Detection Method of Marine Floating Garbage based on Improved Faster R-CNN

Jing Bai [a], Shujie Yang [b]

School of Ocean Engineering Equipment, Zhejiang Ocean University, Zhoushan, 316022, China

[a] S20080200009@zjou.edu.cn, [b] shujie.yang@zjou.edu.cn

Abstract. Traditional marine floating garbage cleaning is implemented manually. Workers have to suffer from the poor environment and intensive tasks, which reduce their efficiency. Neural network can help automate the process, and thus free workers from the heavy labor. This paper proposes a neural network network-based method to improve the accuracy of garbage identification and classification, especially for the small-sized garbage. The method combines the improved Faster R-CNN target detection model and Resnet50. According to the characteristics of marine floating garbage, the parameters of fast r-cnn are determined through repeated experiments, which improves the accuracy of small target detection. Experimental results show that the method provides a comprehensive recognition rate of up to 79,36%.

Keywords: Object Detection; Floating Garbage Detection; Deep Learning; Resnet.

1. Introduction

In recent years, marine debris pollution has become a major economic, political and environmental issue widely discussed internationally. Marine debris refers to "persistent, man-made or processed solid waste in the marine and coastal environment". Most marine debris contain substances with a very slow degradation rate, including petroleum and its products, household garbage, heavy metal substances and radionuclides. Caused serious harm, many animals in the ocean were accidentally injured or killed, seabirds, turtles, fish and marine mammals swallowed plastic waste or became entangled in waste fishing nets. In addition, persistent and cumulative pollutants in marine debris may also harm humans through the biological chain. If nothing is done, the oceans will be unloaded and humans will not be able to survive. Therefore, how to realize the automatic detection and identification of floating garbage has become the primary problem to be solved urgently.

In this paper, the improved Faster R-CNN algorithm is used to achieve high-precision detection of multiple types of marine floating debris targets. Since no publicly available datasets have been found in research on marine floating debris identification, the primary research focus of this paper is to produce datasets. Secondly, considering the influence of different backbone networks on the recognition accuracy, speed and generalization performance, this paper proposes a deep convolutional neural network algorithm based on the ResNet50 network structure to classify and identify 5 kinds of floating garbage pictures, thereby improving garbage recognition accuracy. In addition, according to the characteristics of marine floating debris, this paper determines the parameters of Faster R-CNN through repeated experiments, which improves the accuracy of small target detection.

2. Datasets

Figure 1. Datasets

The data used in this project comes from the floating garbage captured by the lakes and coasts near Zhejiang Ocean University. A total of 600 images were collected. In order to ensure the number of training samples required by the ResNet50 network, a Python program is used to perform data enhancement processing on the collected 600 images. The processing methods include horizontal flipping and upside-down reversal. Through sample set enhancement, the data set images are expanded to 2400. Then put the processed images in a folder named JPEGImages, and each image is named 00xxxx.jpg (xxxx ranges from 1-2400). Fig.1 shows some of the images in the data.

Use LabelImg to manually calibrate the floating garbage in the picture, including these five categories of floating garbage: Bottle, milk-box, plastic-bag, ball, leaf. LabelImg is a visual image calibration tool. The data sets required by target detection networks such as Faster R-CNN, YOLO, and SSD all need to use this tool to calibrate the target in the image. The generated XML file follows the format of PASCAL VOC.

3. Floating Garbage Detection Algorithm

The structure of the marine floating debris detection algorithm based on improved Faster R-CNN designed in this paper is shown in Fig.2:

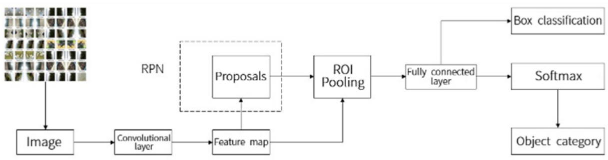

Figure 2. Faster R-CNN network architecture diagram for marine garbage identification

3.1 Feature Extraction

The feature extraction network of the Faster R-CNN algorithm used in this project is the ResNet50 network. Compared with the original VGG-16 network, the ResNet50 network improves the recognition accuracy. Train the ResNet50 network model on the data set processed in the previous step, and input a 416×416×3 picture. According to the network structure settings, the picture will be expanded to 600×600×3.

Then define the most important residual module bottleneck module, which contains three convolutions. The convolutional layer is one of the core components of the ResNet50 network. It can perform convolution operations on the mapping area to extract corresponding features. Therefore, the convolutional layer Also known as feature extraction layer. In the convolution operation, the convolution kernel and the convolution stride are two very important parameters. The convolution operation refers to the process of convolution traversal of the image with the convolution kernel. Different convolution kernels can obtain different feature maps through the convolution operation. The convolution kernel refers to an m×m square matrix, and the size is usually a matrix of 1×1, 3×3, and 5×5. The convolution kernel moves with a fixed step size, weighted and summed with the pixel values of the image, plus a bias, and a value is obtained. The sequential operations will get different values, and these different values will be placed in order to form a matrix of feature maps. The specific process of the convolution operation is shown in Fig.3.

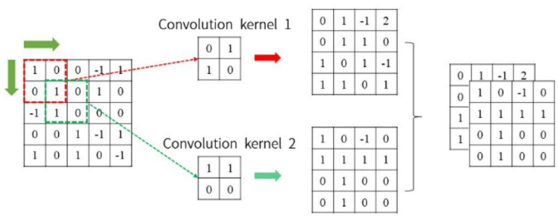

Figure 3. The process of convolution operation

The formula for the convolution operation is as follows:

$$x_j^l = f\left(\sum_{i\in M_j} x_i^{l-1} \cdot k_{ij}^l + b_j^l\right) \tag{1}$$

The convolution kernels of 1×1, 3×3, and 1×1 is used here, which are used to compress dimensions, convolution processing, and restore dimensions, respectively. Inplane is the number of input channels, plane is the number of output channels, and expansion is the output channel. Multiplier of the number of channels. The task of bottleneck is to compress and then enlarge the number of channels. Therefore, plane no longer represents the number of output channels, but the number of channels compressed inside the block, and the number of output channels becomes plane*expansion.

Then there is the main body of the network, which defines the ResNet object. Resnet50 has five stages. The convolution kernel of the first stage is 7×7, stride is 2, 64 convolution kernels, padding=3.

Then there is the pooling process. After the input data is subjected to the convolution operation to obtain the feature map, the feature map will be input to the pooling layer, which is also called the down sampling layer. The main purpose of adding a pooling layer is to reduce the dimensionality of the input data, reduce the training parameters of the network, and further extract the features of the input data. The principle of realizing the operation of the pooling layer is similar to the operation principle of the convolution layer. The values are obtained on the feature map by moving the square matrix, but the square matrix in the pooling layer and the convolution kernel in the convolution layer are used. The algorithm is different, and the pooling layer uses the maximum value or the average value instead. Therefore, the pooling operation is mainly divided into maximum pooling and average pooling, and this paper adopts maximum pooling. The advantage of max pooling is that it can extract the strongest features in the feature map without worrying about the interference of other features.

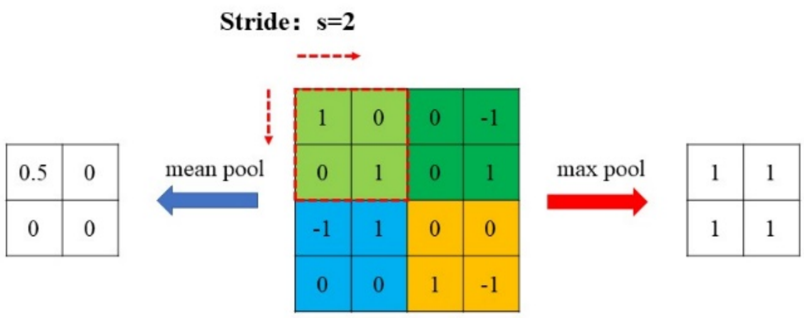

Figure 4. Average pooling and max pooling

These two pooling methods are shown in Fig.4, and the formula for the pooling operation is as follows:

$$x_j^l = f\left(down\left(x_j^{l-1}\right) + b_j^l\right) \tag{2}$$

In the formula, down () is the sampling function.

After the pooling process, the size of the feature map has become 150×150×64, and then there are four stages, which are layer1, layer2, layer3, layer4 in the code. Here, the make_layer function is used to generate four layers, and the user needs to input the number of blocks for each layer. From the first stage to self.layer3, a 38×38×1024 feature layer is finally obtained. self.layer4 is used in the classifier model.

Next is the class definition of the ResNet50 model, call the above ResNet object, the input block type is bottleneck structure, which is convenient for model storage and calculation, and the number of input blocks is [3, 4, 6, 3]. The hierarchical structure of ResNet50 is shown in Table 1.

Table 1. Hierarchical structure of ResNet50

Method	Resnet50
Convolution 1	***Convolution, 7×7, 64, stride 2***
Convolution 2_x	***Max pooling, 3x3, stride 2***
	$\begin{bmatrix} 1\times 1 & 64 \\ 3\times 3 & 64 \\ 1\times 1 & 256 \end{bmatrix}\times 3$
Convolution 3_x	$\begin{bmatrix} 1\times 1 & 128 \\ 3\times 3 & 128 \\ 1\times 1 & 512 \end{bmatrix}\times 4$
Convolution 4_x	$\begin{bmatrix} 1\times 1 & 256 \\ 3\times 3 & 256 \\ 1\times 1 & 1024 \end{bmatrix}\times 6$
Convolution 5_x	$\begin{bmatrix} 1\times 1 & 512 \\ 3\times 3 & 512 \\ 1\times 1 & 2048 \end{bmatrix}\times 3$
Average pooling	

3.2 Region Proposal Networks (RPN)

First, we receive a 38×38×1024 feature map from ResNet50. We traverse the entire feature map space through a 3×3 sliding window and convert it to a 1024-d form, so each point of the feature map is equivalent to 1024-dimensions. We need to create Anchor boxes for each point. All Anchor boxes are based on the anchor point as the center. Each anchor point will have nine Anchor boxes of different sizes. This project uses ratios= [0.5, 1, 2], anchor_scales= [8, 16, 32] nine anchor boxes are shown in Fig.5.

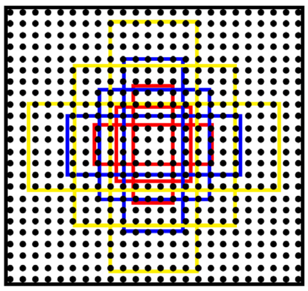

Figure 5. Nine Anchor boxes

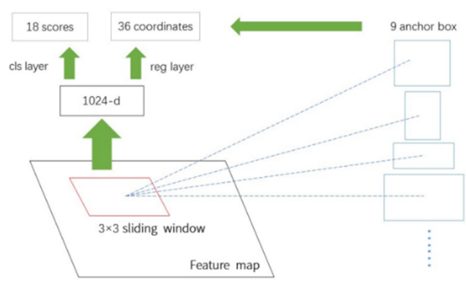

Figure 6. RPN network

With the Anchor Box, the next step is to use the RPN network to obtain candidate boxes. Starting from 1024-d, two layers are generated through the fully connected layer. One is the cls layer and the other is the reg layer. The role of the cls (classify) layer is to classify whether the area is foreground or background, and give the corresponding scores (which can also be understood as relative probabilities), that is, each point generates 18 scores. The reg(regression) layer predicts the position of the x, y, w, and h of the box, that is, each generates 36 coordinates. Then, by summarizing the results of the two layers, the preliminary screening of the anchors is realized. First, the anchors that cross the boundary are eliminated, and then according to the cls results, the non-maximum suppression (NMS) method is used to filter out the anchors that have a large number of overlaps. Finally, according to the reg result is regressed. At this time, the output change is called Proposal (candidate box), and it is sent to the ROI Pooling layer. The RPN network is shown in Fig.6.

3.3 ROI Pooling, Classification Network and Bounding Box Regression

Many proposals (candidate regions) of different sizes output by RPN are input to ROI Pooling together with the original feature map, and output into a 7×7 uniform size feature map, which is for the convenience of the following cls and reg.

The feature map is then flattened through a series of fully connected layers to obtain prediction results. The fully connected layer is equivalent to a classifier, and its structure is shown in Fig.7. As can be seen from the figure, the fully connected layer is a multi-layer perceptron.

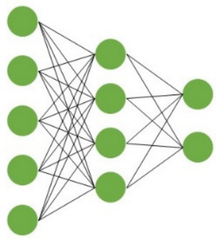

Figure 7. The structure of the fully connected layer

The output layer generally uses the Softmax layer, which normalizes the classification results of the fully connected layer into a probability distribution. The output value of the Softmax layer can be regarded as the probability value of the corresponding sample. The highest probability value is the final classification value. All probability the values add up to a probability of 1, and the expression for Softmax is as follows:

$$P_i = Softmax(y_i) = \frac{e^{y_i}}{\sum_{i=1}^{n} e^{y_i}} \tag{3}$$

In the formula, Pi represents the probability value belonging to the i-th class; yi represents the output value reaching the last fully connected layer; n represents the total number of classes of faults.

Fig.8 explains the operation principle of Softmax. The figure shows the probability of each sample number z1, z2, z3, of which the probability of z2 is the largest.

The general expressions of the fully connected layer and the Softmax layer are as follows:

$$y^k = softmax\left(w^k * x^{k-1} + b^k\right) \tag{4}$$

In the formula,k represents the k-layer network; yk is the output of the fully connected layer; wk is the weight coefficient; xk-1 is the input of the fully connected layer; bk is the bias.

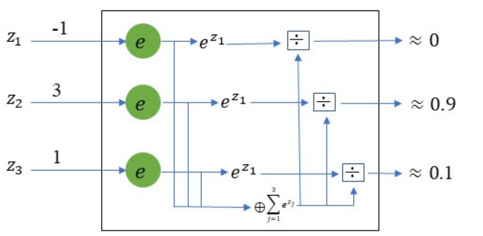

Figure 8. Softmax calculation flow chart

Fig.9 shows a schematic diagram of common connections between the fully connected layer and the output layer.

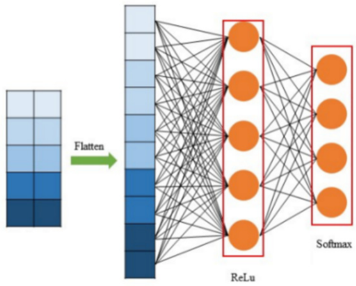

Figure 9. Fully connected layer and softmax output layer

The classification part calculates which category each proposal belongs to (Bottle, milk-box, plastic-bag, ball, leaf) through the fully connected layer and softmax, and outputs the cls_prob probability vector; at the same time, the bounding box regression is used again to obtain the positional bias of each proposal. The shift amount bbox_pred is used to regress a more accurate target detection frame.

3.4 Configuration Policy

According to the characteristics of marine floating debris, this paper determines the parameters of Faster R-CNN through repeated experiments, which improves the accuracy of small target detection. The specific parameter configuration is shown in Table 2.

Table 2. Parameter configuration of Faster R-CNN target detection algorithm

Parameter name	Parameter value
Learning rate	***0.00005***
Learning rate drop factor	***0.96***
Number of problem loops where the learning rate drops	***1000***
Updated weights	***0.9***
The maximum number of iterations	***2000***
IoU value	***0.7***
Positive sample threshold for RPN	***0.8***
Negative sample threshold for RPN	***0.2***

4. Experimental Results and Analysis

Perform image detection on the video to obtain the type, confidence and frame coordinates of the detected garbage. Fig.10 and Fig.11 show the classification results, confidence rates and bounding box coordinates.

Figure 10. Classification results and confidence rates

```
b'bottle 1.00' 427 161 559 291
b'bottle 0.99' 561 71 627 204
b'bottle 0.96' 489 260 545 419
b'milk-box 0.94' 84 394 133 457
b'plastic-bag 0.83' 119 291 183 385
b'plastic-bag 0.66' 375 285 450 375
```

Figure 11. Target detection frame coordinates

The experimental results show that the average detection accuracy of this method can reach 79.36%, as shown in Fig.12.

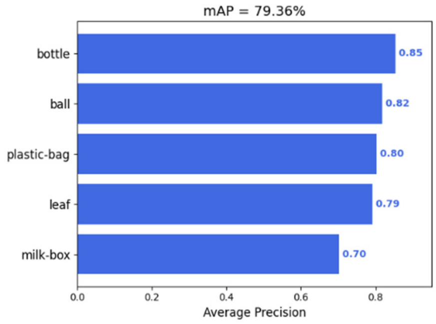

Figure 12. Average detection accuracy

5. Conclusion

In view of the harsh environment, heavy task and low efficiency of manual cleaning of marine floating garbage, this paper uses the existing Faster R-CNN model to propose a ResNet algorithm-based recognition and detection algorithm for marine floating garbage to solve the problem of manual garbage cleaning. The Resnet network is a deep residual network. The network directly skips the data output of a certain layer of the previous layers and introduces it into the input part of the subsequent data layer, which means that the content of the later feature layer will be partially changed from the previous one. The linear contribution of a certain layer, the deep residual network can effectively solve the problem that the learning efficiency is reduced due to the deepening of the network and the accuracy cannot be effectively improved. The average detection accuracy of this method can reach 79.36%. However, there are still shortcomings. For example, the processing speed of video is very slow. The next step will be based on the improvement of the network and the training model, to study this problem, in order to make a more accurate detection speed and a more practical network model.

References

[1] Sahu Geet, Seal Ayan, Krejcar Ondrej, Yazidi Anis. Single image dehazing using a new color channel[J]. Journal of Visual Communication and Image Representation,2021,74.

[2] Chaitanya B.S.N.V., Mukherjee Snehasis. Single image dehazing using improved cycleGAN[J]. Journal of Visual Communication and Image Representation,2021,74 (prepublish).

[3] Li Pengyue, Tian Jiandong, Tang Yandong, Wang Guolin, Wu Chengdong. Deep Retinex Network for Single Image Dehazing. [J]. IEEE transactions on image processing: a publication of the IEEE Signal Processing Society,2020, PP.

[4] K. He, J. Sun, X. Tang. Single image haze removal using dark channel prior[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(12):2341-2353.

[5] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, Gaussian KD-trees for fast high-dimensional filtering[J]. ACM Transactions on Graphics, 2009, 28(3):1-8.

[6] G. F. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan. Efficient image dehazing with boundary constraint and contextual regularization[C]. 2013 IEEE International Conference on Computer Vision. Sydney, NSW, Australia: IEEE, 2013:617-624.

[7] J. Li, H. Zhang, D. Yuan, M. Sun. Single image dehazing using the change of detail prior[J]. Neurocomputing, 2015, 156:1-11.

[8] Q. Zhu, J. Mai, L. Shao. A fast single image haze removal algorithm using color attenuation prior[J]. IEEE Transactions on Image Processing, 2015, 24(11):3522-3533.

[9] W. Ren, S. Liu, H. Zhang, et al. Single image dehazing via multi-scale convolutional neural networks[C]. European conference on computer vision. Amsterdam, Netherlands: Springer,2016:154-169.

[10] Y. Song, J. Li, X. Wang, et al. Single image dehazing using ranking convolutional neural network[J]. IEEE Transactions on Multimedia, 2018,20(6):1548-1560.

[11] B. Li, X. Peng, Z. Wang, et al. Aod-net: All-in-one dehazing network[C]. Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy: IEEE, 2017:4770-4778.

[12] B. Cai, X. Xu, K. Jia, et al. DehazeNet: An End-to-End System for Single Image Haze Removal [J]. IEEE Transactions on Image Processing, 2016,25(11):5187-5198.

[13] Z. Luan, Y. Shang, X. Zhou, et al. Fast single image dehazing based on a regression model[J]. Neurocomputing, 2017,245:10-22.