electronics

Article

Semantic Segmentation of Side-Scan Sonar Images with Few Samples

Dianyu Yang, Can Wang, Chensheng Cheng, Guang Pan and Feihu Zhang *

School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
* Correspondence: feihu.zhang@nwpu.edu.cn; Tel.: +86-15596611656

Abstract: Underwater sensing and detection still rely heavily on acoustic equipment, known as sonar. As an imaging sonar, side-scan sonar can present a specific underwater situation in images, so the application scenario is comprehensive. However, the definition of side scan sonar is low; many objects are in the picture, and the scale is enormous. Therefore, the traditional image segmentation method is not practical. In addition, data acquisition is challenging, and the sample size is insufficient. To solve these problems, we design a semantic segmentation model of side-scan sonar images based on a convolutional neural network, which is used to realize the semantic segmentation of side-scan sonar images with few training samples. The model uses a large convolution kernel to extract large-scale features, adds a parallel channel using a small convolution kernel to obtain multi-scale features, and uses SE-block to focus on the weight of different channels. Finally, we verify the effect of the model on the self-collected side-scan sonar dataset. Experimental results show that, compared with the traditional lightweight semantic segmentation network, the model's performance is improved, and the number of parameters is relatively small, which is easy to transplant to AUV.

Keywords: side-scan sonar; segmentation; CNN; SE-block; multi-channel

Citation: Yang, D.; Wang, C.; Cheng, C.; Pan, G.; Zhang, F. Semantic Segmentation of Side-Scan Sonar Images with Few Samples. *Electronics* **2022**, *11*, 3002. https://doi.org/10.3390/electronics11193002

Academic Editor: Byung Cheol Song

Received: 29 August 2022
Accepted: 19 September 2022
Published: 22 September 2022

1. Introduction

With the continuous improvement of the technical level, robot perception and recognition have begun to develop toward intelligence and automation in the underwater research field. Recognition and perception rely on front-end equipment capturing environmental features, which is sonar for the underwater environment. Therefore, correlation analysis and processing methods of sonar images have received extensive attention in recent years [1–3]. Side-scan sonar transmits sound waves and receives echoes from underwater objects to image underwater objects and calculate approximate distances [4]. The original sonar image has low resolution, serious noise interference, and a fuzzy target shape, which greatly complicates the recognition work of researchers [5].

However, achieving a lasting effect through a manually designed filtering algorithm in a complex and changeable underwater environment is not easy. If the judgment depends on experienced personnel, it will significantly increase the cost and reduce efficiency. Therefore, it is of great significance to design a feature extraction model for sonar images that can replace, or at least assist, human judgment.

Image processing models based on deep learning algorithms have made great progress recently. Among them, classical image classification models, such as VGG-net [6], GoogLeNet [7], and Resnet [8], have achieved good results on many camera image datasets. Image segmentation models represented by FCN [9], U-net [10], PSPNet [11] have also attracted the attention of many researchers. GAN networks are also widely used in machine learning data generation to solve the problem of insufficient data [12–14]. Given the good results of these algorithms, the researchers hope to apply them to underwater acoustic images, thereby advancing the field of underwater sensing and detection.

Song et al. [15] proposed a preliminary segmentation model of side-scan sonar image based on the FCN network model. Their model divides the image into the target area, shadow area, and seabed reverberation area. Finally, MRF is used to process the classification results to improve accuracy. Chen et al. [16] proposed a semi-supervised CNN network model, which uses many unlabeled or weakly labeled samples and a few densely labeled samples to segment the SAR images. Wu et al. [17] proposed a convolutional neural network model for side-scan sonar named ECNet. The network structure consists of an encoder and a decoder. The encoder obtains contextual features, and the decoder is used for image restoration. In addition, a single-stream deep neural network with multiple side outputs is added to optimize edge segmentation. Huo et al. [18] proposed a semi-synthetic sonar data generation method. For the input optical image, the CNN model combines image segmentation with intensity distribution simulation in different regions to generate synthetic sonar images of the plane and the drowning person to enrich the sonar image data set. Zhou et al. [19] added the Laplacian energy filter based on the CNN model, and the two-channel pulse-coupled neural network was used to fusion the side-scan sonar images and achieved good results. In the work of Połap et al. [20], a method based on a neural network model is proposed to search for target signals in ocean areas and restore areas with low image quality. Zhu et al. [21] used the convolutional neural network model to extract the target features of side-scan sonar images and input them into the trained SVM for classification.

Side-scan sonar is a kind of active imaging sonar. Its imaging principle is to send a short acoustic pulse with a slight horizontal opening angle (about 1 degree) and a large vertical opening angle to one or both sides of the vertical direction of the survey ship. After the pulse reaches the seabed, it is continuously reflected according to the distance from the seabed to the transducer. The sonar image with uneven gray level changes is drawn according to the strength of the reflected signal. Sonar images can be used to observe changes in the seafloor topography, whether there are obstacles to the navigation, and the type of seabed substrate. When the side-scanning sonar emission pulse propagates in water and meets the target, the target scatters the acoustic energy in all directions, and the transducer receives the backscattered echo. In contrast, the acoustic energy is difficult to reach the side and rear of the target (called the blind area).The sonar array moves forward with the carrier, and in the process of moving forward, sonar continues to transmit, receive and form sonar images [22]. As a result, the target (strong echo signal of the target) and its shadow (blind area behind the side of the target) appear at the corresponding position on the sonar image. It can be seen that the side-scan sonar reflects the echo intensity of the detected target so that the side-scan sonar image can be understood as a single-channel gray map, and the target with stronger reflection has greater brightness. However, the difference in brightness of most underwater targets is not apparent, so there must be a particular dimension of the color channel that contains most of the target information in the image.

On this basis, in this paper, a side-scan sonar image segmentation model is proposed based on the CNN network. Compared with camera images, side-scan sonar images are more challenging to acquire and have less data, so the network model needs to control the depth to avoid overfitting. In addition, due to the low color richness of side-scan sonar images, each channel contains a relatively large amount of information, so it is necessary to focus on the information in essential channels.

The main contributions of this paper are as follows:

(1) We introduced the SE module to increase channel attention in the feature extraction process and increase independent weight for each channel so that the more critical channels obtain a higher weight to improve the overall segmentation accuracy.
(2) We increased the convolution kernel size used from 3×3 to 7×7, which proved effective in sonar images with a larger size. Meanwhile, DW convolution was adopted to reduce the number of parameters given the increase in the number of parameters caused by the expansion of the convolution kernel size.

(3) Simply increasing the convolution kernel size cannot effectively improve the quality of feature extraction. Therefore, we constructed a parallel feature extraction channel using a small-size convolution kernel and concatenated its output with the leading network to achieve multi-scale feature extraction.
(4) We used a full convolution layer to restore the output of the decoder to the original image size and output the segmentation results. Then we conducted a contrast experiment with other lightweight CNN.

The rest of this paper is divided into five sections: Section 2 introduces the work of other researchers related to the model design; Section 3 presents the structure and details of the model; Section 4 uses the self-collected side-scan sonar data to verify the performance of the model; and Section 5 gives the conclusion.

2. Related Work

In this section, some essential concepts for model design are introduced, including the basic principle of the CNN network, the U-NET network's design idea, and the SE module's influence.

2.1. Principles of CNNs

Neural network models with CNN were completed by Lecun Y [23] and carried forward by AlexNet [24]. In the classical CNN model, data have two directions: forward propagation and backward propagation. Forward propagation realizes data feature extraction through the convolutional layer, pooling layer, activation function layer, and fully connected layer. The convolution layer is processed by multiple convolution checks to extract high-dimensional feature maps. The pooling layer compresses the parameters while preserving the main features. Finally, the activation function ensures the nonlinearity of the multi-layer network structure, and the last fully connected layer implements the mapping from image features to classification categories. According to the comparison between the output results of the forwarding propagation and label data, backpropagation performs gradient descent on network parameters layer by layer in reverse to improve the network performance. Finally, the network achieves due performance after multiple forward and backward propagation.

2.2. U-Net and FCN

There are many excellent models for semantic segmentation tasks, such as DeepLabV3 [25], hrnet [26], Transformer [27], etc. However, the original design concept of the segmentation model comes from FCN. The initial neural network model can only be applied to the classification task, and the emergence of FCN brought it into the field of image segmentation. Pioneering the model using the convolution layer instead of full connection as the last layer of the network's output solved the problem that the whole connection layer limits the input size. In addition, the model outputs from a one-dimensional probability vector into a two-dimensional probability matrix. That is, every pixel can be classified. FCN uses deconvolution and linear interpolation for image restoration and uses the feature fusion method of skip layer. It concatenates image features of high and low dimensions, which greatly impacts the design idea of the subsequent segmentation model. The structure of the FCN network model is shown in Figure 1.

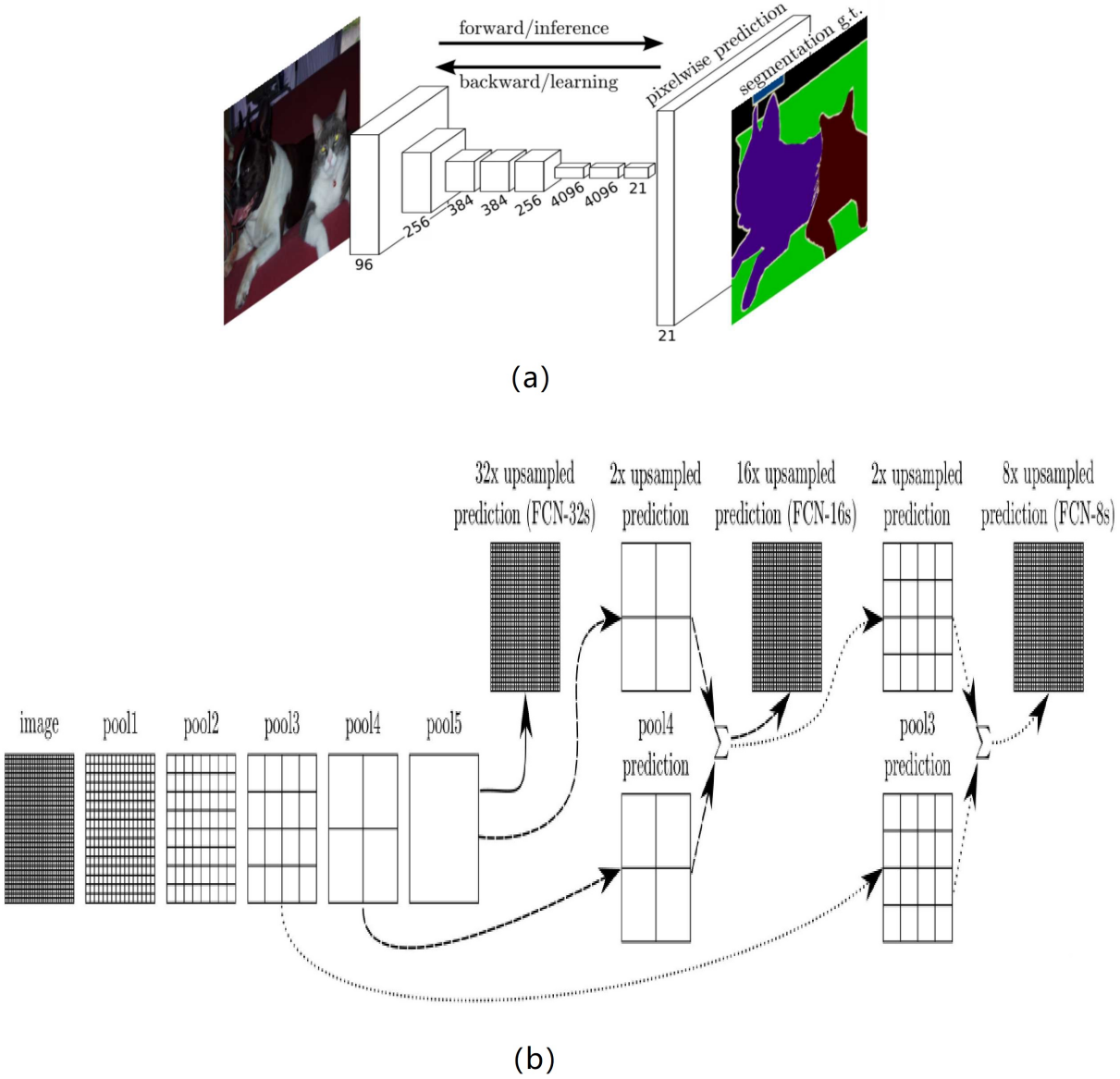


Figure 1. (**a**) FCN model with VGG as a backbone [9]. (**b**) Skip layer of FCN: There are three versions of the FCN network, namely FCN-8S, FCN-16S, and FCN-32S. The 32S version directly performs image restoration after a feature fusion, so the output quality is the lowest, but the number of parameters is the lowest. The 8S version can obtain the highest precision output after three times of feature fusion. 16S is relatively balanced.

U-net is an image segmentation network model that draws on the FCN model. The model still adopts the design idea of deconvolution restoration and full convolution instead of complete connection. However, it gives up using the VGG network as a backbone and designs a symmetric four-layer codec structure instead. At the same time, feature fusion is carried out between encoding and decoding structures at the same level, similar to skip layer. The u-net model is still the mainstream algorithm in all minor sample segmentation problems, such as medical image segmentation, due to its low depth, fewer parameters, and good segmentation effect. The U-NET model structure is shown in Figure 2.

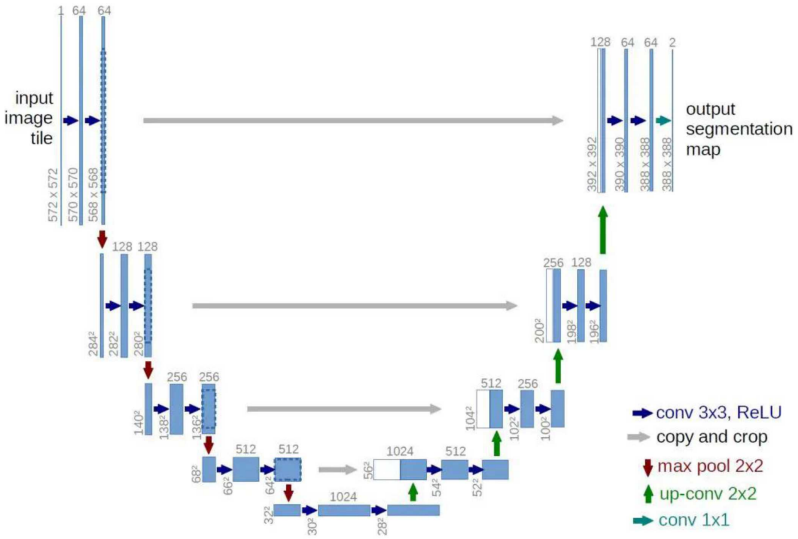


Figure 2. U-Net model [10]: classical symmetric codec structure with feature concatenate.

2.3. The Effect of SE-Block

SENet [28] is the ImageNet 2017 champion model. The SE-block structure is shown in Figure 3. Its full name is squeeze-and-excitation congestion networks. The main contribution is a channel attention extraction module called Se-block that can be added to any network structure.

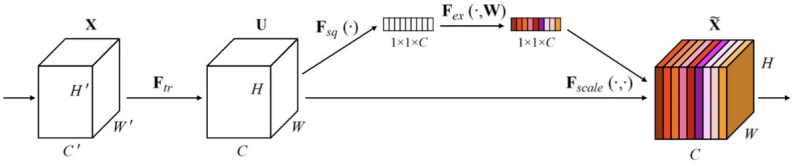


Figure 3. The structure of SE-block [28].

The module consists of two parts: the squeezing part, which compresses the original 3D data input into a one-dimensional vector, implemented mainly by global average pooling (this operation can extract the global features of each channel); and the crimping section, which uses a full connection layer to map the output of the compression module to a predicted weighting sequence, which is multiplied by all the channels for weighting. This module can effectively extract important channel features and ignore minor channels.

3. Method

The design ideas of our model are derived from U-NET, and we adopt a coding–decoding structure similar to U-NET and SENet, as well as the large convolution kernel and re-parameterizing mentioned in RepLKNet [29], but improve it for our downstream tasks. First, we added Se-block to the encoder, namely the feature extraction module, to obtain the weight of different feature channels. The network model will find the channel that significantly impacts the segmentation output result (the channel added after multiple convolutions, rather than the original RGB), increases the weight proportion of its corresponding parameters, and focuses on adjustment. Then, the large and small convolution kernels are used to capture features of different scales in parallel. Finally, after fusion and restoration, the image segmentation results are output.

3.1. Multi-Scale Feature Fusion

Due to the increasing complexity of images, multi-scale feature fusion has become a necessary capability for a qualified segmentation network. The skip layer of FCN, the codec information interaction of U-NET, and the ASPP module of the Deeplab model all belong

to this kind of structure. The RepLKNet model proposes a structure-reparameterization method. The model uses a large convolution kernel (31 × 31) for feature extraction, and a parallel feature extraction channel using a conventional 3 × 3 small convolution kernel is added. After the parameter training of the convolution kernel is completed, the small convolution kernel is directly inserted into the large convolution kernel to realize the feature fusion of different levels of size and scale.

Due to the difficulty of obtaining side-scan sonar images, we cannot provide the massive amount of data required for training large convolutional kernels and deep networks, such as RepLKNet. Therefore, after slightly expanding the size of the convolution kernel, we did not insert the small convolution kernel directly into the large convolution kernel because this would destroy the feature extraction ability of the large convolution kernel itself. Instead, we use the concatenate method to incorporate features of different scales before restoring images using deconvolution.

3.2. Depthwise Separable Convolution

The concept of depthwise separable convolution was first proposed by MobileNet [30]. The standard convolution operation is decomposed into two steps: the first step is deep convolution, and the second step is point convolution. A specific example is used to compare the difference between this method and standard convolution: assuming that the size of the input image is 12 × 12 × 3 (3 represents three channels), and the desired output result is 8 × 8 × 128, so 128 5 × 5 × 3 convolution kernels are needed for convolution, and the number of operations in the whole process is 9600.

If deep convolution is used first, three 5 × 5 × 1 convolutions are used to convolve the three channels of the image, and the output result of 8 × 8 × 3 is obtained. Then point convolution is used, 128 1 × 1 × 3 convolution kernels (equivalent to one pixel containing three channels) are used to convolve the previous output results again, and finally, the output results of the same size are obtained. Still, the number of operations is reduced to 5 × 5 × 3 + 1 × 1 × 3 × 128 = 469.

The deep separable volume reduces the amount of network computation at the cost of increasing the depth of the network, which may affect the output results of the network while speeding up the calculation speed. Therefore, this practice may not play a positive role for networks mainly using small convolution kernels, but it is indispensable for our model.

3.3. Model Structure

The structure of our model is borrowed from the design of U-NET, and the central part is the four-layer codec, shown in Figure 4.

The encoder consists of four layers in total, and each layer contains an encode block. Each encode block uses a convolution kernel size of 7 × 7 (DW convolution is used to improve the operation rate while adding padding). The number of output channels in each layer is 32, 64, 128, and 256. Meanwhile, SE-blocks are added parallel to each layer to predict the channel weights.

Another parallel feature extraction channel uses a small-size convolution kernel; the main structure is similar to the central part.

The decoder input is the high-dimensional feature map extracted by the encoder, and the channel is 256. The decoder uses deconvolution to up-sample layer by layer. First, concatenate with the same dimensional features output by the feature channel using a small convolution kernel, then convolve twice and input to the next layer. After four repetitions, the image segmentation results are obtained through the full convolutional layer.

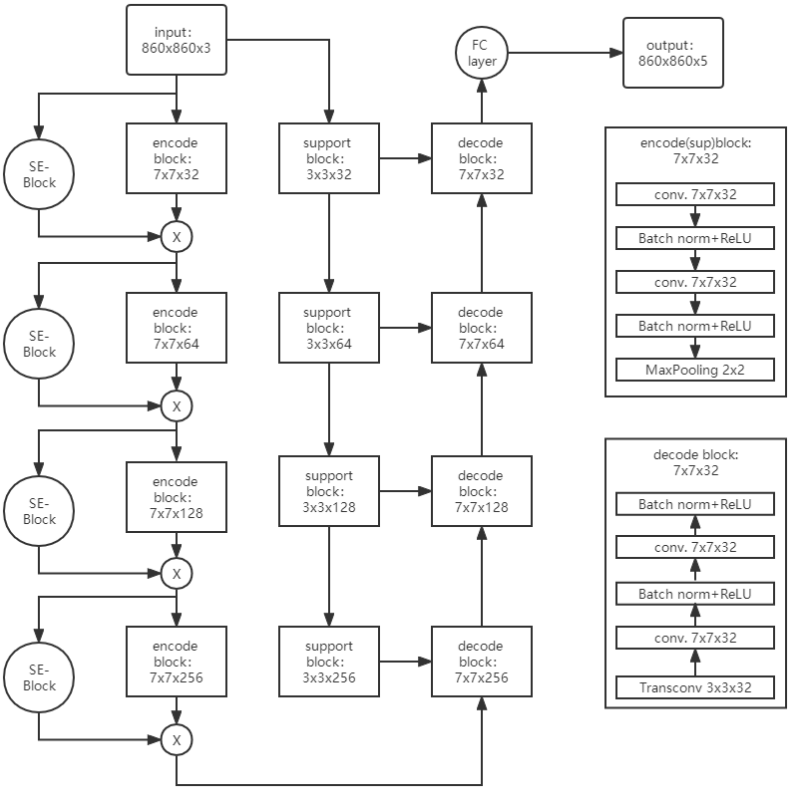


Figure 4. The structure of our model.

4. Experiment and Analysis

All experiments were conducted with Intel Core i9-10900F CPU@2.8 Ghz × 20, 64 GB RAM, Nvidia Geforce 3090 GPU, 24 GB of video memory, by CUDA Toolkit 11.3, CUDNN V8.2.1, Python 3.6, PyTorch-GPU 1.10.1, Ubuntu18.04.operating system.

4.1. Dataset Collection

We used Hydro 3060 dual-frequency side-scan sonar to collect sonar data needed for the experiment in the Lake District of Jiande, Hangzhou, China. The original image captured frame by frame was 960 × 960 pixels in size, and its effect is shown in Figure 5.

The side-scan sonar is mounted on an AUV and emits sound waves to both sides as the subject moves, collecting echoes from underwater objects to build an image. The bright parts of the image represent the targets with strong echoes, such as rocks and metals, while the parts without echoes will appear black, such as water bodies and blocked parts.

Our model is based on supervised learning, which requires manually annotated accurate data labels as training data. We annotated the data using LabelMe, open-source software on the Ubuntu platform. For the whole dataset, we divided the data into five categories (not every image contains labels from all five categories): (1) water; (2) the mountain part; (3) the land; (4) shaded part; and (5) unmarked area (background). The unlabeled area mainly refers to the debris area left after the first four types of image labeling. The labeled image is shown in Figure 6.

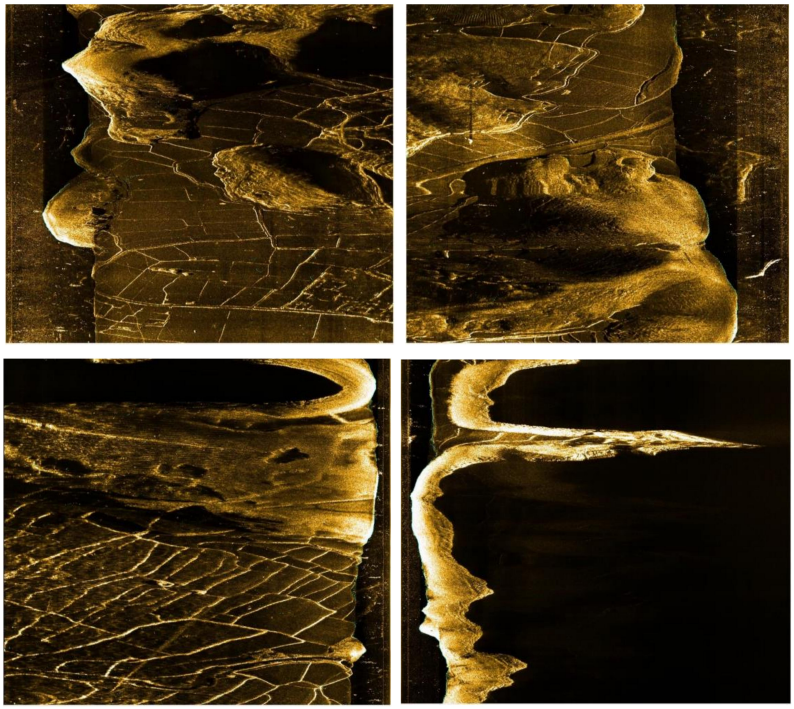
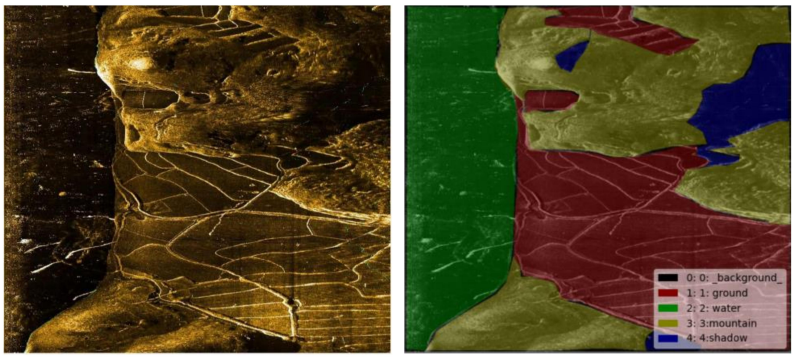


Figure 5. The original sonar image (each sonar image is cropped down the middle into two images).



(a) (b)

Figure 6. (**a**) Original image, (**b**) label.

4.2. Data Augmentation

As mentioned before, collecting side-scan sonar data is challenging, so the amount of data is not very rich. Therefore, we adopted the method of data amplification to increase the number of samples to ensure the training effect, and the method used is shown in Figure 7.

(1) The most common method is to flip the image at different angles, amplifying the data but also breaking the location correlation and making the network more generalized.
(2) Image translation is also a standard method, which controls the image translation in four directions by some random numbers, but not too much. Otherwise, it will destroy the feature structure of the image.
(3) By randomly clipping the original image, the size of the image can be reduced while the data are expanded, and the training can be accelerated.

The sonar image is less dependent on shape features but more on color features, so no color data amplification was carried out. The size of the original sonar data collected is 960 × 960, and the number is about 300. After data amplification, the data size is 860 × 860,

and the number is increased by about four times. We randomly selected 60 percent as the training set, and the validation and test sets were 20 percent.

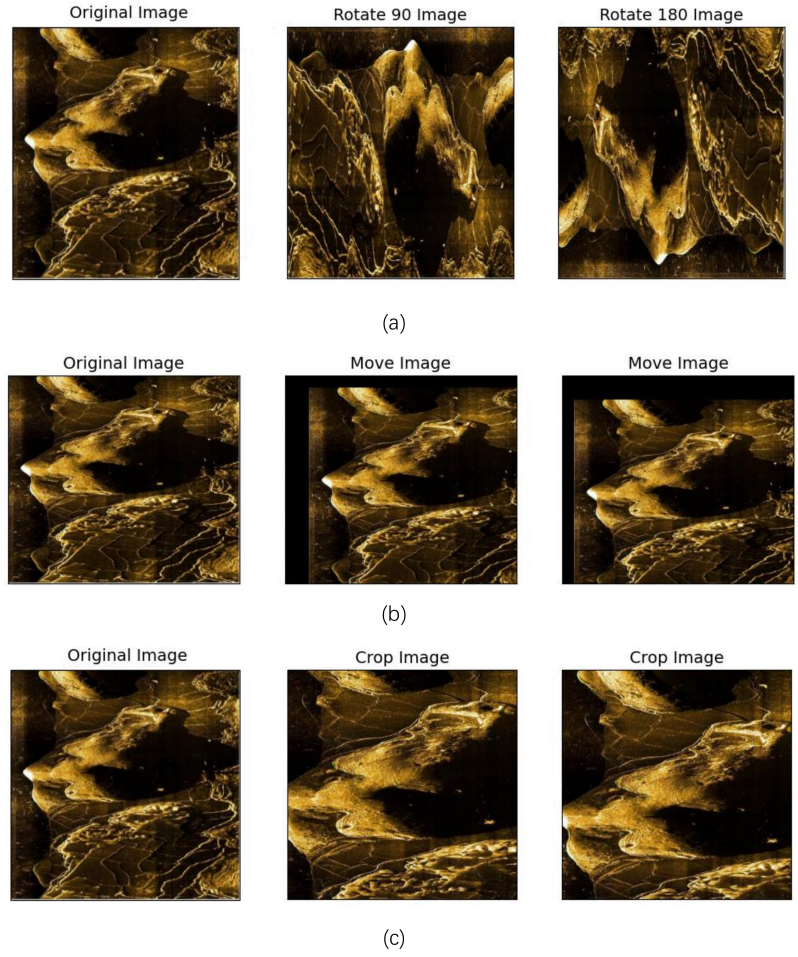


Figure 7. Data augmentation (**a**) image inversion, (**b**) image panning, (**c**) random crop.

4.3. Verification Indicators

We measure the model from two perspectives: the consumption of computing resources, and the model's accuracy. Computing resources are measured by the total number of network parameters and the FLOPs indicator, which refers to floating point operations. More FLOPs mean more computing resources consumed by the model. The calculation formula of the convolution layer FLOPs of the convolutional network is as follows:

$$FLOPs = (2c_{in}k^2 - 1)HWc_{out} \tag{1}$$

c_{in} and c_{out} represents the number of input and output channels in the convolution layer, and k represents the size of the convolution kernel. The size of the output feature graph is $H \times W$.

OA (overall accuracy) and MIoU (mean intersection over union) will measure the model accuracy. The calculation formula of OA is as follows:

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

TP, TN, FP, FN mean true positive (positive sample is judged as a positive sample), true negative (negative sample is judged as a negative sample), false positive (negative sample is misjudged as a positive sample), and false negative (positive sample is misjudged as a negative sample).

The calculation formula of $MIoU$ is as follows:

$$MIoU = \frac{1}{k}\sum_{i=1}^{k}\frac{p \cap g}{p \cup g} \tag{3}$$

P means prediction, and G means ground truth.

4.4. Network Model Training

We use the processed sonar data for network training, and the hyperparameters used in the training process are listed in Table 1. The loss function used in the training process is the cross-entropy loss function, and the training process is shown in Figure 8.

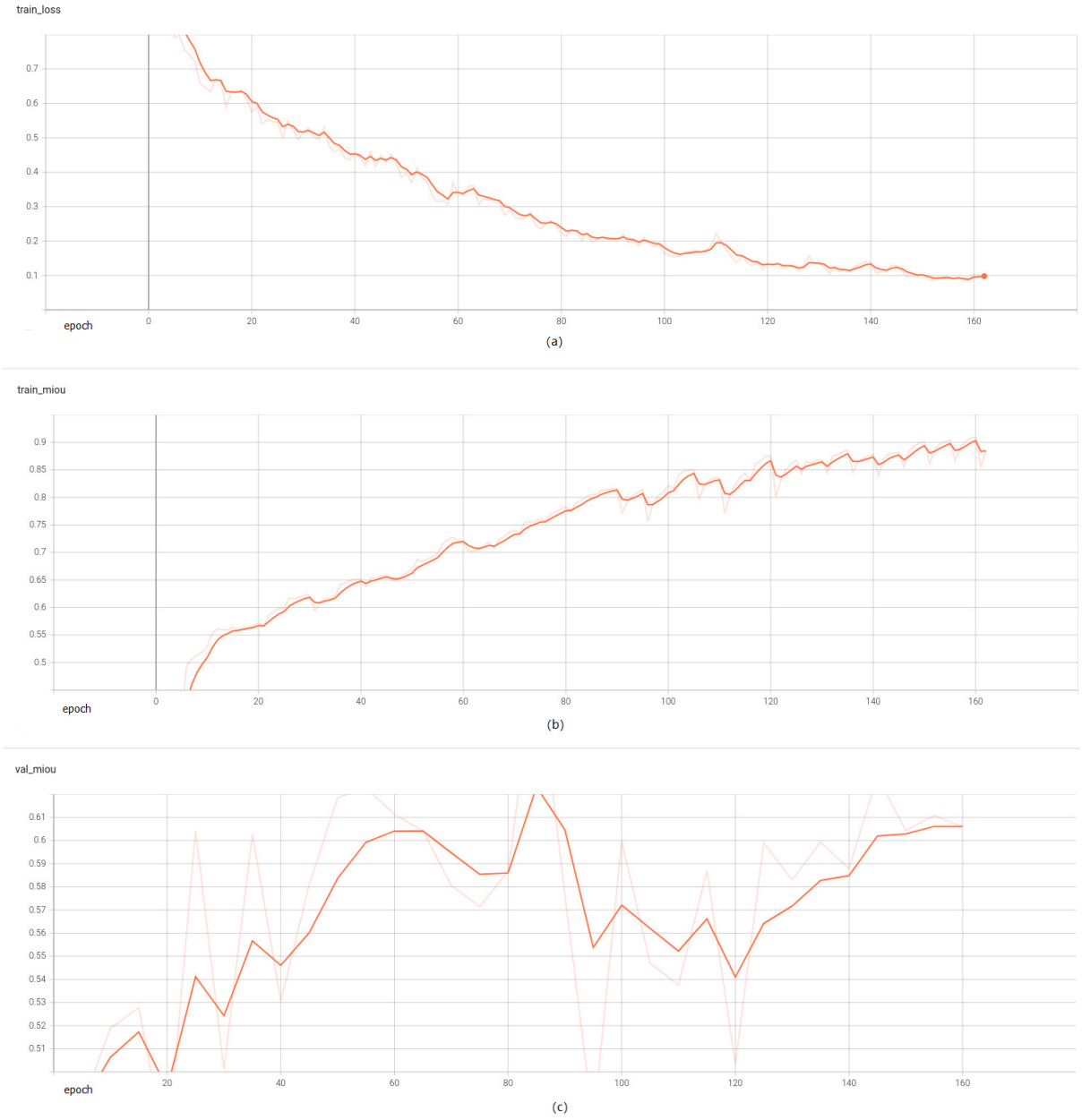


Figure 8. We use TensorBoard to draw the convergence curve of the training process, and the network has basically converged at 200 epochs. (**a**) train loss, (**b**) train MIoU, (**c**) val MIoU.

Table 1. Hyper-parameter.

Type	Value
num of workers	8
batch size	6
optimizer	SGD
learning rate	0.01
learning policy	poly
step size	10,000

$$H(p,q) = -\sum_{i=1}^{n} p(x_i) log(q(x_i)) \quad (4)$$

The cross-entropy function is used to measure the difference between two probability distributions. For example, machine learning tasks represent the difference between the network output and the label.

4.5. Performance and Comparison

In the experiment, the quantitative analysis of the segmentation results of U-Net, FCN, and PSPNet, which are typical lightweight networks, and our method is conducted. The comparison results are shown in the tables, and the recovered images are shown in Figure 9.

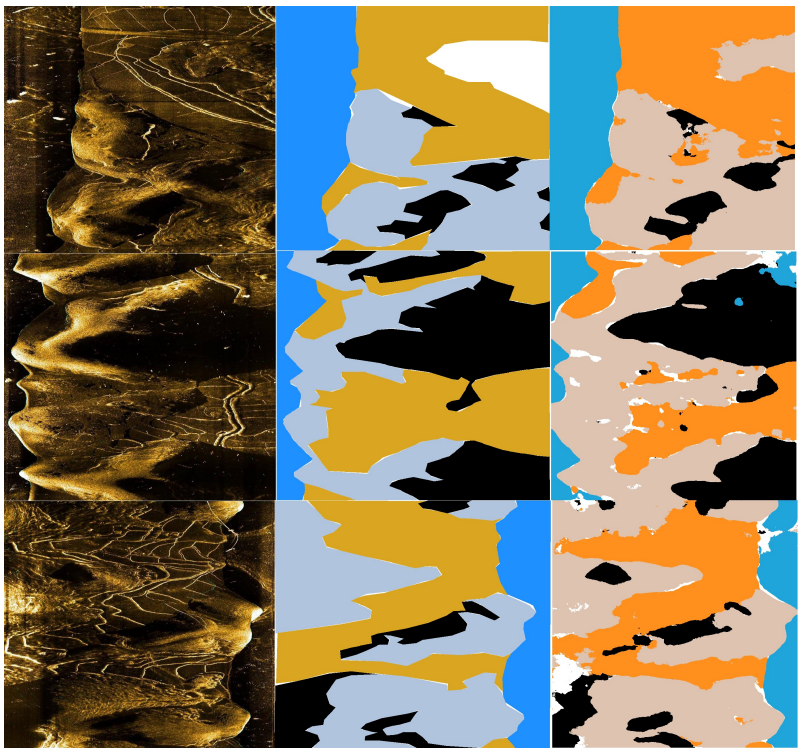


Figure 9. The segmentation results of the model output are shown in the figure. The original image, label, and output result are left to right. The colors in the picture are blue for water, gray for rocks, yellow for flat land, black for shadows, and white for fruitless areas (areas that are hard to distinguish).

Due to the small sample size, we used K-fold cross-validation on the dataset to calculate the model performance indicators. We set the value of K as 5, randomly divided all the data into five parts, and selected one of them as the validation set and the rest as the training set each time. Finally, the results obtained five times were averaged. The model indicators of five-fold cross-validation are shown in Table 2. The results shown in Table 3

show that the average OA and MIoU of our model in the dataset are 0.87159 and 0.67893, the highest of the four models. The total number of parameters is 21,340,813, which was above the average of the four models. The FLOPs are slightly higher because the currently used code and computing devices do not support DW convolution perfectly, and there is still room for further improvement.

Table 2. K-fold cross validation (K = 5).

K	OA	MIoU
1	0.869394	0.685063
2	0.856123	0.678424
3	0.854726	0.656946
4	0.884486	0.699488
5	0.856770	0.668848
avg	0.872299	0.677754

Table 3. Different model performance.

Model	OA	MIoU	Num of Para	FLOPs
FCN	0.864415	0.663187	18,643,845	212.4 G
U-Net	0.871427	0.674909	34,525,391	487.71 G
PSPNet	0.849124	0.651908	65,576,517	673.94 G
Ours	0.872299	0.677754	21,340,813	647.94 G

In order to test the effect of increasing the size of the convolution kernel, we carried out relevant comparative tests and adjusted the size of the convolution kernel from 3×3 to 11×11. The performance changes are shown in Table 4, and it can be found that the parameters currently used are the best ones.

Table 4. Model performance with different kernel size.

Size	OA	MIoU
3×3	0.864976	0.663328
5×5	0.862365	0.667896
7×7	0.872299	0.677754
9×9	0.866372	0.673241
11×11	0.862757	0.658241

5. Conclusions

This paper proposes a semantic segmentation model for side-scan sonar images based on the CNN network. The model uses a symmetric codec structure as the main body, adds a convolution kernel of different scales to extract multi-scale features, adds SE modules to focus on the weight of essential channels, and finally fuses at the output end. We verify the accuracy and reliability of the model on the self-collected sonar data and find that the model has a low computational cost and high portability. Our method achieves multiple classifications of side-scan sonar images at the semantic level. At the same time, most other researchers focus more on the recognition of objects with specific shapes or the simple binary classification of images. In addition, our model also has high portability. The large neural network model proposed by many researchers is inferior in real-time performance on AUV. After loading our model into the AUV control terminal, it can still complete the task and has low dependence on high-performance computers, which is also a significant advantage. In the future, we will consider further increasing the network depth and convolution kernel and find ways to make them effective in a small sample environment.

Author Contributions: D.Y. and F.Z. conceived the study and put forward the methodology. C.C. and C.W. performed the data collection and pre-processing. D.Y. carried out the software for the experiments and wrote the first draft of the manuscript. F.Z. and G.P. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (52171322), the National Key Research and Development Program (2020YFB1313200), and the Fundamental Research Funds for the Central Universities (D5000210944).

Acknowledgments: The authors would like to thank Songxiang Wang, Xijun Zhou, and Liyuan Chen et al. for their help during the experiment.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study.

References

1. Petrich, J.; Brown, M.F.; Pentzer, J.L.; Sustersic, J.P. Side scan sonar based self-localization for small autonomous underwater vehicles. *Ocean. Eng.* **2018**, *161*, 221–226. [CrossRef]
2. Reed, S.; Petillot, Y.; Bell, J. An automatic approach to the detection and extraction of mine features in sidescan sonar. *IEEE J. Ocean. Eng.* **2003**, *28*, 90–105. [CrossRef]
3. Acosta, G.G.; Villar, S.A. Accumulated ca–cfar process in 2-d for online object detection from sidescan sonar data. *IEEE J. Ocean. Eng.* **2015**, *40*, 558–569. [CrossRef]
4. Zhang, X.; Tan, C.; Ying, W. An imaging algorithm for multireceiver synthetic aperture sonar. *Remote Sens.* **2019**, *11*, 672. [CrossRef]
5. Wang, Z.; Guo, J.; Huang, W.; Zhang, S. Side-scan sonar image segmentation based on multi-channel fusion convolution neural networks. *IEEE Sens. J.* **2022**, *22*, 5911–5928. [CrossRef]
6. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
7. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2014.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
9. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 640–651.
10. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer International Publishing: Cham, Switzerland, 2015.
11. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2016.
12. Tian, C.; Zhang, X.; Lin, J.C.W.; Zuo, W.; Zhang, Y. Generative Adversarial Networks for Image Super-Resolution: A Survey. *arXiv* **2022**, arXiv:2204.13620.
13. Tian, C.; Yuan, Y.; Zhang, S.; Lin, C.W.; Zuo, W.; Zhang, D. Image Super-resolution with an Enhanced Group Convolutional Neural Network. *arXiv* **2022**, arXiv:2205.14548.
14. Tian, C.; Xu, Y.; Zuo, W.; Lin, C.W.; Zhang, D. Asymmetric CNN for image superresolution. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 3718–3730. [CrossRef]
15. Song, Y.; Zhu, Y.; Li, G.; Feng, C.; He, B.; Yan, T. Side scan sonar segmentation using deep convolutional neural network. In Proceedings of the OCEANS 2017, Anchorage, AK, USA, 18–21 September 2017.
16. Chen, J.; Summers, J.E. Deep convolutional neural networks for semi-supervised learning from synthetic aperture sonar (SAS) images. In Proceedings of the 173rd Meeting of Acoustical Society of America and 8th Forum Acusticum, Boston, MA, USA, 25–29 June 2017.
17. Wu, M.; Wang, Q.; Rigall, E.; Li, K.; Zhu, W.; He, B.; Yan, T. ECNet: Efficient convolutional networks for side scan sonar image segmentation. *Sensors* **2019**, *19*, 2009. [CrossRef] [PubMed]
18. Huo, G.; Wu, Z.; Li, J. Underwater object classification in sidescan sonar images using deep transfer learning and semisynthetic training data. *IEEE Access* **2020**, *8*, 47407–47418. [CrossRef]
19. Zhou, P.; Chen, G.; Wang, M.; Liu, X.; Chen, S.; Sun, R. Side-scan sonar image fusion based on sum-modified Laplacian energy filtering and improved dual-channel impulse neural network. *Appl. Sci.* **2020**, *10*, 1028. [CrossRef]
20. Połap, D.; Wawrzyniak, N.; Włodarczyk-Sielicka, M. Side-scan sonar analysis using roi analysis and deep neural networks. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–8. [CrossRef]
21. Zhu, P.; Isaacs, J.; Bo, F.; Ferrari, S. Deep learning feature extraction for target recognition and classification in underwater sonar images. In Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia, 12–15 December 2017.

22. Burguera, A.; Oliver, G. High-resolution underwater mapping using side-scan sonar. *PLoS ONE* **2016**, *11*, e0146396. [CrossRef] [PubMed]
23. Lecun, Y.; Bottou, L. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
24. Technicolor, T.; Related, S. Imagenet Classification with Deep Convolutional Neural Networks 2012. [50]. Available online: https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (accessed on 20 August 2022).
25. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
26. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Xiao, B. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef]
27. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for semantic segmentation. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021.
28. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2011–2023. [CrossRef]
29. Ding, X.; Zhang, X.; Zhou, Y.; Han, J.; Ding, G.; Sun, J. Scaling up your kernels to 31 $\times$ 31: Revisiting large kernel design in CNNs. *arXiv* **2022**, arXiv:2203.06717.
30. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.