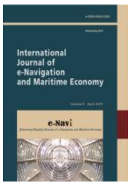
Available online at http://www.e-navi.kr/
e-Navigation Journal



Original article

Research on aquatic target image recognition based on convolutional neural networks

ChaoYu Lu[a], FengGuang Jia[b], Li Min Yu[c]

[a] College of Naval Architecture and Port Engineering, Shandong Jiaotong University, China, LL17852178133@163.com

[b] College of Naval Architecture and Port Engineering, Shandong Jiaotong University, China, dz_jfg@163.com, Corresponding Author

[c]College of Naval Architecture and Port Engineering, Shandong Jiaotong University, China, Yulimin05@163.com

Abstract

Image recognition is not very effective in the water environment due to multiple factors, such as high scattering and high scattering in the water column. This is why the relevant parameters in the Faster R-CNN network model need to adjust continuously to improve the effectiveness of water detection. The control variable method adjusts the program's learning rate by tuning the network model's parameters. Then, the number of training rounds is adjusted according to the loss function of each round, and finally, we can get the number of matches with the minimum loss function. Based on the experimental results on the dataset, it is shown that the proposed method not only selects the learning rate with the best detection results but also has the strongest robustness and achieves a 96%-99% recognition rate for passenger ships, cargo ships, warships, and bridges compared with other learning rates. Experiments show that the Faster R-CNN network model detects water targets with significant results, and the best network model learning rate parameter is 6×10-3.

Keywords: Aquatic target detection, Convolutional Neural Networks, Artificial Intelligence

https://doi.org/10.52820/j.enavi.2022.19.001

1. Introduction

In recent years, with the continuous growth of foreign economic trade, ships in and out of ports have been frequent, and various maritime traffic accidents and maritime disasters occur from time to time. Effective identification of vessels plays a vital role in the safe driving of ships and naval traffic safety management but also helps to improve port navigation, and the ability of cruise rescue and has significant application value to national maritime safety. Researchers have targeted convolutional neural networks in deep learning to rapidly make accurate detection and classification of arbitrary aquatic targets while minimizing human and material costs.

According to Shaofeng Jiang et al. (2014), they Proposed a SAR commercial ship classification algorithm based on structural features, which can classify bulk carriers, container ships, and fishing vessels; with the rise of neural network methods, Jin Xiong Liang (2015) used BP neural network to identify infrared images of six types of ships, namely aircraft carriers, destroyers, frigates, passenger ships, container ships, and oil tankers; A few years later, Katie Rainey (2016) designed a Convolutional Neural Networks (CNN) for satellite ship image classification and achieved a better classification result. Compared with SAR images and infrared images, digital images can provide richer visual information. Zhao Liang et al. (2016) used convolutional neural networks to extract features from digital ship images, then fused HOG and HSV features to construct ship image features, and then used the Support Vector Machine (SVM) method to classify container ships, passenger ships, fishing ships, warships, sailboats. Proia, N et al. (2010) used Bayesian decision-making to identify small vessels. Yokoya N et al. (2015) used the Hough transform for ship detection.

Although this research has achieved better recognition results, they mainly aim at ship image libraries with significant differences in ship shapes and single backgrounds. Many transportation ship images taken in real ports and channels with complex environments, slight differences in ship shapes, and high similarity caused by shooting angles make the traditional methods for classification and recognition of ship images not get better results. The Faster R-CNN network model is tuned to improve the recognition effect in this thesis.

2. Related Work

2.1. Convolutional Layer

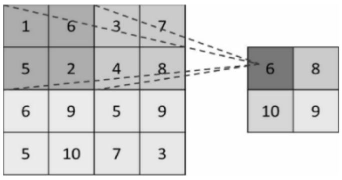
A convolutional layer designs a set of learnable convolutional kernels whose primary function is to extract image features from an image. The height and width of the convolutional kernels are usually relatively small. In the forward propagation of the network, the convolutional kernel of the convolutional layer performs convolutional operations with the input data according to the set stride, and the result of the operations generates the feature map of the layer through the nonlinear activation transform, and the output feature map of the convolutional layer can be expressed as:

$$X_j^l = f(\Sigma_{i \in M_j} X_i^{l-1} * K_{ij}^l + b_j^i) \quad (1)$$

Where X_j^l denotes the j th feature map X_j^{l-1} of the output layer (layer l), the i th feature map of the input layer (layer l-1), M_j denotes the selected combination of input feature maps, K_{ij}^l denotes the convolution kernel between the input and output feature maps, * denotes the convolution operation, b_j^i is the bias term corresponding to the feature maps, and $f(x)$ means the activation function in the convolutional network.

2.2. Pooling Layer

The pooling layer, also called the aggregation layer, is mainly used to reduce the dimensionality of the feature map. The pooling layer aggregates the values of the regions in the feature map and maps the importance of an area into one value, thus reducing the size of the feature map. The most used pooling method is Max pooling, and there are also Stochastic pooling, Mean pooling, etc. Based on Scherer D (2010), Figure 1 shows an example of Max pooling and Mean pooling.



1	6	3	7
5	2	4	8
6	9	5	9
5	10	7	3

6	8
10	9

Figure 1:Example diagram of maximum pooling

2.3. Fully Connected Layer

Fully Connected Layer is the structure of a traditional neural network, which refers to the interconnection between the nodes of two adjacent layers of neurons.

Fully connected layer converts a multidimensional feature map into a one-dimensional vector while retaining helpful information about the features. In convolutional neural network architectures for image classification tasks, the fully connected layer is usually placed at the tail of the network to feed parts into the classifier.

2.4. Classification Layer

The classification layer of convolutional neural networks is most commonly used in Softmax regression, which is a generalization of logistic regression to multi-classification tasks. For the data set $\left\{\left(\left(x^{(1)},y^{(1)}\right),\ldots,\left(x^{(i)},y^{(i)}\right),\ldots,\left(x^{(m)},y^{(m)}\right)\right)\right\}$ of Softmax regression, where the class of y is k, that is $y^{(i)}=\{1,2,\ldots,k\}$, for the input x, it is necessary to use the hypothesis function to find the probability value $p(y=j\mid x)$ of x for each outcome of category j. Here the k probability values are represented by a k-dimensional vector, so the hypothesis function takes the form

$$h_\theta\left(x^{(i)}\right)=\begin{bmatrix}p\left(y^i=1\mid x^{(i)};\theta\right)\\ p\left(y^i=2\mid x^{(i)};\theta\right)\vdots\\ p\left(y^i=k\mid x^{(i)};\theta\right)\end{bmatrix}=\frac{1}{\sum\limits_{j=1}^{k}e^{\theta_j^T x(i)}}\begin{bmatrix}e^{\theta_1^T x(i)}\\ e^{\theta_2^T x(i)}\\ e^{\theta_k^T x(i)}\end{bmatrix} \qquad (2)$$

Where $\theta_1,\theta_2,\ldots,\theta_k\in R^{n+1}$ are the parameters and $\frac{1}{\sum\limits_{j=1}^{k}e^{\theta_j^T x(i)}}$ is normalized to the probability distribution ,and all their probabilities sum to 1. The loss function of Softmax can be written as

$$J(\theta)=-\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k}1\left\{y^{(i)}=j\right\}\log\frac{e^{\theta_j^T x^{(i)}}}{\sum\limits_{l=1}^{k}e^{\theta_1^T x^{(i)}}}\right] \qquad (3)$$

Where $1\left\{y^{(i)}=j\right\}$ is the indicator function that takes the value of 1 if $y^{(i)}=j$ and 0 vice versa. The probability of input x being classified as category j in Softmax regression is calculated as

$$p(y^{(i)}=j\mid x^{(i)};\theta)=\frac{e^{\theta_j^T x(i)}}{\sum\limits_{l=1}^{k}e^{\theta_1^T x(i)}} \qquad (4)$$

2.5. Faster R-CNN Networks

The Faster R-CNN algorithm is an upgraded algorithm of the two-step target detection algorithms R-CNN and Fast R-CNN. The flow of the Faster R-CNN algorithm is shown in Figure 2. According to Simonyan k (2020) and Zisserman A (2020), They proposed the VGG network architecture and the ResNet residual network structure, respectively. First, the features of the input image are extracted using VGG network architecture or ResNet residual network structure, and the candidate regions are generated on the extracted feature maps using Region Proposal Network (RPN). The fully connected layer achieves the target detection to classify and regress the candidate regions after non-maximal value suppression.

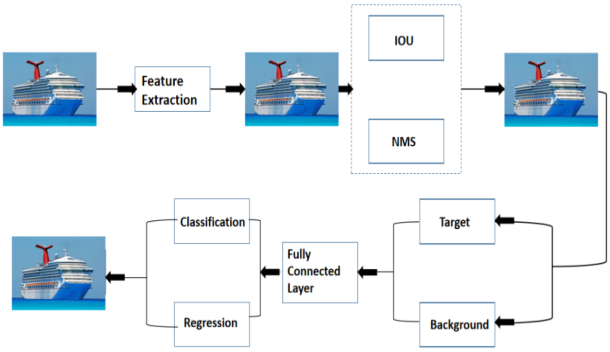


Figure 2: Faster R-CNN algorithm flow

The Faster R-CNN algorithm introduces the Anchor mechanism and edge regression to generate nine frames of different sizes (three areas, three aspect ratios) using sliding windows with Anchor as the center, and each candidate frame is determined to contain target information. The frame with the highest intersection ratio to the actual value is selected as the detection result and regressed to achieve target detection. The structure of the RPN network is shown in Figure 3.

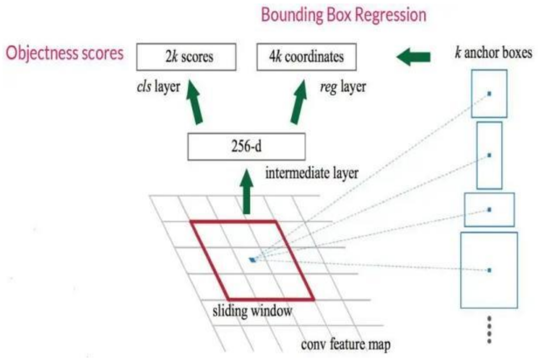


Figure 3: RPN network structure

The purpose of introducing the RPN network structure in the Faster R-CNN algorithm is to extract the possible regions of the target from the image, replacing the selective search method used in the previous algorithm, and allowing end-to-end training of the entire network. The classification layer mainly predicts the confidence score of the target, and the regression layer calculates the position coordinate offset of the target. Since the Faster R-CNN algorithm directly uses the RPN network to generate the detection frame, it can effectively solve the problem of slow generation of candidate regions. However, it still detects the target at a single scale.

3. Experiments

3.1. Experimental Dataset

Before importing the dataset into the network model, images must be collected and data pre-processed. In this paper, we collect 560 valid images from different original pictures of ships and bridges as raw image data. In the process of neural network operation, to prevent unknown errors, all images are pre-processed to a uniform size of 480×320 in this paper. The image samples are shown in Figure 4. This paper performs detection tasks for aquatic targets, and the targets are roughly divided into four categories: passenger ships, cargo ships, warships, and bridges. The dataset images are divided into two categories: 448 photos in the training set and 112 appearances in the test set, and each pack contains four types of images. When multiple candidate targets appear in the source image, multiple ROIs will be tagged to improve the neural network model's target detection accuracy and reliability parameters.

Figure 4:Sample of source image dataset classification

3.2. Training

The parameters are adjusted by controlling the variables, and assuming that the comparison of the image type of the source image dataset is a step, the learning rate can handle the step. Then the loss function of each round is constantly observed. If the loss function shows small fluctuations up and down, but the overall trend is down, then the operation can continue, and the experiment can be completed. If the loss function becomes larger and larger, it means overfitting, and the number of rounds needs to be adjusted to test that the minimum loss function can be obtained under this parameter. After completing all the operations, we can keep the final loss function value and run the test set to observe whether the unknown pictures produce results.

The value of the loss function can be obtained by calculating Equation (5). The smaller the result indicates, the higher accuracy of the output result.

$$L(\{p_i\},\{t_i\})=\frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i,p_i^*\right)+\lambda\frac{1}{N_{reg}}\Sigma_i\, p_i^* L_{reg}\left(t_i,t_i^*\right) \quad (5)$$

4. Experimental results and Discussion

4.1. Experimental results

By setting different learning rates, we can get the loss function results and the test set's accuracy, and the data are shown in Tables 5 and 6.

Table 5:Values of loss function

Loss Function Learning Rate	Loss Detector classifier	Loss Detector regression
2×10^{-4}	0.421	0.181
2×10^{-2}	1.29	0.8
6×10^{-3}	0.25	0.136

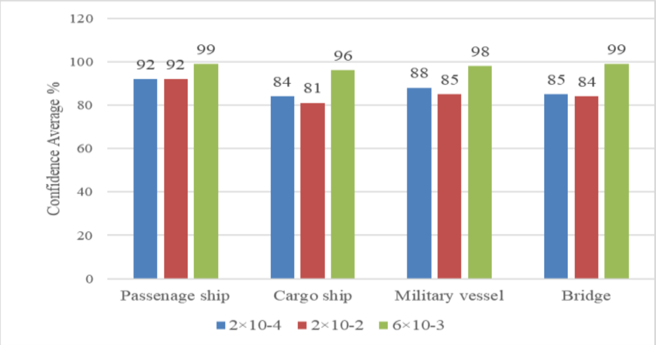


Figure 6:Mean confidence values for each category in the three learning rates

4.2. Experimental Discussion

By comparing the overall confidence mean, the faith means of each category and the loss function values of these three learning rate parameters, we can analyze the optimal parameters of the Faster R-CNN network model for this dataset.

As shown in Figure 7, by comparing the overall confidence mean values under the learning rate parameters of 2×10^{-4}, 2×10-^{2}, and 6×10^{-3}, we found that the faith implies a value of 6×10^{-3} is 98%, which is the highest value of the three parameters. Therefore, we can determine that this neural network's best learning rate parameter is around 6×10^{-3}.

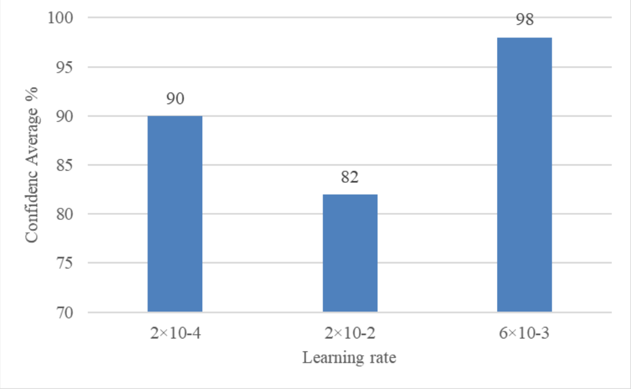


Figure 7:Comparison of confidence means of three learning rate

In Figure 6, comparing the mean confidence values of the four categories under the three learning rate parameters, the various confidence levels of 2×10^{-2} are generally low, and the confidence values of 6×10^{-3} are above 95 for all classes with tiny errors between the predicted and actual values. Therefore, this parameter has the best learning effect among the three learning rate parameters.

As seen from Figure 8, among the loss function values of the three learning rates, the actual output value differs the least from the desired output value for the learning rate parameter of 6×10^{-3}; the difference for the parameter of 2×10^{-4} is the second. The actual output value differs the most from the desired output value for the parameter of 2×10^{-2}, so 6×10^{-3} is more accurate. We further verified the optimal learning rate parameters by comparing the loss function values of the three learning rate parameters.

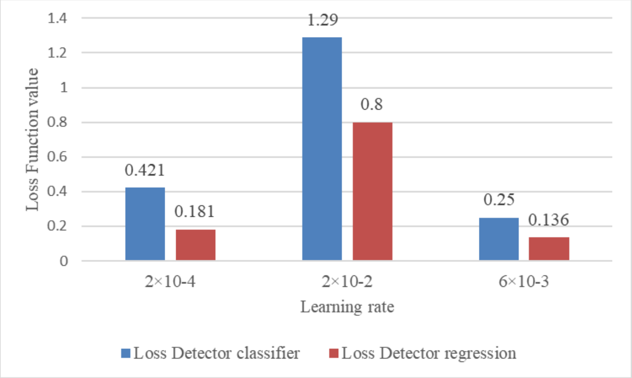


Figure 8:Loss function values with various learning parameters

5. Conclusion

In this paper, we use the Faster R-CNN algorithm for ship detection recognition and adjust the parameters on the original basis to achieve the best recognition effect. The first learning rate parameter selection can be used to roughly determine the detection accuracy of the picture based on visual observation. If the image is easier to detect, the learning rate parameter is significant, the step length is long, and the detection is faster, and vice versa, a smaller learning rate needs to be selected. By testing different learning rates, observing the confidence and loss function values of the test results, and adjusting the size of the learning rate according to the size of the difference, we find the learning rate with the best test results. The experiment results show that the model has the highest confidence and the strongest robustness when the learning rate parameter is 6×10^{-3}, and 6×10^{-3} is analytically selected as the optimal parameter to adapt the network model to this dataset.

References

He, K., Zhang, X., Ren, S., & Sun, J. J. I. (2016). Deep Residual Learning for Image Recognition.

Proia, N., Page, V. J. I. G., & Letters, R. S. (2010). Characterization of a Bayesian Ship Detection Method in Optical Satellite Images. *7*(2), 226-230.

Jinxiong Liang, & Keqi Wang. (2015). Ship Science and Technology. *BP neural network based ship target recognition and classification.* 37(03), 206-209.

Liang Zhao, Xiaofeng Wang, & Yitao Yuan.(2016). Ship Science and Technology. *Research on ship identification method based on deep convolutional neural network.* 38(15), 119-123.

Sadjadi, F. A., Mahalanobis, A., Rainey, K., Reeder, J. D., & Corelli, A. G. (2016). *Convolution neural networks for ship type recognition*. Paper presented at the Automatic Target Recognition XXVI.

Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In *Artificial Neural Networks – ICANN 2010* (pp. 92-101).

Shaofeng Jiang, Chao Wang, Fan Wu, Bo Zhang, Yixian Tang, & Hong Zhang.(2014). Remote Sensing Technology and Application. *COSMO-SkyMed image commercial ship classification algorithm based on structural feature analysis*. *29*(04), 607-615.

Simonyan, K., & Zisserman, A. J. a. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition.

Yokoya, N., Iwasaki, A. J. I. J. o. S. T. i. A. E. O., & Sensing, R. (2015). Object Detection Based on Sparse Representation and Hough Voting for Optical Remote Sensing Imagery. *8*(5), 2053-2062.

Received 20 October 2022

Accepted 23 December 2022