Ship Detection Framework Based on Deep Learning Network

Jing YE¹*, Yu-fen SUN¹, Gang LIU¹ and Lei LIU²

¹School of Computer Science and Technology, Wuhan University of Technology, Wuhan, Hubei, China
²National Engineering Research Center for Water Transport Safety, Wuhan University of Technology, Wuhan, Hubei, China
*Corresponding author

Keywords: Ship detection, Feature extraction, Multi-scale predictions.

Abstract. Ship detection and tracking have been recognized as a challenging task in the maritime administration. This paper focuses on the maritime traffic situation, inspects and produces the ship dataset. We improved the existing deep learning method through experiments. It is mainly reflected in the addition of feature reused dense blocks which are used in the feature extraction stage and the addition of contextual information in the low-scale feature map which is used in the multi-scale prediction stage. The improved network model can effectively identify and calibrate the ship image in the dataset, thus improving maritime surveillance efficiency.

Introduction

Ship detection is an important technology for intelligent navigation and unmanned driving. It has broad application prospects in many fields such as military field, defense safety, port vessel dispatching, ecological security, ecological monitoring, and traffic management. In the application of maritime traffic, the ship detection of the camera becomes very important. The shipborne camera needs real-time object detection and tracking of ships coming and going, which can effectively provide potential collision warnings, avoid maritime traffic accidents, and book the best navigation route of the ship.

The main processes of traditional ship detection and recognition methods are object feature extraction (horizon detection, background subtraction, foreground segmentation), object recognition, and object location. The features used here are all artificially designed features such as SIFT², HOG³ features, and so on. However, for the object detection of the ships, the traditional ship detection system is usually relatively large, the dependent sensor is expensive, and the technology used is complicated. The conventional ship detection method cannot meet the requirements of the camera's object accuracy and real-time performance.

Fortunately, in recent years, object detection and tracking technology have developed rapidly. Object detection and recognition methods based on deep learning have become mainstream. They can be mainly expressed as depth feature extraction of images and object recognition and localization based on deep neural networks. The deep learning model used is mostly the Convolutional Neural Network (CNN)¹. By ship detection, many follow-up processes can be used to improve the safety of intelligent driving, such as predicting the potential threat to the current driving route; performing behavior identification on the detected ships and predicting their paths; avoid traffic accidents; use RNN to generate textual descriptions of ship orientations. Therefore, the research work of this subject has proper theoretical significance and practical application value.

Given the accuracy and real-time of the ship detection and tracking, this paper summarises the research from the currently known deep learning model and optimizes the existing network model through experiments. The main contributions are as follows:

Dense blocks were added in the feature extraction stage, which introduced the idea of DenseNets⁹, making the network more “parameter efficient,” promoting the reuse of features and avoiding the disappearance of gradients;
In the object detection phase, contextual information is introduced into the low-scale feature map which enriches the information of the low-scale feature map and helps the model detect the small ships.

![Figure 1. The test results.](image)

**Related Work**

At present, the existing deep learning-based object detection algorithms can be divided into two categories: object classification and localization algorithms based on region proposal, and representative R-CNN\(^4\), Fast-RCNN\(^5\), Faster-RCNN\(^6\); regression-based object recognition and localization algorithm, typically YOLO\(^7\), SSD\(^8\).

The advantages of ship object detection based on a convolutional neural network\(^1\) are as follows: 1. No artificial intervention is required to preprocess the image, and raw images can be directly input, and the system still has good effects; 2. System expansion strong can be modified and increased on the CNN network to get different output, to achieve various functions; 3. the system has strong migration ability, the ship detection module can be modified after slightly modifying and using different datasets. Other modules that are similar in function to object detection are used in other systems.

**Object Classification and Localization Algorithms Based on Region Proposal**

The object detection and recognition algorithm based on region selection is the most mature and widely used framework at this stage. It simplifies the entire recognition process into a classification task and uses the superiority of deep learning methods to classify large-scale complex data to improve detection accuracy. Once this framework emerges, it defeats all the traditional object detection methods.

Ross Girshick's R-CNN\(^4\) is a pioneering work to introduce the CNN method into the field of object detection, which significantly improves the object detection effect. It can be said that the primary research ideas in the area of object detection have been changed. However, every proposal regions in R-CNN needs to enter the CNN network calculation. There are a large number of overlapping of thousands of Regions. Repeated feature extraction makes the calculation of CNN feature extraction very redundant, which brings substantial computational waste. The detection speed is mostly limited, and the whole algorithm is not efficient. Subsequently, Ross Girshick proposed the Fast-RCNN\(^5\) algorithm based on R-CNN\(^4\). The main contribution was to accelerate the R-CNN and suggested the regions of interest (RoI) strategy to avoid inputting a picture...
continuously which classification and regression use the multitasking loss layer to improve the accuracy of the algorithm. However, there are still many areas to be identified, and this part of the algorithm is temporarily unable to integrate into the GPU, and the efficiency of the whole algorithm is still not high. Based on the assumption that candidate frame extraction can be performed on the feature map, Shaoqing Ren et al. proposed the Faster R-CNN\cite{6} algorithm, which introduces a new concept, the regions proposal network (RPN). In the RPN network design for generating the proposal window, Faster-RCNN\cite{6} implements the extraction of the candidate frame for the object classification and the frame regression by sliding the window on the feature map. The object detection and recognition of Faster-RCNN\cite{6} are integrated into a network. The whole model is an end-to-end process, which dramatically improves the overall performance, especially concerning detection speed.

Of course, these three methods directly process the object candidate region and the depth feature of the object. This method simplifies the object detection and recognition task into a classification task by introducing object candidate regions, which makes perfect use of the robust performance of CNN. After continuous improvement, the accuracy is getting higher and higher, but it cannot be applied to the real-time detection system regarding time consumption.

Object Classification and Localization Algorithms Based on Regression

In order to enable real-time detection and recognition, deep learning begins to be applied simultaneously to the identification and detection steps. Therefore, the object detection and recognition algorithm that directly returns without generating the object candidate region is made, mainly including the YOLO\cite{7} algorithm proposed by Joseph Redmon et al. and the SSD\cite{8} algorithm proposed by Wei Liu et al.

For humans to observe a picture, detecting and identifying the object in the image is very fast and accurate. Therefore, Joseph Redmon et al. proposed the YOLO\cite{7} (You Only Look Once) algorithm. The machine only needs to look at it once. It does not require the candidate frame extraction similar to RPN but directly returns the whole graph. It is a new object detection. Identify the framework. YOLO\cite{7} algorithm speed has been significantly improved. The overall structure is relatively simple, can achieve 45 frames per second under Titan X GPU, fast version can achieve 150 frames per second, can achieve video object detection and recognition. However, the positioning accuracy is worse than the object detection and recognition method based on the regional recommendation. The boxes regression feature loss is more severe than Faster R-CNN. The detection efficiency is not good when some small objects or objects are close to each other. Because the algorithm only results in regression of fixed-size, fixed-position image blocks.

The SSD proposed by Wei Liu et al. combines the regression idea in YOLO with the anchor mechanism in Faster R-CNN, and uses multiple layers of features on different feature maps to detect and identify objects in various positions in the image. This algorithm not only ensures that the detection and recognition technology has the speed to be recognized in the camera video but also provides that the recognition and classification object has an accuracy comparable to that based on the regional recommendation method. However, since the receptive field in shallow feature maps in the SSD is small, the missed detection of small objects is more serious.

Since the introduction of neural networks, the object detection framework has become more and more mature, but for ship detection, to avoid maritime traffic accidents, we hope that the detection of ship objects should be faster and more accurate. In the field of computer vision, better new energy usually means deeper networks, but deeper networks often bring huge computational redundancy. Therefore, we focus on improving the accuracy of the model while ensuring speed and enhancing the detection of small objects. In this article, we try to integrate various new methods to improve the performance of the network structure.

The ShipNet Architecture

In this chapter, we detail our ShipNet architecture and its approach to optimization improvements. The method proposed in this paper is a multi-scale feature detection framework similar to SSD.
divided the entire ShipNet network architecture into two parts: a backbone network for primary feature extraction and a multi-scale predictions network for predicting multiscale feature maps. The backbone network draws on DenseNet and consists of a stem layer, four dense block layers, and four transition layers. Due to the increasingly close integration of low-scale upsampling fusion and high-scale dense structure, multi-scale predictions' network is more abundant. The complete network architecture and parameter settings are shown in Figure 2 and Table 1. We will elaborate on each of the improvements in the following sections.

**Figure 2. The ShipNet architecture.**

**Table 1. The ShipNet backbone.**

| LAYERS             | FILTERS | SIZE | OUTPUT   |
|--------------------|---------|------|----------|
| STEM               | Convolution | 64   | 3x3/2    | 150x150  |
|                    |          | 64   | 3x3      | 150x150  |
|                    |          | 128  | 3x3      | 150x150  |
| DENSE BLOCK(1)     | Convolution | k=48 | 1x1      | 75x75    |
| x6                 |          |      | 3x3      | 75x75    |
| TRANSITION         | Convolution | 256  | 1x1      | 75x75    |
|                    |          |      | 3x3/2    | 38x38    |
| DENSE BLOCK(2)     | Convolution | k=48 | 1x1      | 38x38    |
| x8                 |          |      | 3x3      | 38x38    |
| TRANSITION         | Convolution | 512  | 1x1      | 19x19    |
|                    |          |      | 3x3/2    | 19x19    |
| DENSE BLOCK(3)     | Convolution | k=48 | 1x1      | 19x19    |
| x8                 |          |      | 3x3      | 19x19    |
| TRANSITION         | Convolution | 1024 | 1x1      | 19x19    |
| DENSE BLOCK(4)     | Convolution | k=48 | 1x1      | 19x19    |
| x8                 |          |      | 3x3      | 19x19    |
| TRANSITION         | Convolution | 512  | 1x1      | 19x19    |
| PREDICTION         |          |      |          |          |

**Stem**

According to darknet-53 and Inception-v3\textsuperscript{[10]}\textsuperscript{[11]} and Inception-v4\textsuperscript{[11]}, the stem of the network structure is defined as a series of 3 x 3 convolution layers with step length of 1 and 3 x 3 convolution layers with step length of 2. Through experiments, we found that compared with the original design of DenseNet (7 x 7 conv-layer, stride = 2 followed by a 3 x 3 Max pooling, and stride = 2), adding this simple structure
in the network structure can significantly improve the detection performance. We can see in table 2 that it is imperative to add this structure.

Deep Connections

In deep learning networks, as the depth of the network deepens, classification loss and location loss become more apparent. This article draws on the idea of DenseNets, starting with the feature, through the ultimate use of the feature to achieve better results and fewer parameters, we introduced the Dense Block. As described in DenseNets, in each dense block, any two layers with the same feature map size are directly connected, and each layer's input contains feature maps of all previous layers, and it The output is passed to each subsequent layer. These feature maps together through depth concatenation can effectively solve the gradient disappearance problem, enhance feature propagation, support feature reuse, and significantly reduce the number of parameters. DenseNets tends to produce improvements in accuracy as parameters increase, and there is no performance degradation or over-fitting, which has been verified in [9]. The growth rate indicates the number of feature maps output per layer in each dense block, expressed in k. To avoid the network becoming very wide, here we use k = 48.

Transition Layer

In DenseNet's original design, each Transition Layer contains a pooling operation to downsample the sampled map features. We introduced this layer to change the number of parameters of the network without reducing the final feature map resolution. The entire network is divided into four Dense Blocks, and we refer to the convolutional layer between the block and the block as the Transition Layer. Usually, the downsampling uses the max pooling layer to do the sample of the feature map, for example [9]. However, we have found through experiments that in the feature extraction stage, it is better to replace the whole convolution layer with a convolution kernel of 3x3 and a step size of 2. This result is because the max pooling layer directly violently reduces the image to half, losing much valid information related to the object. After the conv layer replaces the max pooling layer, the relevant image information can be effectively preserved, and the loss of the object feature after the primary network is avoided, and the small object feature is not apparent when the object feature is extracted at a multi-scale later.

Multi-scale Predictions

The image with a size of 300 x 300 was input into SSD, which was detected on the feature graph of six scales after the basic network was passed. Shallower layers have contextual information, but they do not have semantic information. Therefore, we need to combine these two types of information and combine the low-scale feature map First with the resolution of 38x38 and the feature map Second with the resolution of 19x19 through sampling on linear interpolation to form a new feature map for detecting small objects. For the other five feature graphs with smaller and smaller resolution, add a plain transition layer with the matching structure (a 1 x 1 conv-layer for reducing the number of feature maps plus a 3 x 3 conv-layer) between every two adjacent scales. As proposed in [12], learning half and reusing half. Each later scale is directly transited from the adjacent previous scale. In addition to the First scale feature map, half of the channels in each scale feature map were learned through the previous conv-layers, and the other half were sampled directly from the previous continuous high-resolution feature map. In this paper, the lower sampling block was followed by a convolution kernel of 2 x 2, Max pooling layer of stride = 2 followed by a 1 x 1, and stride = 1 conv-layer. The convolution kernel is 1 x 1, and the conv-layer of stride = 1 is to reduce the number of channels to reduce the calculation cost. The convolution kernel is 2 x 2, and the Max pooling layer of stride = 2 aims to reduce the size of the feature graph of the shallow layer by half and that of the latter layer to match and fuse in the connected multi-scale feature graph. This approach of structural reuse can improve accuracy with fewer parameters and achieve the best results with higher efficiency.
Experiment

Dataset

Work on maritime image processing typically uses military-owned or proprietary data sets that cannot be used for research purposes. The dataset MarDCT 1 can be used online for academic and research purposes, but this dataset only has images from the visible and infrared band sensors and video. Other datasets used in image processing competitions, such as SAR images, are based on radar overhead imaging. Considering that there is no video image data for the river or the actual traffic on the sea, we need to go to the cargo ship on the river bank to install equipment and collect data in the field. The data collected includes various images of ships with different attitudes and views. Some of the ships are shielded from each other, some are not shielded, some contain a single ship, some do not include a ship, and not only the large ships with a close distance but also the small ships with long distance or low resolution. According to the video images collected onshore ships, JPG format images containing ship images should be manually captured in the video data. After uniform editing and renaming of the images, image files should be placed according to the structure of VOC data set to form the ship data set.

Training

We implement our testing based on the Caffe\textsuperscript{[13]} architecture. All models were trained on the NVidia Ti1080 GPU using the SGD algorithm. Our training strategy fundamentally inherits SSD, including data augmentation, scale, aspect ratios, and loss functions, until we have our learning rate and minimum batch size settings.

Result

According to the improvement proposed in Section 3, the collected data sets were trained and verified respectively, and the obtained mAP (Mean Average Precision) results are shown in Table 2. It can be seen from Table 2 and Table 3 that the improved and optimized ship detection model in this paper has the highest mAP in the existing deep learning method.

Table 2. Effectiveness of various designs on MyDataset test set(with IOU=0.5).

| STEM? | DENSE BLOCK? | TRANSITION? | FULLY CONNECTIONS LAYER? | FEATURE FUSION? | MAP |
|-------|--------------|-------------|--------------------------|-----------------|-----|
| √     | √            | √           | √                        |                 | 0.59 | 0.62 | 0.66 | 0.69 | 0.70 | 0.74 |

Table 3. MyDataset test detection results.

| METHOD         | BACKBONE     | MAP  |
|----------------|--------------|------|
| FASTER RCNN    | ResNet-101\textsuperscript{[13]} | 52.3 |
| SSD300         | VGGNet\textsuperscript{[16]}    | 57.6 |
| SSD512         | VGGNet       | 59.0 |
| YOLO           | VGGNet       | 38.3 |
| YOLOV2         | Darknet-19   | 59.2 |
| MINE           | ShipNet      | 74.2 |

Conclusion

This paper proposes a multi-scale ship detection framework based on depth supervision (The ShipNet), which shows excellent results on the datasets obtained from our field surveys. Faster than the exact Faster RCNN, with fewer parameters, higher precision, and better detection of small objects than advanced SSDs’. The ShipNet has not only excellent effects in the single maritime surface area but also has great potential for different fields such as medicine and agriculture and forestry images.
Acknowledgement

This research is supported by the open fund from National Engineering Research Center for Water Transport Safety.

Reference

[1] Uijlings J R R, Van De Sande K E A, Gevers T, et al. Selective search for object recognition[J]. International journal of computer vision, 2013, 104(2): 154-171.

[2] David G.Lowe. Distinctive Image Features from Scale-Invariant Keypoints. In International Journal of Computer Vision. 2004,60(2):91-110.

[3] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 1: 886-893.

[4] Van De Sande K, Gevers T, Snoek C. Evaluating color descriptors for object and scene recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 32(9): 1582-1596.

[5] Girshick R. Fast R-CNN[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[6] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016.

[8] W. Liu, D. Anguelov, D. Erhan, et al. Ssd: Single shot multi-box detector. In ECCV, 2016.

[9] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In CVPR, 2017.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, et al. Rethinking the inception architecture for computer vision. In CVPR, 2016.

[11] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In ICLR workshop, 2016. 3, 4.

[12] Zhiqiang Shen, Zhuang Liu, Jianguo Li, et al. DSOD: Learning Deeply Supervised Object Detectors from Scratch. In ICCV, 2017.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” Proceedings of the 22nd ACM international conference on Multimedia, 675-678, (2014)

[14] K. He, X. Zhang, S. Ren, J. Sun, “Deep residual learning for image recognition,” Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778, (2016)