Smart Boat Detection Based on Feature Pyramid Network and Deformable Convolution

Kaipeng Li*, Yunge Wang, Yuan Shao and Xingxiao Wu
Xiangshan Power Industry Co., Ltd., Zhejiang, China

*Corresponding author: kaipengli@zj.sgcc.com.cn

Abstract. The submarine cable guarantees the electricity and communication of the island residents. The operation of fishing boats poses a major threat to the submarine cable. Due to the complex environment, manual monitoring has defects such as strong subjective factors and easy fatigue. The paper adopts the intelligent monitoring method, using the object detection algorithm based on deep learning and the camera to monitor the boats on the sea. Use feature pyramid network to enhance the detection of smaller and farther boats. Use deformable convolution to solve the problem of few samples. Experimental results show that model can detect boats. The detection ability of the feature pyramid network is stronger, especially for the distant and smaller boat targets. Using deformable convolution can improve the accuracy of models trained on small dataset.

Keywords: Object Detection, Deep Learning, Feature Pyramid Network, Deformable Convolution.

1. Introduction

Submarine cables are laid on the seabed. Cables provide power to islands. China has a long coastline and many islands. Submarine cables are of great significance to the production and life on the island.

The main threats to cables are fishing gear and anchor operations. Fishing nets can pierce the seabed and break the submarine cables by boat trawling. Anchors can hook the cable when stop in the sea. The crew even directly enters the water to cut the submarine cable. In order to ensure the safe and stable operation of submarine cables, relevant state departments have formulated many measures. Set up submarine cable signs on the coast; use loudspeakers to warn passing boats; even set up observation rooms directly on the coast and send special personnel to observe the sea. Real-time video monitoring is one of the very basic and very important measures.

Generally, submarine cable monitoring uses cameras to replace human. Traditional monitor does not have the boat detection function and consumes a lot of manpower. After working for a long time, the human eye will be very tired, and the abnormal situation that occurs at this time. Moreover, people's subjective consciousness will also miss the abnormal situation. Real-time detection of boats can reduce the workload of monitoring personnel, overcome the shortcomings and shortcomings of huma, reduce the probability of accidents, and play an important role in ensuring the safe and stable operation of the power grid.
2. Method

2.1. Object Detection
Object detection is to find all the needed objects on image and gets its position [1]. The project needs to find out the position of the boat in the sea area of the submarine cable, and then judge its threat to the submarine cable. Therefore, it is considered very suitable to use object detection instead of human. At present, object detection has entered the era of deep learning from traditional algorithms [2]. In 2012, AlexNet based on deep learning defeated traditional algorithms with a huge advantage in the ILSVRC [3]. Since then, more researchers have participated in deep learning research. Deep learning is the use of very deep artificial neural networks to learn the inherent laws of sample data. When inference, the network is used to extract the features of the data, and the network output is regressed or classified according to the actual problem. When learning, first initialize the weights and biases, and then enter the independent variables in the dataset to obtain the error between the dependent variable and the true dependent variable in the dataset. The gradient descent algorithm and backpropagation are used to reduce the error and modify the weights and biases, so that the network inference is getting closer and closer to the real dependent variable.

In the field of computer vision, the most commonly used deep learning model is Convolutional Neural Network (CNN) [4]. CNN is a network based on receptive fields, consisting of a convolutional layer, a pooling layer, and a fully connected layer. The main function of the convolutional layer is to extract features. This process may be accompanied by down-sampling. The effect of down-sampling is to increase the receptive field and reduce the parameters [5]. The function of the pooling layer is to retain the main characteristics, increase the receptive field, and reduce the parameters [6]. The fully connected layer mainly classifies the features.

2.2. Faster R-CNN
The most commonly used object detection based on deep learning are Faster R-CNN [7], YOLO [8]. Faster R-CNN detection speed is slow, but the accuracy is high; YOLO is the opposite. The project requires real-time monitoring of sea vessels, and the YOLO may be more suitable for this project. However, the monitoring scene is huge, and the speed of the boat is very slow compared to the entire monitoring screen, so it is also feasible to use Faster R-CNN. The resolution of the camera selected for this project is 1920×1080. The resolution used by Faster R-CNN is 1333×800. Low-resolution will lose some small object information. Therefore, Faster R-CNN is more suitable for this project.

In the Faster R-CNN framework, a powerful backbone is required for feature extraction, and CNN is generally used as backbone. At present, the most commonly used CNNs in the industry are ResNet [9], VGG [10], GoogLeNet [11]. ResNet draws lessons from VGG and GoogLeNet, and uses a large number of 1×1 and 3×3 convolution stacks to construct the BottleNeck structure [12]. Researchers found that after the network reaches a certain depth, the effect cannot be further improved, but the network convergence becomes slower. ResNet uses a residual structure to solve this problem. The feature extraction ability of ResNet is significantly better than VGG and GoogLeNet. Therefore, this project finally chose ResNet as BackBone.

RoiPooling turns the feature map extracted by Region Proposal into a fixed size, generally 7×7. Due to the inconsistency of the input image size, RoiPooling encounters a feature map that cannot be divisible by 7, so it uses the upper left coordinate and the lower right coordinate to use the upward rounding strategy. This invisibly introduces a lot of errors, especially for small targets, where the feature is not a few pixels. When RoiAlign [13] encounters a floating-point number, the area proposal will still be divided equally. For half of the pixels that appear in the merged range, the pixels are re-updated using bilinear interpolation to avoid errors caused by quantization operations.

In order to improve the generalization ability of the model, methods such as data enhancement and transfer learning are usually used. Generally, the more training data, the stronger the generalization ability. Data Augmentation is a way to expand a dataset [14]. Generally, Data Augmentation is performed by means of flipping, cropping, Gaussian noise, etc. Transfer Learning is the application of
knowledge learned in a certain field to other related fields [15]. At present, training a neural network from scratch requires a lot of time and a lot of data, which is impractical. The parameters trained by others can be transferred to this project.

3. Model

3.1. Feature Pyramid Network

There are a large number of down-sampling in CNN. These down-sampling increase the receptive field, but at the same time reduce the resolution of the image, especially for some small objects. However, down-sampling must exist. Feature Pyramid Network (FPN) can retain both down-sampling and resolution [16]. As shown in Fig. 1, FPN is composed of three processes: bottom-up, lateral connections, and top-down. C2, C3, C4, and C5 are the feature maps of the second, third, fourth, and fifth blocks of ResNet50. The border thickness of the feature map indicates the strength of feature expression.

![Feature Pyramid Network](image)

Figure 1. Feature Pyramid Network.

Bottom-up is the forward propagation process of CNN, which forms a natural feature pyramid structure. The C2, C3, C4, and C5 feature maps are transmitted to the right through 1×1 convolution by lateral connections, and the feature maps of different channel numbers are uniformly compressed to 256 channels. This can reduce a large number of parameters, make FPN calculation faster and take up less memory. The C5 layer obtains M5 through 1×1 convolution. M5 is up-sampled twice and merged with horizontally connected C4 to form M4, and so on.

The RoIPooling layer of FPN develops different scales for different levels of pyramids. Roi of different scales use different feature layers as the input of the RoIPooling layer. Large-scale Rois use deep pyramid features, such as P5; small-scale Rois use shallow pyramid features, such as P2. Judge by:

\[
k = \lfloor k_0 + \log_2(\sqrt{wh/224}) \rfloor \quad (1)
\]

224 is the standard image input size of AlexNet. \(k_0\) is set to 5, which represents the output of the P5 layer. \(w\) and \(h\) are the length and width of the Roi area. Assuming that the Roi is \(112 \times 112\), then \(k = 4\), which means that the Roi should use the P4 feature layer. The \(k\) value should be rounded to prevent the result from being an integer.

FPN combines strong expressive ability, low-resolution deep feature maps and weak expressive ability, high-resolution feature maps into strong expressive, high-resolution feature maps. The monitoring area of the project is huge, and the boat far away from the camera is very small. Therefore, FPN is added to the detection module.

3.2. Deformable Convolution

The purpose of the convolution kernel is to extract features. The size of ordinary convolution is fixed and square (for example, 3×3, 5×5). The receptive field size of the common convolution kernel of the same layer is the same, but different positions of the feature map correspond to features of different sizes or different shapes. These features need to automatically adjust the size and shape of the receptive field.
Original convolution has poor adaptability to changes in feature size and shape, and its generalization ability is not strong.

The current method to solve the above problems is to train through a large number of samples to improve the generalization ability of the convolution kernel. Because it is impossible to obtain a large number of data samples in the project, consider using Deformable Convolution (DCN) [17].

Square convolution is not the best shape. The network must learn its shape to adapt to different features. In fact, it is not necessary to change the shape of the convolution kernel to complete the above operation. The convolution operation is essentially that the convolution sampling points act on the pixels on the corresponding feature map. DCN adds an offset to each convolution sampling point to achieve the effect of shape change.

A 3×3 convolution has 9 sampling points, where (−1, −1) represents the lower left sampling point, and (−1,0) represents the left sampling point.

\[ R = \{(-1, -1), (-1,0), \ldots, (0,1), (1,1)\} \] (2)

The output of the conventional 3×3 convolution is expressed by:

\[ y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n) \] (3)

As shown in formula (4), where \( p_n \) belongs to position \( R \), \( w(p_n) \) to indicate the weight of \( p_n \), and \( x(p_0 + p_n) \) to indicate the pixel value in the feature map corresponding to \( p_n \). If you need to change the shape of the convolution kernel, essentially you only need to change the pixel position on the sampled feature map. After \( x(p_0 + p_n + \Delta p_n) \) is added to \( \Delta p_n \), the sampling position has changed, but \( w(p_n) \) has not changed.

\[ y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \] (4)

The final result is that the deformable convolution operation can be completed as long as the sampled pixel position is shifted.

![Figure 2. Deformable Convolution.](image)

As shown in Fig. 2, an additional convolutional layer is used to learn the offset. DCN generates a set of offsets for each pixel position of the feature map. Taking 3×3 convolution as an example, it is necessary to generate 3×3×2=18 offsets, and 18 represents the offset of the 9 sampling points of the position in the x direction and the y direction of the pixel. Then the feature map and offset are used as input together, and the sampling point is shifted first, and then convolution is performed.

DCN can adapt to the shape and size of features, without a large number of samples, and can learn features well. This is very friendly to scenarios where it is difficult to obtain samples.
4. Experiment

4.1. Dataset
The dataset is a self-made 698 boat images, which is manually labeled using LabelImg software, and the labeling information is saved using XML. The format of the entire boat dataset is consistent with the VOC2007, and then converted to the COCO dataset format [18]. Among them, 600 images are randomly selected as the training, and the remaining 98 images are used as the testing. There are no duplicate images in the trainset and testset.

4.2. Result
During the training process, each epoch will randomly select 50% of the images in the training set to be flipped horizontally or vertically for data enhancement. The optimizer uses a combination of SGD and Momentum [19]. The learning rate decay method is a piecewise constant decay. A total of 15 epochs were trained, and the initial learning rate was 0.01. When the training reaches the 9th epoch, the learning rate is reduced to 1/10 of the previous one. When the training reaches the 13th epoch, the learning rate is reduced to 1/10 of the previous one. The gradient descent algorithm with a small learning rate makes the value of the error approach the minimum value of the error function.

The order of the images is shuffled and input into the model, and the order of the images input in each epoch is not consistent. The batch size is 1, and one image is input to the model each time during training. Because the batch size is too small, Batch Normalization [20] is disabled. There are images with a resolution of more than 1080 in the dataset, and images with a resolution of less than 500. They are uniformly scaled to a size from 800 to 1333 which the shortest side is 800, and the longest side is 1333. And keep the original aspect ratio. Transfer the parameters trained on COCO to this model, and then fine-tune the parameters from the C3 layer.

Original model is denoted as R_50_C4, FPN is denoted as R_50_FPN, and DCN is denoted as R_50_FPN_DCN. Since the feature map of the C5 layer of ResNet50 is too small for the target detection task, the ordinary R-CNN generally uses the feature map of the C4 layer. In FPN, the feature maps of the C2–C5 layers will be used. The control variable method ensures that all other parameters remain unchanged.

![Figure 3. Comparison of AP](image)

Figure 3. Comparison of AP
The final experimental results are shown in Fig. 3. Taking IoU=0.5:0.95 as an indicator, the three tend to stabilize at the 10th epoch. At this time, it can be considered that the model has been fully trained. The AP value of R_50_C4 is about 0.45; the AP value of R_50_FPN is about 0.48; the AP value of R_50_FPN_DCN is about 0.51. It can be seen that the detection effect of R_50_FPN_DCN is better. As shown in Fig. 4, APsmall is an evaluation index for small objects, and R_50_FPN_DCN is still best.

5. Conclusions
The project uses the Faster R-CNN model as the basic framework. Add FPN, DCN and other modules. The experimental results show that the model can detect boats on the sea very well, and it still has a good effect in the small objects and in the case of a small dataset.

Acknowledgments
This work was financially supported by STATE GRID ZHEJIANG ELECTRIC POWER CORPORATION Collective Enterprise Science and Technology Project (NBGC20201127-03) fund.

References
[1] Wu Y, Chen Y, Yuan L, Liu Z, Wang L, Li H, Fu Y. Rethinking Classification and Localization for Object Detection [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2020: 10186-10195.
[2] LeCun Y, Bengio Y, Hinton G. Deep Learning [J]. Nature, 2015, 521(7553): 436-444.
[3] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks [C]. International Conference on Neural Information Processing Systems. 2012:1097-1105.
[4] Radosavovic I, Kosaraju R P, Girshick R, et al. Designing Network Design Spaces [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2020: 10428-10436.
[5] Hubel D H, Wiesel T N. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex [J]. The Journal of Physiology, 1962, 160(1): 106-154.
[6] He K, Zhang X, Ren S, Sun J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition [C]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015: 1940-1946.
[7] Ren Shaoqing, He Kaiming, Ross Girshick, Sun Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [C]. Advances in Neural Information Processing Systems. 2015: 91-99.
[8] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 779-788.
[9] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.
[10] Ding X, Zhang X, Ma N, et al. RepVGG: Making VGG-Style ConvNets Great Again. ArXiv Preprint ArXiv: 2101.03697: 1-10.
[11] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.
[12] Min L, Chen Q, Yan S. Network In Network [C]. Advances in International Conference on Learning Representations. 2014: 1-9.
[13] Lin T Y, Dollár P, Girshick R, et al. Feature Pyramid Networks for Object Detection [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 2117-2125.
[14] He K, Gkioxari G, Dollár P, et al. Mask R-CNN [C]. Proceedings of the IEEE International Conference on Computer Vision. 2017: 2961-2969.
[15] Yun S, Han D, Oh S J, Yoo Y, Choe J. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features [C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 6023-6032.
[16] Han K, Vedaldi A, Zisserman A. Learning to Discover Novel Visual Categories via Deep Transfer Clustering [C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 8401-8409.
[17] Everingham M, Van Gool L, Williams C K I, et al. The Pascal Visual Object Classes (VOC) Challenge[J]. International Journal of Computer Vision, 2010, 88(2): 303-338.
[18] Kingma D, Ba J. Adam: A Method for Stochastic Optimization [C]. Advances in International Conference on Learning Representations. 2015: 1-13.
[19] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [C]. International Conference on Machine Learning. 2015: 448-456.
[20] Dai J, Qi H, Xiong Y, et al. Deformable Convolutional Networks [C]. Proceedings of the IEEE International Conference on Computer Vision. 2017: 764-773.