Research on Water Surface Environment Perception Method Based on Visual and Positional Information Fusion

Qin Na, Zhe Zuo, Ning Xu, ZhenYu Zhang(✉), and Yi Lu

Beijing Institute of Technology, Beijing 100081, People's Republic of China
zuzeus@bit.edu.cn

Abstract. The water surface environment characterised by complexity and variability, is heavily influenced by weather. To address this problem, this paper proposes a water surface environment perception network based on the fusion of visual and positional information, and proposes an encoder-decoder based semantic segmentation neural network for classifying the pixel points of the input image into three categories: water, sky and environment (obstacles).

Keywords: Semantic segmentation · Positional information · Attentional mechanisms

1 Introduction

A good surface environment perception method is an important guarantee to help surface ships realize autonomous unmanned navigation in waters. It is difficult to understand the complex and variable features of different water surface objects on the basis of traditional methods, while research concerning deep learning methods lacks some practical and a priori knowledge in traditional methods.

To address this problem, this paper proposes a water surface environment sensing technique in accordance with the fusion of visual and positional information, which combines the advantages of both traditional methods and deep learning methods. In the model structure, residual network acts as an encoder to extract the information and features of different scale images. An attention mechanism and a feature fusion module are used in the decoder enabling the network to focus on locally focused information and feature fusion at different scales. The bit-position information is encoded into feature vectors of the neural network and fused with it, and the features of the encoder and decoder merge at different stages of the decoder. After that, the model designed in this paper is compared with the latest SOTA model in the field of semantic segmentation, in order to qualitatively and quantitatively analyze the advantages and disadvantages of different models, and to confirm the effectiveness and advantages of Swan-Net in the field of water surface environment sensing.

Compared with the existing studies, the platform position information obtained from the inertial measurement unit (IMU) is applied as the priori

B. Xin et al. (Eds.): IWACIII 2023, CCIS 1931, pp. 50–62, 2024.
https://doi.org/10.1007/978-981-99-7590-7_5

knowledge of the water boundary and encoded into feature channels to be fused with different scale image features. After fusing the position information, the network can effectively improve the accuracy of water boundary estimation in low-contrast environments.

2 Swan-Net

In this paper, a semantic segmentation model named Swan-Net is designed. The concept and local substructure of the design are mainly refer SOTA networks in the field of semantic segmentation in recent years. The network is also contains an a priori water boundary knowledge encoding of the positional information, which is obtained through an inertial measurement unit (IMU). The overall structure of the model is shown in Fig. 1.

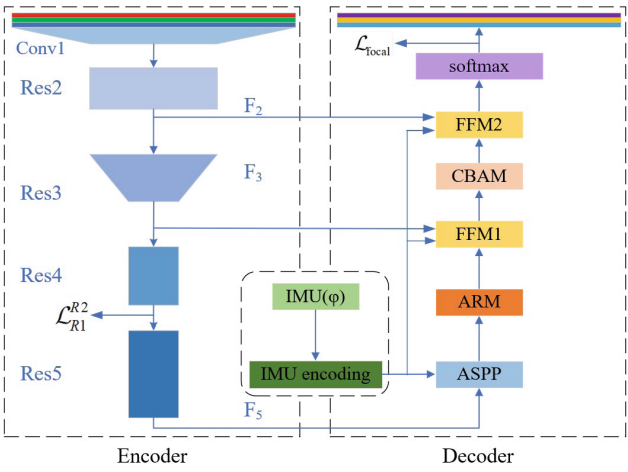

Fig. 1. General structure of the model

The general structure of the model is shown in the figure above, which consists mainly of an encoder and a decoder. The model accepts an input image of size $480 \times 640 \times 3$ (height $\times$ width $\times$ number of channels) and assigns each of these pixels a category label, water, sky or environment. The final output is of the same resolution as the input, avoiding the loss of detailed information as much as possible.

2.1 Feature Extraction Module

The main function of the encoder is to accept the input image and extract its features at different scales. The main composition is a ResNet101 [1] neural network. That is mainly composed of a pre-convolutional layer (Conv1), four residual convolutional blocks (Res2,Res3,Res4,Res5), unlike traditional ResNet101,

the dilated convolutions is used in Res4 and Res5. Through controlling the sampling step of the convolution kernel, dilated convolutions is achieved so as to increase the convolution kernel field of perception and reduce the number of parameters. It can also be realized by inserting zeros in the middle of the convolution kernel or by leaving the convolution kernel unchanged and sampling the input at equal intervals.

2.2 Position Information Feature Encoding

The IMU usually consists of a three-axis accelerometer and a three-axis gyroscope. By fusing and solving the accelerometer and gyroscope data, it can measure the current attitude information of the platform. The attitude information includes roll angle, pitch angle and yaw angle, referring to the angel of tilt relative to the horizontal plane, the angel of front-to-back tilt, and the angel of rotation of the axis perpendicular to the horizontal plane respectively. This positional information can be regarded as a priori knowledge of the semantic components in the image, since the position of the horizon in the image is usually related to the current attitude of the platform. When the roll angle changes, the tilt angle of the sea antenna also changes.

Suppose X^{usv}denotes the 3D coordinates of a point in the coordinate system of the amphibious platform, and let R_{cam}^{usv} denote the rotation matrix describing the rotation between the platform and the camera coordinate system. Point X_i^{usv} is projected to the image plane of the camera according to the following Equation:

$$\lambda_c x_i = K R_{cam}^{usv} X_i^{usv} \tag{1}$$

K is the camera calibration matrix, which is estimated during the calibration process. In this method, the points X_i^{usv} constituting the sea antenna are obtained from the IMU measurements. It is assumed that R_{cam}^{usv} denotes the rotation matrix of the IMU relating to the platform, while R_{imu} denotes the rotation matrix of the IMU with respect to the water surface. A reasonable assumption can be made that the Z-axis of the IMU and the camera are approximately aligned. In principle, these geometric relationships are sufficient to compute the vanishing point and can be used directly to estimate the sea antenna.

However, it has been shown that projecting vanishing points into the input image leads to inaccuracies, due to the fact that the vanishing points are likely to be projected outside the image boundaries and the calibrated radial aberration model can reliably estimate the aberrations of points located only inside the image. The sea antenna can therefore be obtained by projecting two points, $\left\{X_1^{imu}, X_2^{imu}\right\}$being two points in the XZ plane of the IMU coordinate system that are located at a horizontal angle $\pm\alpha_h$ and at a finite distance Z=l_{dist}. These points are rotated into a plane parallel to the water surface by following Equation:

$$X_i^{usv} = R_{imu}\left(R_{usv}^{imu}\right)^{-1} X_i^{imu} \tag{2}$$

X_i^{imu} and X_i^{usv} denote the points before and after the rotation, respectively. The rotated points $\{X_1^{imu}, X_2^{imu}\}$ are projected into the image using Eq. 1 while considering radial distortion. The sea antenna is estimated by fitting the projected radial distortion points to a line. Through the above method, the attitude angle information of the platform can be converted into a projection of the sea antenna in the image, in order to fuse the projection information into the neural network.

2.3 Feature Fusion Module

The task of the decoder is to fuse the image features extracted by the encoder module with the information from the IMU, and after feature refinement and upsampling, to produce the final semantic segmentation output. The decoder accepts features from the three modules in the encoder (Res2, Res3, Res5) as well as the encoded IMU feature channels, utilising both the more detailed information from the high resolution features and the global semantic information captured at lower resolutions. First, the output features from the last layer of the encoder are fed into a spatial pyramid pooling module (ASPP) with 4 dilated convolutions, and the output of the ASPP module passes through the attention refinement module, which reweights the features.

The feature fusion module in the decoder aims to integrate the F3 and F2 features from the encoder with the IMU information features to achieve effective fusion of the different path features and to take full advantage of the different features.

The CBAM module calculates both attention on the channel and attention on the space. The inclusion of this module will effectively help the decoder network to learn spatially focused features.

ARM Module. The attention refinement module ARM is derived from BiSeNet [3], a convolutional neural network module for image classification and semantic segmentation, and is primarily used to enhance the representational power of feature maps. It automatically learns relevant features in the input feature map and applies these features to enhance the representational power of the model. The ARM module enables the enhancement or weakening of the feature representation at different locations by performing a channel-by-channel weighted summation of the input feature map. By introducing the ARM module, the accuracy and robustness of the model can be significantly improved, especially when dealing with areas such as detail and edges in an image. The overall structure of the ARM module is shown in Fig. 2.

The tensor output from the ASPP module goes one way through the raw feature input channel without any processing. The other way goes through the attention vector computation channel, which first goes through a global averaging pooling layer. In global averaging pooling, for each channel feature map, the features of all pixels are averaged into a single value that represents the statistical features of the entire feature map. The feature maps are then subjected to batch normalisation [4] and Sigmoid units [5] to compute the attention vector.

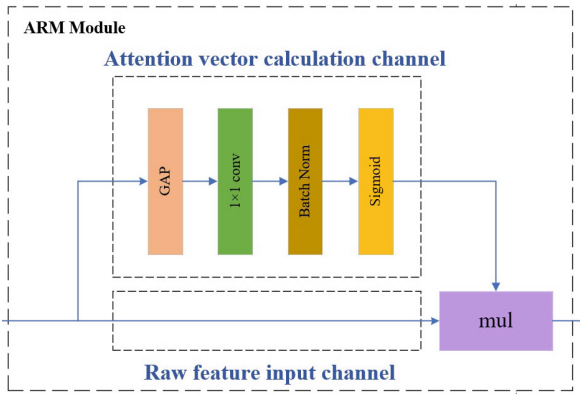

Fig. 2. ARM Module diagram

After computing the attention vector, it is multiplied with the original feature map on a channel-by-channel point-by-point basis. The original features will be re-weighted by the attention vector, enhancing the important features and weakening the less important ones, thus making the extracted features more directional.

FFM Module. The main role of the feature fusion module FFM is to fuse feature maps from multiple scales and levels to obtain richer and more complete image feature information. Typically, the FFM module consists of several branches, each of which uses different convolution kernels and pooling strategies to extract features at different scales and then integrate these features together. Feature maps at different scales and levels complment each other to obtain more comprehensive and accurate image information.

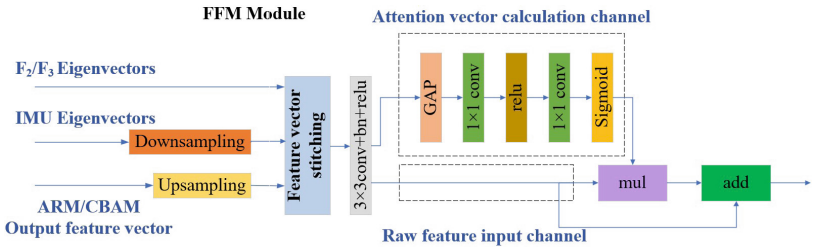

Fig. 3. Feature Fusion Module diagram

The FFM module used in this paper is shown in Fig. 3. First the FFM module stitches the output of the ARM module with the ASPP module, followed by further feature extraction using a 3×3 convolution, batch regularisation and the

ReLU activation function [6]. Thereafter a similar attention vector re-weighting method as in the ARM1 module was utilised to further extract and fuse features from different paths.

CBAM Module. The design of the CBAM module is derived from the literature [7]. The purpose of this module is mainly used to enhance the perceptual field of each location in the feature graph, thus advancing the performance of the network. The overall structure of CBAM is shown in Fig. 4. The CBAM module consists of two main parts: the channel attention module and the spatial attention module. The former is mainly used to perform attention computation in the channel dimension in order to learn the importance of different channels in the feature map adaptively, so as to better utilize the information of different channels. The latter is used to perform attention computation in the spatial dimension in order to adaptively learn the importance of different positions in the feature map for making use of the information of different positions in the feature map.

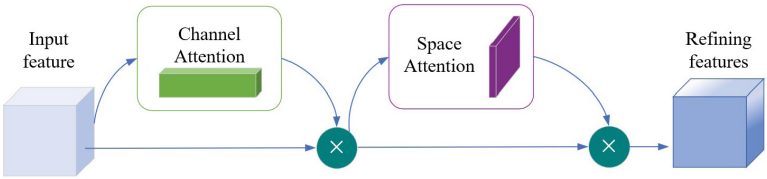

Fig. 4. CBAM Attention Module diagram

The results of the channel attention module are shown in Fig. 5. In the channel attention module, a one-dimensional vector is obtained by compressing the feature map in the spatial dimension. In the compression, both global average pooling and global maximum pooling are considered. The average pooling and maximum pooling can be used to aggregate the spatial information of the feature map, send to a shared network, compress the spatial dimension of the input feature map, sum and merge element by element then normalize the weights using the Sigmoid activation function, to obtain the channel attention map.

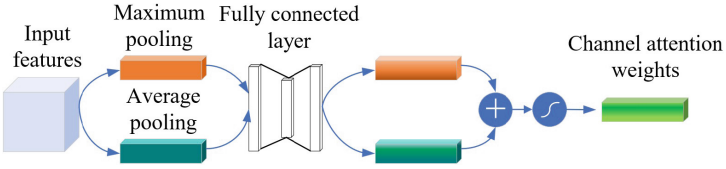

Fig. 5. CBAM Channel Attention

The channel attention computation process can be expressed by the following equation:

$$\begin{aligned} M_c(F) =& \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ =& \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \end{aligned} \tag{3}$$

where F denotes the input feature map, w_0 and w_1 represent the parameters of the two layers in the multilayer perceptron model, and $\sigma(\cdot)$denotes the Sigmoid activation function.

The results of the spatial attention module are shown in Fig. 6, differing from the channel attention module in the dimensionality of the processed features. The spatial attention module uses average pooling and maximum pooling in the channel dimension to put the two obtained H×W×1 feature descriptions stitched together according to the channels. Then, after a 7×7 convolutional layer is reduced to a single channel and a Sigmoid activation function is used to obtain the weight vector.

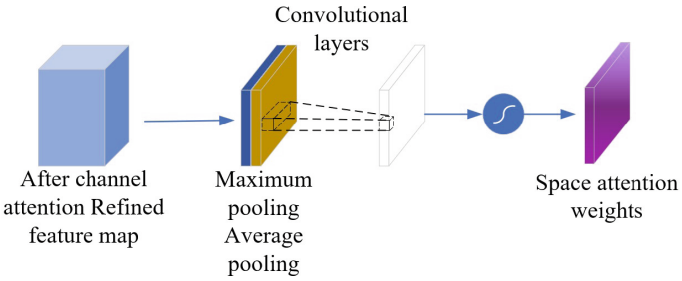

Fig. 6. CBAM Spatial Attention

The channel attention calculation process can be expressed by the following equation:

$$\begin{aligned} M_s(F) =& \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \\ =& \sigma(f^{7\times7}([F^s_{avg}; F^s_{max}])) \end{aligned} \tag{4}$$

2.4 Loss Function

The aquatic environment differs from the usual environmental dataset in that although some obstacles may be large, the majority of pixels in a typical aquatic scene belong to water or sky, which leads to an imbalance in the categories, and this imbalance makes the classical cross-entropy loss inapplicable [8]. Therefore, a weighted Focal Loss applicable to segmentation is used, calculated as follows:

$$L_{foc} = -\alpha_t(1 - p_t)^{\gamma} log(p_t) \tag{5}$$

where p_i denotes the prediction probability of the model for the sample, α_t denotes the category weights, and γ is a moderator. When $\gamma = 0$, Focal Loss

degenerates to the ordinary cross-entropy loss function. When $\gamma > 0$, Focal Loss decreases the loss contribution of easy-to-categorize samples and increases the weights of hard to categorize samples, thus making these samples receive more attention.

3 Experimental Methods and Analysis of Results

The experimental framework in this paper uses Pytorch 1.8, CUDA 11.4, TITAN XP graphics card, cosine annealing strategy for learning rate setting, adam optimizer [9] to update the model parameters during training, and 50 training rounds. The training data are input using a data normalization strategy that normalizes the pixel values of each channel of the input image to a mean of 0 and a variance of 1. The benefit of normalization is to ensure that the pixel values of all channels are in the same range of values, preventing a particular channel from having too much influence on the model training.

3.1 Training Dataset

The publicly available datasets used in this paper for training and testing are: MaSTr1325 [10], MODD2, SMD [11] and USV Inland [12], and a dataset MyDataset collected during the experiments in this paper. Images in the dataset are shown in Fig. 7.

The network designed in this paper, as well as all other networks used for comparison, is trained on the MaSTr1325 dataset, which contains 1325 unique images taken over a 24-month period. Three semantic components are manually annotated on a pixel-by-pixel basis: water, sky and environment (obstacle).

In this paper, data augmentation is used in the training. Two types of data enhancement are chosen to suit the water environment: horizontal mirroring and luminance transformations. An elastic distortion is also applied to the water component of the training image to artificially simulate waves and curls, increasing the diversity of local textures in the training set. The effect of data enhancement is shown in figure. The final result after applying data enhancement is a total of 48724 training images.

The performance of the model is evaluated in MyDataset, as well as two publicly available ocean datasets, MODD2 and SMD, and an inland unmanned vessel dataset, USV Inland. MODD2 is recorded in the Adriatic coastal area, consisting of 28 different time series, all collected by the camera and synchronized with IMU measurement times. It is recorded using an unmanned surface vessel. SMD dataset is recorded at different locations in the port of Singapore, which consists of 66 sequences containing the following. The USV Inland is more different from the previous dataset and is the first inland unmanned boat dataset with multiple sensors and weather conditions in real scenarios.

Fig. 7. Training dataset

3.2 Model Structure Ablation Experiment

In deep neural networks, ablation experiments are performed by removing certain component parts of the network and observing the change in the performance of the neural network to determine the effect of each substructure on the model performance. In the Swan network designed in this paper, the model that does not contain ASPP module, ARM module and FFM module is noted as Baseline. In Baseline the decoder only upsamples and splices the final output with the features from the encoder at different stages. The following table shows the impact of different structures on the model.

The Table 1 shows that the ARM, FFM, and ASPP structures of the model all positively influence the final F1 score, and that the first ASPP module has the greatest improvement in model performance, followed by the FFM and ARM modules.

3.3 Performance Comparison of Different Models

To confirm the effectiveness of the model designed in this paper, this section compares some previous research works and the accuracy metrics of the latest semantic segmentation models on the same dataset. A total of three current state-of-the-art segmentation networks RefineNet [12], BiSeNet [13] and U-Net [14] are compared, which achieve the best results in segmentation tasks in either the self-driving car domain or the medical domain, with different encoder and

Table 1. Model substructure ablation experiment

Method	F1-score
Baseline	73.3
Baseline+ASPP	78.7
Baseline+ASPP+ARM	83.0
Baseline+ASPP+ARM+FFM1	86.8
Baseline+ASPP+ARM+FFM1+CBAM	89.7
Baseline+ASPP+ARM+FFM1+CBAM +FFM2	93.3

decoder architectures. The Table 2 summarizes the number of different model parameters and inference times.

Table 2. Comparison of the number of parameters and inference time of different models

Model	N_{param}	δ_t (ms)	FPS
RefineNet	85.7M	130	7
U-Net	28.0M	45	22
BiSeNet	47.5M	68	15
Swan	66.5M	100	10

The Table 3 summarizes the results of all models tested on MODD2 and SMD. Swan greatly outperforms all competing networks in terms of water edge estimation. The second best is RefineNet, with an accuracy about two pixels lower, followed closely by BiSeNet.

Table 3. Test results of different models

Model	TP_{100} (times)	FP_{100} (times)	F1(%)
U-Net	39.9	17.5	69.2
BiSeNet	48.4	12.1	83.8
RefineNet	49.0	2.2	91.6
Swan	51.1	3.8	93.3

The traditional target detection task generally compares the prediction results of the model with the real annotation when evaluating the metrics, and if the intersection ratio of the output rectangular box to the real annotated box reaches a certain threshold and the classification is correct, it is considered as

TP. This approach is fine for detecting small obstacles (e.g., buoys, small boats) on the water surface. However, large vessels are usually integrated with the water boundary, and the model can segment the vessels but the output cannot identify this part of the vessels.

In this paper, we use the method proposed in the literature [15] in calculating the obstacle detection index. We determine whether the detection is correct by calculating the proportion of correctly classified pixel points in the labeled area, and when the proportion exceeds the set threshold of 20%, it is considered as a TP detection, otherwise it is considered as a FP detection.

In terms of TP and FP metrics, Swan receives the best recall score because of the highest true positive (TP) detection rate, followed by RefineNet and BiSeNet.

The Fig. 8 shows the comparison of prediction results of different networks. From the Fig. 8, it can be concluded that in the presence of mist on the sea antenna (first row of the figure), U-Net and BiSeNet are more inaccurate in estimating the water edge, which usually results in over- and underestimation. RefineNet is better, but the expected water is still underestimated, while Swan neither overestimates nor underestimates its position, and Swan has a significant improvement.

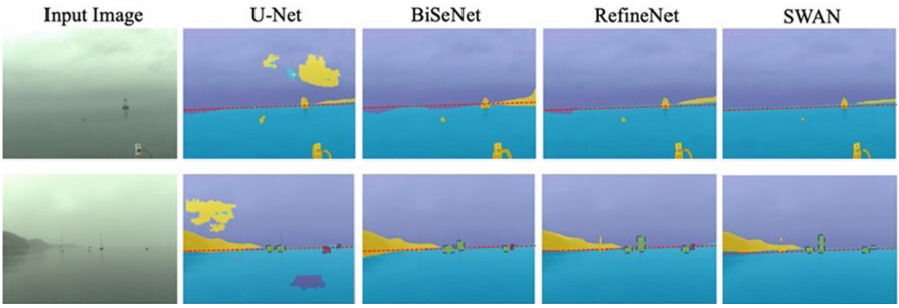

Fig. 8. Comparison of different model forecasts

In the presence of small objects in the distance of the image and in the absence of contrast (second row of the Fig. 8), Swan detects smaller obstacles more accurately than the other networks, and the other three networks showe missed detections, while Swan accurately detectes all the objects in the figure, And it has the most TP detections and achieves the best recall, followed by RefineNet, then BiSeNet and U-Net.

When evaluating the value of a model for engineering applications, the metric for detecting obstacles is usually an important part. A high Precision means that the model can detect the obstacles more accurately, while a high Recall means that the model is better able to detect more obstacles. In practice, in order to achieve uninterrupted autonomous navigation, it is necessary to maintain both a certain recall to ensure that all obstacles can be found as fully as possible and that the platform does not collide.

It is also necessary to maintain a certain level of accuracy to prevent the model from misreporting too many obstacles and affecting the normal operation and obstacle avoidance of the platform, so there is also a trade-off between the number of FP and TP detections, which is measured by the F1 score. The best performing methods based on F1 are Swan, RefineNet, BiSeNet and U-Net. The quality of segmentation masks can be further understood based on the accuracy of obstacle detection overlap thresholds, recall and F1 score plots. The plots of the four best performing networks are shown in Fig. 9.

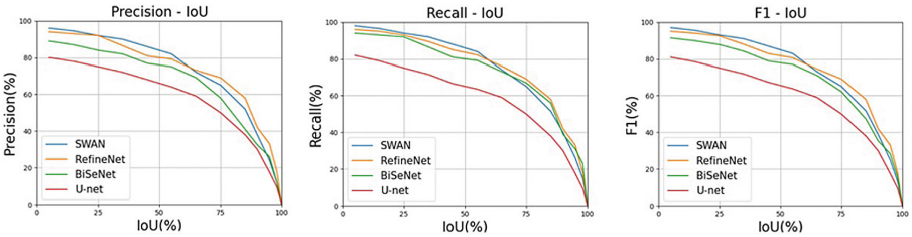

Fig. 9. Plot of model metrics and obstacle overlap threshold

The Swan network proposed in this paper has the highest accuracy and recall scores at medium overlap threshold, followed by RefineNet, which means that Swan detects more obstacles with fewer FPs. Further analysis shows that most of the TP detected by Swan but not detected by other networks are small objects with area less than 900 pixels, proving that Swan is more advantageous for detecting small targets.

However, RefineNet performs better when the overlap threshold is relatively high (above 70%), predicating that the localization of RefineNet is more accurate than that of Swan. However, RefineNet can show poor detection of isolated obstacles and miss some small obstacles, leading to relatively low TP rate, that is dangerous in real autonomous navigation. As the threshold rises above 65%, the curves of all models drop faster, indicating that these models still have shortcomings in localization. Thus accurate obstacle segmentation is very challenging for all models, and the models still have room for improvement upwards.

4 Conclusion

In this paper, we propose a semantic segmentation neural network, Swan-Net, which fuses the pose information to extract the environmental semantic information from the image, and classifies the pixels in the image into three categories: water, sky, and environment, that can be used to provide passable area information for surface ships and vessels. In this paper, we first integrate the positional information into a semantic segmentation neural network as an a priori water boundary approach, encode the platform positional information obtained from the inertial measurement unit (IMU) into a feature channel, and apply it to

the decoder to fuse with the image features. The decoder design makes full use of multi-scale feature fusion, channel attention, and spatial attention mechanisms, and the effectiveness of the model design is demonstrated by ablation experiments and comparison experiments. Its advantages in the field of water surface environment perception are illustrated through the comparison with the existing excellent networks in the field of semantic segmentation. The next step needs to improve the segmentation performance of the network on the inland river scenario, so more data collection and application of new model optimization methods needs to be considered to enhance the model performance in this scenario.

References

1. He, K., et al.: Deep Residual Learning for Image Recognition. IEEE (2016)
2. Bovcon, B., Perš, J., Kristan, M.: Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation. Robot. Auton. Syst. **104**, 1–13 (2018)
3. Yu, C., et al.: Bisenet: bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 325–341 (2018)
4. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. JMLR.org (2015)
5. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back propagating errors. Nature **323**(6088), 533–536 (1986)
6. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**(2) (2012)
7. Woo, S., et al.: CBAM: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
8. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**(2) (2012)
9. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. Comput. Sci. (2014)
10. Bovcon, B., et al.: The mastr1325 dataset for training deep USV obstacle detection models. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3431–3438. IEEE (2019)
11. Prasad, D.K., et al.: Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey. IEEE Trans. Intell. Transp. Syst. 1993–2016 (2017)
12. Cheng, Y., et al.: Are we ready for unmanned surface vehicles in inland waterways the USV inland multi-sensor dataset and benchmark. IEEE Robot. Automat. Lett. **99**, 1 (2021)
13. Lin, G., et al.: RefineNet: multi-path refinement networks for high-resolution semantic segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
14. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
15. Bovcon, B., Kristan, M.: WaSR-a water segmentation and refinement maritime obstacle detection network. IEEE Trans. Cybernet. **99** (2021)