curve for *P. pelagicus* was closer to the upper right corner, suggesting that this category maintained a high precision rate even at lower recall values. The average precision was 0.873 for tests with a confidence threshold level greater than or equal to 0.5. A remarkable mAP value of 0.992 was

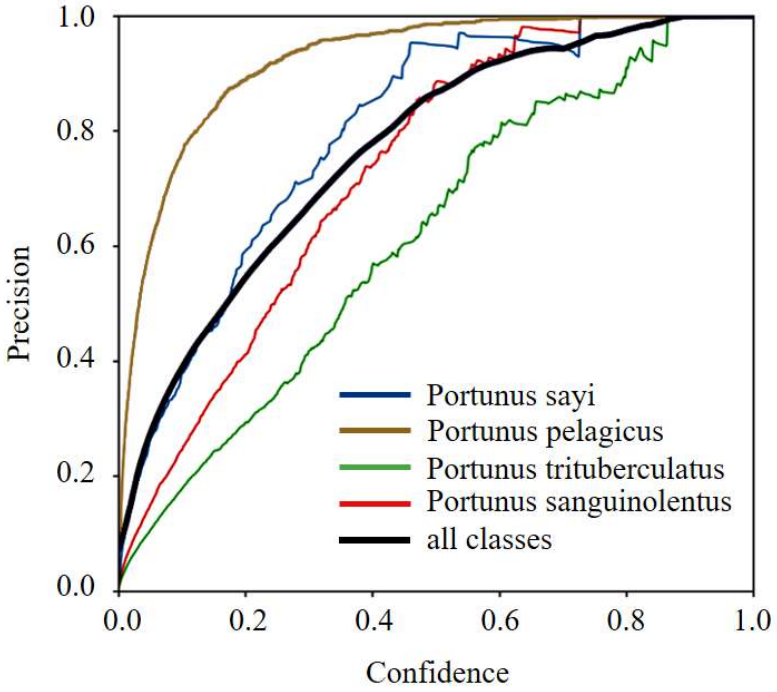

FIGURE 15. Precision-confidence curves of the YOLOv5(CBAM) model

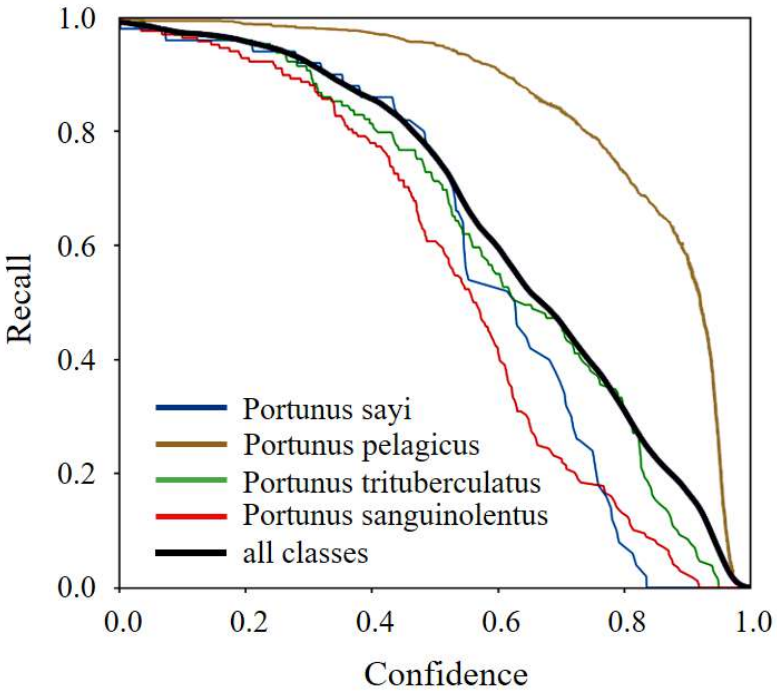

FIGURE 16. The recall-confidence curves of the YOLOv5(CBAM)

achieved for *P. pelagicus*, while the mAP values for *P. trituberculatus*, *P. sanguinolentus*, and *P. sayi* were 0.756, 0.833, and 0.913, respectively.

Figure 18 showcases the mAP values for the YOLOv8(CBAM) algorithm. An impressive mAP value of 0.9978 was obtained for *P. pelagicus*, 0.9812 for *P. trituberculatus*,

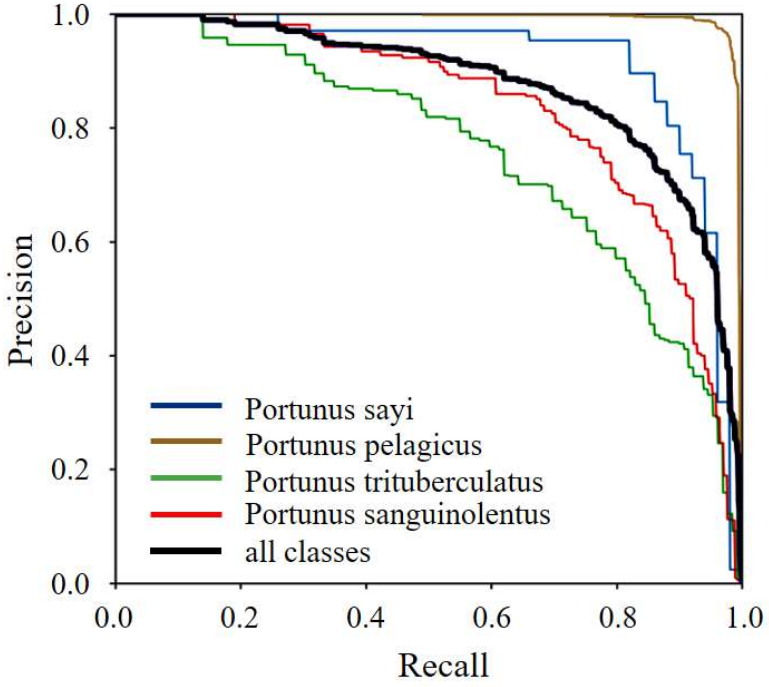

FIGURE 17. Precision-recall curves of the YOLOv5(CBAM)

0.9868 for *P. sanguinolentus*, and 0.75 for *P. sayi*. The YOLOv8(CBAM) algorithm enhanced the mAP values significantly for the majority of the targets while reducing the mAP value by 0.163 for *P. sayi*.

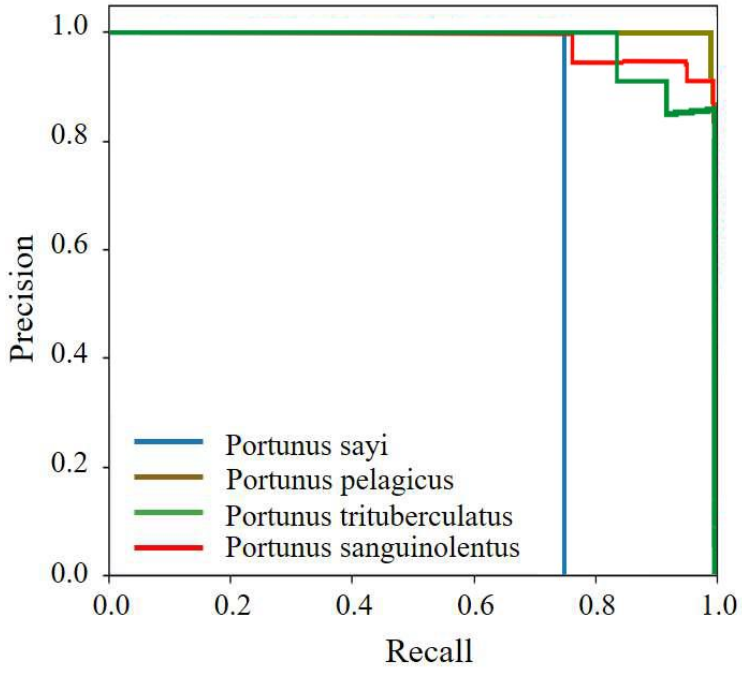

FIGURE 18. The precision-recall curves of the YOLOv8(CBAM)

Figure 19 presents the normalized confusion matrix, representing the performance evaluation results of the YOLOv8(CBAM) detection algorithm across different categories.

The recall for the *P. sayi* category was 0.94, indicating a 94% detection accuracy for samples belonging to this category. For the *P. pelagicus* category, the model achieved a recall of 0.99, signifying even better performance with a 99% detection accuracy. The recall values for *P. trituberculatus* and *P. sanguinolentus* were 0.94 and 0.93, respectively.

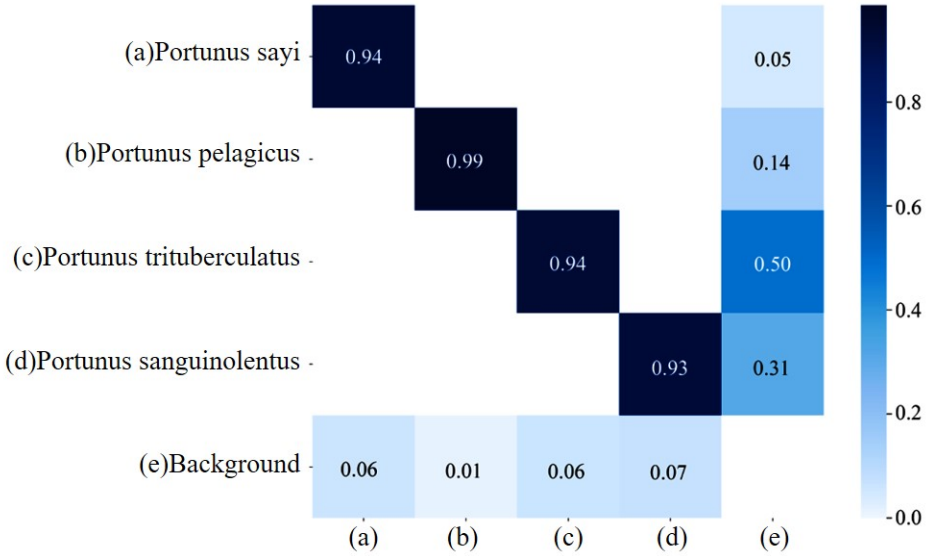

FIGURE 19. Normalized confusion matrix of the YOLOv8(CBAM)

The analysis results highlighted misclassification cases in the last "background" line, where values 0.06, 0.01, 0.06, and 0.07 indicated the likelihood of the algorithm incorrectly predicting true background samples as other categories. Similarly, values 0.05, 0.14, 0.50, and 0.31 in the last column suggested the probability of the algorithm incorrectly classifying samples from other categories as background.

The F1 score is widely recognized as a crucial metric for evaluating the overall performance of a model across different levels of confidence, as it harmoniously combines precision and recall, thereby offering a more comprehensive reflection of the model's true performance.

Figure 20 illustrates that the YOLOv5(CBAM) algorithm achieved a commendable balance between precision and recall values, registering an overall F1 score of 0.82 at a confidence threshold of 0.455. Notably, the *P. pelagicus* category consistently exhibited the highest F1 scores across all confidence thresholds, whereas the F1 scores for the *P. sanguinolentus* and *P. trituberculatus* categories were observed to be the lowest. Additionally, the F1 scores for the *P. sayi* category demonstrated a rapid decline post a confidence level of 0.5.

Figure 21 delineates the variation in the F1 score in relation to the confidence level for the YOLOv8(CBAM) algorithm. A notable F1 score of 0.99 was achieved for *P. pelagicus*, while the scores for *P. trituberculatus*, *P. sanguinolentus*, and *P. sayi* were recorded at 0.88, 0.93, and 0.67, respectively.

The YOLOv8(CBAM) algorithm underwent training on a meticulously compiled pike crab dataset, with the confidence threshold for network training uniformly set to 0.001. An exploration into the impacts of batch size and calendar elements on the effectiveness of network training under various hyperparameter values yielded significant results, as depicted in Table 2. These findings underscored that the model attained its peak mean Average Precision (mAP) value of 92.2% at a batch size of 128 and an epoch number of 75.

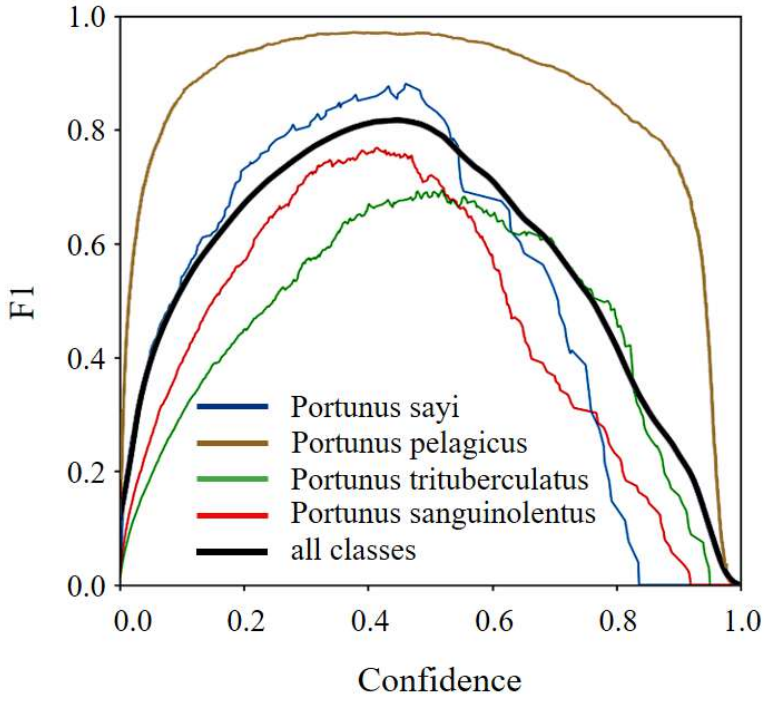

FIGURE 20. The F1-confidence curves of the YOLOv5(CBAM)

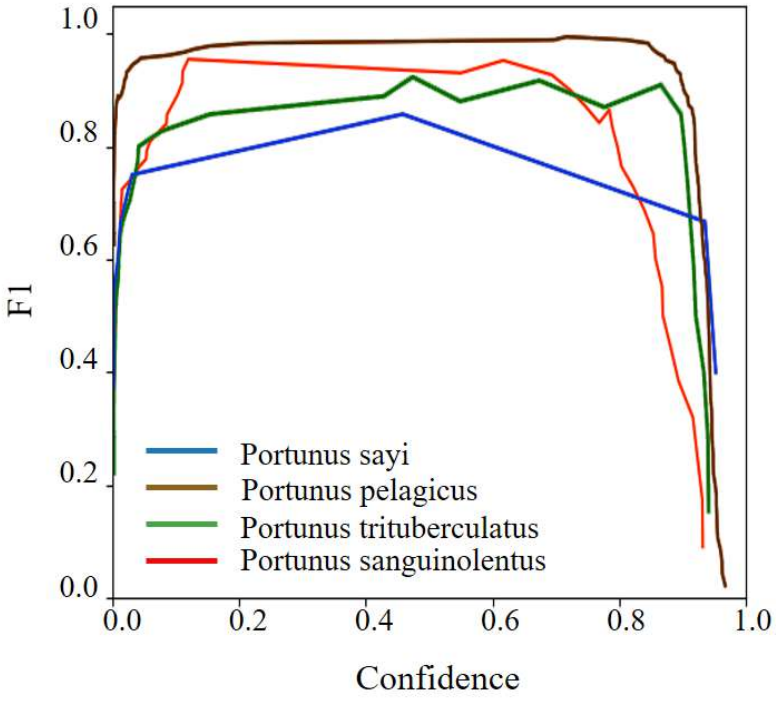

FIGURE 21. F1-confidence curves of the YOLOv8(CBAM)

6. **Comparison with other algorithms.** This study further conducted a horizontal comparison of the proposed algorithm with several existing algorithms, including the SSD, Efficientdet, and Faster R-CNN algorithms. Some of the properties of the algorithms are shown in Table 3. The YOLOv8(CBAM) model, with the addition of the CBAM attention mechanism, had a 185-layer neural network structure with 11,167 learnable parameters.

TABLE 2. Experimental results demonstrating the influence of batch size and epoch number on the mAP value

Hyperparameter	**Epoch number**			
Batch size	**25**	**50**	**75**	**100**
8	37.6%	56.2%	78.9%	80.3%
16	33.1%	66.2%	87.2%	88.6%
32	43.9%	54.8%	90.0%	83.9%
64	33.4%	66.9%	80.7%	91.4%
128	41.3%	51.6%	**93.2**%	92.89%

The computational complexity of this model was 28.817 GFLOPs, meaning that 2.8817 billion floating-point operations were performed during one forward propagation (inference). This algorithm used only 10 MB of memory. The Efficientdet model was the only algorithm that surpassed the YOLOv8(CBAM) and YOLOv5(CBAM) model in terms of FLOPs. In addition, YOLOV8(CBAM) takes up less memory than YOLOV5. This indicates that the YOLOv8(CBAM) algorithm's detection accuracy significantly exceeded that of the others, as shown in Figure 21.

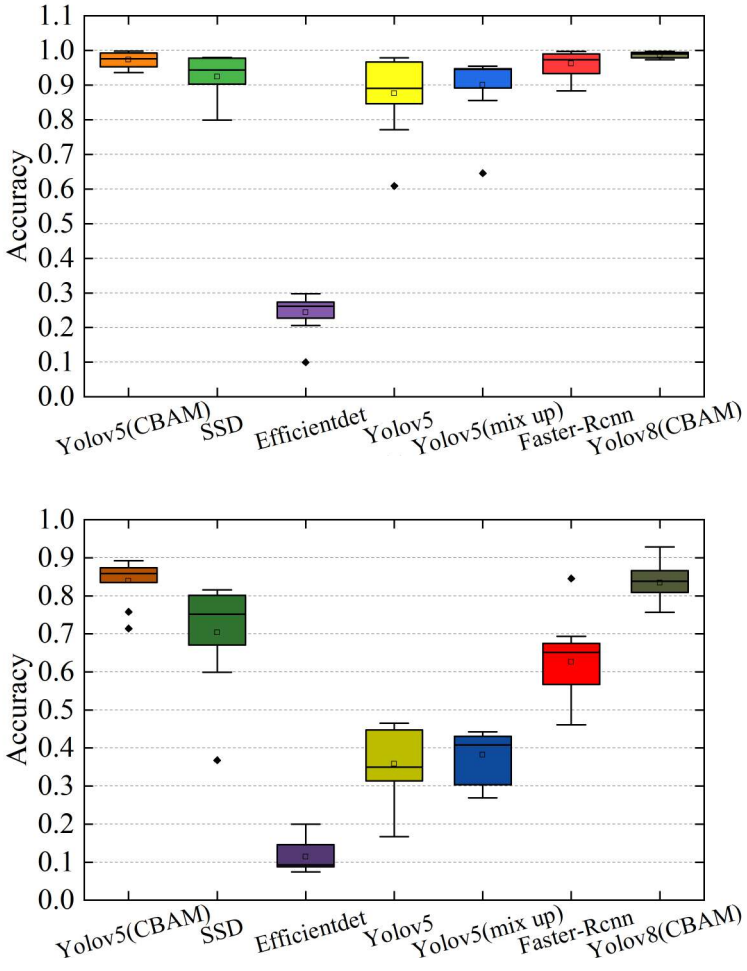

FIGURE 22. Test accuracies of different algorithms

TABLE 3. Parameters of different algorithms

Algorithm	Layer number	Parameter number	GFLOPs	Volume size (10^6)
SSD	56	26,285,486	62.747	92.78
Efficientdet	312	3,874,217	5.234	10.3
Faster R-CNN	40	137,098,724	370.210	10.32
YOLOv5	484	47,056,765	115.918	10.15
YOLOv5 (Mixup)	484	47,056,765	115.918	10.36
YOLOv5 (CBAM)	226	7,063,828	16.0	10.21
YOLOv8 (CBAM)	185	11,167	28.817	10

Considering all factors comprehensively, the proposed YOLOv8(CBAM) algorithm demonstrated several advantages for the *P. pelagicus* category, including high accuracy and low computational complexity.

A lower loss on the validation dataset indicated that this weight exhibited a more desirable generalization on the validation dataset. However, this was not directly related to the accuracy of the model's predictions. Five sets of weights with lower losses were selected for each algorithm separately. The accuracy results for the two runs are shown in Figure 21. The mean and standard deviation could not describe this distribution because it was non-Gaussian. Possible reasons for this could include the dataset containing photos of *Portunus* crabs captured under low-visibility conditions and the fact that the contours of *Portunus* crabs were not always prominent.

In Figure 22, the red diamonds represent outliers, and the blue dots represent the mean values. Figure 22(a) shows the recognition accuracy of each algorithm for the *P. pelagicus* category. The results demonstrated that the SSD algorithm achieved an accuracy of 97.8%, the Efficientdet had an accuracy of 29.8%, and the Faster-Rcnn had an accuracy of 99.0%. Therefore, the difference in accuracy between the algorithms was small. The EfficientDet had outliers at 9.9%, the YOLOv5 had outliers at 60.9%, and the YOLOv5 (Mixup) had outliers at 64.53%, indicating a certain degree of instability of all three algorithms. Figure 22(b) shows the recognition accuracy of the algorithms for the recognition of a category as a whole. The overall recognition accuracies of the SSD, Efficientdet, and Faster-Rcnn algorithms were all less than 85% for the upper bound digit and median of the recognition accuracies. Although the YOLOv5(CBAM) algorithm had two outliers after optimization in terms of the overall model prediction, that did not affect the excellent recognition prediction capability of the YOLOv5(CBAM) algorithm. YOLOv8(CBAM) demonstrates even better stabilization than YOLOv5(CBAM).

7. **Conclusion.** In this study, Convolutional Neural Network (CNN) models are used for the regression and classification of different species. The appearance characteristics of the four pike crab species, including *P. pelagicus*, *P. sanguinolentus*, *P. trituberculatus*, and *P. sayi* are described in detail.

The main contributions of this study are as follows:

(1) The YOLOv5 model and data-enhanced model were used separately for the recognition of *P. pelagicus*, and good recognition results were obtained. The F1 score values of 0.79 and 0.89 and mAP values of 88.88% and 94.67% were achieved by the YOLOv5 model and data-enhanced model, respectively. The two algorithms exhibited an average overall pike crab identification ability, with the mean average precision (mAP) values not exceeding 50%. Therefore, additional efforts are necessary to enhance the algorithms for multi-species, multi-category detection capabilities further to improve their recognition performance.

(2) The Convolutional Block Attention Module (CBAM) was incorporated into the framework based on the YOLOv5 and YOLOv8 algorithms to enhance the performance of deep recognition algorithms. The number of *P. pelagicus* samples in the dataset was 865, and for *P. pelagicus*, the proposed model achieved an mAP value of 99.2% and 99.9% respectively. The proposed YOLOv8 model had the highest overall mAP value of 93.2% at a batch size of 128 and an epoch number of 75. The overall average accuracy of the proposed model was 0.9289. This implied that the proposed model could reduce the false alarm rate during the recognition process and effectively differentiate between the target and non-target objects.

(3) Boundary conditions for the model parameters were explicitly defined. The modified YOLOv8 had a 185-layer neural network structure with a computational complexity of 28.817 GFLOPs, and the algorithm used only 10 MB of memory. Compared to the SSD, Faster-Rcnn, and enhanced YOLOv5 algorithms, the enhanced YOLOv8 showed significant improvements. The proposed algorithm was comparable to the Efficientdet algorithm in terms of computational complexity and the number of learnable parameters; however, its mAP was improved by 63.09%.

In summary, the YOLOv8 algorithm with the CBAM attention mechanism could achieve outstanding performance in recognizing distant sea pike crabs. This algorithm had certain prospects for scientific research and practical applications and could be applied to deep learning tasks in ecological monitoring, conservation efforts, and related fields.

Acknowledgment. This study was funded by the Fujian Industrial Technology Development and Application Project (Grant No. 2021I0024); Fujian Provincial Key Laboratory of Marine Fishery Resources and Eco-environment (Grant No. 202301).

REFERENCES

[1] Y. Ma, Y. Peng, and T. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.

[2] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A spectral convolutional neural network model based on adaptive Fick's law for hyperspectral image classification," *Computers, Materials & Continua*, vol. 79, no. 1, 2024.

[3] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, p. 40, 2019.

[4] K. M. Knausgård, A. Wiklund, T. K. Sørdalen, K. T. Halvorsen, A. R. Kleiven, L. Jiao, and M. Goodwin, "Temperate fish detection and classification: a deep learning based approach," *Applied Intelligence*, pp. 1–14, 2022.

[5] G. Guénard, J. Morin, P. Matte, Y. Secretan, E. Valiquette, and M. Mingelbier, "Deep learning habitat modeling for moving organisms in rapidly changing estuarine environments: A case of two fishes," *Estuarine, Coastal and Shelf Science*, vol. 238, p. 106713, 2020.

[6] F. Han, J. Yao, H. Zhu, and C. Wang, "Marine organism detection and classification from underwater vision based on the deep cnn method," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–11, 2020.

[7] J. A. Jose, C. S. Kumar, and S. Sureshkumar, "A deep multi-resolution approach using learned complex wavelet transform for tuna classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6208–6216, 2022.

[8] S. C. Mana and T. Sasipraba, "An intelligent deep learning enabled marine fish species detection and classification model," *International Journal on Artificial Intelligence Tools*, vol. 31, no. 01, p. 2250017, 2022.

[9] M. Tan, D. Langenkämper, and T. W. Nattkemper, "The impact of data augmentations on deep learning-based marine object classification in benthic image transects," *Sensors*, vol. 22, no. 14, p. 5383, 2022.

[10] C. R. Purcell, A. J. Walsh, A. P. Colefax, and P. Butcher, "Assessing the ability of deep learning techniques to perform real-time identification of shark species in live streaming video from drones," *Frontiers in Marine Science*, vol. 9, p. 981897, 2022.
[11] L. Zhang, B. Xing, W. Wang, and J. Xu, "Sea cucumber detection algorithm based on deep learning," *Sensors*, vol. 22, no. 15, p. 5717, 2022.
[12] M. Al Duhayyim, H. M. Alshahrani, F. N. Al-Wesabi, M. Alamgeer, A. M. Hilal, and M. A. Hamza, "Intelligent deep learning based automated fish detection model for uwsn," *CMC-Computers Materials & Continua*, vol. 70, no. 3, pp. 5871–5887, 2022.
[13] S. Cao, D. Zhao, X. Liu, and Y. Sun, "Real-time robust detector for underwater live crabs based on deep learning," *Computers and Electronics in Agriculture*, vol. 172, p. 105339, 2020.
[14] J. T. Ridge, P. C. Gray, A. E. Windle, and D. W. Johnston, "Deep learning for coastal resource conservation: automating detection of shellfish reefs," *Remote Sensing in Ecology and Conservation*, vol. 6, no. 4, pp. 431–440, 2020.
[15] G. Piazza, C. Valsecchi, and G. Sottocornola, "Deep learning applied to sem images for supporting marine coralline algae classification," *Diversity*, vol. 13, no. 12, p. 640, 2021.
[16] I. Martinsen, A. Harbitz, and F. M. Bianchi, "Age prediction by deep learning applied to greenland halibut (reinhardtius hippoglossoides) otolith images," *Plos One*, vol. 17, no. 11, p. e0277244, 2022.
[17] W. Vickers, B. Milner, D. Risch, and R. Lee, "Robust north atlantic right whale detection using deep learning models for denoising," *The Journal of the Acoustical Society of America*, vol. 149, no. 6, pp. 3797–3812, 2021.
[18] P. C. Bermant, M. M. Bronstein, R. J. Wood, S. Gero, and D. F. Gruber, "Deep machine learning techniques for the detection and classification of sperm whale bioacoustics," *Scientific Reports*, vol. 9, no. 1, p. 12588, 2019.
[19] M. Zhong, M. Castellote, R. Dodhia, J. Lavista Ferres, M. Keogh, and A. Brewer, "Beluga whale acoustic signal classification using deep learning neural network models," *The Journal of the Acoustical Society of America*, vol. 147, no. 3, pp. 1834–1841, 2020.
[20] B. Han, Z. Hu, Z. Su, X. Bai, S. Yin, J. Luo, and Y. Zhao, "Mask_lac r-cnn for measuring morphological features of fish," *Measurement*, vol. 203, p. 111859, 2022.
[21] C.-C. Chang, Y.-P. Wang, and S.-C. Cheng, "Fish segmentation in sonar images by mask r-cnn on feature maps of conditional random fields," *Sensors*, vol. 21, no. 22, p. 7625, 2021.
[22] V. Kandimalla, M. Richard, F. Smith, J. Quirion, L. Torgo, and C. Whidden, "Automated detection, classification and counting of fish in fish passages with deep learning," *Frontiers in Marine Science*, vol. 8, p. 2049, 2022.
[23] A. Lumini and L. Nanni, "Deep learning and transfer learning features for plankton classification," *Ecological Informatics*, vol. 51, pp. 33–43, 2019.
[24] C. R. Conrady, Ş. Er, C. G. Attwood, L. A. Roberson, and L. de Vos, "Automated detection and classification of southern african roman seabream using mask r-cnn," *Ecological Informatics*, vol. 69, p. 101593, 2022.
[25] N. F. F. Alshdaifat, A. Z. Talib, and M. A. Osman, "Improved deep learning framework for fish segmentation in underwater videos," *Ecological Informatics*, vol. 59, p. 101121, 2020.
[26] W. Xu, J. Niu, W. Gan, S. Gou, S. Zhang, H. Qiu, and T. Jiang, "Identification of paralytic shellfish toxin-producing microalgae using machine learning and deep learning methods," *Journal of Oceanology and Limnology*, vol. 40, no. 6, pp. 2202–2217, 2022.
[27] S.-S. Baek, J. Pyo, Y. S. Kwon, S.-J. Chun, S. H. Baek, C.-Y. Ahn, H.-M. Oh, Y. O. Kim, and K. H. Cho, "Deep learning for simulating harmful algal blooms using ocean numerical model," *Frontiers in Marine Science*, vol. 8, p. 729954, 2021.
[28] M. Martin-Abadal, A. Ruiz-Frau, H. Hinz, and Y. Gonzalez-Cid, "Jellytoring: real-time jellyfish monitoring based on deep learning object detection," *Sensors*, vol. 20, no. 6, p. 1708, 2020.