Ship Object Detection Method Based on Multi-Scale Neural Networks

Authors: Wu Yuanyuan, Zang Baihan, Wang Xinghua, Liu Shuai, Zongliang Zhang, Wang Xinghua

Date: 2024-03-23T00:00:00+00:00

Abstract

In recent years, the advancement of intelligent shipping has imposed increasingly stringent requirements on the accuracy of ship target detection and classification. Accurate detection and identification of ship categories, as well as precise localization of vessels, constitute critical safeguards for maritime navigation safety. The optical imaging of ship targets is highly susceptible to external environmental disturbances—including wind, currents, precipitation, and fog—which degrades the performance of deep learning-based detection algorithms. Moreover, the considerable diversity in ship types, morphological variations, and geometric dimensions further compounds the challenges inherent in ship target detection and recognition. To address these issues, this paper proposes a multi-scale neural network-based object detection methodology to enhance the detection accuracy of ships in optical imagery. The approach employs Convolutional Neural Networks (CNNs) for feature extraction, utilizing an improved CSPDarkNet backbone integrated with a multi-scale network architecture to enable accurate detection of surface vessels via shipborne optical cameras, thereby improving model performance on small and densely packed targets. Additionally, label smoothing is employed to prevent overfitting, while non-maximum suppression mitigates duplicate detections. Experimental results demonstrate that the proposed method achieves a Mean Average Precision (mAP) of 84.80% on the Ship-Detection dataset, delivering superior detection performance and enhanced potential for practical application compared with existing methods such as Faster-RCNN and CO-DETR.

Full Text

A Multi-Scale Neural Network-Based Approach for Ship Target Detection in Optical Imagery

Wu Yuanyuan[2]†, **Zang Bohan**[3]†, **Wang Xinghua**[1]*, **Liu Shuai**[2], **Zhang Zongliang**[3]

[1]Navigation College, Jimei University, Xiamen 361021, Fujian, China

[2]College of Computer Engineering, Jimei University, Xiamen 361021, Fujian, China

†Co-first authors, *Corresponding author

Abstract

In recent years, the development of ship intelligence has driven increasing demands for higher accuracy in ship target detection and classification. Accurate detection, classification, and localization of vessels are critical for safe maritime navigation. However, the performance of deep learning-based ship detection algorithms is often degraded by environmental factors such as wind, currents, rain, and fog that affect optical imaging. Additionally, the wide variety of ship types, diverse morphologies, and varying geometric sizes pose significant challenges for reliable detection and identification. To address these issues, this paper proposes a multi-scale neural network-based target detection method to improve the accuracy of ship detection in optical imagery. Our approach employs Convolutional Neural Networks (CNNs) for feature extraction, utilizing an improved CSPDarkNet backbone combined with a multi-scale network architecture to enable accurate detection of waterborne vessels from ship-borne optical cameras, with enhanced precision for small and densely-packed targets. We also incorporate label smoothing to prevent overfitting and non-maximum suppression to reduce duplicate detections. Experimental results demonstrate that our method achieves a mean Average Precision (mAP) of 84.80% on the Ship-Detection dataset, outperforming previous approaches such as Faster R-CNN and CO-DETR, and offering greater potential for practical applications.

Keywords: ship detection; deep learning; neural network; multi-scale neural network

Introduction

With the rapid development of waterway transportation and the marine economy, advancing ship intelligence has become an urgent societal need. As critical transportation carriers and military assets, ships require accurate detection, classification, and recognition to improve navigation safety and crew efficiency,

with significant application value and strategic importance in maritime traffic regulation, protection of national maritime rights, and ocean security.

Traditional optical imagery-based ship detection methods primarily fall into three categories: corner-based, edge feature-based, and region feature-based approaches. For corner detection, Smith [1] proposed the SUSAN algorithm, which uses a 37-pixel circular template around a core point to count pixels with similar brightness values, applying non-maximum suppression to initial corner points to obtain final detections. Edge-based detection typically involves smoothing input images to reduce or eliminate noise, followed by edge detection operators (Roberts [2], Canny [3], etc.) to extract boundary points. Region-based methods primarily combine edge and region processing, using connected grayscale binary images and multi-threshold processing to achieve target detection.

In recent years, with the increasing depth of big data applications and continuous improvements in computational processing speeds, intelligent target detection methods based on Convolutional Neural Networks (CNNs) [4] have made tremendous progress. Current CNN-based detection methods are mainly divided into two strategies: One-Stage and Two-Stage [5]. One-Stage approaches perform direct target detection on input images, while Two-Stage methods incorporate a Region Proposal Network (RPN) [6] to constrain target locations. In CNN-based detection research, Alexander et al. successively proposed classic neural network structures including R-CNN, Fast R-CNN, and Faster R-CNN [6], which laid the foundation for intelligent optical imagery target detection. Neural networks provide an end-to-end model that avoids the enormous cost of manual feature design, and as network models deepen, target detection accuracy continues to improve.

However, due to structural limitations of the R-CNN series algorithms, traditional R-CNN network models exhibit low detection accuracy for small targets, making them inadequate for certain application scenarios. To address this issue, this paper proposes a multi-scale neural network-based ship detection algorithm. Sermanet pioneered the multi-scale concept, which effectively addresses the impact of feature targets at different scales on algorithm performance while improving model robustness. The multi-scale neural network model algorithm proposed in this paper achieves high accuracy with good detection speed, meeting the precision and real-time requirements of ship target detection.

Based on the above analysis and considering that optical data from ship-borne sensors contains ship targets of varying sizes (pixel areas), this paper investigates multi-scale ship target detection algorithms. We analyze the network structure and loss functions of multi-scale detection networks, construct a ship target detection dataset, perform improvements and optimizations tailored to ship characteristics, and conduct multiple rounds of comparative experiments to validate the algorithm' s detection effectiveness and performance.

---

1 Multi-Scale Ship Target Detection Network

Our network model architecture consists of four components: an input layer, backbone network, neck network, and output layer. The overall model is a One-Stage target detection framework, as illustrated in [Figure 1: see original paper]. For the backbone network, we adopt a CSPDarkNet-based framework and utilize more powerful basic building blocks (see Section 1.1) to enhance model accuracy, adjusting parameters such as depth, width, and resolution in the neck network accordingly (see Section 1.2). Prior to training, we apply Mosaic data augmentation in the input layer (see Section 1.3). Additionally, we employ label smoothing as a regularization technique, while using Focal Loss [12] and GIoU [13] as loss functions to optimize the model (see Section 1.4). The overall network architecture is shown in [Figure 1: see original paper].

1.1 Improved Backbone Network

We utilize an improved CSPDarkNet [16] as our backbone network. Traditional basic building blocks, as shown in Figure 2(a), consist of $1\$\times 1 and 3\times 3 convolutional layers. Considering that ships are sometimes densely distributed and that large eff ec$ depthwise convolutions to increase the effective receptive field, as shown in Figure 2(b). Since the improved basic building block increases the number of convolutional layers, which would reduce detection speed, we decrease the number of building blocks used and make certain modifications to the overall network to achieve optimal performance. The overall structure of our backbone network is shown in [Figure 3: see original paper].

[Figure 2: see original paper] Different basic building blocks in backbone networks. (a) Basic building block used in CSPDarkNet, consisting of $1\$\times 1 and 3\times 3 convolutional layers. (b) Our improved basic building block, which introduces 5\times\$5$ depthwise convolutions to increase the effective receptive field while reducing computational cost.

Each convolutional layer in the figure is followed by a BatchNorm [10] and ReLU [11] activation function, computed as:

$$F(X) = ReLU(BN(Conv2d(X)))$$

where BN denotes batch normalization. Although data normalization preprocessing is applied in the input layer, parameter updates during deep neural network training can still cause dramatic parameter variations that typically affect the final trained model' s performance, necessitating additional normalization after each convolution operation. The ReLU activation function is expressed as:

$$G(X) = Max(0, X)$$

Compared to other activation functions, ReLU is computationally simpler [11]

and can zero out some parameter outputs, reducing inter-parameter dependencies and improving target detection speed.

[Figure 3: see original paper] Overall architecture of our adopted backbone network

1.2 Neck Network

For target detection tasks, multi-scale feature pyramids are essential. Multi-scale approaches involve sampling signals at different granularities, enabling observation of different features at various scales to accomplish detection tasks for objects of different sizes. In our ship target detection task, the neck network performs further feature fusion based on features extracted by the backbone network, helping the network perceive targets at different scales and providing more contextual information.

To accommodate changes to the backbone network and consider training speed factors, we expand the basic building blocks in the neck network, placing more computational operations in the neck network to achieve a better trade-off between speed and accuracy.

1.3 Data Augmentation

We employ the Mosaic method to process the dataset. Its main idea is to randomly crop four images and concatenate them onto a single image for training. This approach not only increases data diversity and enables training with more original images than the batch size, but also enhances model robustness and generalization capability.

1.4 Loss Function

[Figure 4: see original paper] Ship detection images using Mosaic augmentation

Since images contain ships of different categories with extremely imbalanced class distribution (e.g., very few sailing ships), this paper adopts Focal Loss [12] as our loss function, computed as follows:

$$\begin{cases} -\alpha(1-p)^{\gamma}\log_{10}(p) & \text{if } y=1 \\ -(1-\alpha)p^{\gamma}\log_{10}(1-p) & \text{if } y=0 \end{cases}$$

where γ is set to 2 and α is set to 0.25. Focal Loss focuses on large-sample data; other loss functions only constrain annotated samples, leaving boundary regions of output images unconstrained, which can cause gradient explosion when numerous unannotated data exist within the same training sample.

For constraining the predicted bounding boxes, we employ GIoU Loss [13] to reduce the metric distance between model-predicted boundaries and expert-annotated boundaries, computed as:

$$IoU\ Loss = -\log_{10} \frac{Intersection(y, \hat{y})}{Union(y, \hat{y})}$$

$$GIoU = IoU - \frac{C \setminus Intersection(y, \hat{y})}{C}$$

where $Intersection$ denotes the intersection between two bounding boxes, $Union$ denotes their union, and C represents the area of the smallest enclosing box. Using the Intersection over Union (IoU) as a constraint reduces the metric distance between predictions and actual objects.

---

2 Additional Techniques

2.1 Non-Maximum Suppression

In target detection tasks, numerous bounding boxes are typically generated, with many overlapping boxes for the same instance. Non-Maximum Suppression (NMS) selects the optimal bounding box from a series of overlapping boxes, retaining only the box with the highest probability within a certain range. The computation is as follows:

$$boxlist = [y_1, y_2, \ldots, y_n]$$

$$s_i = \begin{cases} s_i & \text{if } IoU(M, b_i) < N_t \\ s_i(1 - IoU(M, b_i)) & \text{if } IoU(M, b_i) \geq N_t \end{cases}$$

When the IoU value exceeds the set threshold (commonly 0.5, often 0.7 in target detection), the overlapping boxes are suppressed by setting their detection scores to 0. After each round, the highest-scoring box among the remaining boxes is selected, and boxes with IoU values above the threshold are suppressed. This process continues until only non-overlapping boxes remain, effectively ensuring each target is represented by a single detection box.

2.2 Label Smoothing

In traditional deep learning networks using one-hot encoded labels, the optimization objective drives each category' s constraint direction infinitely close to 1 for the target class and 0 for non-target classes. This results in the target class probability approaching 1 in the final network output, creating excessive variance between correct and incorrect labels that reduces model robustness and can lead to overfitting or gradient explosion, preventing entropy-based loss functions from operating efficiently. Therefore, we employ label smoothing:

$$P_i = \begin{cases} (1-\epsilon) & \text{if } i = y \\ \frac{\epsilon}{K-1} & \text{if } i \neq y \end{cases}$$

3 Experiments

3.1 Dataset Construction

Datasets play a decisive role in deep learning, determining the final detection performance of the model while requiring sufficient samples for the network to fully learn target features. The constructed ship dataset is summarized in .

Distribution of the total dataset (Sample counts: Total dataset / Training set / Test set)

3.2 Model Training

We adopt PyTorch [14] as our deep learning framework. During training, we use a Step learning rate schedule that multiplies the learning rate by 0.1 every 20 epochs, with an initial learning rate of 0.0003 and the Adam [15] optimizer. The training batch size is 8, and our experimental method performs 300 iterations on a CUDA 11.6 platform with PyTorch version 1.12.

3.3 Performance Comparison

We conduct comparative experiments between our proposed multi-scale ship target detection algorithm and other deep learning-based detection methods. To ensure training consistency, we employ identical training parameters (learning rate, optimizer, weight decay strategy, etc.). presents the comparison results between our proposed model and other mainstream algorithms. Note that the data in the table is for relative reference only, as the validation dataset annotations contain certain errors. We use mean Average Precision (mAP) as the evaluation metric, computed as:

$$mAP = \frac{TP}{TP + FP}$$

where TP represents true positive rate (predicted true and actually true), FP represents false positive rate, and n denotes the total number of samples.

Comparison results between our proposed model algorithm and other mainstream methods (Faster R-CNN [6], Sparse R-CNN [22], CO-DETR [19], DINO [20], DDQ [21]) with metrics: mAP@.5:.95, mAP@0.5, mAP@0.75

3.4 Detection Results

We evaluate the algorithm' s prediction results and output model predictions. The detection effects are as follows: top-left shows Faster R-CNN, top-center shows CO-DETR, top-right shows Sparse R-CNN, bottom-left shows DDQ, bottom-center shows DINO, and bottom-right shows our algorithm. Sparse R-CNN and DDQ algorithms are insensitive to small ship targets, exhibiting missed detections. Faster R-CNN shows duplicate candidate boxes and false positives in complex scenarios; additionally, its region proposal network' s search algorithm runs only on CPU, resulting in low inference efficiency on CPU-limited machines. While CO-DETR and our multi-scale detection model both achieve correct detections, CO-DETR performs less effectively on overlapping targets compared to our algorithm—the ship Ground Truth in the background only reaches 50.4, and can exhibit missed/false detections with suboptimal training. Our algorithm optimizes for these issues of inference speed, spatial cell limitations, and overlapping targets by reducing region proposal network complexity, replacing cell constraints with non-maximum suppression, and employing multi-scale algorithms for regional detection to improve accuracy on overlapping targets.

[Figure 5: see original paper] Comparison of detection effects between our multi-scale ship detection model and mainstream models

Based on the multi-scale target detection algorithm model, this paper proposes a real-time ship target detection method that extracts regions of interest through backbone and neck networks, with improvements using label smoothing and other optimization strategies. Using a self-constructed ship image dataset, we train the designed model on an Ubuntu server and compare it with Faster R-CNN, CO-DETR, Sparse R-CNN, and other algorithms. Experimental results demonstrate that our improved algorithm achieves an mAP of 84.80%, outperforming other methods and exhibiting good detection performance for small targets and overlapping imagery.

References

[1] Smith S M, Brady J M. SUSAN: A New Approach to Low Level Image Processing[J]. Int. Journal of Computer Vision, 1997, 23(1):45-78.
[2] L. Roberts Machine Perception of 3-D Solids, Optical and Electro-optical Information Processing, MIT Press
[3] Canny, J., A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.
[4] Fukushima, K. (2007). "Neocognitron" . Scholarpedia. 2(1): 1717. Bibcode: 2007SchpJ···2.1717F. doi:10.4249/scholarpedia.1717.
[5] Burke, D. L., & Ensor, J. (2017). Meta-Analysis Using Individual Participant Data: One-Stage and Two-Stage Approaches, and Why They May Differ. Tutorial in Biostatistics, 36(5), 855–875. doi:https://doi.org/10.1002/sim.7141.

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. doi:10.48550/ARXIV.1506.01497.
[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. doi:10.48550/ARXIV.1506.02640.
[8] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. ECCV 2016. doi:10.1007/978-3-319-46448-0_2.
[9] Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. doi:10.48550/ARXIV.1612.08242.
[10] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. doi:10.48550/ARXIV.1502.03167.
[11] Agarap, A. F. (2018). Deep Learning using Rectified Linear Units (ReLU). doi:10.48550/ARXIV.1803.08375.
[12] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. Focal Loss for Dense Object Detection. Proceedings of the IEEE International Conference on Computer Vision (ICCV).
[13] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. doi:10.48550/ARXIV.1902.09630.
[14] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ··· & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems,
[15] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. doi:10.48550/ARXIV.1412.6980.
[16] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. doi:10.48550/ARXIV.2004.10934.
[17] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). CenterNet: Keypoint Triplets for Object Detection. doi:10.48550/ARXIV.1904.08189.
[18] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In NeurIPS, 2016.
[19] Zong, Zhuofan, Guanglu Song and Yu Liu. "DETRs with Collaborative Hybrid Assignments Training." 2023 IEEE/CVF International Conference on Computer Vision (ICCV) (2022): 6725-6735.
[20] Zhang, Hao, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun-Juan Zhu, Lionel Ming-shuan Ni and Heung-yeung Shum. "DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection." ArXiv abs/2203.03605 (2022): n. pag.
[21] Zhang, Shilong, Wang xinjiang, Jiaqi Wang, Jiangmiao Pang, Chengqi Lyu, Wenwei Zhang, Ping Luo and Kai Chen. "Dense Distinct Query for End-to-End Object Detection." 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023): 7329-7338.
[22] Sun, Pei, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang and Ping Luo.

"Sparse R-CNN: End-to-End Object Detection with Learnable Proposals." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020): 14449-14458.

Note: Figure translations are in progress. See original paper for figures.