Journal of *Marine Science and Engineering*

MDPI

Article

Improved UNet-Based Shoreline Detection Method in Real Time for Unmanned Surface Vehicle

Jiansen Zhao [1], Fengchuan Song [1,*], Guobao Gong [2] and Shengzheng Wang [1]

[1] Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China; jszhao@shmtu.edu.cn (J.Z.)
[2] Navigation College, Dalian Maritime University, Dalian 116026, China
* Correspondence: 202130110062@stu.shmtu.edu.cn

Abstract: Accurate and real-time monitoring of the shoreline through cameras is an invaluable guarantee for the safety of near-shore navigation and berthing of unmanned surface vehicles; existing shoreline detection methods cannot meet both these requirements. Therefore, we propose an improved shoreline detection method to detect shorelines accurately and in real time. We define shoreline detection as the combination of water surface area segmentation and edge detection, the key to which is segmentation. To detect shorelines accurately and in real time, we propose an improved U-Net for water segmentation. This network is based on U-Net, using ResNet-34 as the backbone to enhance the feature extraction capability, with a concise decoder integrated attention mechanism to improve the processing speed while ensuring the accuracy of water surface segmentation. We also introduce transfer learning to improve training efficiency and solve the problem of insufficient data. When obtaining the segmentation result, the Laplace edge detection algorithm is applied to detect the shoreline. Experiments show that our network achieves 97.05% MIoU and 40 FPS with the fewest parameters, which is better than mainstream segmentation networks, and also demonstrate that our shoreline detection method can effectively detect shorelines in real time in various environments.

Keywords: water surface segmentation; attention mechanism; edge detection; shoreline detection

Citation: Zhao, J.; Song, F.; Gong, G.; Wang, S. Improved UNet-Based Shoreline Detection Method in Real Time for Unmanned Surface Vehicle. *J. Mar. Sci. Eng.* **2023**, *11*, 1049. https://doi.org/10.3390/jmse11051049

Academic Editor: Sergei Chernyi

Received: 24 March 2023
Revised: 22 April 2023
Accepted: 9 May 2023
Published: 15 May 2023

1. Introduction

Unmanned surface vehicles (USVs) have dramatically developed in recent years thanks to technical advancements. These intelligent devices can be navigated by manual or programmed control to accomplish a variety of tasks. Real-time and accurate monitoring of the shoreline is important when using these autonomous surface vehicles, both for the safety of berthing and near-shore navigation.

On large vessels, several types of sensors are installed to monitor the surrounding environment, such as cameras [1,2] and radar [3,4]. These devices can provide various forms of environmental information to the ship, but considering the limitations of USVs themselves in terms of carrying capacity and energy supply, they cannot equip huge or a large number of sensors. As a result, visual sensors such as cameras that are lighter and more energy-efficient while still offering extensive environmental information are better-suited for USVs. Based on this analysis, a visual-based shoreline detection method is crucial for USVs.

Depending on their technical means, existing visual-based shoreline detection methods can be classified into traditional image-based methods and deep-learning-based methods. Traditional methods include local binary patterns combined with the gray-level co-occurrence matrix method [5], column-by-column logistic regression combined with the polynomial spline modeling method [6], the calculation of vertical gradients in gray space combined with the random sample consensus (RANSAC) algorithm fitting method [7], and the calculation of morphological gradients on HSV color space combined with the watershed algorithm and edge detection method [8]. The abovementioned methods are subject to

limitations in their use and are susceptible to water surface reflections, light changes, waves, and long processing times, making them unable to meet the need for accurate and real-time detection of shorelines. In recent years, artificial intelligence has been greatly developed and been widely used in the maritime fields, such as for ship detection [9,10], ship trajectory analysis [11] and prediction [12], marine accident risk quantification [13], and environmental perception [14]. As for shoreline detection, some researchers have attempted to introduce semantic segmentation techniques into this field [15] by first extracting the water surface area with a semantic segmentation network and using an edge detection algorithm on the obtained result to detect the shoreline. Based on this, some works have improved existing semantic segmentation models, making them more suitable for water surface area segmentation [16–18], and other studies have introduced pretrained methods [19] or the use of transfer learning [20] to solve the problem of an insufficient amount of data for training, as well as use self-supervised training approaches [21] to address the problem of insufficient labeled data. The trained models have strong robustness, which is good for solving the interference of environmental factors present in traditional image-based approaches but still cannot address the processing speed problem due to the use of a large network architecture, the large scale of the feature maps, etc.

To address the abovementioned problem, i.e., that existing visual-based shoreline detection methods cannot meet the requirement of shoreline detection both accurately and in real time, we constructed a better method to achieve real-time and accurate shoreline detection. We define shoreline detection as the combination of water surface segmentation and edge detection. According to our definition, the key to shoreline detection is water surface segmentation, which directly determines the accuracy and inference speed of our method. Edge detection has almost no impact on either of these factors. Therefore, to achieve the abovementioned target, we propose an improved U-Net network to perform water surface segmentation accurately and in real time. This network is based on U-Net combined with a residual network [22] and a squeeze-and-excitation (SE) attention module [23] to increase the segmentation accuracy and processing efficiency and named the Residual Squeeze-and-excitation U-Net (RS-UNet). According to experimental verification, the network proposed in this paper achieved a processing speed of 40 FPS and 97.05% MIoU, outperforming some mainstream methods of semantic segmentation, meeting the demand for real-time and accurate water surface area segmentation, and shoreline detection when combined with an edge detection algorithm. Other experiments also demonstrate the generalization capability of our method. Specifically, the contributions of this paper include:

- An encoder is built based on ResNet-34 to enhance the feature extraction capability of the network in complicated environments, with the introduction of transfer learning using pretrained ResNet-34 weights to improve the training efficiency and solve the problem of insufficient training data;
- To reduce the amount of computation, a lightweight decoder is built, and an attention mechanism is added to the decoder to force the network to pay more attention to the data in the critical part throughout the segmentation process, increasing the computational speed and maintaining the segmentation quality;
- Construction of a shoreline detection method based on the proposed RS-UNet, which can accurately detect the shoreline in real time and be applied in various environments.

The remainder of this paper is organized as follows. Section 2 presents a brief review of related works. Section 3 explains our proposed method. Section 4 shows the experimental results and analyses of these results. Section 5 provides a summary of our work and directions for future work. We also provide a list of abbreviations in Abbreviations for a better reading experience.

2. Related Work

2.1. Traditional Shoreline Detection Method

Traditional shoreline detection methods detect the shoreline through image processing; for example, Kristan et al. [24] proposed the imposition of weak structural constraints and

a Markov random field to account for the semantic structure of the marine environment in real time. At present, the accuracy of this method is relatively low. Kröhnert [6] proposed the use of column-by-column logistic regression and polynomial spline modeling to detect shorelines, which has a good detection effect for nearly straight shorelines but a relatively poor effect for more curved shorelines. Wei and Zhang [5] used local binary patterns and a gray-level co-occurrence matrix to calculate river texture information and used structure detection to eliminate the effects of wind and light, which is also a suitable method for more flat shorelines. Zhan et al. [7] calculated the vertical gradient on a gray image with the background texture removed, obtained the water shoreline candidate points from the vertical gradient of each column, and fitted the shoreline from the candidate points using the RANSAC algorithm. The processing speed of this method is faster, but it is susceptible to the influence of water reflection or texture. Feng et al. [8] computed morphological gradients in HSV color space to highlight edges, then used the watershed algorithm to segment the image area, combined with the use of a filtering operator to detect a river shoreline, which achieved real-time shoreline detection but was still subject to the influence of ambient lighting to some extent. Peng et al. [25] analyzed the differences in the characteristics of images in HSV color space under different lighting conditions. Different regions in the image were segmented, and the shoreline was detected based on the differences in saturation and brightness between land and water areas. This method is less stable and easily affected by environmental changes and lighting variations.

2.2. Deep-Learning-Based Shoreline Detection Method

The key to the deep-learning-based method is the segmentation of the water surface area. After obtaining the segmented result, the corresponding shoreline can be obtained using the edge detection algorithm, so it is essentially a semantic segmentation problem. For example, Steccanella et al. [15] used a fully convolutional neural network to detect the water surface area and obtained a high segmentation accuracy rate. However, this method could not meet the requirement of real-time processing. Steccanella et al. [16] further improved this method and achieved 98.8% pixel segmentation accuracy and 10 FPS on a 160 × 160 image. To address the problem of an insufficient amount of data for water surface segmentation training, Adam et al. [19] demonstrated that the use of a pretrained backbone can significantly improve the network's ability to segment water surface regions, and Vandaele et al. [20] proposed the use of a transfer learning approach that completes pretraining on the COCO dataset and is then fine-tuned on the water surface segmentation dataset. Zhan et al. [21] combined a semantic segmentation network with conditional random fields (CRFs) and superpixel mapping to propose an adaptive water surface segmentation network that effectively solves the training problem on datasets with limited labeled data. Shen et al. [18] used improved DeepLab v3+ to acquire water surface area segmentation results combined with an edge detection algorithm to detect the shoreline, which effectively overcomes the interference of factors such as reflection, although the processing speed of this method is 8 FPS, which is far from meeting the demand for real-time detection. Yao et al. [26] proposed ShrelineNet to detect shorelines for USVs, which segments the entire image into sky, land, and water sections. Then, the shoreline is detected based on the water region. Yin et al. [17] applied the improved PSPNet to the water surface segmentation task and later used the Canny edge detection algorithm to detect the shoreline, obtaining a segmentation MIoU of 96.87% on the USVInland dataset [27].

3. Method

3.1. U-Net Network

The U-Net network [28], which is a fully convolutional network (FCN) [29], achieves outstanding segmentation performance on small sample datasets for the job of segmenting medical images. As shown in Figure 1, the distinctive characteristic of U-Net is fully symmetric encoder–decoder composition. The feature maps input at each stage of the encoder and decoder are subjected to two consecutive 3 × 3 convolution processes without

padding, while each downsampling in the encoder corresponds to a 2 × 2 upsampling in the decoder. In addition, U-Net crops the feature maps of each layer of the encoder for the decoding process by Crop and Copy operation to supplement part of the information lost in the downsampling and upsampling process.

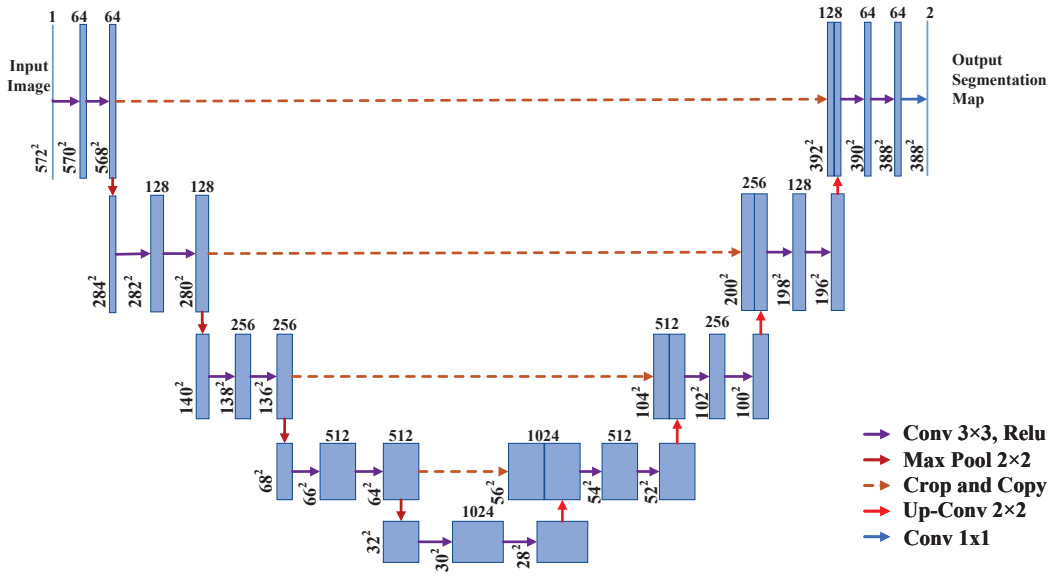


Figure 1. U-Net network architecture [28]. The input is a gray image, and the output is the probability that each pixel belongs to each category. The input image is downsampled four times to extract features of different levels and upsampled an equal number of times to recover the original resolution. Crop and Copy operation is used to first crop the feature map of different layers of the encoder, then copy them to the decoder for semantic segmentation.

3.2. RS-UNet Network

The RS-UNet network proposed in this paper is based on U-Net owing to the similarity in nature between water surface area segmentation and medical image segmentation. Medical image segmentation is essentially a binary segmentation task that segments lesion regions or other regions of interest in the input images. The water surface area segmentation problem that we wish to solve is also a binary segmentation task that involves segmenting the water surface area in the input image. Both tasks are simultaneously plagued by the problem of a limited quantity of training samples. For this reason, we think that in this study, we can take design inspirations from U-Net to build a water surface region segmentation network.

This network maintains the encoder–decoder architecture and employs equal amounts of downsampling and upsampling, as shown in Figure 2. To ensure that the final segmentation result is consistent with the resolution of the input image and that the feature maps of the corresponding stages of the encoder and decoder have the same spatial resolution, the convolution operation of each stage is padded according to the filter size of the convolution layer. This enables us to fully utilize the output of various stages of the encoder to make up for the information loss caused by sampling and to make comprehensive use of the contextual information at various scales to better complete the task of water surface area segmentation by using a skip connection instead of Crop and Copy opreation as in the original U-Net.

For the water surface area segmentation task, the effectiveness of the extracted features from the input image directly affects the segmentation results. Natural images used for this task are more complicated and contain more information than medical images, so the 10 layer convolutional network in the original U-Net encoder is unable to extract a sufficient amount of useful features from such complex scenes. To improve the feature extraction capability of the encoder while controlling the FLOPs of the network, we use

ResNet-34 [22] with the final average pooling layer and fully connected layer removed as the backbone to extract features. Because of the deeper network architecture and fewer FLOPs of the backbone, the encoder of RS-UNet can effectively extract more high-level and richer contextual features, which lays a foundation for the network to better complete the segmentation task with less inference time. In addition, the utilization of ResNet provides conditions for the introduction of the transfer learning approach, which solves the problem of insufficient training data and improves the training efficiency of the network.

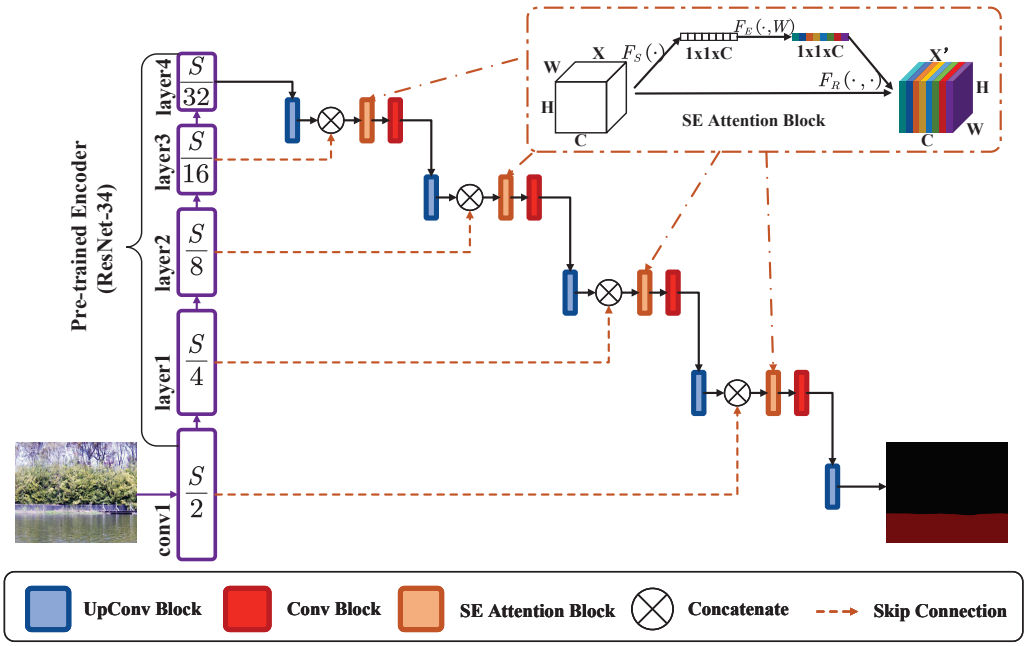


Figure 2. RS-UNet network architecture. *S* indicates the spatial resolution of the input image. The input is an RGB image, and the output is the segmentation result. The Encoder is the pretrained ResNet-34 and is fine-tuned during training. There are two input feature maps with different resolutions for each layer in the decoder. The lower feature is upsampled by the UpConv block and concatenated with the bigger larger feature through a skip connection. Then, interdependence between channels is modeled through the SE attention block and processed by the Conv block. At the end of the network, one UpConv block recovers the spatial resolution to the resolution of the input image and classifies each pixel into a category.

Theoretically, the real-time processing capability means that the network has low FLOPs. According to this theory, we built a very simple decoder for the network. The main computational body of each layer contains only one transposed convolution operation and one convolution operation; the former is responsible for recovering the spatial resolution, and the latter is responsible for processing the concatenated feature map. This architecture guarantees that the decoder contains low FLOPs, but it also leads to a loss of the computational capacity of the decoder. To overcome this weakness, the SE attention module [23] is introduced to each layer of the decoder. By modeling the interdependence between different channels of the concatenated feature map, this module can help the decoder focus its limited computational capacity on important features to improve performance. The additional computation required to introduce this kind of module is almost negligible. Table 1 shows the details of the network architecture.

Table 1. The architectural details of the network. Up: upsampling multiplier; *Channels*: the number of channels of each input and output module; In and Out: spatial resolutions of the feature maps; $Input$: input content of the module; $\otimes$: concatenation operation; S: spatial resolution of the original image; $F(B)$: the output corresponding to block B.

Encoder						
Block	**Filter Size**	**Stride**	**Channels**	**In**	**Out**	**Input**
conv1	7×7	2	3/64	S	$S/2$	Input image
MaxPooling	2×2	2	64/64	$S/2$	$S/4$	$F(conv1)$
layer1	3×3	1	64/64	$S/4$	$S/4$	$F(MaxPooling)$
layer2	3×3	2	64/128	$S/4$	$S/8$	$F(layer1)$
layer3	3×3	2	128/256	$S/8$	$S/16$	$F(layer2)$
layer4	3×3	2	256/512	$S/16$	$S/32$	$F(layer3)$
Decoder						
Block	**Filter Size**	**Up**	**Channels**	**In**	**Out**	**Input**
de4upconv	2×2	2	512/256	$S/32$	$S/16$	$F(layer4)$
de4se		1	512/512	$S/16$	$S/16$	$F(de4upconv \otimes layer3)$
de4conv	3×3	1	512/256	$S/16$	$S/16$	$F(de4se)$
de3upconv	2×2	2	256/128	$S/16$	$S/8$	$F(de4conv)$
de3se		1	256/256	$S/8$	$S/8$	$F(de3upconv \otimes layer2)$
de3conv	3×3	1	256/128	$S/8$	$S/8$	$F(de3se)$
de2upconv	2×2	2	128/64	$S/8$	$S/4$	$F(de3conv)$
de2se		1	128/128	$S/4$	$S/4$	$F(de2upconv \otimes layer1)$
de2conv	3×3	1	128/64	$S/4$	$S/4$	$F(de2se)$
de1upconv	2×2	2	64/64	$S/4$	$S/2$	$F(de2conv)$
de1se		1	128/128	$S/2$	$S/2$	$F(de1upconv \otimes conv1)$
de1conv	3×3	1	128/64	$S/2$	$S/2$	$F(de1se)$
upconv	3×3	2	64/2	$S/2$	S	$F(de1conv)$

The semantic segmentation task is essentially a pixel-level classification task, so the most commonly used loss function is the cross-entropy loss function. However, considering the potential positive and negative sample imbalance problem in the water segmentation task, a joint loss function (Equation (3)) based on Dice loss (Equation (1)) and focal loss (Equation (2)) is constructed in this paper to replace the cross-entropy loss function to supervise the training of the network.

$$L_{dice}(X,Y) = 1 - \sum_{k=0}^{K} \frac{2\omega_k \sum_{i=1}^{H}\sum_{j=1}^{W} p(X(i,j),k)g(Y(i,j),k)}{\sum_{i=1}^{H}\sum_{j=1}^{W} p(X(i,j),k) + \sum_{i=1}^{H}\sum_{j=1}^{W} g(Y(i,j),k)} \quad (1)$$

where H and W denote the height and width of the image, respectively; X and Y denote the predicted result of the network and the ground truth, respectively; K denotes the number of categories except the background; w_k represents the weight of each category; $p(X(i,j),k)$ denotes the probability of $X(i,j)$ being predicted as category k; and $g(Y(i,j),k)$ denotes the truth label of $Y(i,j)$ corresponding to category k.

$$L_{focal}(X,Y) = -\frac{1}{H \times W} \sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{K} (1 - p_t(X(i,j),k))^{\gamma} g(Y(i,j),k) \log\left(p_t(X(i,j),k)\right) \quad (2)$$

where $p_t(X(i,j),k)$ denotes the probability of $X(i,j)$ being predicted as category k, and γ is the focusing parameter.

$$L(X,Y) = \omega_{dice} L_{dice}(X,Y) + \omega_{focal} L_{focal}(X,Y) \quad (3)$$

where ω_{dice} and ω_{focal} denote the weight coefficients of Dice loss and focal loss in the loss function, respectively.

3.3. Shoreline Detection

The flow chart of our shoreline detection method is shown in Figure 3. We define shoreline detection as the combination of water surface segmentation and edge detection. The shoreline, which is the edge of the water surface area, can be obtained using the eight-neighborhood Laplace edge detection algorithm [30] based on the results of water surface segmentation. The extracted shoreline is then superimposed on the original picture for display. The outcome of shoreline detection is shown in Figure 4, demonstrating that a simple eight-neighborhood Laplace operator can effectively detect the complete shoreline based on the segmentation results, with only a small amount of additional computation generated.

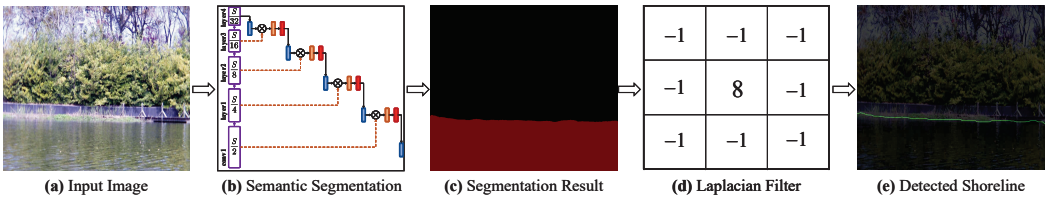


Figure 3. Flow chart of shoreline detection; from left to right: (**a**) input image; (**b**) semantic segmentation; (**c**) obtained segmentation result; (**d**) process executed by the Laplacian edge detection algorithm; (**e**) display of the detected shoreline.

(**a**) Original image. (**b**) Detected shoreline.

Figure 4. Shoreline detection results. (**a**) The original image; (**b**) the detected shoreline (green lines).

4. Results and Discussion

4.1. Experimental Implementation

Dateset and Evaluation Metrics According to our definition of shoreline detection introduced in Section 3.3, the key to shoreline detection is the semantic segmentation of water surface areas, the accuracy of which can be equivalent to the accuracy of shoreline detection. To enhance the water segmentation capability of the network and its adaptability to different scenes, a new dataset was constructed for the water surface segmentation task. The dataset consists of 433 images of various scenes; the resolutions of these images range from 220 × 165 to 5792 × 4344. These images were collected through the Internet or photographed by ourselves. The dataset includes a wide range of shooting angles and lighting conditions to make the dataset more representative and more widely applicable.

As the accuracy of shoreline detection can be equivalent to the accuracy of water surface segmentation, we directly quantitatively evaluated the performance of our shoreline

detection method using the metrics of semantic segmentation, that is, the Dice coefficient (Dice), mean intersection over union (MIoU), category mean pixel accuracy (MPA), and pixel accuracy (PA).

Training Setting For network training, the dataset is divided into a training set and a test set according in a 9-to-1 ratio. We set $\omega_k = \frac{1}{K+1}$ in Equation (1), $\gamma = 2$ in Equation (2), and $\omega_{dice} = \omega_{focal} = 1$ in Equation (3) according to the theory that Dice loss (Equation (1)) and focal loss (Equation (2)) constrain the network updating toward the same target from different perspectives and that their equal status benefits the capability of the network and saves on computation during training. We used adaptive moment estimation (Adam) [31] for optimization. Since pretraining weights were introduced, the combination of freeze training and unfreeze training was adopted. In a total of 150 epochs of the training process, we first went through 30 epochs of freeze training, in which the weights of the backbone were not updated, with an initial learning rate of 10^{-3} and a batch size of 4. The following 120 epochs trained the backbone, together with the rest of the network, with an initial learning rate of 10^{-4} and a batch size of 2. In addition, a learning rate decay coefficient of 0.96 was used throughout the whole training process.

For data augmentation, we employed some fundamental data augmentation techniques, such as random flipping, random scaling, and augmentation through HSV color space. Specifically, each input image was rescaled to between 25% and 200% of its original resolution, then horizontally flipped with a probability of 50%. Finally, its hue, saturation, and value were randomly adjusted to between 50% and 150% of the original value. All these processes were executed automatically and randomly.

Training was conducted on one NVIDIA GeForce RTX 3080 GPU.

4.2. Ablation Studies

We hypothesized that the potential imbalance of positive and negative samples in the segmentation task would affect the training effect of the network, so the joint loss function of Dice loss, which measures the similarity of segmentation results, and focal Loss, which boosts the weights of small samples, was employed in training to replace the cross-entropy loss function. In this section, experiments were conducted to compare the impact of different loss functions on the training of the network, which used only the cross-entropy loss function or the joint loss function. As shown in Table 2, the semantic segmentation performance of the network trained with the joint loss function is better. This indicates that considering the positive and negative sample imbalance problem is more beneficial to training semantic segmentation networks than simply measuring the pixel-level classification accuracy in the water surface area segmentation task.

Table 2. The impact of different loss functions on the training of the network. Best results are in bold.

Loss Function	Dice	MIoU (%)	MPA (%)	PA (%)
Cross-entropy loss	0.9748	96.65	98.27	98.37
Joint loss	**0.9763**	**97.05**	**98.49**	**98.56**

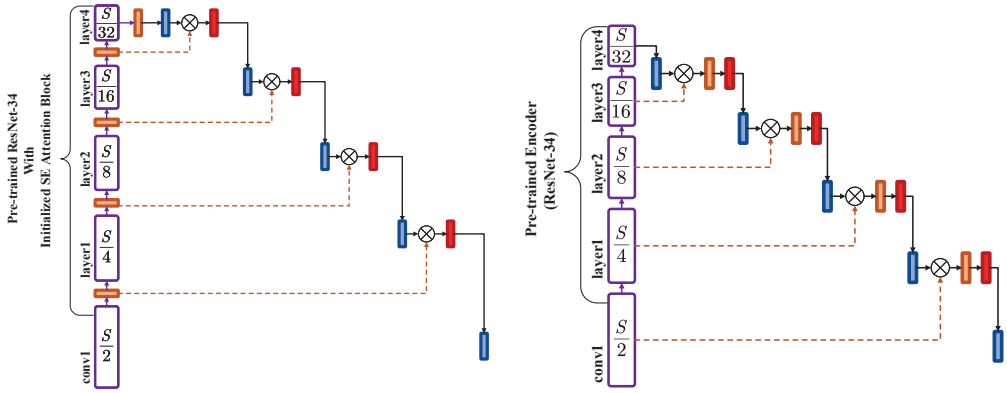
The effect of introducing attention mechanisms at different nodes on the network's performance is shown in Table 3. The two networks, RS-UNet-1 and RS-UNet-2, are depicted in Figure 5 as Figure 5a,b, respectively. RS-UNet-1 adds the attention mechanism to the pretrained backbone used for feature extraction, while RS-UNet-2 adds the attention mechanism to the decoder (ours). For training, the same hyperparameters are employed. According to the experimental results, integrating the attention mechanism into the pretrained decoder is preferable to integrating it into the backbone in an interpolated manner.

Owing to this phenomenon, we found that because the backbone was pre-trained on ImageNet, while the added SE attention module was not pre-trained but only initialized, this kind of interleaved combination of pre-trained module and non-pre-trained module destroyed the consistency of weight in the encoder. When training the network, the same learning rate cannot have a uniform effect on both parts, making the network converge to

a local optimum instead of a global optimum, which results in a suboptimal final output. During training, the convergence speed of RS-UNet-1 also lags behind that of RS-UNet-2, which is considered to have the same cause.

Table 3. Effect of adding the attention mechanism to different locations on network performance. The network architectures of RS-UNet-1 and RS-UNet-2 are shown in Figure 5. Best results are in bold.

Architecture	Dice	MIoU (%)	MPA (%)	PA (%)
RS-UNet-1	0.9735	96.66	98.26	98.37
RS-UNet-2	**0.9763**	**97.05**	**98.49**	**98.56**



(**a**) RS-UNet-1 (**b**) RS-UNet-2

Figure 5. RS-UNet-1 and RS-UNet-2 network architectures. RS-UNet-1 adds the attention mechanism in the encoder (integrated into the backbone), and RS-UNet-2 adds the attention mechanism in the decoder. All symbols are the same as in Figure 2.

4.3. Experimental Results and Analysis of Water Surface Segmentation

The comparison between our network and some mainstream convolution-based semantic segmentation networks for water surface segmentation performance and parameters is shown in Table 4. All the networks compared here were retrained on our dataset with its best hyperparameters. Note that all the networks were trained on one NVIDIA GeForce RTX 3080 GPU and tested on one NVIDIA GeForce RTX 3050Ti GPU; the spatial resolution of the images used in the test of processing speed was uniform, at 640 × 320. The improvement over [28,32,33] validates the effectiveness of introducing the attention mechanism. The improvement over [32–35] favorably validates the effectiveness of integrating contextual information at all scales using a skip connection. In summary, the combination of the attention mechanism and skip connection help the network successfully overcome the influence of inherent properties of the water surface, such as reflection and irregular boundary shapes. We can see that the processing speed of our network is the fastest due to the concise network architecture.

Table 4. Comparison of the water surface segmentation performance and parameters of different networks. We compare our network against some mainstream segmentation networks. All the numbers reported here are from our experiment. Best results are in bold.

Network	Params (M)	Dice Coefficient	MIoU (%)	MPA (%)	Pixel Accuracy (%)	FPS
U-Net [28]	31.0	0.8791	80.16	88.24	89.60	9.5
PSPNet [32]	65.6	0.9052	88.83	93.82	94.36	20
DeepLab v3+ [33]	59.3	0.9449	92.22	95.90	94.36	35
DANet [34]	66.6	0.9705	95.61	97.70	97.93	34
CondNet [35]	44.1	0.9664	95.36	97.67	97.80	37
RS-UNet(ours)	23.7	**0.9763**	**97.05**	**98.49**	**98.56**	**40**

In addition to the quantitative comparison, some qualitative results are also presented in Figure 6 to visually show the differences between these networks after being trained on the same dataset. The first column is the original input image, the associated ground truth is displayed in the second column, and the following columns are the segmentation results corresponding to different methods. The result of our network is the closest to the ground truth. Other networks have obvious incorrect segmentation problems due to the inherent properties of the water surface.

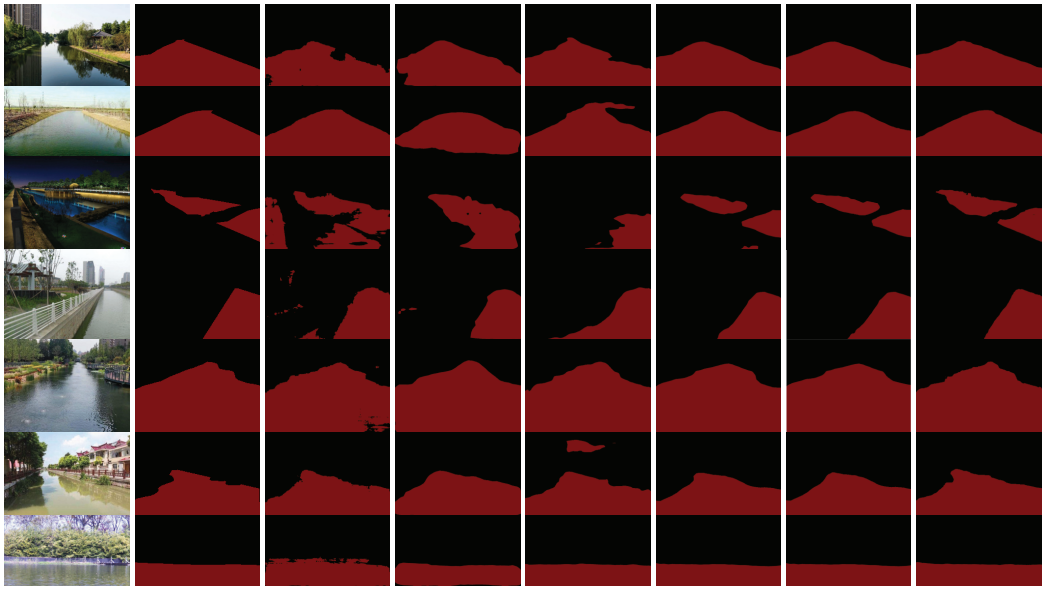


Figure 6. Some qualitative results of water surface segmentation of different networks. The first column is the input image, the second column is the corresponding ground truth, and the other columns are the segmentation results of the water surface area obtained by different methods. This figure visually shows the performance differences between different networks after training on the same dataset. Our results match the ground truth best.

4.4. Comparison with Other Shoreline Detection Methods

In addition to the comparison of water surface area segmentation results with those of mainstream segmentation methods, we also compared our shoreline method with other shoreline detection methods in the professional field. Due to the lack of open-source resources, here, our method was only compared with that of Yin et al. [17], one of the newest and best shoreline detection methods, on the USVInland dataset [27]. We trained our network on this dataset with the same hyperparameters as in Section 4.1 and followed their dataset division strategies. The results reported in [17] were directly used here. For fairness, this experiment was conducted on one RTX2080Ti. Because in these two methods, the segmentation performance is directly related to the shoreline detection performance, Table 5 only shows the comparison of these two methods in water surface segmentation.

Table 5. Comparison with other shoreline detection methods in segmentation performance and inferring speed. The data reported in the first row are from [17]. Best results are in bold.

Method	MIoU (%)	PA (%)	FPS
Yin et al. [17]	96.87	98.49	-
Ours	**97.21**	**98.60**	32

It can be seen that when compared with the other methods in the professional field, our method still shows a more favorable result. The inferring speed of our method is

very attractive, which satisfies the need for real-time shoreline detection. Furthermore, comparison of the improvements in MIoU and PA also demonstrates the effectiveness of the use of joint loss during training.

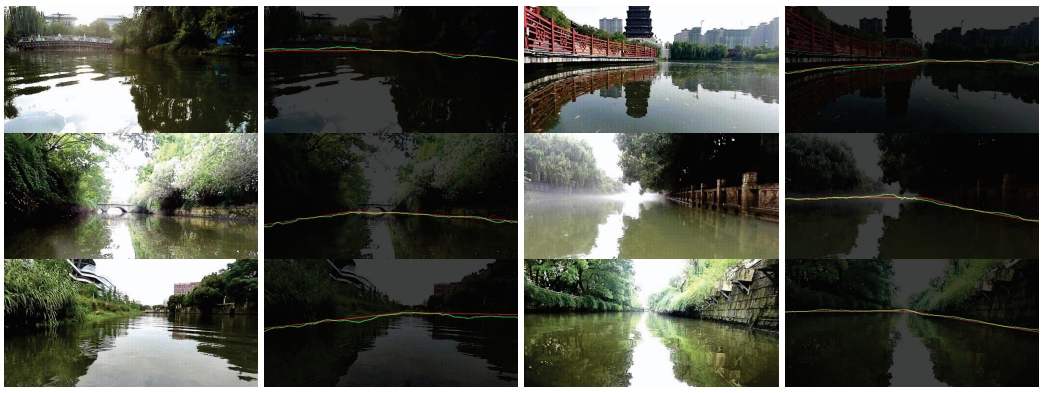
4.5. Experiment on Generalization Capability

As discussed in Section 4.3, the inherent properties of the water surface, such as reflection, and irregular boundary shapes, have a significant impact on the detection result. However, these common features are easily influenced by the environment; for example, the landscape on the shore affects the reflection. This causes the same feature to behave differently in different environments, which challenges the generalization capability of the network. To verify the generalization ability of our method in different scenes, we tested it on the USVInland dataset [27] and the port scenes collected by our team without any fine tuning. The segmentation performance of RS-UNet on these two datasets is shown in Table 6. Obviously, our network generalizes well on the two datasets.

As for the shoreline detection performance, some qualitative results on the USVInland and port datasets are shown in Figures 7 and 8, respectively. The green line indicates the shoreline detected by our method, the red line is the artificially delineated reference shoreline, and the yellow line indicates the overlap of the two. It can be seen that, although there are reflection interference problems in the USVInland dataset [27] and the port scene is not included in the training data, the shoreline detected by our method matches the reference shoreline closely in both environments. This excellent result demonstrates that our shoreline detection method can generalize well in various environments.

Table 6. The performance of RS-UNet on other datasets. The network was not re-trained on these datasets, just use the weights trained in Section 4.3.

Dateset	Dice	MIoU (%)	MPA (%)	PA (%)
USVInland [27]	0.9763	95.60	97.73	97.75
Port	0.9830	97.26	98.68	98.76



(**a**) Original Image (**b**) Detected Shoreline (**c**) Original Image (**d**) Detected Shoreline

Figure 7. The results of our method on the USVInland dataset for shoreline detection. The green line is the detected shoreline, the red line is the reference shoreline, and the yellow portions represent the overlap between the two.

(**a**) Original Image (**b**) Detected Shoreline (**c**) Original Image (**d**) Detected Shoreline

Figure 8. The results of our method on the port dataset for shoreline detection.The green line is the detected shoreline, the red line is the reference shoreline, and the yellow portion represents the overlap between the two.

5. Conclusions

In this paper, we discussed the current state of research in the field of shoreline detection and analyzed the reasons for the shortcomings of these methods in accurately detecting shorelines in real time. Accordingly, we constructed a more accurate and real-time shoreline detection method based on the proposed RS-UNet network. Our method defines shoreline detection as the combination of water surface area segmentation and edge detection. As the key of our method, we proposed RS-UNet as the water segmentation network, which can segment the water area accurately and in real time. Experiments show that our RS-UNet achieves a 97.05% MIoU and 40 FPS processing speed in the task of segmenting the water surface area, which is better than some existing mainstream semantic segmentation methods and deep-learning-based shoreline detection methods. We also demonstrated the generalization capability of our method through experiments. In summary, our shoreline detection method can accurately detect shorelines in real time and in various environments.

Although our method performs well in shoreline detection, it is still to some limitations. The main limitation is the insufficient amount of training data with annotations, which limits the generalization ability of our method. Our future work will be focused in two directions. The first direction is exploring training our network in an unsupervised manner. By employing an unsupervised training process, we can use a large number of images without annotations for training and significantly reduce the impact of a lack of training data. Another direction is to treat this work as a foundation, integrating the proposed method with other downstream tasks. such as obstacle detection on the water surface, automatic visual positioning, and the monitoring of distance between USVs and the shore, in order to provide guarantees for the safety of navigation of USVs.

Author Contributions: Conceptualization, J.Z. and F.S.; Data curation, F.S.; Funding acquisition, J.Z. and S.W.; Investigation, G.G. and F.S.; Methodology, J.Z. and F.S.; Project administration, J.Z. and S.W.; Resources, J.Z. and G.G.; Software, F.S.; Supervision, S.W.; Validation, F.S.; Visualization, F.S.; Writing—original draft, F.S.; Writing—review and editing, J.Z., S.W. and F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Key Research and Development Program, China (Grant no. 2021YFC2801004) and the National Natural Science Foundation of China (Grant nos. 52071199 and 52102397).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Access to the data will be considered upon request by the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

As there is a large number of acronyms and terms in our manuscript, we provide an abbreviations list here for a better reading experience.

Abbreviation	Full Name
MIoU	Mean intersection over union
FPS	Frames per second
USV	Unmanned surface vehicle
RANSAC	Random sample consensus
HSV	Hue saturation value
SE	Squeeze and excitation
RS-UNet	Residual squeeze-and-excitation U-Net
CRF	Conditional random fields
FCNs	Fully convolutional Networks

References

1. Yang, P.; Song, C.; Chen, L.; Cui, W. Image Based River Navigation System of Catamaran USV with Image Semantic Segmentation. In Proceedings of the 2022 WRC Symposium on Advanced Robotics and Automation (WRC SARA), Beijing, China, 20 August 2022; pp. 147–151.
2. Sinisterra, A.J.; Dhanak, M.R.; Von Ellenrieder, K. Stereovision-based target tracking system for USV operations. *Ocean Eng.* **2017**, *133*, 197–214. [CrossRef]
3. Ji, X.; Zhuang, J.Y.; Su, Y.M. Marine radar target detection for USV. In Proceedings of the Advanced Materials Research, Lille, France, 26–30 May 2014; Trans Tech Publications: Bach, Switzerland, 2014; Volume 1006, pp. 863–869.
4. Wang, Z.; Zhang, Y. Estimation of ship berthing parameters based on Multi-LiDAR and MMW radar data fusion. *Ocean Eng.* **2022**, *266*, 113155. [CrossRef]
5. Wei, Y.; Zhang, Y. Effective waterline detection of unmanned surface vehicles based on optical images. *Sensors* **2016**, *16*, 1590. [CrossRef] [PubMed]
6. Kröhnert, M. Automatic waterline extraction from smartphone images. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 857. [CrossRef]
7. Zhan, W.; Xiao, C.; Yuan, H.; Wen, Y. Effective waterline detection for unmanned surface vehicles in inland water. In Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 28 November–1 December 2017; pp. 1–6.
8. Feng, T.; Xiong, J.; Xiao, J.; Liu, J.; He, Y. Real-time riverbank line detection for USV system. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 2546–2551.
9. Liu, R.W.; Yuan, W.; Chen, X.; Lu, Y. An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system. *Ocean Eng.* **2021**, *235*, 109435. [CrossRef]
10. Chen, P.; Li, Y.; Zhou, H.; Liu, B.; Liu, P. Detection of Small Ship Objects Using Anchor Boxes Cluster and Feature Pyramid Network Model for SAR Imagery. *J. Mar. Sci. Eng.* **2020**, *8*, 112. [CrossRef]
11. Liang, M.; Liu, R.W.; Li, S.; Xiao, Z.; Liu, X.; Lu, F. An unsupervised learning method with convolutional auto-encoder for vessel trajectory similarity computation. *Ocean Eng.* **2021**, *225*, 108803. [CrossRef]
12. Feng, H.; Cao, G.; Xu, H.; Ge, S.S. IS-STGCNN: An Improved Social spatial-temporal graph convolutional neural network for ship trajectory prediction. *Ocean Eng.* **2022**, *266*, 112960. [CrossRef]
13. Chen, X.; Liu, S.; Liu, R.W.; Wu, H.; Han, B.; Zhao, J. Quantifying Arctic oil spilling event risk by integrating an analytic network process and a fuzzy comprehensive evaluation model. *Ocean Coast. Manag.* **2022**, *228*, 106326. [CrossRef]
14. Xue, H.; Chen, X.; Zhang, R.; Wu, P.; Li, X.; Liu, Y. Deep Learning-Based Maritime Environment Segmentation for Unmanned Surface Vehicles Using Superpixel Algorithms. *J. Mar. Sci. Eng.* **2021**, *9*, 1329. [CrossRef]
15. Steccanella, L.; Bloisi, D.; Blum, J.; Farinelli, A. Deep learning waterline detection for low-cost autonomous boats. In Proceedings of the International Conference on Intelligent Autonomous Systems, Singapore, 1–3 March 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 613–625.
16. Steccanella, L.; Bloisi, D.D.; Castellini, A.; Farinelli, A. Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring. *Robot. Auton. Syst.* **2020**, *124*, 103346. [CrossRef]
17. Yin, Y.; Guo, Y.; Deng, L.; Chai, B. Improved PSPNet-based water shoreline detection in complex inland river scenarios. *Complex Intell. Syst.* **2022**, *9*, 233–245. [CrossRef]
18. Shen, J.; Tao, Q.; Xiao, Z. Shoreline detection algorithm based on the improved Deeplab v3+ network. *J. Image Graph.* **2019**, *23*, 2174–2182.
19. Adam, M.A.M.; Ibrahim, A.I.; Abidin, Z.Z.; Zaki, H.F.M. Deep Learning-Based Water Segmentation for Autonomous Surface Vessel. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Kuala Lumpur, Malaysia, 20–21 October 2020; IOP Publishing: Bristol, UK, 2020; Volume 540, p. 012055.

20. Vandaele, R.; Dance, S.L.; Ojha, V. Automated water segmentation and river level detection on camera images using transfer learning. In Proceedings of the DAGM German Conference on Pattern Recognition, Tubingen, Germany, 28 September–1 October 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 232–245.
21. Zhan, W.; Xiao, C.; Wen, Y.; Zhou, C.; Yuan, H.; Xiu, S.; Zou, X.; Xie, C.; Li, Q. Adaptive semantic segmentation for unmanned surface vehicle navigation. *Electronics* **2020**, *9*, 213. [CrossRef]
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27 June–1 July 2016; pp. 770–778.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
24. Kristan, M.; Sulić Kenk, V.; Kovačič, S.; Perš, J. Fast Image-Based Obstacle Detection From Unmanned Surface Vehicles. *IEEE Trans. Cybern.* **2016**, *46*, 641–654. [CrossRef] [PubMed]
25. Peng, M.; WANG, J.; WEN, X.; Cong, X. Shoreline detection method by combining HSV spatial water image feature. *J. Image Graph.* **2018**, *23*, 526–533.
26. Yao, L.; Kanoulas, D.; Ji, Z.; Liu, Y. ShorelineNet: An Efficient Deep Learning Approach for Shoreline Semantic Segmentation for Unmanned Surface Vehicles. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 5403–5409.
27. Cheng, Y.; Jiang, M.; Zhu, J.; Liu, Y. Are we ready for unmanned surface vehicles in inland waterways? The usvinland multisensor dataset and benchmark. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3964–3970. [CrossRef]
28. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
29. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
30. Wang, X. Laplacian operator-based edge detectors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 886–890. [CrossRef] [PubMed]
31. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
32. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
33. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
34. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3146–3154.
35. Yu, C.; Shao, Y.; Gao, C.; Sang, N. CondNet: Conditional classifier for scene segmentation. *IEEE Signal Process. Lett.* **2021**, *28*, 758–762. [CrossRef]