Ocean Front Detection With Bi-Directional Progressive Fusion Attention Network

Jing Zhu, Qingyang Li, Cui Xie, *Member, IEEE*, and Guoqiang Zhong, *Member, IEEE*

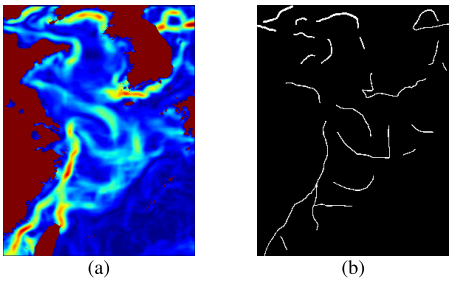
***Abstract*— Ocean fronts are a mesoscale phenomenon in the ocean. It is important for fisheries, environmental protection, and military activities. Therefore, more and more attention has been attracted to ocean front detection. However, the distribution of front and nonfront pixels is highly unbalanced in remote-sensing images, and it is not easy to establish an effective ocean front detection algorithm with high accuracy. To alleviate these problems, we model the problem of detecting ocean fronts as an edge detection task and design a new end-to-end bi-directional progressive fusion attention network (BPFANet). Specifically, BPFANet consists of an effective backbone and a bi-directional path. The whole backbone has four-stage detection blocks (SD blocks), which capture the ocean front features at different scales. Each SD block contains a side branch structure, which includes a deep residual dilated convolution (DRDC) module to enrich multiscale edge information and an attention module (AM) to enhance the feature representation in both the channel and spatial dimensions. In addition, the bi-directional path can progressively fuse the four SD blocks of ocean front information. To evaluate BPFANet, we perform experiments on the OFDS365 dataset and show its advantages over existing ocean front detection methods.**

***Index Terms*— Deep learning, edge detection, ocean front.**

I. Introduction

OCEANFRONTS are the transition regions between water bodies of different properties, such as temperature and density. Ocean fronts are often accompanied by material transport, energy exchange, and fish activity. Hence, it is crucial to detect ocean fronts in some applications, such as remote sensing and marine fisheries.

Most previous work [1] explore ocean front detection based on gradient thresholding methods. However, these methods rely on the manual tuning of algorithm parameters, resulting in low accuracy. Recently, some machine-learning algorithms are applied for recognizing the position of the ocean front and nonfront regions [5], [6]. However, these methods cannot accurately detect ocean fronts.



(a) (b)

Fig. 1. Examples of the used OFDS365 dataset. (a) SST gradient image and (b) ground truth. In our work, we formulate ocean front detection as an edge detection problem.

There are two main challenges in implementing ocean front detection with deep-learning models. First, the training of supervised deep neural networks requires a large amount of labeled data. Second, because of the highly unbalanced distribution of ocean fronts in remote-sensing images, their detection is fairly difficult. Therefore, it is not easy to build an effective algorithm that can accurately detect ocean fronts.

To tackle the above challenges, Li et al. [7] proposed a weak edge identification network (WEIN) to detect ocean fronts in sea surface temperature (SST) gradient images. To improve the ocean front detection performance based on [7], we model ocean front detection as an edge detection problem. An example of the ocean front image and its corresponding ground truth is illustrated in Fig. 1. We build a bi-directional progressive fusion attention network (BPFANet) that consists of four-stage detection blocks (SD blocks), a top-down path, and a bottom-up path. Each block contains a side branch structure with an embedded deep residual dilated convolution (DRDC) for capturing multiscale features, and an attention module (AM) for removing background noise from the channel dimension and spatial dimension, thus achieving cross-channel interaction. Furthermore, the top layer of BPFANet helps to capture weak edge information of ocean fronts, and the bottom layer extracts large-scale ocean fronts. Hence, the bi-directional paths fuse rich multiscale ocean front information.

The main contributions of this work are as follows.

我们将海洋检测作为边缘检测问题建模，并在此基础上提出BPFANet。

1) We model ocean front detection as an edge detection problem and based on it propose BPFANet.
2) The designed bi-directional network can progressively fuse multiscale feature representations. Moreover, we propose the DRDC that can fully exploit the

Manuscript received 9 November 2022; revised 9 March 2023; accepted 19 April 2023. Date of publication 25 May 2023; date of current version 16 June 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100400, in part by the Hui-Yan Action Project under Grant LZY2022033004, in part by the Natural Science Foundation of Shandong Province under Grant ZR2020MF131 and Grant ZR2021ZD19, in part by the Project of the Marine Science and Technology Cooperative Innovation Center under Grant 22-05-CXZX-04-03-17, in part by the Science and Technology Program of Qingdao under Grant 21-1-4-ny-19-nsh, and in part by the Project of Associative Training of Ocean University of China under Grant 202265007. *(Corresponding author: Guoqiang Zhong.)*

The authors are with the College of Computer Science and Technology, Ocean University of China, Qingdao 266100, China (e-mail: jingzhu@stu.ouc.edu.cn; 1194094543@qq.com; spring@ouc.edu.cn; gqzhong@ouc.edu.cn).

Digital Object Identifier 10.1109/LGRS.2023.3279783

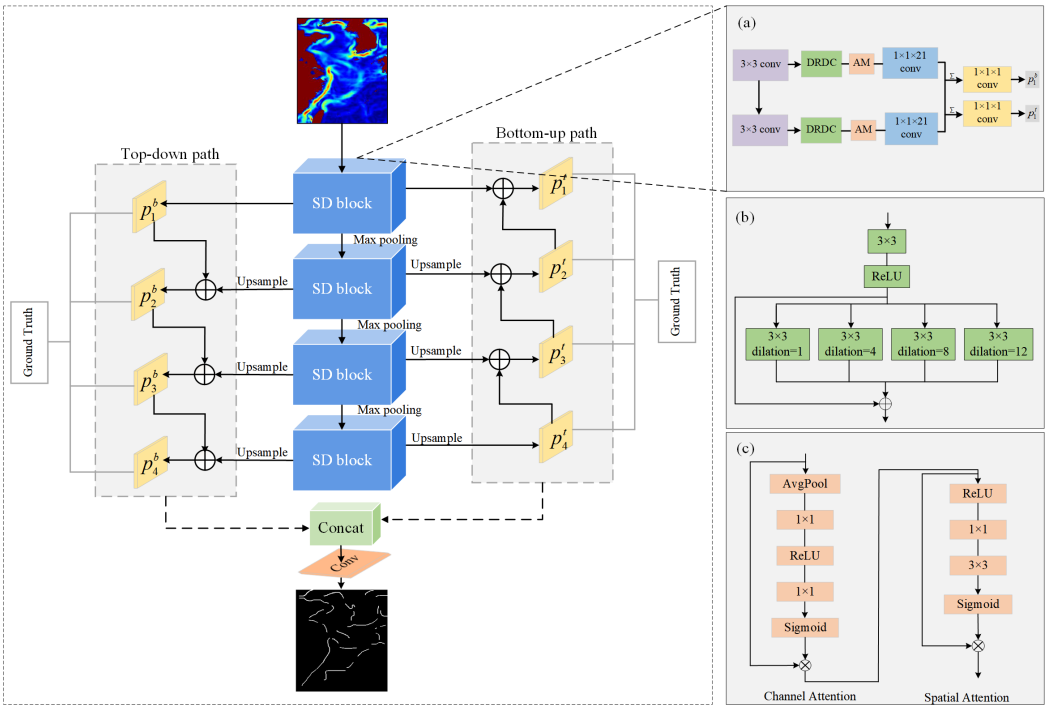


Fig. 2. Overall architecture of BPFANet. p_m denotes the prediction map at the mth ($1 \leq m \leq 4$) SD block. The ocean front prediction maps obtained at the top layer can be progressively fused to the bottom layer, while the bottom-up paths are similar, only with a different direction.

multiscale ocean front features and the AM that can refine the feature map.

3) Experiments on the OFDS365 dataset demonstrate the superiority of BPFANet over existing ocean front detection methods.

The rest of this letter is organized as follows. Section II summarizes the related work. Section III presents the network design of BPFANet. Section IV shows the experimental results. Finally, Section V concludes this work with remarks and future work.

II. Related Work

A. Ocean Front Detection Algorithms

Early ocean front detection methods mainly focus on calculating the image gradients to obtain ocean fronts. For example, Belkin and O'Reilly [1] use contextual filters to remove noise on the images and then use the gradient method to detect ocean fronts; Kirches et al. [2] proposed to combine the gradient method with a histogram algorithm to detect ocean fronts. However, the gradient-based methods are sensitive to noisy data and have low accuracy in detecting complex ocean fronts. Recently, many machine-learning algorithms have been proposed for ocean front detection. Sun et al. [5] used a cooperative game theory-based random forest method to achieve the ocean front classification. Lima et al. [6] identified ocean fronts in both color and gray-scale images by fine-tuning AlexNet [8]. Li et al. [7] proposed WEIN based on a convolutional neural network (CNN). However, the ocean fronts detected by these models generally contain noise.

B. Edge Detection Algorithms

In early work, the Sobel operator [3] computes the gradient map and then extracts the edges. The Canny operator [4] utilizes a dual-thresholding method to detect the edges. However, these algorithms rely on hand-crafted features, thus limiting the ability to accurately detect meaningful edges. In recent years, many CNN-based edge detection methods have been proposed. HED [9] fuses multilevel features to achieve edge detection. RCF [10] improves the skip layer structure on [9] based on predicting each block independently. BDCN [11] introduces a bi-directional cascade network to detect edges. PiDiNet [12] integrates the traditional edge detection operator into the convolution operation in CNNs. To address the issue of ocean front detection, this letter models it as an edge detection problem and designs an effective ocean front detection model.

III. Method

A. BPFANet

The overall architecture of BPFANet is illustrated in Fig. 2. The max-pooling layer is connected after each SD block, which progressively reduces the dimension of the feature map while increasing its receptive field.

To enrich the hierarchical features, each SD block uses a side branch structure to generate the ocean front prediction map. As shown on the right side of Fig. 2(a), the side branch structure of the SD block consists of the convolutional layers, a DRDC, and an AM. The DRDC obtains the output of the 3×3 convolution layer and fuses the multiscale features captured by the dilated convolution with different sampling rates through the residual connection. The output of DRDC is then fed to AM to calculate the attention weights of the feature maps. The $1 \times 1 \times 21$ convolutional layer is used to reduce the number of channels in the output feature map, where 21 is the number of channels. Finally, the 1×1 convolution compresses the output prediction map into a single channel. In addition,

the last three SD blocks upsample the ocean front prediction map to the original size. Specifically, the detailed description of DRDC and AM is as follows.

1) *Deep Residual Dilated Convolution:* To learn multiscale edge representations and make full use of the contextual information, we have designed the DRDC module. Inspired by the fact that dilated convolution can increase the receptive field, DRDC adopts dilated convolution to improve the ability to extract multiscale features. At the same time, we use residual connections to enhance the use of detailed information. Concretely, $s \in \mathbb{R}^{H\times W}$ and $f \in \mathbb{R}^{h\times w}$ represent the input feature map and the convolution filter, respectively. Then, A_{ij} is the output of the dilated convolution at the pixel (i, j), calculated as follows:

$$A_{ij} = \sum_{u,v}^{h,w} s_{[i+d\cdot u, j+d\cdot v]} \cdot f_{[u,v]} \tag{1}$$

where d is the dilation rate and $d = 1$ represents the standard convolution.

2) *Attention Module:* To extract clear ocean fronts and reduce background interference, AM enhances the ocean front features from both channel and spatial dimensions. Specifically, channel attention assigns a weight to each feature map on each channel and models the importance between features to effectively extract the ocean fronts. The spatial attention focuses on the spatial location information of the features, thereby reducing the background noise and helping to generate clear ocean fronts. To aggregate spatial information, the input feature $A \in \mathbb{R}^{H\times W\times C_{\text{in}}}$ is first mapped to the vector $A_c \in \mathbb{R}^{1\times 1\times C}$ by taking the global average pooling (GAP) in the channel dimension, which is illustrated in Fig. 2. Next, after a 1×1 convolutional layer, the A_c space dimension is transformed to $\mathbb{R}^{1\times 1\times C/r}$, where r, as the reduction rate, is set to 8. After that, it goes through the ReLU activation function. Finally, a 1×1 convolution reshapes the feature volume to C, followed by a sigmoid function to create the channel attention map X. Furthermore, the output feature map is fed to the spatial attention and we obtain the feature map $A_s \in \mathbb{R}^{H\times W\times 4}$ by compressing the channel dimension using 1×1 convolution. After that, the spatial information is extracted by performing a 3×3 convolution, and then the dimension is normalized to $\mathbb{R}^{H\times W\times 1}$ by the sigmoid function to obtain the attention map Y. The above process is computed as

$$\begin{aligned} X(A) &= \sigma\left(f^{1\times 1}(\text{GAP}(A))\right) \\ Y(A) &= \sigma\left(f^{3\times 3}\left(f^{1\times 1}(A)\right)\right) \end{aligned} \tag{2}$$

where f represents a convolution operation, the right superscript is the convolution kernel size, and σ denotes the sigmoid function.

Let $A' \in \mathbb{R}^{H\times W\times C_{\text{out}}}$ denotes the output feature map. $X \in \mathbb{R}^{1\times 1\times C}$ and $Y \in \mathbb{R}^{H\times W\times 1}$ denote the channel attention map and the spatial attention map, respectively.

The output of the AM is as follows:

$$A' = Y(A) \otimes (X(A) \otimes A) \tag{3}$$

where $\otimes$ denotes element-wise multiplication.

Each SD block generates two ocean front predictions p_m^t and p_m^b, which are fused by the bottom-up and top-down paths, respectively. Finally, all the ocean front predictions are concatenated as the final output.

In BPFANet, as the ocean front information captured by each SD block becomes coarser with the increase of its receptive field, the ocean front prediction scales from the four SD blocks are different. Therefore, inspired by [11], we decompose the ground truth into different scales for deep supervision of ocean front prediction. Specifically, using G to denote the ground truth, we define the supervision for the mth SD block as follows:

$$\begin{aligned} G_m^t &= G - \sum_{i>m} p_i^t \\ G_m^b &= G - \sum_{i<m} p_i^b \end{aligned} \tag{4}$$

where p_i denotes the ocean front prediction at the ith SD block. The superscripts t and b denote the bottom-up and top-down paths, respectively. G_m^t and G_m^b denote the supervised image of the bottom-up path and the top-down path at the mth SD block, respectively.

B. Loss Function

Considering that the ocean front image contains more non-front pixels, the distribution of positive and negative samples is unbalanced. Therefore, we adopt the Dice loss [13] to calculate the similarity difference between the two image sets, which is defined as

$$L_{\text{Dice}}\left(b_i, \hat{b}_i\right) = \frac{\sum_i^I b_i^2 + \sum_i^I \hat{b}_i^2}{2\sum_i^I b_i \hat{b}_i} \tag{5}$$

where I denotes the total number of pixels on the map and $b_i, \hat{b}_i$ denote the pixels from the ith prediction and ground truth, respectively.

Moreover, to minimize the distance between each prediction map and the ground truth from the pixel level, we use the binary cross-entropy loss for each prediction map. The binary cross-entropy loss for an image that has I pixels is as follows:

$$L_{\text{BCE}}\left(b_i, \hat{b}_i\right) = -\sum_i^I \left[b_i \log \hat{b}_i + (1 - b_i)\log\left(1 - \hat{b}_i\right)\right]. \tag{6}$$

By combining these two loss functions, the loss function for each prediction is given by

$$L = L_{\text{Dice}} + \mu L_{\text{BCE}} \tag{7}$$

where $\mu \geq 0$ is a hyperparameter used to balance the two terms.

During the training, the final loss of the ocean front prediction can be defined as

$$L_{\text{final}} = w_{\text{side}} \cdot L_{\text{side}} + w_{\text{fuse}} \cdot L_{\text{fuse}}$$

$$L_{\text{side}} = \sum_{m=1}^{M} \left(L\left(p_m^b, G_m^b\right) + L\left(p_m^t, G_m^t\right) \right)$$
$$L_{\text{fuse}} = L(p, G) \tag{8}$$

where w_{side} and w_{fuse} represent the weight of side branch structure loss and the weight of fusion loss, respectively. With $M = 4$ in (8), each SD block has two prediction p_m^t and p_m^b. In addition, p denotes the fusion map that aggregates all prediction maps.

IV. Experiments

A. Dataset

We used a labeled OFDS365 dataset [7] for ocean front detection. This dataset was collected from the region 118.025°E ∼ 130.975°E and 23.025°N ∼ 39.975°N, containing 365 images calculated from day-by-day SST in 2014, all of which have a resolution of 340 × 260 × 3 pixels. In our experiment, we selected 305 images as the training set and 60 images as the test set, where the test set was composed of randomly selected five images from each month.

B. Implementation Details

We implemented BPFANet using the Pytorch library. In detail, we initialized the backbone using the VGG16 [14] pretrained model. To compare with other deep-learning methods, we set the same parameters as follows: batch size was set to 10, initial learning rate to 1e-6, the momentum to 0.9, weight decay to 2e-4, training for 30k iterations, and the learning rate decreased to ten times the original after every 5 k iterations. The hyperparameter μ for balancing two loss functions of (7) was set to 0.001. In (8), the weight w_{side} was set to 0.5, and w_{fuse} was empirically set to 1.1. To avoid overfitting, we used batch normalization and dropout for model regularization. We used SGD as an optimizer and trained the model on an NVIDIA GeForce1080Ti GPU.

C. Performance Metrics

We used Precision, Recall, F_1-score, intersection over union (IoU), and structure similarity (SSIM) as evaluation metrics. Precision and Recall are obtained by true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The F_1-score is computed based on Precision (P) and Recall (R), where $P = (\text{TP})/(\text{TP} + \text{FP})$, $R = (\text{TP})/(\text{TP} + \text{FN})$, and $F_1 = (2 \times P \times R)/(P + R)$. IoU is calculated as the ratio of the intersection and union of ground truth and predicted edges, where $\text{IoU} = (\text{TP})/(\text{TP} + \text{FN} + \text{FP})$. SSIM is a measure of the similarity of two images, which can capture the structural information of the image. The higher the score of these five metrics, the better the performance.

D. Ablation Study

We evaluated the importance of each part of BPFANet by conducting a series of ablation studies. The results are summarized in Table I. Ours-w/o-DRDC refers to BPFANet without the proposed DRDC module, and Ours-w/o-AM refers to BPFANet without the proposed AM. First, we measured the detection effect of the baseline network (Ours-w/o-DRDC-w/o-AM) and obtained the F_1-score, IoU, and SSIM were 0.748, 0.816, and 0.829, respectively. Then, only using BPFANet with DRDC and BPFANet with AM can improve the baseline performance. Furthermore, we combine DRDC and AM in BPFANet to test its performance. It is easy to see that BPFANet achieves the best performance on the five metrics compared to other approaches.

TABLE I
Effects of Components in BPFANet

Method	P	R	F_1-score	IoU	SSIM
Ours-w/o-DRDC-w/o-AM	.780	.718	.748	.816	.829
Ours-w/o-AM	.802	.721	.759	.830	.863
Ours-w/o-DRDC	.797	.729	.762	.833	.875
Ours	**.815**	**.732**	**.772**	**.847**	**.882**

It is easy to understand that, with DRDC and AM, BPFANet can obtain extra performance gains over its variants. Specifically, the dilated convolution used in the DRDC module can learn the features of multiple scales of receptive fields. Therefore, BPFANet can achieve promising accuracy on ocean front detection tasks.

E. Comparison With the State-of-the-Art

We compared our model with traditional edge detection-based methods, including Sobel detector [3] and Canny detector [4], edge detection methods using CNNs, including HED [9], RCF [10], BDCN [11] and PiDiNet [12], and deep-learning-based ocean front detection methods, including WEIN [7]. The Sobel and Canny detector focus on computing image gradients. HED and RCF perform edge detection by multiscale full convolutional networks. BDCN uses a cascade architecture and PiDiNet computes pixel difference convolution for edge detection. WEIN has a single-direction multistage network for ocean front detection. Table II shows the quantitative results of our method and other methods on the test set, and we can observe that BPFANet achieves better performance than others. Specifically, the results on IoU and SSIM show that BPFANet has superiority over the state-of-the-art approaches to ocean front detection.

Fig. 3 shows the detection results of our method and other ocean front detection methods on the OFDS365 dataset. The ocean fronts obtained by the Sobel detector are blurry and even contain many unrelated regions, such as the boundary between land and ocean. The Canny detector performs slightly better than the Sobel detector in the ocean front detection task. However, its detection results contain many discontinuous nonfront regions. In general, traditional edge detection methods based on gradients are more sensitive to sharp brightness changes in images, which is not beneficial to obtaining accurate ocean front detection. Moreover, deep-learning-based methods promote learning meaningful ocean front features under deep supervision to make predictions. Nevertheless, HED, RCF, and BDCN detect fronts that are

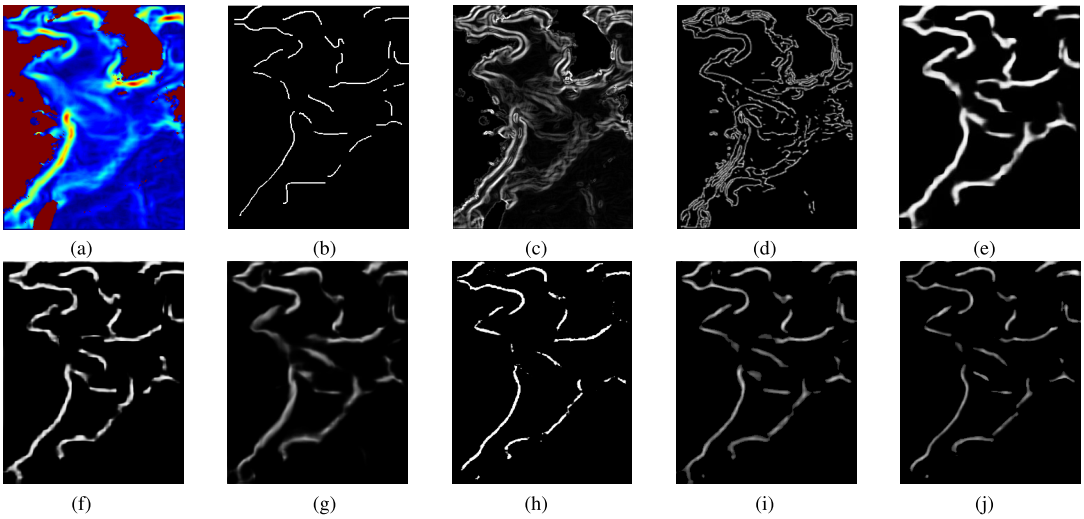


Fig. 3. Comparison of the results of different ocean front detection methods on the OFDS365 dataset. (a) Input image. (b) Ground truth. (c)–(j) Detection outputs obtained by Sobel, Canny, HED, RCF, BDCN, PiDiNet, WEIN, and BPFANet, respectively.

TABLE II
PERFORMANCE OF BPFANET AND COMPARISON METHODS ON THE TEST SET OF THE OFDS365 DATASET

Method	P	R	F_1-score	IoU	SSIM
Sobel [3]	.675	.403	.451	.522	.289
Canny [4]	.704	.468	.518	.610	.390
HED [9]	.803	.580	.622	.727	.701
RCF [10]	.812	.622	.661	.761	.740
BDCN [11]	.810	.634	.689	.771	.819
PiDiNet [12]	.808	.649	.701	.781	.795
WEIN [7]	.814	.673	.707	.803	.851
BPFANet	**.815**	**.732**	**.772**	**.847**	**.882**

still blurry and noisy. PiDiNet ignores some details of ocean fronts. Additionally, WEIN is less accurate in detecting ocean fronts compared to BPFANet. In contrast, BPFANet has an advantage over WEIN in prediction quality, which generates accurate and fine-grained ocean fronts.

V. CONCLUSION

This letter formulates ocean front detection as an edge detection problem. To detect ocean fronts in remote-sensing images, we build the BPFANet model. BPFANet includes a progressive fusion network with top-down and bottom-up paths, which effectively extracts detailed textures and large-scale features of ocean fronts. Moreover, BPFANet employs a DRDC module in its SD block to enrich the multiscale ocean front features and an AM module to refine the features around the ocean fronts and reduce the background noise. Besides, the loss function guides the proposed BPFANet to fit the ocean front distribution. Finally, the experimental results on the OFDS365 dataset show that BPFANet obtains the best results compared to other approaches. In the future, we will try to collect more ocean front data to supplement ocean front detection and optimize our proposed model for applying it to other tasks, such as semantic segmentation on remote sensing images.

ACKNOWLEDGMENT

The authors would like to thank "Qingdao AI Computing Center" and "Eco-Innovation Center" for providing inclusive computing power and technical support for MindSpore during the completion of this letter.

REFERENCES

[1] I. M. Belkin and J. E. O'Reilly, "An algorithm for oceanic front detection in chlorophyll and SST satellite imagery," *J. Mar. Syst.*, vol. 78, no. 3, pp. 319–326, Oct. 2009.
[2] G. Kirches, M. Paperin, H. Klein, C. Brockmann, and K. Stelzer, "GRADHIST—A method for detection and analysis of oceanic fronts from remote sensing data," *Remote Sens. Environ.*, vol. 181, pp. 264–280, Aug. 2016.
[3] J. Kittler, "On the accuracy of the Sobel edge detector," *Image Vis. Comput.*, vol. 1, no. 1, pp. 37–42, Feb. 1983.
[4] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[5] J. Sun, G. Zhong, J. Dong, H. Saeeda, and Q. Zhang, "Cooperative profit random forests with application in ocean front recognition," *IEEE Access*, vol. 5, pp. 1398–1408, 2017.
[6] E. Lima, X. Sun, J. Dong, H. Wang, Y. Yang, and L. Liu, "Learning and transferring convolutional neural network knowledge to ocean front recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 354–358, Mar. 2017.
[7] Q. Li, G. Zhong, C. Xie, and R. Hedjam, "Weak edge identification network for ocean front detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[9] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.
[10] Y. Liu, M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5872–5881.
[11] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "BDCN: Bi-directional cascade network for perceptual edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 100–113, Jan. 2022.
[12] Z. Su et al., "Pixel difference networks for efficient edge detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5097–5107.
[13] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945.
[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.