Identifying Oceanic Eddy With an Edge-Enhanced Multiscale Convolutional Network

Renlong Hang, *Member, IEEE*, Gang Li, Mei Xue, Changming Dong, and Jianfen Wei

***Abstract*—Identifying oceanic eddy from remotely sensed sea surface height (SSH) data is challenging, mainly because of its large-size variations. This article proposes an automatic identification model upon convolutional neural networks for dealing with this issue. The proposed network is comprised of two branches: an eddy identification branch and an edge extraction branch. Both of them adopt encoder–decoder frameworks, and the encoder is shared with each other. The eddy identification branch simultaneously uses multiscale convolution modules in the encoder and skip-layer connections between the encoder and the decoder to learn multiscale features, thus effectively identifying eddies with different sizes. Differently, the edge extraction branch is designed to learn the edge information of eddies, which is not fully captured by the eddy identification branch. To sufficiently evaluate the identification performance of our proposed model, several experiments are conducted on a public eddy identification dataset named SCSE-Eddy, and the results indicate that the proposed model is capable of achieving higher performance than those of the existing identification models.**

***Index Terms*—Convolutional neural network, eddy identification, edge extraction, multiscale features.**

I. Introduction

OCEANIC eddy is a pervasive phenomenon in global ocean currents, which has an important impact on the primary production of the ocean [1], [2], [3], [4]. It has variable translational and rotational speeds, and its shape is usually asymmetric [5], [6]. According to the direction of rotation, oceanic eddy can be classified as cyclonic eddy and anticyclonic eddy. The top-down vertical mixing and dispersion motion of cyclonic eddy can transport nutrients from the seafloor to the surface, which enhance phytoplankton photosynthesis [7], [8], [9]. In contrast, the sinking motion of anticyclonic eddy can reduce nutrients at the surface, which weaken phytoplankton photosynthesis. Meanwhile, the primary production enhanced by oceanic eddy can also affect the distribution of higher level marine populations (e.g., marine fishes) [10]. This suggests that oceanic eddy is of great importance in local ecosystems. Hence, accurately identifying oceanic eddy is beneficial to the study of the physical properties of ocean and is also important to marine science.

In the past few decades, numerous approaches have been designed to identify oceanic eddy. For instance, Isern-Fontanet et al. [11] first used Okubo Weiss (OW) parameters for identifying eddies on the basis of their physical properties, which assumed that the OW of the oceanic eddies is less than a specific threshold. Subsequently, Chelton et al. [12] optimized the calculation of OW by using the geostrophic current velocity component. Based on different physical parameters adopted, many methods have emerged, such as Q-criterion [13], Ω-criterion [14], Δ-criterion [15], and λ_2-criterion [16]. Similarly, these methods need to set thresholds in advance and calculate relevant physical parameters to identify oceanic eddies. Doglioli et al. [18] got their inspirations from [17], and proposed a wavelet analysis-based identification model, which decomposes the relative sea surface vorticity into orthogonal wave packets. The method has been successfully applied to identify eddies in the Cape Basin. However, it is also constrained by the threshold value and difficult to be applied to all sea areas.

In addition to the aforementioned physical models, another popular kind of methods is dependent on the geometric properties of eddies. Chaigneau et al. [19] were the first to propose a winding angle (WA) method for identifying eddies, which defines the WA as the rotation angle of the streamline with respect to the eddy center. This method is mainly dependent on the global topological property of the flow field and does not depend on the parameter settings. Compared with the OW parameter methods, the WA method is more capable of identifying eddies [19]. However, its computational complexity is high; thus, the identification efficiency is relatively low. Dong et al. [20] used the vector geometry (VG)-based algorithm to identify eddies. The VG algorithm calculates the eddy center based on the motion patterns of the velocity vector, and the eddy size is calculated from the closed contours of the stream function. Similar to the WA method, although the VG algorithm improves the accuracy of eddy identification, its computation efficiency also needs to be improved. In [11] and [21], the researchers

Manuscript received 1 September 2022; revised 7 October 2022; accepted 9 October 2022. Date of publication 19 October 2022; date of current version 2 November 2022. This work was supported in part by the Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai) funded project under Grant SML2020SP007 and in part by the Natural Science Foundation of China under Grant 61906096. *(Corresponding author: Jianfen Wei.)*

Renlong Hang, Gang Li, and Mei Xue are with the Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, School of Computer, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with the Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), Zhuhai 519000, China (e-mail: renlong_hang@163.com; 20201249093@nuist.edu.cn; meixue@njucm.edu.cn).

Changming Dong is with the Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), Zhuhai 519000, China, and also with the Oceanic Modeling and Observation Laboratory, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: cmdong@nuist.edu.cn).

Jianfen Wei is with the School of Atmospheric Science and Remote Sensing, Wuxi University, Wuxi 214105, China (e-mail: weijianfen0425@163.com).

Digital Object Identifier 10.1109/JSTARS.2022.3215696

considered the advantages of physical and geometric hybrid approaches to improving the performance of eddy identification.

With its fast advancement recently, deep learning technology has been successfully employed in remote sensing data analysis, such as urban planning [22], [23] and land-cover and land-use classification [24]. Thus, many scholars attempted to apply it to identify oceanic eddies. For example, Franz et al. [25] proposed an encoder–decoder network with several convolutional layers to identify eddies. Liu et al. [26] designed a fully convolutional network (FCN) using eight convolutional layers and one upsampling layer for eddy detection. However, different from other objects, the size of eddies ranges largely. Specifically, its diameter often changes from a few hundred meters to hundreds of kilometers. The individual use of convolutional layers is not enough to simultaneously identify different-size eddies. To deal with this issue, some work tried to extract multiscale features. Lguensat et al. [27] got their inspirations from UNet [28] and proposed Eddynet for eddy identification. Similarly, Fan and Zhong [29] proposed a SymmetricNet, which also drew on the idea of UNet. The SymmetricNet not only uses the sea surface height (SSH) data as input, but also combines sea surface temperature data and other related information. Inspired by ResNet, Santana et al. [30] introduced the residual structure into UNet to deepen the network while obtaining better identification results. All of these models adopt skip-layer connections to combine different-scale features from different layers. Different from them, Xu et al. [31] used PSPnet [32] to extract multiscale features in a single layer through the pyramid pooling operator. Similarly, Liu et al. [33] also used a pyramid pooling structure to learn the multiscale features of eddies. Sun et al. [34] employed a modified Deeplabv3+ network proposed in [35] to learn multiscale features. Although these networks are capable of identifying most eddies, they often lose the detailed edge information of eddies caused by the adopted pooling and convolutional operations.

To address the above issues, this article proposes a two-branch convolutional neural network (TBCNN) for identifying oceanic eddies. It mainly consists of an eddy identification branch and an edge extraction branch (EEB). The former one aims to identify eddies with different sizes, while the purpose of the latter one is enhancing the eddy edges, which is not well captured by the former one. Both of these two branches employ the encoder–decoder frameworks, and the encoder is shared by each other. For the eddy identification branch, we adopt multiscale convolutional layers [36] in the encoder and skip-layer connections between the encoder and the decoder to learn multiscale features. Considering the existence of redundancy information between the encoder and the decoder, a difference enhancement module (DEM) is designed during the skip-layer connections. Such a simultaneous use of single-layer and multiple-layer method can extract more effective features, benefiting the identification of eddies with different sizes. For the EEB, the skip-layer connections are also applied to transfer the spatial information between the encoder and the decoder, thus helping the extraction of detailed edge information. To sum up, the primary contributions of this article can be summarized as follows.

1) For effectively identifying the oceanic eddies with different sizes, we propose a two-branch identification network, which decomposes the whole process into an EEB and an eddy identification branch. The edge extraction one is capable of enhancing the edge information of identified eddies.
2) Different from most existing eddy identification networks, we use single-layer and multiple-layer feature fusion methods simultaneously to learn multiscale features more effectively. For fusing multiple-layer features, DEMs are designed to reduce redundancy information between different layers.
3) Comprehensive experiments are performed upon a publicly available dataset named SCSE-Eddy, and the experimental results indicate that the proposed network is capable of achieving higher performance than those of several existing models, which certifies the effectiveness of it.

The rest of this article is organized as follows. Section II presents the detailed network structure of our eddy identification model. In Section III, the adopted dataset for our experiments and the obtained results of different identification models are introduced. Finally, Section IV concludes this article.

II. Methodology

A. Overall Structure

Fig. 1 demonstrates the flowchart of our proposed TBCNN model. It is comprised by two parallel branches. Assume that $X \in R^{w \times h}$, where h and w denote the height and the width of X, respectively, is the input SSH data. It is simultaneously fed into the two branches. The upper branch is designed for identifying oceanic eddies with different sizes. To achieve this goal, an encoder–decoder structure is employed. The encoder is equipped with three multiscale convolution modules (MCMs) to learn multiscale features from X:

$$F^i = \mathbf{MT}(F^{i-1}) \tag{1}$$

where $i \in \{1, 2, 3\}$, $F^0 = X$, and $\mathbf{MT}$ denotes the multiscale convolutional operator, which will be presented in Section II-B. After that, F^3 is enhanced by a convolution block, which contains two successive convolutional layers, to derive the final feature representation.

In contrast with the encoder, the decoder aims at progressively extracting high-resolution features for eddy identification. It contains three DEMs, each of which is followed by a convolutional block. The DEM is able to incorporate the shallow features from the encoder to the decoder, thus fusing different scale features effectively. Specifically, for the ith module, its output D^i is formulated as

$$D^i = \mathbf{DE}\left(\mathbf{ConvB}\left(D^{i-1}\right), F^{4-i}\right) \tag{2}$$

where $\mathbf{ConvB}$ represents the two convolutional operators in the convolutional block, $D^0 = F^3$, and $\mathbf{DE}$ denotes the difference enhancement operator, which will be presented in Section II-C.

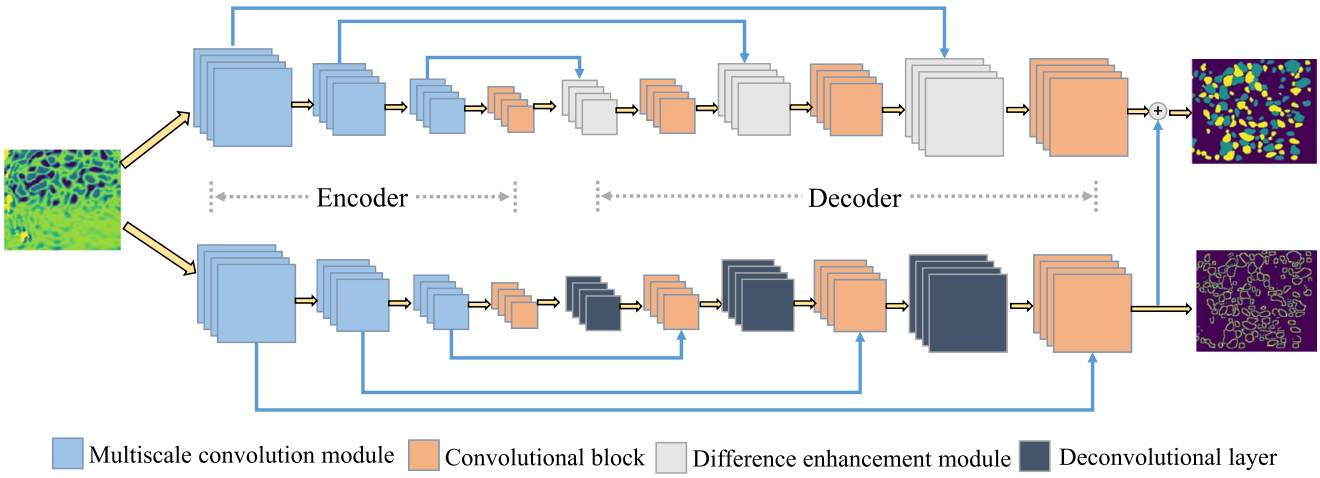

Fig. 1. Flowchart of our proposed TBCNN for oceanic eddy identification.

Besides the upper branch, there also exist a bottom branch in Fig. 1. Similarly, it is built upon the encoder–decoder framework. The encoder has the same network structure to extract features, i.e., three MCMs and a convolutional block. Considering that the shallow features usually have powerful generalization ability across different tasks, we make the encoder share parameters between the upper and the bottom branches, thus reducing the network parameters to train. For the decoder, three deconvolutional layers are applied to learn high-resolution features sequentially. Following each deconvolutional layer, a convolutional block is used to fine-tune the learned features. Since the purpose of the bottom branch is outputting an eddy edge map, its high-resolution features should contain discriminative information for eddy edges. Therefore, we incorporate the final feature into the upper branch via a summation operator to obtain an edge-enhanced eddy identification result.

The detailed configuration of our proposed TBCNN is summarized in Table I, where the kernel size, the number of convolutions or deconvolutions in each module, and the output size after each operator are demonstrated. Take the first MCM as an instance, the operator "[Conv3 $\times$ 3] $\times$ 5" means that there are five convolutional layers in the module and each one has a kernel size of 3 $\times$ 3. In addition, their stride is set to 1 because $S = 1$ in the "Stride" column.

B. MCM

As discussed in Section I, existing eddy identification networks often employ single-layer or multiple-layer fusion methods to extract multiscale features. Our proposed TBCNN differs from them by using both fusion methods at the same time. Fig. 2 presents the architecture of our MCM to extract multiscale features in a single layer. Specifically, for an input feature map F^{i-1}, we first apply a convolutional layer, whose kernel size is 3×3 to increase its channel number by four times. Then, we equally divide the new feature map into four feature subsets, denoted as $x^j, j \in \{1, 2, 3, 4\}$. They will undergo different operators. For the first subset x^1, we directly forward it to the next layer. For the other subsets, we sequentially adopt convolutional operators,

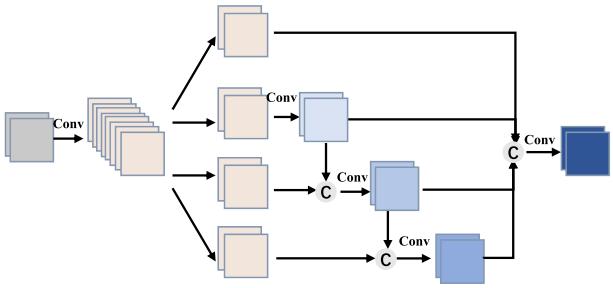

Fig. 2. Detailed structure of the MCM.

whose kernel size is 3×3, to extract features. Assume that the corresponding output feature of x^j is y^j, the computation process can be expressed as

$$y^j = \begin{cases} x^j, & j = 1 \\ \mathbf{Conv}(x^j), & j = 2 \\ \mathbf{Conv}(\mathbf{Cat}(y^{j-1}, x^j)), & j = \{3, 4\} \end{cases} \tag{3}$$

where **Conv** is a convolutional operator, and **Cat** represents a concatenation operator along the channel dimension. Since y^j has different receptive fields when j changes, multiscale features are derived after (3). To fuse these features, a concatenation operator and a convolutional layer are used in sequence. Thus, the final feature after the MCM is

$$\mathbf{MT}(F^{i-1}) = \mathbf{Conv}(\mathbf{Cat}(y^1, y^2, y^3, y^4)). \tag{4}$$

It is worth noting that for each convolutional operation in the MCM, a padding operator is followed to ensure the size of outputs not changed.

C. DEM

In order to combine multiscale features from different layers, skip-layer connections are adopted across the encoder to the decoder. Traditional methods often directly fuse them together via concatenation or summation operators. However, such operators will contain redundant information because the features in

TABLE I
DETAILED CONFIGURATION OF OUR PROPOSED TBCNN

Branches	Layer name	Operator	Stride	Output size
Encoder	MCM0	$[\text{Conv}3\times3]\times5$	$S=1$	$168\times200\times32$
	Maxpool0	$[2\times2]$	$S=2$	$84\times100\times32$
	MCM1	$[\text{Conv}3\times3]\times5$	$S=1$	$84\times100\times32$
	Maxpool1	$[2\times2]$	$S=2$	$42\times50\times32$
	MCM2	$[\text{Conv}3\times3]\times5$	$S=1$	$42\times50\times32$
	Maxpool2	$[2\times2]$	$S=2$	$21\times25\times32$
	Conv block	$[\text{Conv}3\times3]\times2$	$S=1$	$21\times25\times32$
Decoder (eddy identification branch)	DEM0	$\begin{bmatrix}\text{DeConv}3\times3\\ \text{Conv}3\times3\end{bmatrix}$	$\begin{bmatrix}S=2\\ S=1\end{bmatrix}$	$42\times50\times64$
	Conv block0	$[\text{Conv}3\times3]\times2$	$S=1$	$42\times50\times32$
	DEM1	$\begin{bmatrix}\text{DeConv}3\times3\\ \text{Conv}3\times3\end{bmatrix}$	$\begin{bmatrix}S=2\\ S=1\end{bmatrix}$	$84\times100\times64$
	Conv block1	$[\text{Conv}3\times3]\times2$	$S=1$	$84\times100\times32$
	DEM2	$\begin{bmatrix}\text{DeConv}3\times3\\ \text{Conv}3\times3\end{bmatrix}$	$\begin{bmatrix}S=2\\ S=1\end{bmatrix}$	$168\times200\times64$
	Conv block2	$[\text{Conv}3\times3]\times2$	$S=1$	$168\times200\times32$
Decoder (EEB)	Deconv0	$[\text{Deconv}3\times3]$	$S=2$	$42\times50\times32$
	Conv block0	$[\text{Conv}3\times3]\times2$	$S=1$	$42\times50\times32$
	Deconv1	$[\text{Deconv}3\times3]$	$S=2$	$84\times100\times32$
	Conv block1	$[\text{Conv}3\times3]\times2$	$S=1$	$84\times100\times32$
	Deconv2	$[\text{Deconv}3\times3]$	$S=2$	$168\times200\times32$
	Conv block2	$[\text{Conv}3\times3]\times2$	$S=1$	$168\times200\times32$

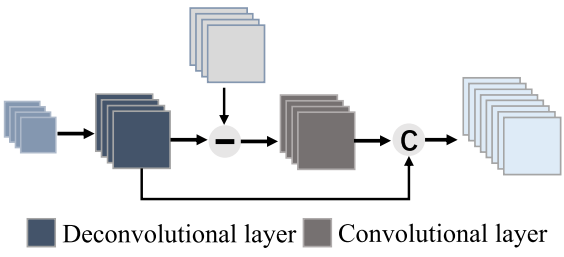

Fig. 3. Detailed structure of the DEM.

the decoder are progressively extracted from the corresponding shallow ones in the encoder. To deal with this issue, we propose a DEM, shown in Fig. 3. Take ith module as an instance, a deconvolutional layer is first designed to extract a high-resolution feature $\tilde{D}^i$ from the input, which can be described as

$$\tilde{D}^i = \mathbf{DeConv}\left(\mathbf{Conv}\left(D^{i-1}\right)\right) \tag{5}$$

where $\mathbf{DeConv}$ denotes a deconvolutional operator. Then, a subtraction operator between $\tilde{D}^i$ and F^{4-i} in the encoder is employed to obtain a complementary feature. Such feature is enhanced via a convolutional layer. Finally, a concatenation operator is used to get the output feature D^i. The computation procedure of the DEM can be formulated as

$$D^i = \mathbf{Cat}\left(\tilde{D}^i, \mathbf{Conv}\left(\tilde{D}^i - F^{4-i}\right)\right). \tag{6}$$

D. Loss Function

There are two branches in the proposed TBCNN. They are responsible for different tasks. In order to train them effectively,

we design two loss functions. One is $\mathcal{L}_{\text{eddy}}$ for the eddy identification branch, and the other one is $\mathcal{L}_{\text{edge}}$ for the EEB. Since the entire network is trained via an end-to-end way, the final loss function $\mathcal{L}$ needs to combine these two loss functions together:

$$\mathcal{L} = \mathcal{L}_{\text{eddy}} + \lambda \mathcal{L}_{\text{edge}} \tag{7}$$

where λ is a hyperparameter to tradeoff the effects of different losses. Assume Y_{eddy} and $\tilde{Y}_{\text{eddy}}$ are the groundtruth eddies and the predicted ones from the upper branch, we adopt the widely used dice loss [37] to build the loss function, which is described as

$$\mathcal{L}_{\text{eddy}} = \frac{2|Y_{\text{eddy}} \bigcap \tilde{Y}_{\text{eddy}}|}{|Y_{\text{eddy}}| + |\tilde{Y}_{\text{eddy}}|} \tag{8}$$

where $Y_{\text{eddy}} \bigcap \tilde{Y}_{\text{eddy}}$ computes the intersection between Y_{eddy} and $\tilde{Y}_{\text{eddy}}$, and $|\cdot|$ represents the sum of elements in the matrix. The value of $\mathcal{L}_{\text{eddy}}$ is between 0 and 1, and the larger value corresponds to the greater correlation. Similarly, we can derive the $\mathcal{L}_{\text{edge}}$. Note that during the computation of $\mathcal{L}_{\text{edge}}$, the edge groundtruth for eddies is derived from Y_{eddy} via a widely used canny edge detection algorithm.

Considering that there often exist different categories of eddies (e.g., cyclonic eddies and anticyclonic eddies) to identify and the available eddy numbers change a lot between them, we further design a class weighted loss function across different categories to alleviate this issue. Therefore, $\mathcal{L}_{\text{eddy}}$ can be replaced by

$$\mathcal{L}_{\text{eddy}} = \sum_i \gamma_i \mathcal{L}^i_{\text{eddy}} \tag{9}$$

where $\mathcal{L}^i_{\text{eddy}}$ and γ_i represent the dice loss and the category weight for the ith category, respectively. The weight γ_i is expressed as

$$\gamma_i = \frac{1/|Y^i_{\text{eddy}}|}{\sum_i 1/|Y^i_{\text{eddy}}|} \tag{10}$$

where Y^i_{eddy} is the groundtruth for the ith category eddies. Similar to $\mathcal{L}_{\text{eddy}}$, we can also design a weighted loss function for $\mathcal{L}_{\text{edge}}$ because the numbers of edge and nonedge are distributed unbalanced.

III. Experiment

A. Data Introduction

In order to evaluate the identification performance of TBCNN, we perform experiments upon a public eddy identification dataset named SCSE-Eddy [34]. As demonstrated in the red box of Fig. 4, SCSE-Eddy covers the entire South China Sea as well as its eastern sea areas (SCSE), ranging from 105.5° to 150° in east longitude and 4° to 30° in north latitude [34]. The whole dataset spans 17 years, i.e., from September 23, 2003, to September 23, 2018, with a total of 5480 pairs of images. Each pair consists of SSH data and the corresponding eddy data. Fig. 5 shows three examples of pairs of images, where the yellow and blue colors correspond to anticyclonic and cyclonic eddies, respectively. All the pairs are separated into two different sets,

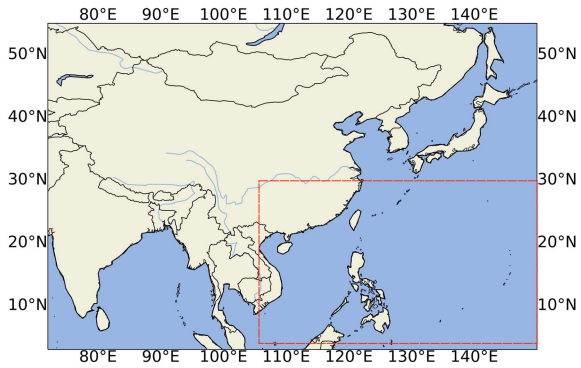

Fig. 4. Spatial extent (i.e., the red box) covered by the SCSE-Eddy dataset.

including a training set and a testing set. The former one is comprised of 4750 image pairs acquired in the first 15 years, while the latter one is made up of 730 image pairs collected in the last two years. To better train the proposed model, we choose 20% data (i.e., 950 image pairs) from the training set as a validation set to avoid overfitting. As exhibited in Fig. 5, most of the eddies are concentrated on the side away from the coastline. To improve the training speed and be consistent with the work in [34], we select the region away from the coastline to conduct experiments. The spatial resolution and size of each image are 0.25° and 168×200 pixels, respectively.

B. Evaluation Metrics

In the experiments, we choose three widely employed metrics [38], namely precision, recall, and F1-score, to quantitatively test the identification performance achieved by TBCNN, which are computed as

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

where TP and TN denote the numbers of correctly identified eddies and background samples, respectively, FP and FN describe the numbers of misidentified eddies and background samples, respectively, Precision indicates the proportion of correctly identified ocean eddies among all identified eddies, Recall represents the proportion of correctly identified eddies numbers to the total numbers of eddies in the groundtruth, F1-score considers both the Precision and Recall of the model, and is used to indicate the comprehensive evaluation value of the eddy identification model.

C. Experimental Setup

In addition to our TBCNN model, we also conduct experiments on some state-of-the-art deep learning models for eddy identification to compare their performance. They are Eddynet,

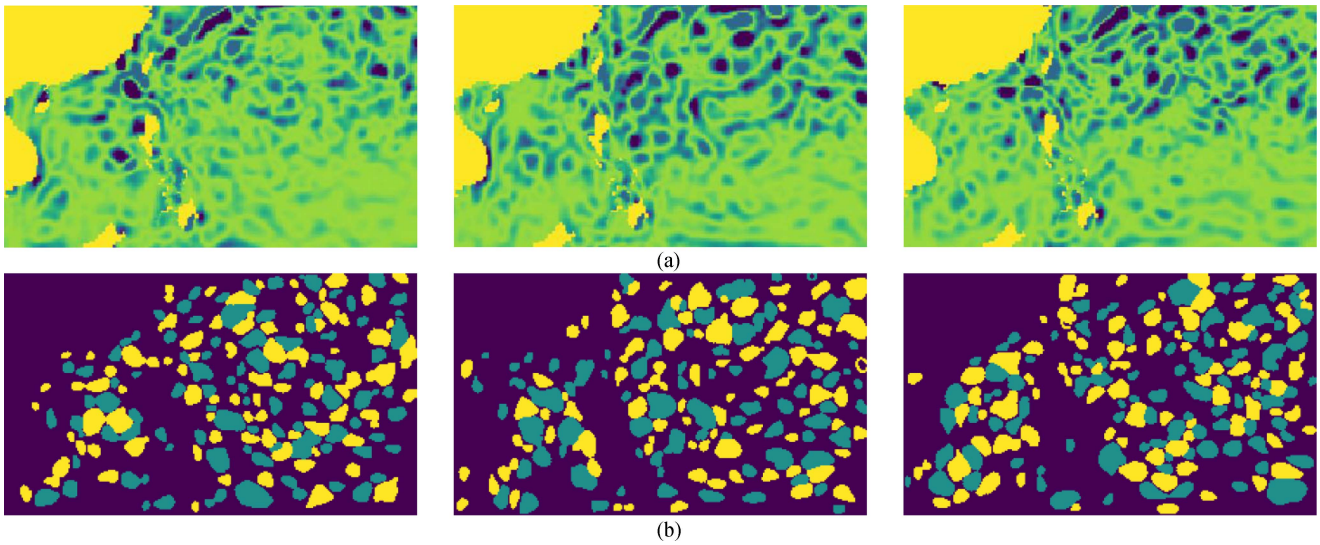

Fig. 5. Visualizations of some examples in the SCSE-Eddy dataset. (a) Input SSH data. (b) Groundtruth data.

FCN-8s, PSPnet, Segnet, and Deeplabv3+. In the following, we provide some descriptions about these comparative models.

1) *Eddynet:* The same as the work in [27], Eddynet adopts the encoder–decoder structure like UNet. The encoder consists of six convolutional layers as well as three max-pooling layers, and the decoder contains three deconvolutional layers and six convolutional layers.
2) *FCN-8s:* As in the work of [39], FCN uses VGG16 [40] as the backbone, and selects bilinear interpolation to recover the output to their original size.
3) *PSPnet:* The same as the work in [31], we adopt ResNet50 as the backbone of PSPnet, which uses a pyramid pooling structure to fuse the multiscale features of different layers.
4) *Segnet:* As in the work of [41], the encoding module of Segnet uses the first 13 convolutional layers of VGG16. Each encoding layer corresponds to a decoding layer.
5) *Deeplabv3+:* The same as the work in [34], Xception [42] is used as a backbone and Deeplabv3 is adopted as an encoder to extract features, which are then fused with the corresponding features in the backbone, and finally upsampled again to restore the original size.

For implementing the aforementioned identification models, we adopt the widely used Keras framework. The optimizer for them is Adam with default parameters. They are deployed on an NVIDIA TITAN XP GPU. The batch size is chosen as 8. The training inputs have a size of 168×200. The basic learning rate is set to 0.01, and the model is trained by 300 iterations.

D. Analysis on TBCNN

1) *Effects of hyperparameters λ:* There is a regularization parameter in (7). To validate its effects on the identification performance, we predefine a candidate set $\{10^{-2}, 10^{-1}, 0, 0.5, 1, 10, 100\}$, and choose values from it to conduct experiments. Fig. 6 shows the identification performance of TBCNN using different λ values. It can be observed that when $\lambda = 0.01$, the highest precision value is obtained.

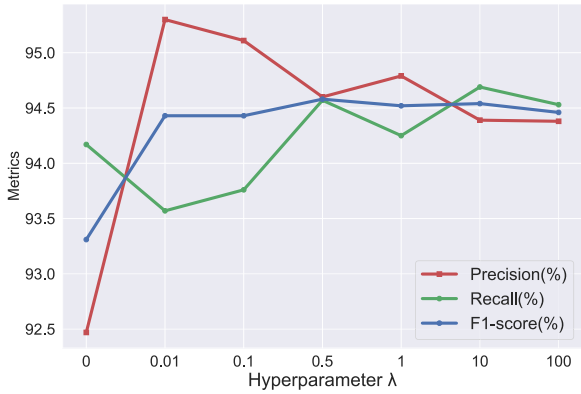

Fig. 6. Eddy identification performance with different λ values.

However, the corresponding recall value is worse. Similar phenomenon occurs for the recall value. The best recall value appears when $\lambda = 10$, but the precision value is not good. In comparison with precision and recall, F1-score is a better indicator to balance the precision and recall values. Since the highest F1-score is acquired when $\lambda = 0.5$, it is reasonable to choose this value for the following experiments.

2) *Effects of the MCMs:* To explore the effects of numbers of MCMs, we conduct four experiments by modifying the module numbers from 1 to 4. As shown in Fig. 7, the F1-score varies when the module number changes. Specifically, when the number changes from 1 to 4, the F1-score first increases and then decreases. The highest value is obtained when the module number is 3. Therefore, three MCMs are selected to extract multiscale features in the encoder of TBCNN.

3) *Effects of different components:* Compared to the existing encoder–decoder identification framework, our TBCNN has three new components. They are the EEB, the DEM, and the MCM. In order to test their effects on the identification performance, we conduct ablation studies. Table II demonstrates the detailed identification performance with

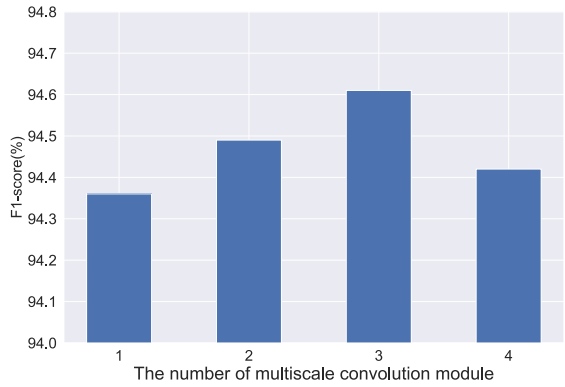

Fig. 7. Eddy identification performance with different numbers of MCMs.

TABLE II
EFFECTS OF DIFFERENT COMPONENTS ON THE EDDY IDENTIFICATION PERFORMANCE

EEB	DEM	MCM	F1-score
–	–	–	0.9331
√	–	–	0.9383
–	√	–	0.9370
–	–	√	0.9353
√	√	–	0.9441
√	–	√	0.9405
–	√	√	0.9418
√	√	√	0.9458

different components. In this table, "√" and "–" represent with and without the corresponding component, respectively. For the first row, a baseline network without any components achieves 93.31% F1-score. By adding the EEB, we get 93.83% F1-score with an improvement of 0.52%. By adding the DEM, we get 93.70% F1-score with an improvement of 0.39%. Additionally, adding the MCM improves the F1-score with 0.22%. More interestingly, the F1-score obtained by the combination of any two components is greater than 94%. The best combination is EEB and DEM, whose F1-score is 94.41%. Finally, we fuse the three components together, and the F1-score is further improved to 94.58%. All these improvements show the effectiveness of each component in our proposed TBCNN for eddy identification.

E. Performance Comparisons

Table III shows the detailed identification performance achieved by different models with three indicators. For each row, the bold font indicates the best value. From this table, we can observe several conclusions. First of all, the traditional FCN-8s model is capable of achieving satisfactory results, whose precision, recall, and F1-score are higher than 0.85. It indicates the huge potential of deep learning models, especially FCN-based models, to modeling the complex relationships between SSH and eddies. Second, although FCN-8s already sets up a strong baseline, Eddynet, PSPnet, and Deeplabv3+ can still outperform it due to the exploit of multiscale features. To be specific, almost all the identification indicators are above 0.9. Among these three models, Eddynet obtains the best performance, whose precision, recall, and F1-score are 0.9247, 0.9417, and 0.9331, respectively. Nevertheless, in comparison with Eddynet, our proposed TBCNN further improves the performance in all three indicators. In particular, the precision, recall, and F1-score are increased to 0.9460, 0.9457, and 0.9458, respectively. Such results are capable of certifying the superiority of our proposed identification model.

Besides the quantitative results in Table III, Figs. 8 and 9 also demonstrate visualization results on some instances. In Fig. 8, it is obvious to see that FCN-8s often oversegments eddies, making neighboring eddies easily connected together. On the contrary, Segnet generates undersegmentation results, which segment the complete eddy into multiple pieces. Therefore, its identification performance is relatively low, shown in Table III. For the other models, most of eddies are identified correctly and our proposed TBCNN achieves the most consistent results in comparison with the groundtruth. In Fig. 9, we crop some patches from the images to show the identification results in two typical situations: the sparsely distributed case in the first two columns and the densely distributed case in the last two columns. Overall, our proposed TBCNN model identifies almost all eddies in these situations, including the smallest and largest eddies, while the other models often lose or wrongly identify some eddies. Despite the good performance of our TBCNN, there also exist some drawbacks. For example, in the first column, TBCNN identifies an eddy in the bottom-right corner, which does not exist in the groundtruth data.

The radius, which reflects the critical characteristic of oceanic eddies, can also be considered as an indicator of identification performance. Fig. 10 demonstrates the radius distributions of identified eddies by different models. When compared to the groundtruth in this figure, it can be observed that the eddy radius is unevenly distributed, and most of eddies have radius of less than 100 km. For the identification models, FCN-8s works bad when the radius is less than 75 km or larger than 200 km; Segnet does not perform good in most situations, especially when the radius is less than 50 km. The other four models perform relatively well, and our proposed TBCNN achieves the closest number of eddies to the groundtruth in all kinds of eddy radius, which further validates the effectiveness of it.

F. Computational Time

In Table III, the computational time of different models for identifying eddies at both training and testing stages are recorded at the last two rows. Note that we do not use any accelerating strategy during the experiments. Unsurprisingly, the test time of each model is far less than their training time, which indicates that these models run quickly once trained. Owing to the relatively simple structure, FCN-8s takes a minimal amount of time for both training and testing stages, but it gets an F1-Score of only 87.57%. Besides, our proposed TBCNN takes the second least time. The training and testing time are 258.59 and 1.64 s, respectively. Although both values are larger than those of FCN-8s, the F1-score is improved significantly from 87.57% to 94.58%. Therefore, it is valuable to spend more time to train and test our proposed TBCNN.

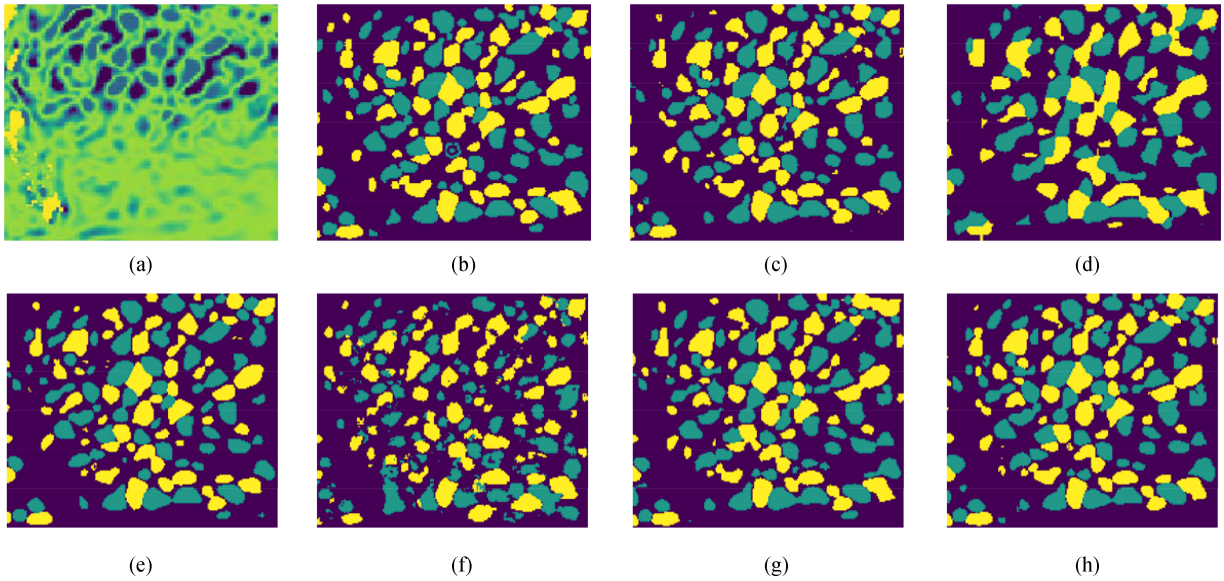

Fig. 8. Visualization of eddy identification results on an instance with different models. (a) SSH. (b) Groundtruth. (c) Eddynet. (d) FCN-8s. (e) PSPnet. (f) Segnet. (g) Deeplabv3+. (h) TBCNN.

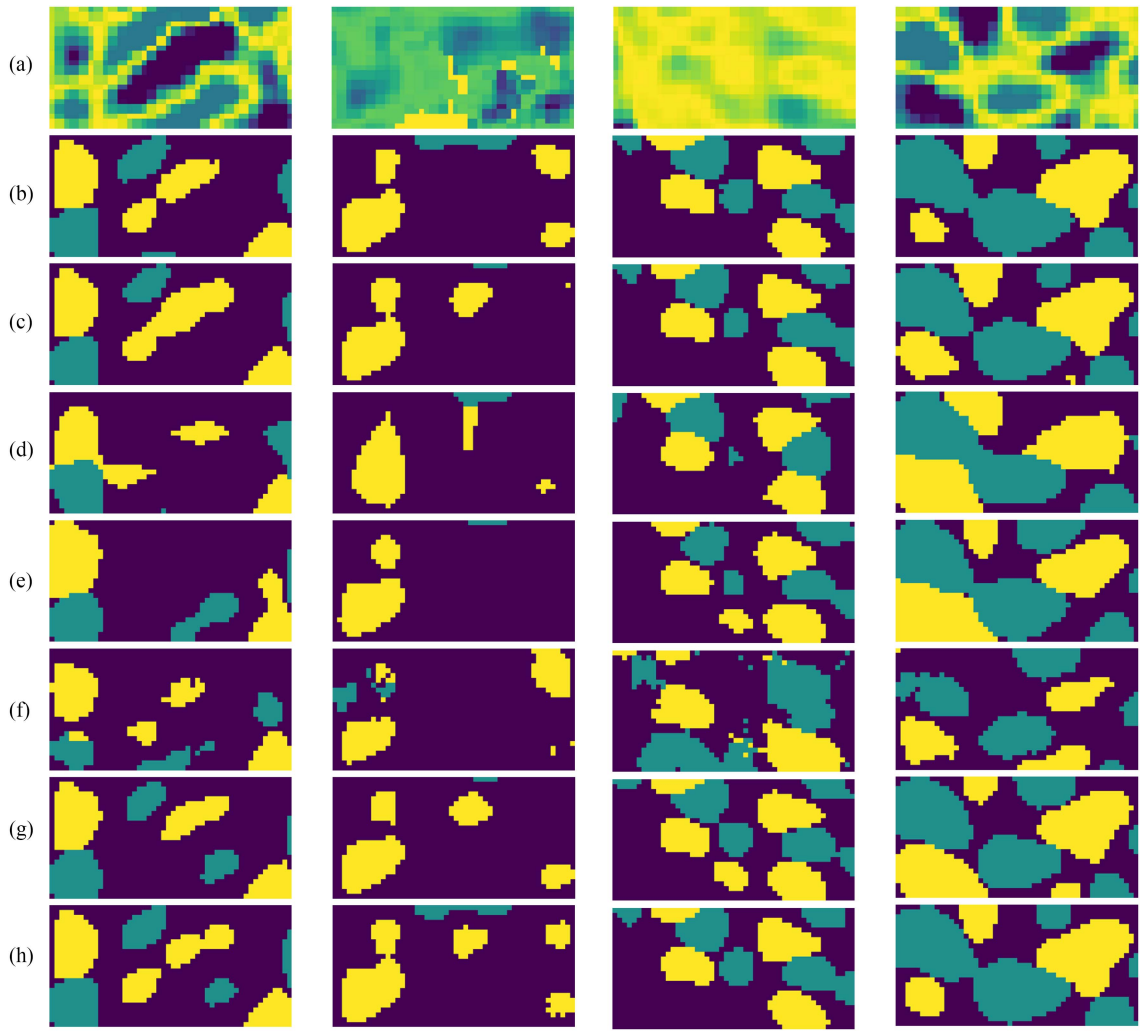

Fig. 9. Visualization of eddy identification results in different situations using different models. (a) SSH. (b) Groundtruth. (c) Eddynet. (d) FCN-8s. (e) PSPnet. (f) Segnet. (g) Deeplabv3+. (h) TBCNN.

TABLE III
EDDY IDENTIFICATION PERFORMANCE ACHIEVED BY DIFFERENT MODELS

Models	Eddynet	FCN-8s	PSPnet	Segnet	Deeplabv3+	TBCNN
Precision	0.9247	0.8659	0.8994	0.7910	0.9068	**0.9460**
Recall	0.9417	0.8857	0.9261	0.8463	0.9278	**0.9457**
F1-score	0.9331	0.8757	0.9126	0.8176	0.9171	**0.9458**
Training time(s)	568.32	**196.26**	284.73	392.79	547.51	258.59
Testing time(s)	2.19	**1.43**	2.02	43.87	2.23	1.64

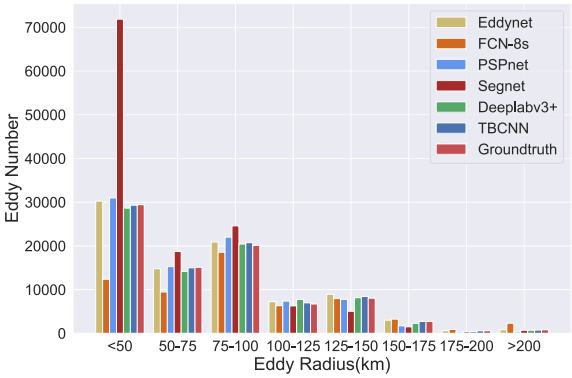

Fig. 10. Sample distribution of identified eddies obtained by different models.

IV. CONCLUSION

In this article, we proposed a TBCNN for oceanic eddy identification. Unlike the existing deep learning models, TBCNN decomposed the complex eddy identification process into an eddy identification branch and an EEB. For the former branch, we adopted single-layer and multiple-layer feature fusion methods simultaneously to identify different sizes of eddies. For the latter branch, it focused on enhancing the edge information of identified eddies, which was not fully explored in the former branch. To evaluate the identification performance of TBCNN, we constructed experiments on a publicly available dataset named SCSE-Eddy and compared it with several eddy identification models. Through quantitative and qualitative comparisons, the effectiveness of TBCNN was demonstrated. In addition, a comprehensive analysis of TBCNN was also provided. Specifically, the effects of different components in TBCNN, different hyperparameter values, and different numbers of MCMs were evaluated. However, based on the SSH data, we can only analyze eddies in the 2-D space. It is difficult to explore the characteristics of eddies under the sea surface. Therefore, in the future, we will collect data and design networks for 3-D eddy identification.

REFERENCES

[1] H. Wang, D. Liu, W. Zhang, J. Li, and B. Wang, "Characterizing the capability of mesoscale eddies to carry drifters in the northwest pacific," *J. Oceanol. Limnol.*, vol. 38, no. 6, pp. 1711–1728, 2020.
[2] L. Deng, Y. Wang, C. Chen, Y. Liu, F. Wang, and J. Liu, "A clustering-based approach to vortex extraction," *J. Visual.*, vol. 23, no. 3, pp. 459–474, 2020.
[3] S. Dandapat and A. Chakraborty, "Mesoscale eddies in the Western Bay of Bengal as observed from satellite altimetry in 1993–2014: Statistical characteristics, variability and three-dimensional properties," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 11, pp. 5044–5054, Nov. 2016.
[4] C. Dong et al., "The near-global ocean mesoscale eddy atmospheric-oceanic-biological interaction observational dataset," *Sci. Data*, vol. 9, no. 1, pp. 1–13, 2022.
[5] F. Nencioli, C. Dong, T. Dickey, L. Washburn, and J. C. McWilliams, "A vector geometry–based eddy detection algorithm and its application to a high-resolution numerical model product and high-frequency radar surface velocities in the Southern California Bight," *J. Atmos. Ocean. Technol.*, vol. 27, no. 3, pp. 564–579, 2010.
[6] A. M. Fernandes, "Study on the automatic recognition of oceanic eddies in satellite images by ellipse center detection–The Iberian coast case," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2478–2491, Aug. 2009.
[7] P. Gaube, D. J. McGillicuddy Jr, D. B. Chelton, M. J. Behrenfeld, and P. G. Strutton, "Regional variations in the influence of mesoscale eddies on near-surface chlorophyll," *J. Geophys. Res.: Oceans*, vol. 119, no. 12, pp. 8195–8220, 2014.
[8] B. Huang, L. Ge, X. Chen, and G. Chen, "Vertical structure-based classification of oceanic eddy using 3-D convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4203614.
[9] P.-C. Hsu, C.-C. Lin, S.-J. Huang, and C.-R. Ho, "Effects of cold eddy on Kuroshio meander and its surface properties, east of Taiwan," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 11, pp. 5055–5063, Nov. 2016.
[10] D. J. McGillicuddy Jr, "Mechanisms of physical-biological-biogeochemical interaction at the oceanic mesoscale," *Annu. Rev. Mar. Sci.*, vol. 8, pp. 125–159, 2016.
[11] J. Isern-Fontanet, E. García-Ladona, and J. Font, "Identification of marine eddies from altimetric maps," *J. Atmos. Ocean. Technol.*, vol. 20, no. 5, pp. 772–778, 2003.
[12] D. B. Chelton, M. G. Schlax, R. M. Samelson, and R. A. de Szoeke, "Global observations of large oceanic eddies," *Geophys. Res. Lett.*, vol. 34, no. 15, 2007, Art. no. L15606.
[13] J. C. R. Hunt, "Vorticity and vortex dynamics in complex turbulent flows," *Trans. Can. Soc. Mech. Eng.*, vol. 11, no. 1, pp. 21–35, 1987.
[14] C. Liu, Y. Wang, Y. Yang, and Z. Duan, "New omega vortex identification method," *Sci. China Phys., Mechanics Astron.*, vol. 59, no. 8, 2016, Art. no. 684711.
[15] M. S. Chong, A. E. Perry, and B. J. Cantwell, "A general classification of three-dimensional flow fields," *Phys. Fluids A: Fluid Dyn.*, vol. 2, no. 5, pp. 765–777, 1990.
[16] J. Jeong and F. Hussain, "On the identification of a vortex," *J. Fluid Mechanics*, vol. 285, pp. 69–94, 1995.
[17] A. Siegel and J. B. Weiss, "A wavelet-packet census algorithm for calculating vortex statistics," *Phys. Fluids*, vol. 9, no. 7, pp. 1988–1999, 1997.
[18] A. M. Doglioli, B. Blanke, S. Speich, and G. Lapeyre, "Tracking coherent structures in a regional ocean model with wavelet analysis: Application to Cape Basin eddies," *J. Geophys. Res.: Oceans*, vol. 112, 2007, Art. no C05043.
[19] A. Chaigneau, A. Gizolme, and C. Grados, "Mesoscale eddies off Peru in altimeter records: Identification algorithms and eddy spatio-temporal patterns," *Prog. Oceanogr.*, vol. 79, no. 2–4, pp. 106–119, 2008.
[20] C. Dong et al., "An oceanic cyclonic eddy on the lee side of Lanai Island, Hawai'i," *J. Geophys. Res.: Oceans*, vol. 114, 2009, Art. no. C10008.
[21] J. Yi, Y. Du, Z. He, and C. Zhou, "Enhancing the accuracy of automatic eddy detection and the capability of recognizing the multi-core structures from maps of sea level anomaly," *Ocean Sci.*, vol. 10, no. 1, pp. 39–48, 2014.
[22] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.
[23] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1424–1436, Feb. 2020.

[24] C. Li, R. Hang, and B. Rasti, "EMFNet: Enhanced multisource fusion network for land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4381–4389, 2021.
[25] K. Franz, R. Roscher, A. Milioto, S. Wenzel, and J. Kusche, "Ocean eddy identification and tracking using neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 6887–6890.
[26] F. Liu, H. Zhou, W. Huang, Y. Tian, and B. Wen, "Cross-domain submesoscale eddy detection neural network for HF radar," *Remote Sens.*, vol. 13, no. 13, 2021, Art. no. 2441.
[27] R. Lguensat, M. Sun, R. Fablet, P. Tandeo, E. Mason, and G. Chen, "Eddynet: A deep neural network for pixel-wise classification of oceanic eddies," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 1764–1767.
[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.
[29] Z. Fan and G. Zhong, "SymmetricNet: A mesoscale eddy detection method based on multivariate fusion data," 2019, *arXiv:1909.13411*.
[30] O. J. Santana, D. Hernández-Sosa, and R. N. Smith, "Oceanic mesoscale eddy detection and convolutional neural network complexity," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 113, 2022, Art. no. 102973.
[31] G. Xu et al., "Oceanic eddy identification using an AI scheme," *Remote Sens.*, vol. 11 no. 11, 2019, Art. no. 1349.
[32] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.
[33] F. Liu, H. Zhou, and B. Wen, "DEDNet: Offshore eddy detection and location with HF radar by deep learning," *Sensors*, vol. 21, no. 1, 2020, Art. no. 126.
[34] X. Sun, M. Zhang, J. Dong, R. Lguensat, Y. Yang, and X. Lu, "A deep framework for eddy detection and tracking from satellite sea surface height data," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7224–7234, Sep. 2020.
[35] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.
[36] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.
[37] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.
[38] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *Proc. Eur. Conf. Inf. Retrieval*, 2005, pp. 345–359.
[39] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[41] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SEGNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[42] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

Renlong Hang (Member, IEEE) received the Ph.D. degree in meteorological information technology from the Nanjing University of Information Science and Technology, Nanjing, China, in 2017.

From 2018 to 2019, he was a Postdoctoral Researcher with the Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City. He is currently an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology. He has authored or coauthored more than 30 peer-reviewed articles in international journals, such as IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. His research interests include machine learning, pattern recognition, and their applications to remote sensing image processing.

Gang Li received the B.S. degree in electrical engineering and automation from Yancheng Normal University, Yancheng, China, in 2019. He is currently working toward the M.S. degree in control engineering with the School of Automation, Nanjing University of Information Science and Technology, Nanjing, China.

His research interests include deep learning and remote sensing image processing.

Mei Xue received the Ph.D. degree in information and communication engineering from the Nanjing University of Information Science and Technology, Nanjing, China, in 2021.

She is currently a Lecturer with the School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing. Her research interests include deep learning and meteorological big data.

Changming Dong received the Ph.D. degree in physical oceanography from Columbia University, New York, NY, USA, in 2002.

From 2002 to 2004, he was a Postdoc with Princeton University and University of California, Los Angeles (UCLA). Since 2004, Prof. Dong was a research professor working at UCLA. In 2014, he joined the Faculty, Nanjing University of Information Science and Technology (NUIST), Nanjing, China, as a Professor. He is currently a Professor with the School of Marine Sciences, NUIST. He has authored or coauthored more than 160 peer-reviewed papers and six academic books. His research interests include physical oceanography, AI oceanography, and laboratory geophysical fluid dynamics.

Jianfen Wei received the Ph.D. degree in climate system and climate change from the Nanjing University of Information Science and Technology, Nanjing, China, in 2018.

She is currently a Lecturer with the School of Atmospheric Science and Remote Sensing, Wuxi University, Wuxi, China. Her research interests include atmosphere–ice–ocean interactions in polar regions and AI-based sea ice prediction.