Research Article

Classification of Shellfish Recognition Based on Improved Faster R-CNN Framework of Deep Learning

Yiran Feng ^{1,2}, Xueheng Tao ², and Eung-Joo Lee ¹

¹Department of Information Communication Engineering, Tongmyong University, Busan 48520, Republic of Korea

²National Engineering Research Center of Seafood, Dalian Polytechnic University, Qinggongyuan 1, Ganjingzi, Dalian, Liaoning 116034, China

Correspondence should be addressed to Xueheng Tao; xhtao@dlpu.edu.cn and Eung-Joo Lee; ejlee@tu.ac.kr

Received 5 April 2021; Accepted 1 June 2021; Published 9 June 2021

Academic Editor: Francesc Pozo

Copyright © 2021 Yiran Feng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In view of the current absence of any deep learning algorithm for shellfish identification in real contexts, an improved Faster R-CNN-based detection algorithm is proposed in this paper. It achieves multiobject recognition and localization through a second-order detection network and replaces the original feature extraction module with DenseNet, which can fuse multilevel feature information, increase network depth, and avoid the disappearance of network gradients. Meanwhile, the proposal merging strategy is improved with Soft-NMS, where an attenuation function is designed to replace the conventional NMS algorithm, thereby avoiding missed detection of adjacent or overlapping objects and enhancing the network detection accuracy under multiple objects. By constructing a real contexts shellfish dataset and conducting experimental tests on a vision recognition seafood sorting robot production line, we were able to detect the features of shellfish in different scenarios, and the detection accuracy was improved by nearly 4% compared to the original detection model, achieving a better detection accuracy. This provides favorable technical support for future quality sorting of seafood using the improved Faster R-CNN-based approach.

1. Introduction

As a major maritime country, China covers a sea area of up to 4.73 million m², which comprises 280 million ha of shelf fishing grounds, 2.6 million ha of shallow aquaculture ponds, 17.47 million ha of inland waters, and 67 million ha of untapped resources suitable for fisheries such as saline-alkaline lands [1]. Unlike planting and animal husbandry, aquaculture is a weak and lagging industry in China. Thus, it is imperative to increase investment in fisheries in order to strengthen the development and utilization of waters and to improve the comprehensive production capacity of the aquaculture industry. Accelerating the growth mode transformation requires substantial novel technologies and methods by which the means and processes of production can be improved.

Among aquatic products, shellfish output accounts for an increasing proportion of total aquaculture production in China, which holds a world leading position in terms of both

culture area and output. In the case of Zhangzi Island, China's largest scallop bottom aquaculture base, its scallop production in 2019 reached over 200,000 t. Regarding the sorting process in shellfish production, substantial manual input is required, which restricts the large-scale development of the shellfish industry severely. First of all, the manual environment is harsh and has low production efficiency. Secondly, due to the shortage of labor, the demands in shellfish production areas cannot be met during the peak season. Thirdly, corporate development is restricted greatly by the rise in labor wages and the jump in product costs. With the ever-demanding requirements on the quality and production efficiency of aquatic products, more efficient and precise means are needed for completing the shellfish sorting, in order to enhance the degree of production automation. In recent years, computer and robotics technologies have gained rapid development, of which computer vision has been applied extensively in numerous fields of industrial production including automobile, electromechanical, food,

logistics, and manufacturing industries. By using computer vision for the identification and localization of scallops and subsequent sorting, the production efficiency can be improved while ensuring the quality of aquatic products. In the meantime, it also reduces the demand for labor.

The computer vision-based shellfish detection and identification technology is typical image analysis, understanding, and classification problem [2–5]. In-depth investigation and solution of this problem involve theoretical knowledge from multiple disciplines such as pattern recognition, image processing and analysis, computer vision, artificial intelligence, deep learning, and computer graphics. Accordingly, this paper proposes a Faster R-CNN-based algorithm for shellfish classification and detection.

In most cases, the fished aquatic products are manually sorted for classification among different species and grading within the same species. Generally, postharvest processing relies on the manual identification of fish species and appearance quality. To reduce the labor demand and enhance the processing efficiency, computer vision can be applied for noncontact counting and measurement of aquatic products. In this way, the processing efficiency and counting accuracy can be improved without damaging the aquatic products under inspection. Recent years have witnessed many applications in this respect. Through neural network training, automatic identification of marlin biological features was achieved, which simplified computation in the recognition process. According to Cha et al. [6], computer vision-based techniques were developed to overcome the limitations of visual inspection by trained human resources and to detect structural damage in images remotely. In 2021, Long et al. [7] presented a novel deep learning based damage evaluation approach by using speckled images. A deep convolutional neural network (DCNN) for predicting the stress intensity factor (SIF) at the crack tip is designed. Based on the proposed DCNN, the SIF can be automatically predicted through computational vision. In 2013, Costa [8] applied computer vision to automatically classify the size, gender, and skeletal abnormalities of groupers. The least squares modeling-based multielement technology, which integrated image analysis and contour morphology (elliptical Fourier analysis), was applicable for sorting and processing live fish. Applying computer vision, Ma et al. [9] designed a set of equipment for sea cucumber grading and counting. After acquiring the projection images of sea cucumbers on the conveyor belt, their specifications were determined according to the size of the image area, thereby accomplishing grading and counting. In the absence of overlapping images, the identification accuracy reached 100%. Wang et al. [10] obtained the fish contours close to the standard shapes through deformation correction on curved fish images and then used the equidistant polar mapping to obtain the mapping maps, followed by extraction of intervals between local and adjacent extreme points in the maps. Finally, the matching degree was computed algorithmically to accomplish the fish identification. In the object detection task, the region-CNN (R-CNN) proposed by Girshick [11] is an important reference method, with plenty of algorithms borrowing ideas from R-CNN. Combining object region

proposal with CNN classification, the R-CNN model extracts 2,000 candidate regions from the input images using the selective search algorithm and then performs feature extraction through the CNN network. Afterward, it utilizes a trained classifier to determine whether the candidate regions contain the target objects and eventually adjusts the proposals with a regressor. Girshick [12] modified the R-CNN by incorporating the ideas of SPP-net to put forward the Fast R-CNN model. The new model performs feature extraction on the entire image only once and then maps it onto the proposals, thereby avoiding the wastage of time caused by repeated feature extraction. Moreover, it performs training through a combination of Softmax classification and box regression, which eliminates the feature storage and enhances the space and time utilization. In the meantime, convolutional features can be shared among the classification and regression tasks. Ren et al. [13] proposed the use of deep learning for regional proposal network (RPN), where RPN and Fast R-CNN were combined to construct a novel Faster R-CNN model, thereby improving the overall detection performance.

This paper innovatively introduces a Faster R-CNN network for the identification, localization, and experimentation of four shellfish species in order to address the absence of a deep learning algorithm for shellfish identification in real contexts [14–16]. Modification is made on the Faster R-CNN framework based on the features of various shellfish species, the original feature extraction module is replaced with a densely connected network (DenseNet), and the multilevel features of fusion objects are extracted to add expressive power to the features. Meanwhile, Soft-NMS is used instead of the original proposal merging strategy, and an attenuation function is designed to enhance the object box localization accuracy. Furthermore, shellfish datasets are collected and built in the real context, and the identification and localization of various shellfish species are implemented eventually in the working environment through training, which achieves excellent accuracy.

2. Materials and Methods

2.1. Faster R-CNN Architectures. As a current mainstream two-stage detection network, Faster R-CNN is a combination of RPN and Fast R-CNN, which enables the output of detection categories and box positioning at each stage [17, 18]. Depending on the network architecture, the Faster R-CNN can be divided into three parts: the basic feature extraction network, the RPN, and the detection network. The specific steps of the algorithm are described below. Figure 1 presents the algorithmic framework.

2.1.1. Feature Extraction Network. Among the above three parts, the feature extraction network comprises convolutional neural networks (CNNs), whose fundamental architecture includes convolutional, pooling, fully connected, and softmax classification layers. Different CNNs produce varying effects on the accuracy and duration of detection. There are three common feature extraction networks for

FIGURE 1: Faster R-CNN architectures.

Faster R-CNN, namely, ZFNet, VGG-16, and ResNet. (1) ZFNet [19], as a slight improvement on AlexNet, retains more features by reducing the number and stride of convolution kernels. It can also be inferred that with increasing network depth, better performance of feature extraction indicates a more remarkable extraction effect. (2) VGG-16 [20] is formed by repeatedly stacking 3×3 convolution kernels and 2×2 maximum pooling layers, which verifies the relationship between the depth and performance of CNNs. Despite a simple architecture and favorable feature extraction effect, it has large parameters, substantial training features, and high hardware requirements. (3) ResNet [21], short for residual network, solves the gradient disappearance problem accompanying the increase in network depth by designing a residual module, which implements identity mapping by connection route. Capable of extracting deeper features of target objects, the network achieves a remarkable identification effect.

2.1.2. Feature Extraction of the Datasets. There are many aspects of shellfish that can be extracted, such as colour or greyscale features, texture features, shape or form features, structural relationships, frequency features, and boundary area features. How to extract the features of the object to be recognised is the key to the target recognition problem. It was found that the pixel values of the shellfish greyscale map were not significantly different from the interfering objects, but the shapes were significantly different. Therefore, in this study, the contour boundary of the target object is extracted, and the classification features are extracted from the contour boundary.

The RPN is responsible for extracting candidate regions [22, 23], whose architecture is illustrated in Figure 2. It receives the convolutional feature maps from the basic feature extraction network and convolves each 3×3 sliding window into a 256-dimensional feature vector via convolution kernels. Through 1×1 convolution, each sliding window is output as two fully connected layers, i.e., the box-classification layer (cls layer) and the box regression layer (reg layer). The cls layer outputs the probabilities of

belonging to the foreground and background, while the reg layer outputs four parameters, i.e., the centroid coordinates x, y of the prediction range, and the length and width w, h . The presence or absence of the object is determined based on the receptive field corresponding to the sliding window center [24]. Given the length and width discrepancies among objects, nine anchors are generated using three window scales (8, 16, 32) and three aspect ratios (1:2, 1:1, 2:1) at a benchmark window size of 16, in order to perform multi-scale multipoint sampling of feature maps. The principle of using a canny algorithm to extract the grayscale image edge is to calculate the gradient histogram of the image and accumulate the number of pixels along the gradient increment direction, set the gradient value at 80% of the total number of pixels as the high threshold t_h and 40% of the gradient value of the high threshold as the low threshold t_l , mark the points in the gradient histogram with a gradient greater than the high threshold t_l as edge points, and the points with gradient value less than the low threshold t_h as background points. And points with gradient values between t_l and t_h are regarded as edge points if there are already marked edge points in their 8-connected neighbourhood, otherwise they are regarded as background points. In order to reduce the noise while preserving the effective boundary, the high threshold was set to 0.4 and the low threshold was set to 0.1, and the extraction effect was satisfactory.

The loss function of RPN is defined as follows:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{reg}}} \sum_i p_i^* L_{\text{reg}}(t_i, t_i^*), \quad (1)$$

where subscript i stands for the anchor index, p_i represents the probability distribution of each anchor corresponding to $k+1$ categories (k categories + 1 background), p_i^* indicates whether an object is contained (1 if so, and 0 otherwise), N_{cls} denotes the minibatch size (usually 256), and N_{reg} is the number of anchors. λ denotes the balance weight, whose value is 1. Besides, t_i represents the proposal box coordinates $\{t_x, t_y, t_w, t_h\}$ and t_i^* represents the marker box coordinates. The specific parameter values are described below:

FIGURE 2: Region proposal networks.

$$\begin{aligned}
 t_x &= \frac{x - x_a}{w_a}, t_y = \frac{(y - y_a)}{h_a}, \\
 t_w &= \log\left(\frac{w}{w_a}\right), t_h = \log\left(\frac{h}{h_a}\right), \\
 t_x^* &= \frac{x^* - x_a}{w_a}, t_y^* = \frac{y^* - y_a}{h_a}, \\
 t_w^* &= \log\left(\frac{w^*}{w_a}\right), t_h^* = \log\left(\frac{h^*}{h_a}\right).
 \end{aligned} \tag{2}$$

x, x_a, x^* (same for y, w, h) denote the location parameters of the proposal, anchor, and calibration boxes, respectively. The classification loss L_{cls} is a logarithmic loss function of object and nonobject:

$$L_{cls}(p_i, p_i^*) = -\log[p_i p_i^* + (1 - p_i^*)(1 - p_i)]. \tag{3}$$

The regression loss is $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$, where

$$R(t_i - t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2, & |t_i - t_i^*| < 1, \\ |t_i - t_i^*| = 0.5, & \text{others.} \end{cases} \tag{4}$$

The RPN performs box regression via the loss function and merges the prediction boxes output by the detector through nonmaximum suppression, which is then connected to the Fast R-CNN as input [25–27]. The candidate regions generated by RPN are mapped onto the feature map output by the feature extraction network. For the candidate region input of any size, the ROI pooling layer can obtain a fixed-dimensional output, which then obtains the final result through the cls and reg layers.

2.2. Improved Faster R-CNN

2.2.1. Dense Block Network. Although a deeper network allows the extraction of deeper semantic information, there will be an inevitable increase in parameters with the deepening of a network [28, 29]. As a result, a series of problems are brought to the network optimization and the experimental hardware. The datasets built specifically for the shellfish classification and detection algorithm herein have

small sample sizes so that the network training easily leads to overfitting. The use of DenseNet as the feature extraction network helps solve problems [30].

As a novel network architecture, DenseNet draws on the ideas of ResNet. The most intuitive difference between the two architectures lies in the varying transfer functions for various network blocks.

$$x_l = H_l(x_{l-1}) + x_{l-1}, \tag{5}$$

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]). \tag{6}$$

As is clear from (6) describing the transfer function of the ResNet, the l th layer output of the network equals the nonlinear variation of $l-1$ th layer output plus the $l-1$ th layer output. Contrastively, the l th layer output of a DenseNet block is the set of nonlinear transformations output by all previous layers. Figure 3 depicts the Dense Blocks of the DenseNet.

The convolutions in each Dense Block are all interconnected [31]. H indicates that each input is convolved with k -dimensional 3×3 kernels using Batch Norm and ReLU in order to ensure that each node can output feature maps of the same dimension. k denotes the thickness of the feature maps output by each convolution layer. Compared to other networks, whose thickness of output feature maps can reach hundreds or even thousands, the DenseNet has an overall thickness of 32 only. For DenseNet, the dense connection between its blocks allows effective utilization of the shallow and deep layer features so that its efficiency and narrowness can be ensured, and its complexity and computational burden can be reduced greatly. Figure 4 details the parameters for connecting nodes H_l .

In this paper, four 121-layer Dense Blocks are used to constitute the feature extraction network, which accomplishes object identification and localization by connecting the RPN and RoI pooling layer following the removal of the fully connected and classification layers. Table 1 lists the parameters of the four-DenseBlock architecture.

2.2.2. Nonmaximum Suppression. In essence, nonmaximum suppression (NMS) aims to search for the local maximum and suppress the nonmaximum elements, which is an important step of the detection process [32, 33]. Faster R-CNN generates a series of detection boxes $B = \{b_1, \dots, b_N\}$ in an image and the corresponding box score set C_i . NMS algorithm selects the detection box M in the object detection process prior to the maximum score, which is then subjected to intersection over union (IoU) computation with the remaining detection boxes. The detection box M will be suppressed if the result is greater than the set threshold N_t . The NMS algorithm formula is as follows:

$$C_i = \begin{cases} C_i, & \text{IoU}(M, b_i) < N_t, \\ 0, & \text{IoU}(M, b_i) \geq N_t, \end{cases} \tag{7}$$

where the IoU is computed by the following formula:

FIGURE 3: Dense Block architecture.

FIGURE 4: Node H_1 parameter.

TABLE 1: DenseNet structure parameters.

Structure	Parameter
Layers	7 * 7 conv, stride 2
Pooling	3 * 3 max pool, stride 2
Dense block (1)	$\begin{bmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \\ 1 * 1 \text{ conv} \end{bmatrix} * 6$
Transition layers (1)	$2 * 2$ average pool, stride 2
Dense block (2)	$\begin{bmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \\ 1 * 1 \text{ conv} \end{bmatrix} * 12$
Transition layers (2)	$2 * 2$ average pool, stride 2
Dense block (3)	$\begin{bmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \\ 1 * 1 \text{ conv} \end{bmatrix} * 24$
Transition layers (3)	$2 * 2$ average pool, stride 2
Dense block (4)	$\begin{bmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \\ 1 * 1 \text{ conv} \end{bmatrix} * 16$
Transition layers (4)	$2 * 2$ average pool, stride 2
Classification layers	7 * 7 global average pool 1000D fully connected, softmax

$$\text{IoU} = \frac{(A \cap B)}{(A \cup B)}, \quad (8)$$

where A and B represent two overlapping detection boxes:

As is clear from (7), the NMS algorithm zeroes with the detection box that is adjacent to M and greater than the threshold. If an object under detection appears in the overlapping region, the NMS algorithm will fail to detect the object, thereby reducing the accuracy of the detection model.

To address this problem, the conventional NMS algorithm is replaced with the Soft-NMS, where an attenuation function is designed based on the IoU between adjacent detection boxes instead of setting their scores to zero,

thereby ensuring accurate identification of adjacent objects. The Soft-NMS algorithm is expressed as follows:

$$C_i = \begin{cases} C_i, & \text{IoU}(M, b_i) < N_t, \\ C_i(1 - \text{IoU}(M, b_i)), & \text{IoU}(M, b_i) \geq N_t. \end{cases} \quad (9)$$

To accomplish the shellfish classification and detection in real contexts, modifications are made on the front-end feature extractor and the tail-end regressor of the Faster R-CNN detection algorithm. The algorithmic flow is provided in Algorithm 1.

2.3. Experimental Conditions. In order for the experiments to match actual production, the shellfish identification experiments in this paper were carried out on a deep learning based seafood sorting robot production line, as shown in Figure 5.

3. Results and Discussion

3.1. Data Sets Making and Processing. Shellfish datasets are collected independently in this study for verifying the effectiveness of the proposed Faster R-CNN algorithm, which are categorized into 4 species (scallops, mussels, conch, and clams) and contain 4,218 images in total (see Table 2).

The data features different light intensities, occlusions, complex backgrounds, and multiple objects in order to ensure that the detection model covers common real-life shellfish. Furthermore, 50% of the dataset is mirror augmented, and the other 50% is translationally augmented, followed by data annotation on the LabelImg software, as shown in Figure 6. The augmented datasets contain 8,436 images, 90% of which are used as the training sets and 10% as testing sets.

For accurate comparison of the improved algorithm in complex backgrounds, difficult samples are selected from the testing datasets based on the following criteria: there are more than four detection objects in an image, and the shellfish is affected by illumination and occlusion. Table 3 details the difficult sample testing data. Some images in the difficult samples have multiple difficult attributes, as shown in Figure 7(c). Apart from containing multiple shellfish

- (1) Input image A , adjust the image size, and output image B with a specified size $M \times N$.
- (2) Using B as the input of the feature extraction module, obtain a multilevel fused feature map C via DenseNet.
- (3) Using C as the input of RPN, obtain 300 proposals: D by the sliding window method. RPN changes the generated anchors through box regression so that they can be closer to the marker boxes.
- (4) Using C and D as the inputs of RoI, obtain a mapping map E that is between the proposal and the feature map.
- (5) Output E separately to the classifier and the regressor. The classifier achieves classification and identification of E using Softmax, while the regressor further corrects the boxes by Soft-NMS regression. Finally, classify and localize the objects.

ALGORITHM 1: Improved Faster R-CNN process.

FIGURE 5: Seafood sorting robot production line based on deep learning.

TABLE 2: Data parameters.

Data set	Conch	Scallop	Mussel	Clam
Training set	1688	1827	2110	1967
Test set	188	203	234	219
Total	1876	2030	2344	2186

FIGURE 6: Data expansion and labeling.

TABLE 3: Test sets of different scenarios.

Difficult samples	Illumination effect	Occlusion effect	Complex backgrounds	Multi-object
Quantity	81	58	29	153

species, they are also shot at nighttime. These images are both multiobject categorization samples and illumination-affected samples.

3.2. Experimental Parameters and Evaluation Indexes. Given the huge video memory requirements of DenseNet for fusing all feature maps prior to the current stage during

FIGURE 7: ResNet-Faster R-CNN detection result. (a) Normal detection. (b) Occluded. (c) Missed detections.

transmission, efficient memory implementation is adopted. Specifically, two preallocated shared memory storage locations are proposed, which are used for storing the shared feature maps to be connected. All intermediate outputs are allocated to these memory modules during the forward transmission, while during the reverse transmission, the transfer function is recomputed and updated as needed. This strategy enables DenseNet to run in a single GPU with less computational overhead.

Experiments are implemented on the Tensorflow and run on a computer with i7 6700 processor, a GeForce RTX2080Ti GPU (11G video memory), and 32 GB RAM. The experimental data includes the shellfish appearances collected in different scenarios such as factories, Internet, and seafood markets, which are manually annotated using the Labellmg software—experimentation on a deep learning based seafood sorting robot line.

The convolution loss process and test accuracy in the training process are shown in Figure 8. After about 2000 convolutions, the overall loss can be rapidly reduced and stabilized between 0.1 and 0.2, and the test accuracy is also stabilized at about 80%.

The evaluation index is average precision (AP), which refers to the area enclosed by the precision-recall (P-R) curve. In a P-R curve, P represents the precision and R represents the recall rate. They are computed by the following formulas:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad (10)$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}},$$

where N_{TP} (true positives) denotes the number of positive samples that are identified as positive, N_{FP} (false positives) denotes the number of negative samples that are incorrectly identified as positive, and N_{FN} denotes the number of positive samples that are incorrectly identified as negative. Besides, AP represents the identification accuracy of a single category. A higher AP value indicates better performance of the network model. In addition, mAP (mean average

FIGURE 8: Training process of loss and testing process of validation accuracy.

precision) represents the overall identification accuracy of all categories, and its relationship with AP is expressed by

$$mAP = \frac{(\sum_{i=1}^C AP_i)}{C}. \quad (11)$$

3.3. Results Comparison and Analysis. Trainings are performed separately with ResNet and DenseNet as the feature extraction networks, and the network models are evaluated using the testing sets. Table 4 lists the obtained AP values for various shellfish species, whereas Table 5 presents the detection and comparison results of difficult samples.

According to the detection results, the Faster R-CNN with ResNet exhibits an over 77% mAP in various shellfish detections. Figure 7 depicts the partial detection results. In Figure 7(a), distinct object features and sufficient illumination are noted so that the model can achieve remarkable detection performance. The scallop in Figure 7(b) is partially occluded and contains multiple shellfish species, where the detection effect is satisfactory. In contrast, missed detection is present in Figure 7(c). Clearly, despite certain detection capabilities, ResNet still yields some missed and false

TABLE 4: Test results of different network models.

Network	Conch (%)	Scallop (%)	Clam (%)	Mussel (%)
ResNet	71.8	76.2	81.6	79.8
DenseNet	75.1	80.3	86.2	82.9
DenseNet + Soft-NMS	75.1	80.3	88.3	84.6

TABLE 5: Test results of difficult sample.

Network	Multi-object (%)	Complex backgrounds (%)	Illumination effect (%)	Occlusion effect (%)
Original algorithm	76.5	74.9	65.8	64.6
Improved algorithm	78.9	76.8	72.2	73.4

FIGURE 9: Comparison of accuracy with different backbones. (a) Missed detection by complex conditions. (b) False detection. (c) Missed detection.

detections. The reason is that full training of ResNet is impossible due to excessively small data size, which leads to inadequate robustness in complex scenarios.

As is clear from Table 4, the use of DenseNet-121 for feature extraction yields a mAP of up to 83%, which is nearly 4% higher than the ResNet. Greatly improved detection results are noted across three shellfish species (mussels, scallops, and clams). Regarding the reasons, the testing sets of these three species contain samples with multiple objects, occlusions, and complex backgrounds, and DenseNet can extract more features of objects to attain better results. After modifying the detection boxes with Soft-NMS, the accuracy rises by 2% in the mussel and clam datasets containing multiobject samples. Suggestively, the Soft-NMS can avoid the proposal box zeroing caused by a higher repetition of detection boxes than the threshold under multiple and overlapping objects, which achieves a better detection effect. According to Table 5, the improved detection network is more robust in difficult samples than the original version,

with rather distinct improvements in two types of samples, i.e., the illumination-affected and occluded samples. In the presence of multiple objects and complex backgrounds, performance comparisons are made between the improved versus original Faster R-CNN algorithms, as shown in Figure 9.

The first row in Figure 9 presents the detection results with the original network, whereas the second row presents those with the improved network. Clearly, the original network often yields missed detections under complex conditions. Figure 9(b) exhibits a false detection made by the original network, which identifies conch and clam as a mussel. Figure 9(c) shows a case of missed detection, where the detection box fails to contain the shellfish objects precisely. As the comparisons reveal, the Faster R-CNN improved with DenseNet and Soft-NMS outperforms the original network in terms of accuracy, which allows the detection of more shellfish objects and an accurate selection of independent elements under excessively short distances

between adjacent objects. The improved Faster R-CNN exhibits a more prominent improvement in performance under complex backgrounds and multiple objects.

According to a series of experimental comparison findings, the improved Faster R-CNN algorithm has high detection accuracy in shellfish detection tasks and good robustness in different environments. Apart from extending the application range of the original algorithm, it also shows practical applicability.

4. Conclusions

A deep learning algorithm for shellfish identification is proposed in order to address the inefficiency of conventional detection algorithms under different ambient light, diverse backgrounds, and varying occlusion conditions. As a modification based on Faster R-CNN, the algorithm uses DenseNet as the feature extraction network, where the dense connection between blocks allows effective utilization of the shallow and deep layer features, thereby enhancing the shellfish detection accuracy. Meanwhile, the proposal merging strategy is optimized by using Soft-NMS instead of the original algorithm, thereby adding precision to the proposals. Furthermore, shellfish datasets are built in real contexts and then augmented to improve the robustness of the training model. The proposed detection algorithm can achieve multiobject shellfish detection in seafood processors and has preferable accuracies in complicated scenarios like illumination influence, partial occlusion, and complex background, which has a nearly 4% higher detection accuracy than the original model, exhibits a good detection performance. In the next Step I will use an improved Faster R-CNN for seafood quality detection based on the findings of this paper.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Public Science and Technology Research Funds Projects of Ocean (No. 201505029) and Natural Science Foundation of Liaoning (No. 2020-MS-273).

References

- [1] M. Kobayashi, S. Msangi, M. Batka, S. Vannuccini, M. M. Dey, and J. L. Anderson, "Fish to 2030: the role and opportunity for aquaculture," *Aquaculture Economics & Management*, vol. 19, no. 3, pp. 282–300, 2015.
- [2] Y. Liu, Q. Wang, Q. Xu, J. Feng, H. Yu, and Y. Yin, "Non-destructive detection of Flos Lonicerae treated by sulfur fumigation based on hyperspectral imaging," *Journal of Food Measurement and Characterization*, vol. 12, no. 4, pp. 2809–2818, 2018.
- [3] T. Zhang, B. Wang, P. Yan et al., "Nondestructive identification of salmon adulteration with water based on hyperspectral data," *Journal of Food Quality*, vol. 2018, pp. 1–10, 2018.
- [4] H. Wang, K. Wang, B. Wang et al., "Integration of artificial neural network modeling and hyperspectral data preprocessing for discrimination of colla corii asini adulteration," *Journal of Food Quality*, vol. 2018, pp. 1–11, 2018.
- [5] J. Dong, X. Fang, H. Wang, X. Zhang, and X. Tao, "Abalone muscle texture evaluation and prediction based on TPA experiment," *Journal of Food Quality*, vol. 2017, pp. 1–9, 2017.
- [6] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyükoztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.
- [7] X. Y. Long, S. K. Zhao, C. Jiang, W. P. Li, and C. H. Liu, "Deep learning-based planar crack damage evaluation using convolutional neural networks," *Engineering Fracture Mechanics*, vol. 246, Article ID 107604, 2021.
- [8] C. Costa, F. Antonucci, C. Boglione, P. Menesatti, M. Vandeputte, and B. Chatain, "Automated sorting for size, sex and skeletal anomalies of cultured seabass using external shape analysis," *Aquacultural Engineering*, vol. 52, no. 7, pp. 58–64, 2013.
- [9] W. Che, L. Sun, Q. Zhang et al., "Pixel based bruise region extraction of apple using vis-NIR hyperspectral imaging," *Computers and Electronics in Agriculture*, vol. 146, pp. 12–21, 2018.
- [10] B. Li, M. Cobo-Medina, J. Lecourt, N. B. Harrison, R. J. Harrison, and J. V. Cross, "Application of hyperspectral imaging for nondestructive measurement of plum quality attributes," *Postharvest Biol. Technol.*, vol. 141, no. 8–15, 2018.
- [11] R. Girshick, J. Donahue, T. Darrell et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, IEEE, Columbus, OH, USA, June 2014.
- [12] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, IEEE, Santiago, Chile, December 2015.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [14] A. J. Siddiqui, S. G. Musharraf, M. I. Choudhary, and A.-U. Rahman, "Application of analytical methods in authentication and adulteration of honey," *Food Chemistry*, vol. 217, pp. 687–698, 2017.
- [15] M. Huang, M. S. Kim, S. R. Delwiche et al., "Quantitative analysis of melamine in milk powders using near-infrared hyperspectral imaging and band ratio," *Journal of Food Engineering*, vol. 181, pp. 10–19, 2016.
- [16] M. Kamruzzaman, Y. Makino, and S. Oshita, "Rapid and non-destructive detection of chicken adulteration in minced beef using visible near-infrared hyperspectral imaging and machine learning," *Journal of Food Engineering*, vol. 170, pp. 8–15, 2016.

- [17] J. Ma, D.-W. Sun, and H. Pu, "Spectral absorption index in hyperspectral image analysis for predicting moisture contents in pork longissimus dorsi muscles," *Food Chemistry*, vol. 197, pp. 848–854, 2016.
- [18] Amctb No. Analytical Methods Committee, "Z-scores and other scores in chemical proficiency testing-their meanings, and some common misconceptions," *Analytical Methods*, vol. 8, no. 28, pp. 5553–5555, 2016.
- [19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the 13th European Conference on Computer Vision (ECCV)*, pp. 818–833, Springer, Zurich, Switzerland, September 2014.
- [20] M. Peris and L. Escuder-Gilbert, "Electronic noses and tongues to assess food authenticity and adulteration," *Trends in Food Science & Technology*, vol. 58, pp. 40–54, 2016.
- [21] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, IEEE, Las Vegas, NV, USA, July 2016.
- [22] C. Peng, K. Zhao, B. C. Lovell et al., "Faster ILOD: incremental learning for object detectors based on faster RCNN," *Pattern Recognition Letters*, vol. 140, pp. 109–115, 2020.
- [23] Z. Liu, Y. Liu, L. Wang et al., "Detection approach based on an improved faster RCNN for brace sleeve screws in high-speed railways," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4395–4403, 2019.
- [24] X. Sun, P. Wu, S. C. H. Hoi et al., "Face detection using deep learning: an improved faster RCNN approach," *Neurocomputing*, vol. 299, pp. 42–50, 2018.
- [25] M. Nan and Y. Li, "Improved Faster RCNN based on feature amplification and oversampling data augmentation for oriented vehicle detection in aerial images," *Remote Sensing*, vol. 12, no. 16, p. 2558, 2020.
- [26] Q. Liu, K. Sun, J. Peng, M. Xing, L. Pan, and K. Tu, "Identification of bruise and fungi contamination in strawberries using hyperspectral imaging technology and multivariate analysis," *Food Analytical Methods*, vol. 11, no. 5, pp. 1518–1527, 2018.
- [27] X. Ding, K. Yang, and W. Chen, "A semantics-guided graph convolutional network for skeleton-based action recognition," in *Proceedings of the 2020 the 4th International Conference on Innovation in Artificial Intelligence*, vol. 5, pp. 130–136, Xiamen, China, May 2020.
- [28] B. Chen, T. Zhao, J. Liu et al., "Multipath feature recalibration densenet for image classification," *International Journal of Machine Learning and Cybernetics*, vol. 2, pp. 1–10, 2020.
- [29] Q. Liu, X. Xiang, J. Qin et al., "Coverless image steganography based on DenseNet feature mapping," *EURASIP Journal on Image and Video Processing*, vol. 39, p. 2020, 2020.
- [30] Z. Li, Y. Lin, A. Elofsson, and Y. Yao, "Protein contact map prediction based on ResNet and DenseNet," *BioMed Research International*, vol. 2020, pp. 1–12, 2020.
- [31] E. Zhang, B. Xue, F. Cao, J. Duan, G. Lin, and Y. Lei, "Fusion of 2D CNN and 3D DenseNet for dynamic gesture recognition," *Electronics*, vol. 8, no. 12, p. 1511, 2019.
- [32] B. Chen, W. Chen, and X. Wei, "Characterization of elastic parameters for functionally graded material by a meshfree method combined with the NMS approach," *Inverse Problems in Science and Engineering*, vol. 26, no. 4, pp. 601–617, 2018.
- [33] Y. Zhou, Y. Zhang, X. Xie et al., "Image super-resolution based on dense convolutional auto-encoder blocks," *Neurocomputing*, vol. 6, 2020.