Image and Vision Computing 93 (2020) 103824

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

An integrated ship segmentation method based on discriminator and extractor☆,☆☆

Wen Zhang [a], Xujie He [a,*], Wanyi Li [b], Zhi Zhang [a], Yongkang Luo [b], Li Su [a], Peng Wang [b]

[a] College of Automation, Harbin Engineering University of China, Harbin 150001, China
[b] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 11 October 2019
Accepted 5 November 2019
Available online 9 November 2019

Keywords:
Ship segmentation
Sea fog
Classification
Interference Factor Discriminator
Ship extractor

ABSTRACT

Ship segmentation is an important task in maritime surveillance systems. A great deal of research on image segmentation has been done in the past few years, but there appears to be some problems when directly utilizing them for ship segmentation under complex maritime background. The interference factors decreasing segmentation performance usually are from the peculiarity of complex maritime background, such as the existence of sea fog, large wakes and large waves. To deal with these interference factors, this paper presents an integrated ship segmentation method based on discriminator and extractor (ISDE). Different from traditional segmentation methods, our method consists of two components in light of the structure: Interference Factor Discriminator (IFD) and Ship Extractor (SE). SqueezeNet is employed for the implementation of IFD as the first step to make a judgment on what interference factors are contained in the input image. While DeepLabv3 + and improved DeepLabv3 + are employed for the implementation of SE as the second step to finally extract ships. We collect a ship segmentation dataset and conduct intensive experiments on it. The experimental results demonstrate that our method for ship segmentation outperforms state-of-the-art methods in terms of segmentation accuracy, especially for the images contain sea fog. Besides our method can run in real time as well.

1. Introduction

Recently, more attention is paid to the oceans, and maritime surveillance has gradually become an indispensable part of a country's security, representing the first line of defense against intruders, also used for military purposes. As an important segment of maritime surveillance, the result of ship segmentation will impact other maritime surveillance segments such as ship identification [1–3] and 3D reconstruction [4,5] on the oceans to some extent. In other words, once ship segmentation problem is worked out the subsequent interpretations can be clicked into place. Therefore, solving the problem of ship segmentation under complex maritime background is of vital importance.

However, according to the authors' literature survey, it was found that the vast majority of the published papers concentrated on ship segmentation are based on infrared images [6–14], and SAR images [15,16], while only a few papers are based on visible images [17,18]. Shortcomings of methods using infrared images include low contrast between target and background, low signal-to-noise ratio (SNR), and missing target details. When it comes to SAR images, there will appear some another shortcomings, containing massive speckle noise with a distribution of Gamma, laborious to differentiate tiny edges, low contrast and low SNR. When considering applying such images into ship segmentation task, there will occur other particular issues. As is known to all, visible images are better than infrared images and SAR images in SNR, contrast and detail presentation, it is also a fact that the visual effect of visible images bringing to the observers is much better than infrared and SAR ones. Moreover, visible images can be captured everywhere in daily life. These all make our research on ship segmentation based on visible images urgent and significant.

At this point, when it comes to image segmentation on visible images, it is known as a challenging branch of digital image processing, which aims at extracting regions of interest from images and has many potential applications within image compression, image retrieval, medical diagnosis, driverless car, etc. Up to now, generally used segmentation methods can be mainly categorized into five types: threshold-based segmentation [19], edge-based segmentation [20], region-based segmentation [21,22], superpixel-based segmentation [23,24] and relevant theory-based segmentation [25,26]. Owing to the prominent segmentation performance on all categories, neural

☆ This paper has been recommended for acceptance by S. Todorovic.
☆☆ This work was supported by National Key R&D Program Projects [grant number 2018YFB1601502].
* Corresponding author at: Room 4138, Building 61, College of Automation, Harbin Engineering University, NO.145, Nantong Avenue, Nangang District, Harbin City, Heilongjiang Province, China.
E-mail address: hexujie@hrbeu.edu.cn (X. He).

https://doi.org/10.1016/j.imavis.2019.11.002
0262-8856/

network-based segmentation methods stood out in a number of relevant theory-based ones. Consequently, numerous segmentation methods based on neural network have been put forward to the literature in recent years. Long et al. [27] first brought the neural network into the field of image segmentation, showing the powerful feature extraction ability to the world and realizing an end-to-end segmentation model in a real sense. Liu et al. [28] brought global information of images. Noh et al. [29] improved upsample processing to a fully deconvolutional network. Ronneberger et al. [30] used concatenation operation to form thicker features. Lin et al. [31] combined low-resolution features with high-resolution features at different scale levels. Zhao et al. [32] first proposed pyramid pooling module which was used to aggregate context information from different regions to improve the ability of obtaining global information. Chen et al. [33] provided the first use of dilated convolution for feature extraction to enlarge receptive fields. Based on ([33,34] aggregated multi-scale, and proposed using Atrous Spatial Pyramid Pooling (ASPP). Chen et al. [35] utilized a more general framework to improve ASPP and removed the post-processing process of Conditional Random Fields. Chen et al. [36] proposed a new structure called encoder-decoder for segmentation. As a result, the continuous improvement of segmentation methods made the segmentation accuracy increase step by step.

Despite extensive researches on visible image segmentation, there appears to be some new problems when directly implementing state-of-the-art neural network-based segmentation methods (NNSM) for ship segmentation under complex maritime background. Some are given here. (i) For the sea fog that always appears above the sea surface, the segmentation results in this case appear to be unsatisfactory, and sometimes a serious segmentation failure occurs, resulting in a significant reduction in segmentation accuracy, see Fig. 1(a). (ii) The currently ideal segmentation method is still not able to suppress large wakes and large waves when extracting ships, see Fig. 1(b). (iii) Where there is a reflection on sea surface, the segmentation results will decrease, see Fig. 1(c). (iv) For the case of dusk, the segmentation results will decrease, too. See Fig. 1(d). It can be observed that NNSM is not completely suitable for complex maritime background. But in our method, we make NNSM applicable to complex maritime background and manage not to lose much time as a counterpart.

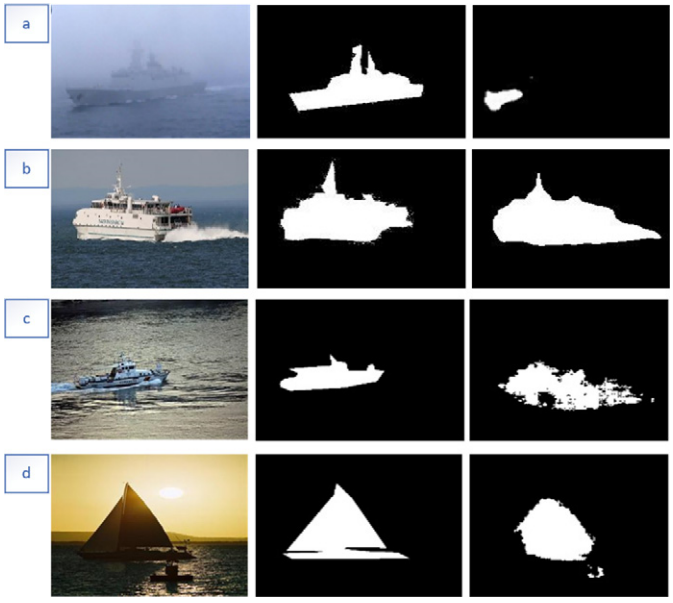


Fig. 1. Problems encountered in ship segmentation under maritime background. The first column displays original images, the second column displays Ground Truths, the third column displays the segmentation results by NNSM. Row (a) is an image under sea fog background, row (b) is an image with large waves, row (c) is an image with reflection, row (d) is an image under dusk background. Best view in color.

In order to solve the problem of segmentation accuracy decreasing caused by interference factors under complex maritime background, we present an integrated ship segmentation method based on discriminator and extractor in this paper. Specifically, our method is composed of two components: Interference Factor Discriminator (IFD) and Ship Extractor (SE). We first employ IFD as a classifier to make a judgment on what kind of interference factors are contained in input images, second we employ SE to extract ships from input images. For the implementation of IFD and SE, neural network-based methods which are SqueezeNet [37] and DeepLabv3 + [36] are finally adopted on account of the complexity and diversity of our specific scenes. Moreover, as part of this procedure, we put forward a new feature extraction network called DenseNet69 on the basis of [38] and embed it into the DCNN partition of DeepLabv3 + as improved DeepLabv3 + to improve its ability of ship extraction.

The contribution of our work can be summarized in three main points.

Firstly, an integrated ship segmentation method based on discriminator and extractor is proposed. The proposed method can provide better ship segmentation results on visible images, especially for the images contain sea fog, which makes up for the vacancy of ship segmentation on visible images.

Secondly, in Ship Extractor part, we proposed a new feature extraction network called DenseNet69 and embed it into the DCNN partition of DeepLabv3 + as improved DeepLabv3 + to improve its ability of ship extraction.

Thirdly, we collected and manually labeled a visible image dataset for ship segmentation. It will be public to the community. Intensive experiments are conducted on our collected dataset. Experimental results demonstrate that our method is more effective for complex maritime background in comparison with DeepLabv3 + which had the best performance in many scenes in the literature before.

In the remainder, we describe the implementation details of our method in Section 2, and some experimental results and comparison will be unfolded in Section 3. We also draw our discussion and conclusion in Sections 4 and 5.

2. Proposed ship segmentation method

As described in previous section, it is found that the existence of sea fog has a serious impact on segmentation results, when using universally applicable segmentation method [36] for ship segmentation task, see Fig. 1(a). In order to improve ship segmentation accuracy, we take sea fog as an example and propose an integrated ship segmentation method. The proposed method is composed of two components: (i) Interference Factor Discriminator (IFD) and (ii) Ship Extractor (SE). In order to make a judgment on what interference factors are contained in the input image, we employ an Interference Factor Discriminator (IFD) based on a classification network SqueezeNet [37]. After going through the classification network, the input image is predicted as a [0,1] probability value indicating whether it is with sea fog or not. In the second component, we employ two Ship Extractors (SE), which are specifically trained for normal maritime background and sea fog background based on DeepLabv3 + [36] and improved DeepLabv3 + to extract ships, more specifically, the input image will be sent into one of the two Ship Extractors according to the probability value attained from IFD.

Fig. 2 gives the procedure of the proposed ship segmentation method, containing Interference Factor Discriminator, Ship Extractor under sea fog background and Ship Extractor under normal maritime background based on neural network. In the following part, we are going to make a detailed explanation for each component.

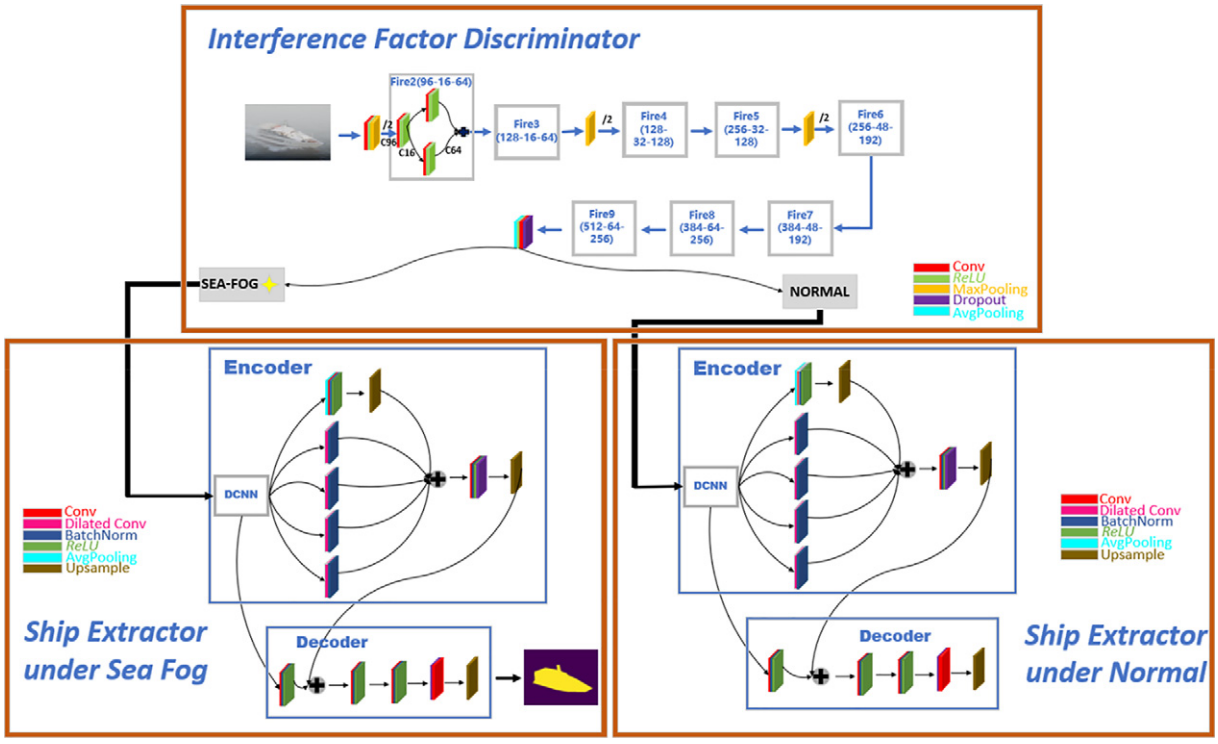


Fig. 2. The overall framework of ISDE. The input image first goes through the Interference Factor Discriminator to attain a judging result on if it is with sea fog nor not, then goes to one of the two Ship Extractors according to the judging result to attain final segmentation result. Best view in color.

2.1. Interference Factor Discriminator (IFD)

As shown in Fig. 2, an Interference Factor Discriminator is required before Ship Extractors. According to a fact that neural networks can approximate functions of any shapes, we finally regarded IFD as a classification problem in the field of deep learning. The reasons that we accomplish our classification task by neural network are given here. (i) It is a common sense in classification field that methods based on neural network are better than those based on traditional methods in many occasions, especially encountering with complex scenes. (ii) The discriminating results are more reliable than those from traditional methods. (iii) More robust. In our method, SqueezeNet [37] was finally adopted as the implementation of IFD. The reasons for adopting SqueezeNet are given here. (i) More effective training; (ii) More lightweight; (iii) Less saving memory. When coming to SqueezeNet, there are three main strategies used in it: (i) Use 1×1 filters instead of 3×3 filters to have few parameters; (ii) Use squeeze layers to decrease input channels; (iii) Delay downsampling to have large activation maps. The first two strategies aimed to judiciously reduce the parameters to make itself more lightweight while preserving accuracy, and the third strategy was derived from the truth that using large strides early could make the layers later have smaller activation maps. Considering the final size of our method and the running speed, we finally choose the one having no bypass among three existing SqueezeNet versions, in other words, the SqueezeNet we used is the smallest version in [37].

Fig. 3 shows the whole structure of IFD which is on the basis of SqueezeNet. As can be observed, there are 8 Fire Modules in total. The adjustment of output categories, which are SEA-FOG class and NORMAL class are performed to satisfy the requirements of complex maritime background.

Next are some details about Fire Module. First we illustrate the structure of a Fire Module and then make an explanation on how to compute filter number in each Fire Module.

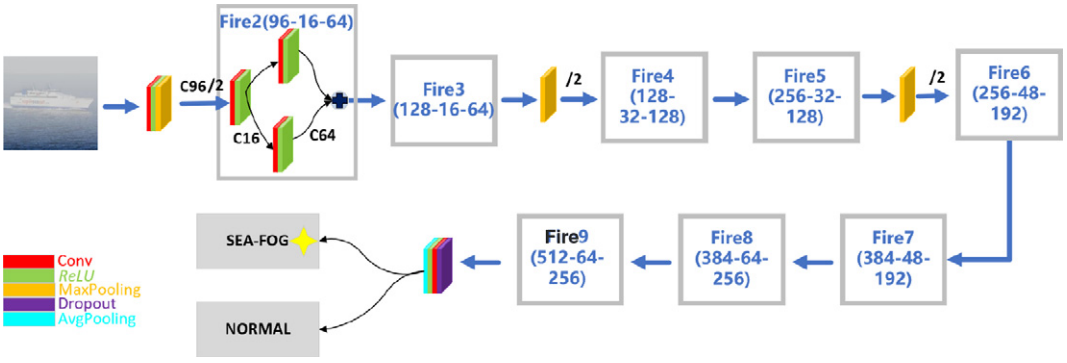


Fig. 3. Structure of IFD. 8 Fire Modules and two classes in total. After going through the whole IFD, the input image is equally converted to a 0–1 probability value. Best view in color.

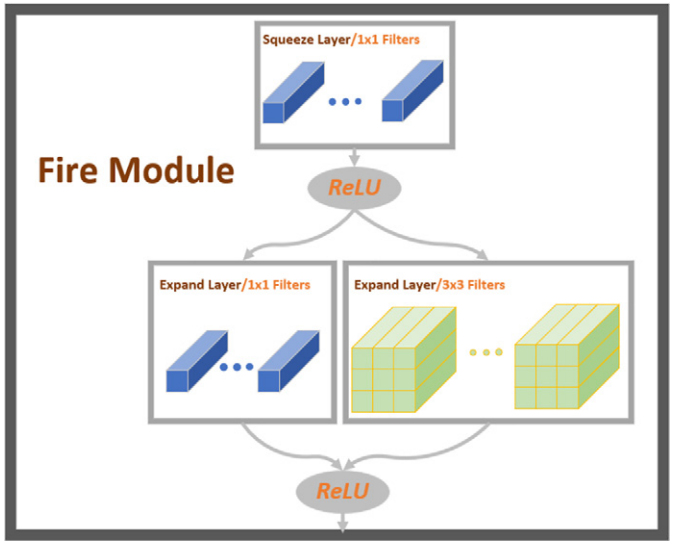


Fig. 4. Fire Module structure. A Fire Module is composed of a Squeeze Layer and an Expand Layer. A Squeeze Layer is composed of a few 1 × 1 filters. An Expand Layer is composed of a few 1 × 1 and 3 × 3 filters. Best view in color.

2.1.1. Fire module

As can be seen in Fig. 3, the Fire Module is a basic structure of SqueezeNet. The aim of using Fire Modules is to get in line with strategy (i) to make the network more lightweight, while preserving final accuracy. There are two main layers in terms of a Fire Module structure: Squeeze Layer and Expand Layer. Fig. 4 shows the structure of a Fire Module, it can be noted that images will go through four processes corresponding the sequence: Squeeze Layer → *ReLU* → Expand Layer → *ReLU*. In other words, for the input feature maps $\boldsymbol{X}$, the output feature maps X can be defined as:

$$\boldsymbol{X}' = ReLU\circ\{E_1[\,ReLU\circ S(\boldsymbol{X})]\oplus E_3[\,ReLU\circ S(\boldsymbol{X})]\} \tag{1}$$

where S denotes Squeeze mapping and E denotes Expand mapping, $\oplus$ denotes concatenating operation and $ReLU\circ$ denotes Rectified Linear Unit function. From Eq. (1), we can clearly figure out that input feature maps first enter a Squeeze Layer, then the output of the Squeeze Layer enters *ReLU*, next the output of *ReLU* enters an Expand Layer which has 1 × 1 and 3 × 3 filters separately, finally concatenate the output of Expand Layer and enter *ReLU* to get the final output of a Fire Module.

(1) The usage of a Squeeze Layer and an Expand Layer

As shown in Fig. 4, an Expand Layer composed of 1 × 1 filters and 3 × 3 filters is closely following Squeeze Layer. The number of parameters from 3 × 3 filters is calculated as:

$$num\_parameter = 3 \times 3 \times i_c \times o_c \tag{2}$$

where i_c denotes the input channel number of 3 × 3 filters and o_c denotes the output channel number of 3 × 3 filters. In order to make the network more lightweight, using a Squeeze Layer before an Expand Layer can squeeze the input channel number of an Expand Layer to a certain range, in this way, the total number of parameters is reduced.

The using of a Squeeze Layer makes the number of parameters reduced drastically, it's an excellent choice for structuring a lightweight network. But here comes a disadvantage meantime, the existence of a Squeeze Layer can also reduce the number of feature maps drastically. In order to avoid degraded performance of feature extraction, an Expand Layer has been employed to expand output channels to make a compromise between network size and feature extraction ability.

(2) Determine filter number

As shown in Fig. 4, there are some bold dots in different colors meaning the filter duplicating operation. There are three newly proposed hyperparameters in a Fire Module that are s_1, e_1, e_3, respectively representing 1 × 1 filter number of a Squeeze Layer, 1 × 1 filter number of an Expand Layer and 3 × 3 filter number of an Expand Layer. But how to precisely know the filter number of each Fire Module? First, we define e^i as the total filter number of an Expand Layer in the ith Fire Module, in other words, same as the calculation described below:

$$e^i = e_1^i + e_3^i \tag{3}$$

Then e^i can be calculated as follows:

$$e^i = e^2 + \lambda \times \left[\frac{i-2}{\beta}\right] \tag{4}$$

where e^2, λ, β are three constants, in our method $e^2 = 128$, $\lambda = 128$, $\beta = 2$. $[\cdot]$ denotes integer function. Pay attention to $i \geq 2$. For example, the filter number of the Expand Layer in the 5th Fire Module is $e^5 = 128 + 128 \times [\frac{5-2}{2}] = 128 + 128 \times [1.5] = 128 + 128 \times 1 = 256$. Second, how to attain the specific number of 1 × 1 filters and 3 × 3 filters in each Expand Layer? We define pce_3 as the percentage of 3 × 3 filters of each Expand Layer. In our method $pce_3 = 0.5$. For example, given the $e^5 = 256$, we can attain the results that are $e_1^5 = 128$, $e_3^5 = 128$. Finally, one value haven't been determined yet, that is the 1 × 1 filter number in each Squeeze Layer which we defined it ass_1 previously. As to this question, s_1 can be calculated by a coefficient defined as *SR* (squeeze ratio), in other words, $s_1^i = SR \times e^i$. For example, in our method $SR = 0.125$, given the $e^5 = 256$, $s_1^5 = 0.125 \times 256 = 32$.

At this point, we can precisely know the filter number of each Fire Module used in our IFD. Please relook at Fig. 3. For Fire Module 2, you may bewilder about (**96-16-64**), while **96** denotes the input channel number of the Fire Module 2, and after going through the first Squeeze Layer, the number of channels is squeezed to **16**, next enter the first Expand Layer, the number of output channels is expanded to **64**, that's what (**96-16-64**) denotes. Of course, the rest Fire Modules can be deduced by analogy.

2.2. Ship Extractor (SE)

With respect to ship extraction problem, there have appeared numerous traditional image segmentation methods which are ideal in speed. However, considering the peculiarity of the segmentation occasion, when encountering with complex scenes, traditional segmentation methods seem to be incapable. However, the successful leading in of neural networks in image segmentation lead us to put our concentration on deep learning based methods. Considering the accuracy requirements, the Ship Extractor component of our method finally adopted DeepLabv3 + [36] because of its best comprehensive performance in many scenes proved in [36]. Meanwhile, an improved DeepLabv3 + is proposed to attain better segmentation results.

In DeepLabv3 +, the main work summarized is using the previous DeepLabv3 as the encoder module and proposing the decoder module, which is called encoder-decoder structure for image segmentation. Fig. 5 shows the overall structure of DeepLabv3 +. It can be noted that, for an input image, features are preliminary extracted by DCNN, the extracted features will be convoluted by dilated convolution with different dilated rates as shown in Fig. 5 with arrowheads in arctic blue (generally using 4 different dilated rates), then concatenate the feature maps from different receptive fields with the features maps which is directly upsampled from the size of 1 × 1 as shown in Fig. 5 with an arrowhead in orange, next execute the composite operations

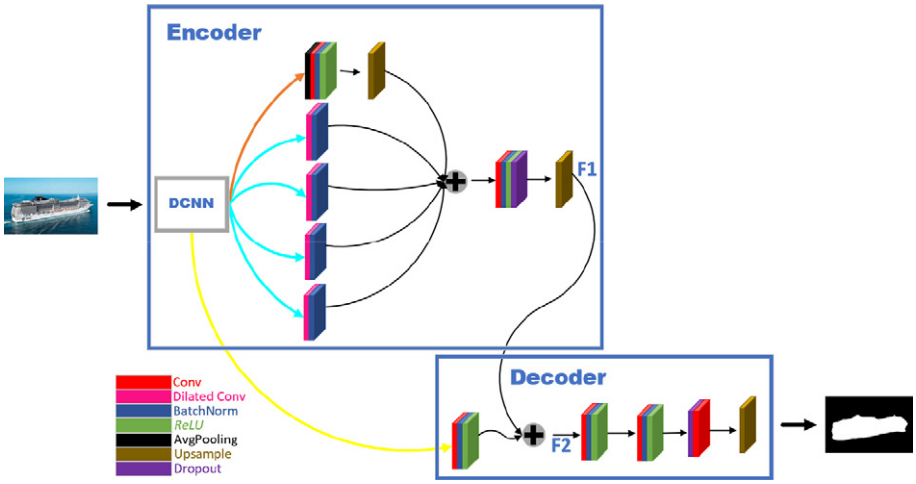


Fig. 5. DeepLabv3 + structure. Best view in color.

of convolution, batch normalization, *ReLU*, dropout and finally upsample them all by a factor of 4 to form new feature maps, we name them F1 here, this segment is also called as Encoder. At the same time, when preliminary extracting features, the feature maps of low level that are 4 times smaller than the input image have also been extracted as shown in Fig. 5 with an arrowhead in yellow, and after executing the convolution operation to squeeze output channels of low level feature maps to a small range, it all will be concatenated with F1 to form new feature maps and we name them F2 here, and after going through a composite operations of convolution, batch normalization, *ReLU* and dropout, the feature maps all will be upsampled by a factor of 4 to reach the size same as input image, this segment is named as Decoder. This is how DeepLabv3 + works.

In the following part, we are going to make an explanation on dilated convolution which is specifically introduced in DeepLab, and then elaborate our improved DeepLabv3 + .

2.2.1. Dilated convolution

When coming to the series of DeepLab, one of the operations that is necessary to be unfolded is dilated convolution. Dilated convolution is also widely called as atrous convolution or convolution with holes. To be exact, the standard convolution process takes a continuous region from the image for convolution, while dilated convolution is quite different. When processing dilated convolution, pixels from the feature maps can be taken every specific rows and columns, which is determined by a new hyperparameter called dilated rate. In order to better understand the dilated convolution and compare it with the standard one, we have drawn it into a figure, see Fig. 6. As depicted in Fig. 6, the process of dilated convolution can be described into a formula:

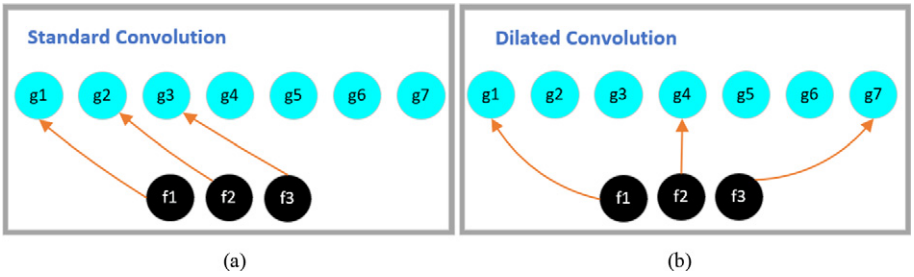
$$output = \sum_{i=1}^{3} f_i \times g_{i+[i-1]\times[\beta-1]} \tag{5}$$

where f_i denotes the index of the filter, g_i denotes a pixel, and $\beta = 3$ denotes dilated rate. From Fig. 6(b), it can be clearly observed that using dilated convolution makes receptive fields enlarged compared with standard convolution, which is of great use and importance for segmentation task.

2.2.2. Improved DeepLabv3 +

In [38], it has been proved a prominent performance of DenseNet on classification problem. Meanwhile, in [39], it has been proved a prominent performance under urban scenes when utilizing improved DenseNet for image segmentation. As a result, aiming at figuring out if DenseNet is adequate to maritime background, we first attempt to replace ResNet101 used in DeepLabv3 + with DenseNet121. Inspired by ResNet101, in order to make DenseNet121 compatible with DeepLabv3 +, two strategies have been taken. (i) The low level feature maps which are 4 times smaller than the input image are drawn forth. (ii) Add another Transition Layer at the end of DenseNet121 to make the feature maps 16 times smaller than the input image. According to [38], we therefore name it as DenseNet122. Unfortunately, when embedding DenseNet122 into DCNN module, here comes another problem, which is exquisitely eating up memory and resulting in a pseudomorph of computer crash from time to time. To reduce the memory requirement, another two strategies have been taken. (iii) Reduce the number of total layers drastically. (iv) Use expansion operation instead of compression operation.

Fig. 7 shows the whole structure of improved DeepLabv3 +, as can be seen, we replace the DCNN with newly proposed DenseNet69. Please pay attention to the output of DenseNet69, we add another composite operation (convolution, batch normalization, *ReLU*) to squeeze the output layer number. More details about DenseNet69 will be unfolded in the following part. In the following part, we first make an explanation on DenseNet69, and then we elaborate the structure of a Dense Block, finally introduce Tran-Expansion Layer.



(a) (b)

Fig. 6. Comparison of two convolutions. (a) indicates Standard Convolution, (b) indicates Dilated Convolution. Note we use a single dimension to make an illustration, same as two or three dimensional images. Best view in color.

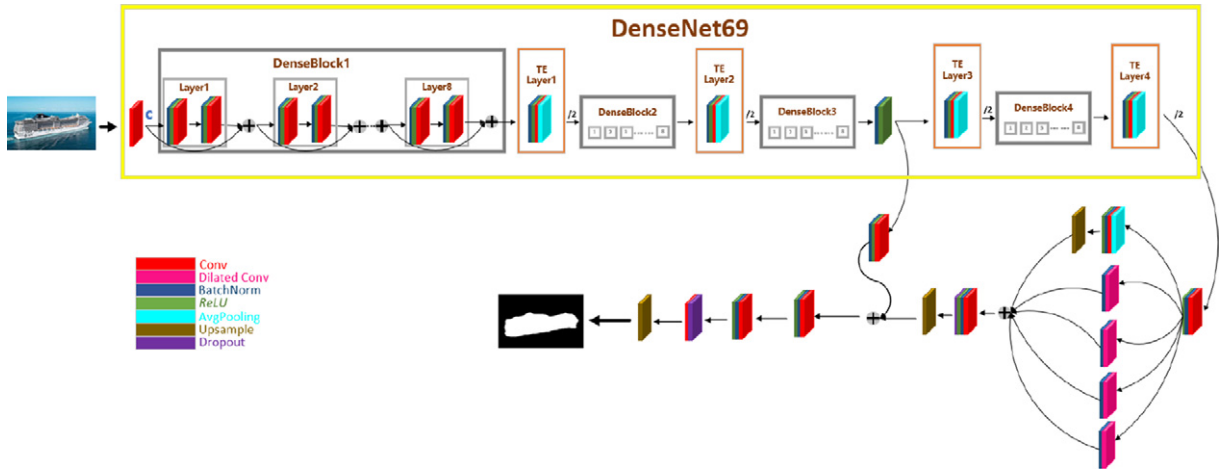


Fig. 7. The whole structure of improved DeepLabv3+. Best view in color.

(1) DenseNet69 structure

In order to make our DenseNet compatible with DeepLabv3 + structure, we first draw forth low level features after Dense Block3 and add TE Layer4 in the end to get in line with strategy (i) and (ii). However, there still exists a problem of memory eating up and computer crash. Aiming at working out these two problems, we utilize 4 Dense Blocks which have 8 Dense Layers in it to drastically reduce total layer number to be consistent with strategy (iii). And utilize 4 Tran-Expansion Layers instead of Transition Layer in total to attain smaller activation maps. Finally, we can attain the total layer number of our new DenseNet is $\text{num\_layer} = 4_{\text{Dense\_Block}} \times 8_{\text{Dense\_Lyer}} \times 2_{\text{Times}} + 4_{\text{TE\_Layer}} + 1 = 69$, therefore we name it as DenseNet69, as shown in Fig. 7.

(2) Dense Block

It can be noted there are 4 Dense Blocks in Fig. 7. The Dense Block is a core structure of the DenseNet which aims at reusing the feature maps long before. Fig. 8 shows the structure of a Dense Block. It can be noted that, (i) a Dense Block contains a certain number of Dense Layers in it, (ii) the input of each Dense Layer is an aggregation of all outputs from previous Dense Layers and the input of the Dense Block. Moreover, different from ResNet, the concatenation operation is adopted to aggregate features in a Dense Block rather than an element-wise addition which can be defined as:

$$f_i = *\langle f_{i-1} \mathrm{U} f_{i-2} \mathrm{U} f_{i-3} \cdots \mathrm{U} f_1 \mathrm{U} f_{in\_d} \rangle \tag{6}$$

where $*\langle \mathrm{U} \rangle$ denotes concatenation operation, f_i denotes the output feature maps of the ith Dense Layer, f_{in_d} denotes the input feature maps of the Dense Block as shown in Fig. 8 with an arrowhead in orange. Meanwhile, it can also be observed each Dense Layer has two same parts composed of convolution, batch normalization and *ReLU*, but the use of them is quite different. Please relook at Eq. (6), with the number of Dense Layer increasing, the input channel number of next Dense Layer would be drastically large. We define num_{d_i} as the input channel number of the dth Dense Layer, define num_{d_o}as the output channel number of the dth Dense Layer and define $num_{f\_in\_d}$ as the input channel number of the dth Dense Block. In other words, in our method $num_{d\_o} = 12$ means the output channel number of each Dense Layer is 12, num_{d_o} is also called as growth rate. For the 5th Dense Layer, the input channel number can thereby be computed as $num_{5\_i} = 12 \times 4 + num_{f\_in\_d}$. Just think about it, if $num_{f\_in\_d}$ is a quite large number, the result of num_{5_i} will be quite large as well, this is adverse for running in real time. In order to solve this problem to reduce computation load, there finally appears a bottleneck which is the first part in each Dense Layer as shown in Fig. 8. The exact use of a bottleneck is to squeeze the input channels of each Dense Layer to a small range to finally improve computational efficiency. In our method we squeeze input channels to 48 channels. However, the second part in a Dense Layer is a standard composite of convolution, batch normalization and *ReLU* mainly used for feature extraction.

(3) Tran-Expansion Layer (TE Layer)

Owing to the drastically reduction of feature maps depicted as strategy (iii), we assume it may impact the feature extraction ability, so we

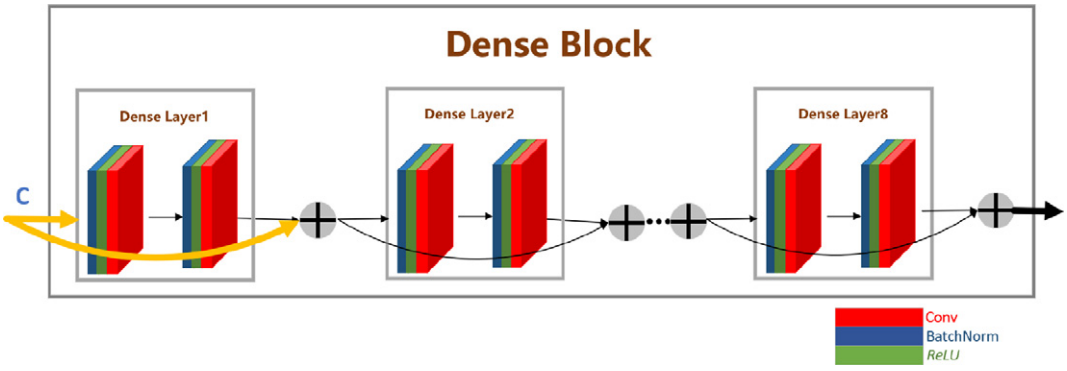


Fig. 8. Dense Block structure. Note there exists 8 Dense Layer in a Dense Block and two same composite operations in each Dense layer. Best view in color.

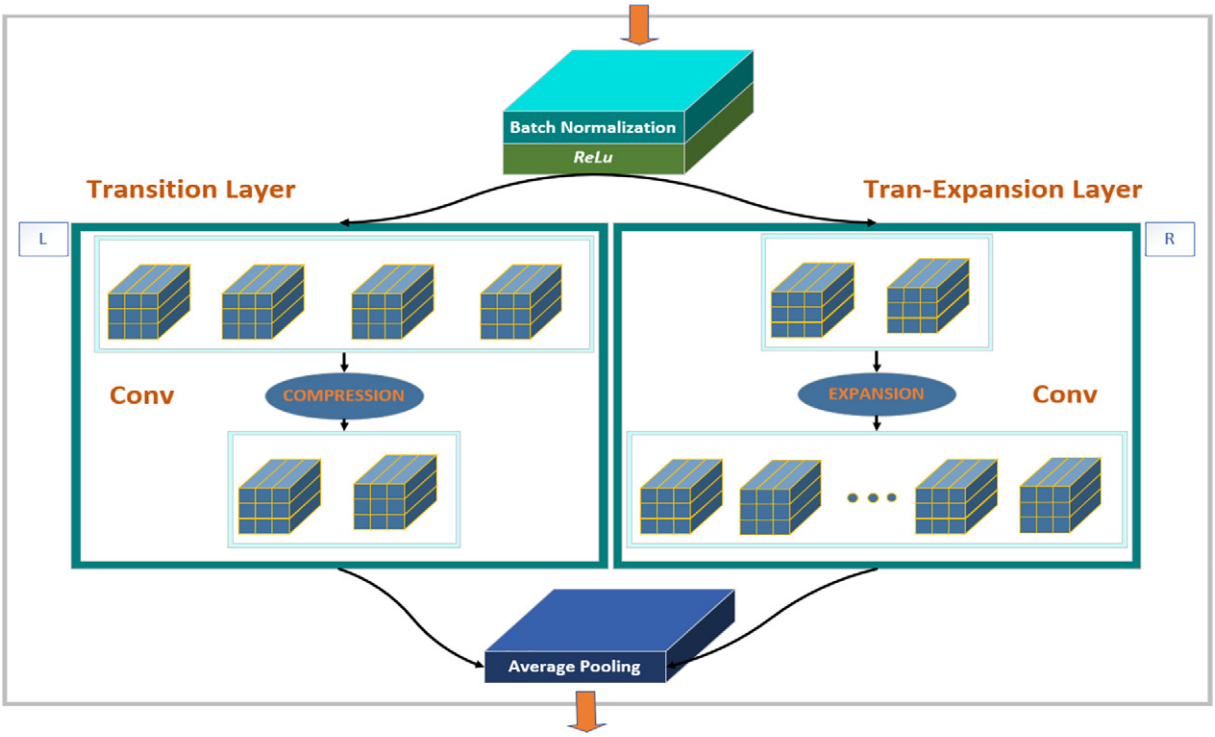


Fig. 9. Comparison of Transition Layer and Tran-Expansion Layer. The left side is Transition Layer proposed before, and the right side is newly proposed Tran-Expansion Layer. Best view in color.

attempt to locate a right place to expand feature maps back to a certain range. In [38], there is a Transition Layer closely following each Dense Block, and the aim of using a Transition Layer is to compress the input channels to a small range due to its thick structure. But in our version, we wish to expand the number of feature maps as a compromise to regain feature extraction ability rather than compress them to a smaller range. Consequently, what we exactly do for strategy (iv) is taking out the Transition Layer and embed Tran-Expansion Layer instead. Fig. 9 on the left shows the process of a Transition Layer proposed in [38], Fig. 9 on the right shows the process of a Tran-Expansion Layer we proposed. As can be clearly observed, in original paper [38], feature maps are compressed to a small range by a factor within 0~1. On the contrary, in our method, feature maps here are expanded to a large range by a factor of ϑ which is greater than 1. In other words, for the input channels having ξ layers, the number of output channels would be expanded to $\vartheta \times \xi$.

3. Experiment

3.1. Dataset and evaluation metrics

3.1.1. Dataset

Since many of the image processing techniques applied in maritime background are used in military and commercial occasions, there is not too much dataset and codes in the community. We collected and manually labeled a visible image dataset for ship classification and ship segmentation task. It will be available at MariShipSeg-HEU. In our collected dataset, there are 3560 maritime images in total including training and testing sets, 10% of images are from the Singapore Maritime Dataset (SMD) [40], 25% of images are from the Maritime Detection, Classification, and Tracking dataset (MarDCT) [41], and the rest are from the Internet. We name our collected ship dataset as MariShipSeg-HEU. Since our method has two components which are IFD and SE needing to be trained separately, the training dataset has to be segmented into two parts. It should be noted that we only took sea fog background as an example to conduct our experiments, because sea fog background has been found have the greatest impact on ship segmentation results.

(1) Dataset for IFD training

Owing to the fact that the IFD part of our method only needs to make a judgment on whether the input image contains sea fog or not, we need a certain number of sea fog images and a certain number of normal maritime images to train IFD. However, the number of sea fog images is relatively small. We tried to collect sea fog images from the Internet, but the amount was still much less than expected. With the help of the methods [42,43], we manually added sea fog into normal maritime images. Finally, we added sea fog into 745 maritime images, see Fig. 10 for some examples. At last, 1669 images were used to train IFD (745 images contain sea fog).

(2) Dataset for SE training

Same as IFD, we only need to train two models which are sea fog model and normal maritime model. As a result, 972 images were used to train SE under normal maritime background and 745 images were used to train SE under sea fog background.

All training and testing images have been manually labeled, our dataset will be public to the community later. Anyone who would like to use can download them from MariShipSeg-HEU. Some details about MariShipSeg-HEU are shown in Table 1.

3.1.2. Evaluation metric

Four widely accepted evaluation metrics are adopted for the final evaluation, they are Pixel Accuracy (PA), Mean Pixel Accuracy (MPA), Mean Intersection over Union (MIoU), and Frequency Weighted Intersection over Union (FWIoU), in addition, we add an evaluation metric named Average to reveal comprehensive segmentation performance. Specific calculation methods are as described below:

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \tag{7}$$

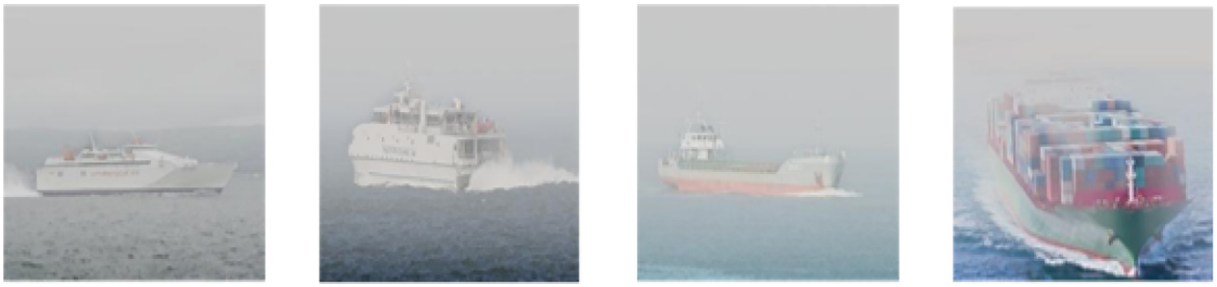


Fig. 10. Sea fog image. Sea fog is manually added into maritime images with related methods. Best view in color.

Table 1
Details of MariShipSeg-HEU. Note that NMB indicates normal maritime background, SFB indicates sea fog background. 3650 images in total for classification and segmentation task. The last column indicates the number of each set. For example, 1669 indicates the number of the training set used for IFD or Classification task.

Name	Total number	Application		Number of images
MariShipSeg-HEU	3650	IFD/classification	Training set	1669
			Testing set	174
		SE/segmentation	Training set for NMB	972
			Training set for SFB	745
			Testing set	174

$$MPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}} \tag{8}$$

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \tag{9}$$

$$FWIoU = \frac{1}{\sum_{i=0}^{k}\sum_{j=0}^{k}p_{ij}}\sum_{i=0}^{k}\frac{p_{ii}\sum_{j=0}^{k}p_{ij}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \tag{10}$$

$$FWIoU = \frac{PA + MPA + MIoU + FWIoU}{4} \tag{11}$$

where p_{ii} denotes the number of pixels belonging to category i and predicted to be category i,p_{ij} denotes the number of pixels belonging to category i but predicted to be category j. Obviously, the higher these five evaluation metrics are, the better segmentation performance is.

3.2. Implementation details

3.2.1. Experimental configuration

The configuration of our experiment is listed in Table 2.

3.2.2. Training details

In our experiments, we use two segmentation networks which are original DeepLabv3+ and improved DeepLabv3+. Both of these two networks were initialized using [44] and trained with Adaptive Moment Estimate (Adam) [45] for 900 epochs. We regularized two networks with a weight decay of 1e-4 and set the initial learning rate to 0.01, and changed learning rate to 0.008 at 300 epochs, 0.005 at 600 epochs and 0.002 at 800 epochs. Considering the GPU memory restriction, we used a mini-batch size of 20 for the original DeepLabv3+ for both training and testing, as for improved DeepLabv3+, owing to its particular structure, we used a mini-batch size of 5 for both training and testing. All input images were resized to 160 × 160 to reduce computation load. Meanwhile, as for training improved DeepLabv3+, we employed checkpoint function from PyTorch to optimize the memory usage, but this would bring 10%~20% extra processing time.

Table 2
Experimental configuration.

Experimental configuration
CPU: AMD Ryzen 5 1600 six core
GPU:Nvidia Geforce GTX 1650
Video Memory: 4G
Operating System: Windows 10 Professional 64-bit (DirectX 12)
Deep Learning Framework: PyTorch
CUDA:NVIDIA CUDA 10.2.120

3.3. Quantitative evaluation

In order to seek out the optimal network parameters for our task, we selected different dilated coefficients and dilated branches to train SE, and attained the corresponding testing results as shown in Tables 3, 4, 5, 6.

Table 3
Testing results of DeepLabv3+/R. Note that R indicates embedding ResNet101 into DCNN. B = 4 indicates the number of dilated branch is 4. Best results according to Avg are reported in bold (similarly hereinafter).

DeepLabv3+/R B = 4	Rates1 [1, 2, 4, 8]	Rates2 [2, 4, 8, 16]	Rates3 [1, 6, 12, 18]	Rates4 [6, 12, 18, 24]
PA (%)	93.05	92.53	92.85	92.86
MPA (%)	87.02	86.09	85.53	86.38
MIoU (%)	77.64	76.60	76.66	77.54
FWIoU (%)	88.11	87.32	87.80	87.85
Avg (%)	**86.45**	85.64	85.71	86.16

Table 4
Testing results of I-DeepLabv3+. Note that I-DeepLabv3+ indicates improved DeepLabv3+ (similarly hereinafter).

I-DeepLabv3+ B = 4	Rates1 [1, 2, 4, 8]	Rates2 [2, 4, 8, 16]	Rates3 [1, 6, 12, 18]	Rates4 [6, 12, 18, 24]
PA (%)	93.75	93.25	93.45	93.18
MPA (%)	88.16	88.18	87.58	86.78
MIoU (%)	79.33	78.32	78.49	77.65
FWIoU (%)	89.27	88.63	88.88	88.41
Avg (%)	87.63	87.10	87.10	86.51

Table 5
Testing results of DeepLabv3+ using 5 dilated branches.

DeepLabv3+/R B = 5	Rates5 [1, 2, 4, 8, 16]	Rates6 [2, 4, 8, 16, 24]	Rates7 [1, 4, 8, 16, 32]
PA (%)	92.50	92.91	92.66
MPA (%)	86.09	87.15	86.80
MIoU (%)	76.78	77.53	76.85
FWIoU (%)	87.42	88.07	87.69
Avg (%)	85.70	86.41	86.00

Table 6
Testing results of I-DeepLabv3+ using 5 dilated branches.

I-DeepLabv3+ B = 5	Rates5 [1, 2, 4, 8, 16]	Rates6 [2, 4, 8, 16, 24]	Rates7 [1, 4, 8, 16, 32]
PA (%)	93.95	93.54	93.40
MPA (%)	88.26	87.56	88.30
MIoU (%)	79.88	79.10	78.89
FWIoU (%)	89.64	89.04	88.87
Avg (%)	**87.93**	87.31	87.36

In order to further observe the experimental results, we drew the experimental results into line graphs, see Fig. 11. From the line graphs, the trend of experimental results with the network parameters changing can be clearly observed. (i) From the Tables 3, 4, 5, 6, it can be noted that, if utilize evaluation metric Avg as a comprehensive reflection of PA, MPA, MIoU and FWIoU, the best performance are DeepLabv3+ (86.45%) and I-DeepLabv3 + (87.93%) as listed in Tables 3 and 6 reported in bold, in other words, the ideal dilated parameters of our new dataset are Rates1 for DeepLabv3+, Rates5 for Improved DeepLabv3+ respectively. (ii) From the Fig. 11, it can be noted that, the testing results of PA, MIoU, FWIoU, Avg from improved DeepLabv3+ are all higher than those from original DeepLabv3 + .

At this point, the optimal dilated coefficients and dilated branch number have been affirmed, which are Rates1 for DeepLabv3+, Rates5 for improved DeepLabv3+ respectively. Then we employed Rates1 for DeepLabv3+ and Rates5 for I-DeepLabv3+ to train segmentation model under sea fog background and segmentation model under normal maritime background, and aggregated IFD altogether to attain final testing results. Owing to the result of IFD is a probability value, we finally set a threshold Γ to make the result of IFD specific, in other words, if the result of IFD is greater than Γ or equal, the input image can be treated as a normal maritime image, if the result of IFD is less than Γ, the input image can be treated as a sea fog image. In our method, for the sake of attaining best segmentation result, we employed 15 different Γ, and testing results are shown in Table 7. In Table 7, we just employ MIoU and Avg to reveal performance because MIoU is the most representative in image segmentation and Avg can comprehensively reveal these four evaluation metrics. Meanwhile, in order to clearly observe its changing tendency, we drew the results in Table 7 to a line graph as shown in Fig. 12, it can be noted that, with Γ increasing, it shows a tendency to initially ascend then slowly descend. Finally, the best Γ for ISDE-DD, ISDE-II, ISDE-ID are 0.020, 0.010 and 0.004 according to Avg respectively.

Table 7
Testing results of ISDE using different Γ. Note that ISDE-DD indicates ISDE using DeepLabv3+/R for both normal maritime background and sea fog background. ISDE-II indicates using I-DeepLabv3+ for both normal maritime background and sea fog background. ISDE-ID indicates ISDE using I-DeepLabv3+ for normal maritime background and DeepLabv3+/R for sea fog background. Best results according to Avg are reported in bold.

Γ	ISDE-DD		ISDE-II		ISDE-ID	
	MIoU (%)	Avg (%)	MIoU (%)	Avg (%)	MIoU (%)	Avg (%)
0.000	77.64	86.45	79.88	87.93	79.88	87.93
0.001	79.08	87.54	80.61	88.47	80.70	88.57
0.002	79.52	87.88	80.65	88.50	81.00	88.81
0.003	79.71	88.02	80.68	88.52	81.18	88.94
0.004	79.80	88.10	80.67	88.52	81.20	**88.96**
0.005	79.84	88.15	80.67	88.53	81.14	88.93
0.010	79.90	88.22	80.67	**88.54**	81.07	88.91
0.015	79.92	88.25	80.65	88.52	81.05	88.90
0.020	79.93	**88.26**	80.61	88.50	81.03	88.90
0.025	79.86	88.22	80.63	88.51	81.00	88.88
0.030	79.88	88.24	80.58	88.48	80.96	88.86
0.035	79.78	88.18	80.57	88.46	80.92	88.84
0.040	79.77	88.18	80.56	88.47	80.90	88.83
0.045	79.75	88.17	80.50	88.42	80.87	88.81
0.050	79.70	88.14	80.49	88.42	80.84	88.79

Since there is a large amount of experimental results in many papers revealing the performance of DeepLabv3+ performed the best in segmentation task in many scenes, what we are supposed to do is just comparing our method with DeepLabv3+. Comparison results and the ultimate experimental results are shown in Table 8. It can be clearly observed that our method outperforms DeepLabv3+ in PA, MPA, MIoU, FWIoU and Avg, and extra time that our method cost is only 10~20 ms.

3.4. Qualitative evaluation

Fig. 13 shows some segmentation results from DeepLabv3+ and ISDE-ID, it can be noted that the segmentation results from ISDE-ID are better than those from DeepLabv3+, especially encountering with sea fog background, see Fig. 13(a), (b), (c), (d), (e). Meanwhile, when

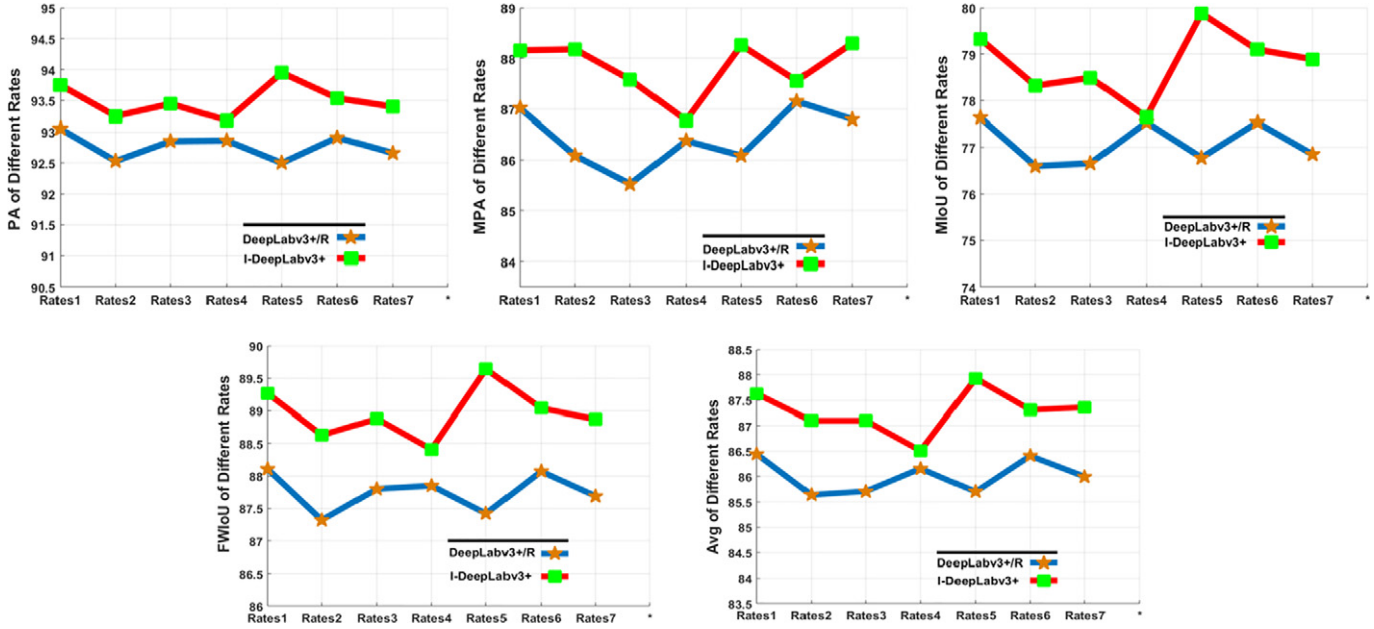


Fig. 11. Comparison of DeepLabv3+/R and I-DeepLabv3+. Best view in color.

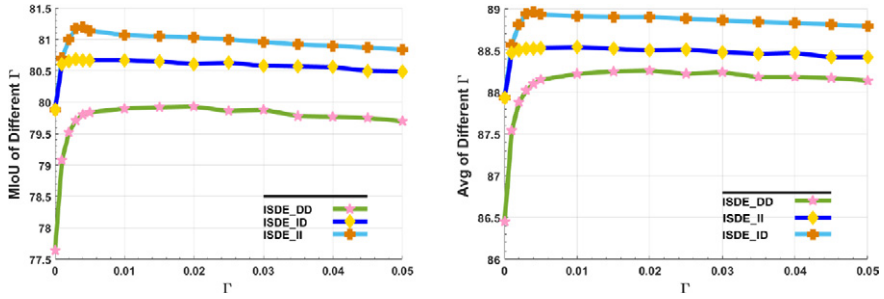


Fig. 12. Testing results of ISDE using different Γ. Best view in color.

encountering normal maritime background, the segmentation results from ISDE-ID are on par with or better than those from DeepLabv3+, see Fig. 13(f), (g), (h), (i) for some examples.

4. Discussion

As shown in Fig. 11, it can be clearly observed that the ship extraction ability of I-DeepLabv3+ is better than DeepLabv3+/R, which benefits from the comprehensive ability of feature extraction and feature reuse of DenseNet69, in [38], it was clearly pointed out the performance of DenseNet would be better as the depth of the model and the value of the growth rate increase. Interestingly, in [39], it was demonstrated that using a composite structure of down-sample and up-sample based on DenseNet could solve the segmentation problem under urban background, this indicates that our I-DeepLabv3+ based on newly proposed DenseNet69 used for complex maritime background makes sense.

As the results revealed in Fig. 12, with the value of Γ increasing, the performance of ISDE-II and ISDE-ID is not prominent as expected. It is important to note, (i) the number of sea fog images is not as much as normal maritime images, it can reduce the generalization ability of the I-DeepLabv3+ to a certain extent, (ii) we set batch size to a small value because of the GPU memory restriction, a small mini-batch size may slow down the converging speed owing to the variety among different mini batches, on the other hand, when training sea fog model, given the variety of sea fog images [42], such as mist, fog and heavy fog, or in sight and out of sight, these all could make I-DeepLabv3+ more difficult to converge.

The results in Table 7 reveal the best Γ for each network would always be a quite small value. Owing to a fact that the number of sea fog images in testing set of MariShipSeg-HEU is relatively small, in other words, sea fog images are just a small part of our testing set, which is the same as reality. In order to avoid degraded performance, we ought to set Γ to a small value to accomplish those with heavy sea fog, the experimental results demonstrate what we think is correct. Meanwhile, it can also be noted that the best Γ for ISDE-ID and ISDE-II are all smaller than for ISDE-DD, the reason is because the segmentation results of I-DeepLabv3+ have been quite great, if we would like to achieve better performance, the results of SE trained for sea fog background ought to be greater, but owing to the aspects explained above, it is more difficult to train SE for sea fog background to have an equal performance with SE trained for normal maritime background, therefore, it is supposed to set Γ to a smaller value to solve those with heavy fog to achieve better performance, the experimental results prove true.

As shown in Table 8, the time our method cost is 10~20 ms longer than DeepLabv3+ does. It should be noted that our method is composed of IFD and SE, as a result, extra time needed is inevitable, but we attempted to narrow it, that's exactly why we chose a lightweight SqueezeNet as IFD. Moreover, please pay attention to checkpoint function used in experiments, the using of checkpoint function could bring 10%~20% extra processing time. Therefore, if configuration allows, take out checkpoint and the predicting time of each image may be shorter, on the other hand, network slimming [46] could also be taken into consideration to slim the network more lightweight. Meanwhile, it should also be noted there are a few images in testing set containing large wakes and large waves, as previously narrated, these factors all could influence the final segmentation accuracy, we just took sea fog as an example and attempted to work it out, fortunately, the experimental results demonstrate that our method works, therefore, if employ our method to solve other interference factors, the segmentation accuracy would be higher.

As shown in Fig. 13, it can be clearly observed the segmentation results from ISDE-ID under sea fog background are better than those from DeepLabv3+, this is because we just took sea fog background as an example in our paper as previously narrated. Interestingly, when encountering with normal maritime background (non sea fog), the segmentation results from ISDE-ID are on par with or better than those from DeepLabv3+, this is mainly because we utilized I-DeepLabv3+ as SE to extract ships, in other words, DenseNet69 works. This can be due to the particular structure of DenseNet69, which is able to extract new features from the features extracted long before and provides more global information.

Table 8
Method comparison. Our three ISDE versions all outperform the DeepLabv3+ in PA, MPA, MIoU and Avg, and extra time our ISDE cost is only 10~20 ms. ISDE-ID performs the best according to Avg. Best results of each evaluation metric are reported in bold.

Method	PA (%)	MPA (%)	MIoU (%)	FWIoU (%)	Avg (%)	Time (ms)
DeepLabv3+	93.05	87.02	77.64	88.11	86.45	**125**
ISDE-DD	93.74	89.94	79.93	89.44	88.26	144
ISDE-II	**94.23**	89.09	80.67	90.16	88.54	143
ISDE-ID	94.22	**90.17**	**81.20**	**90.26**	**88.96**	143

5. Conclusion

In this paper, aiming at solving segmentation accuracy decreasing, we present an integrated ship segmentation method based on discriminator and extractor. Considering the interference factors that input image may contain, we employed SqueezeNet as the implementation of IFD to make a judgment. According to the result of IFD, we employed different SE to finally extract ships. The experimental results demonstrate that our method outperforms state-of-the-art methods in PA, MPA, MIoU, FWIoU and Avg, meanwhile, our method can run in real time as well.

The contribution of our work includes three points. (i) An integrated ship segmentation method based on discriminator and extractor. (ii) A new feature extraction network called DenseNet69, which can be embedded into the DCNN of DeepLabv3+ as I-DeepLabv3+ to improve the ability of ship extraction. (iii) A collected and manually labeled visible image dataset for ship segmentation. It will be public to the community.

Future work will be focused on optimizing the issues of training time. Surely, embedding new networks having better performance in the future literature could also be taken into account.

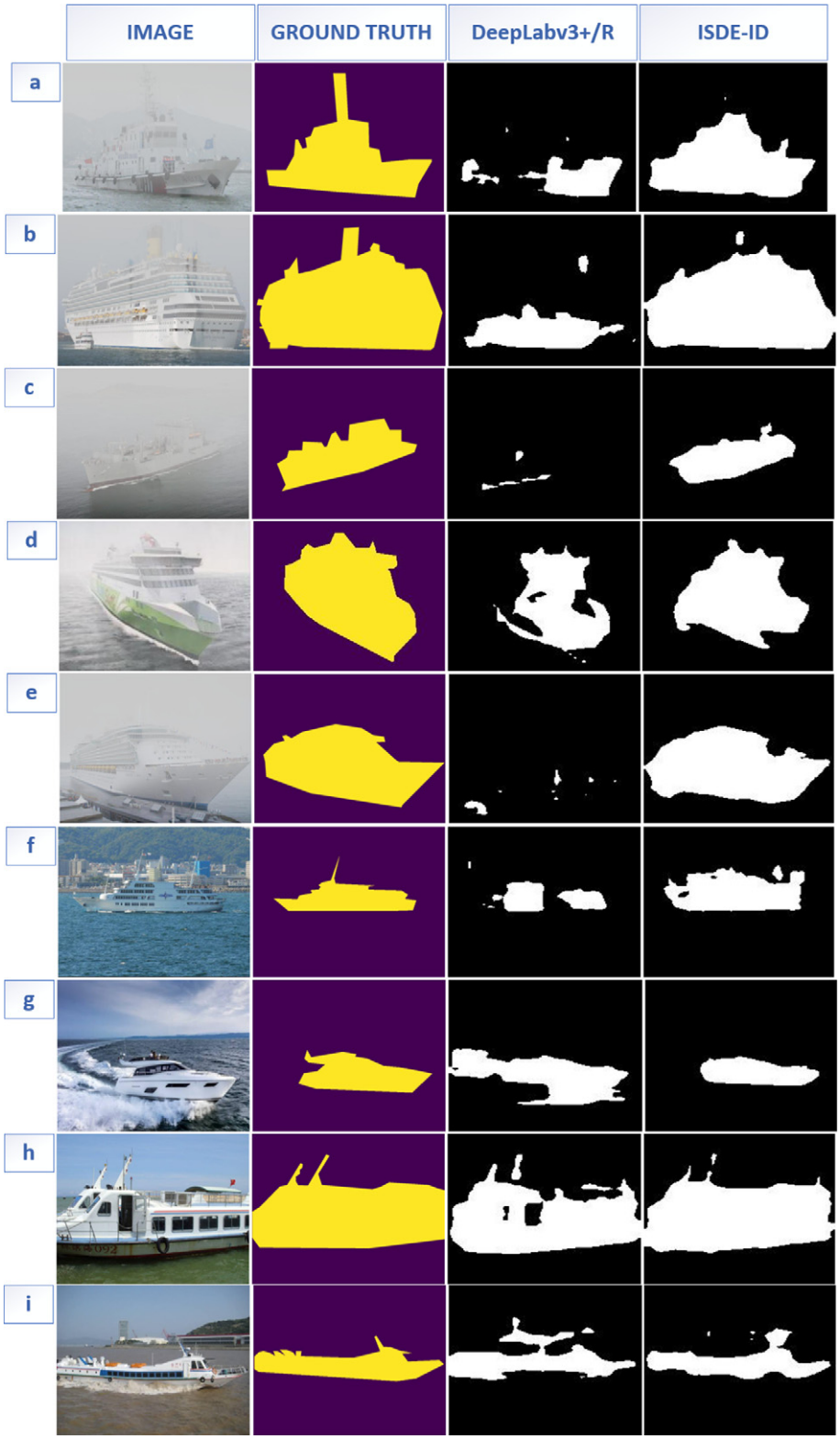


Fig. 13. Segmentation result comparison. Note that original images are listed in the first column, ground truths are listed in the second column, segmentation results from DeepLabv3+/R are listed in the third column, and segmentation results from ISDE-ID are listed in the fourth column. Segmentation results from DeepLabv3+/R are attained from Rates1. Best view in color.

Declaration of competing interest

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

[1] W. Ariza Ramirez, Z.Q. Leong, H. Nguyen, S.G. Jayasinghe, Non-parametric dynamic system identification of ships using multi-output Gaussian Processes, Ocean Eng. 166 (2018) 26–36, https://doi.org/10.1016/j.oceaneng.2018.07.056.

[2] Z. Wang, Z. Zou, C. Guedes Soares, Identification of ship manoeuvring motion based on nu-support vector machine, Ocean Eng. 183 (2019) 270–281, https://doi.org/10.1016/j.oceaneng.2019.04.085.

[3] Y. Zhang, Q.Z. Li, F.N. Zang, Ship detection for visual maritime surveillance from non-stationary platforms, Ocean Eng. 141 (2017) 53–63, https://doi.org/10.1016/j.oceaneng.2017.06.022.

[4] Y. Wei, Z. Ding, H. Huang, C. Yan, J. Huang, J. Leng, A non-contact measurement method of ship block using image-based 3D reconstruction technology, Ocean Eng. 178 (2019) 463–475, https://doi.org/10.1016/j.oceaneng.2019.03.015.

[5] A.P. Wijaya, P. Naaijen, Groesen Andonowati, E. Van, Reconstruction and future prediction of the sea surface from radar observations, Ocean Eng. 106 (2015) 261–270, https://doi.org/10.1016/j.oceaneng.2015.07.009.

[6] X. Bai, Z. Chen, Y. Zhang, Z. Liu, Y. Lu, Infrared ship target segmentation based on spatial information improved FCM, IEEE Trans. Cybern. 46 (2015) 3259–3271, https://doi.org/10.1109/TCYB.2015.2501848.

[7] X. Bai, M. Liu, T. Wang, Z. Chen, P. Wang, Y. Zhang, Feature based fuzzy inference system for segmentation of low-contrast infrared ship images, Appl. Soft Comput. J. 46 (2016) 128–142, https://doi.org/10.1016/j.asoc.2016.05.004.

[8] Liu, Z., Zhou, F., Bai, X., 2013. Infrared ship target segmentation based on region and shape features. Int. Work. Image Anal. Multimed. Interact. Serv. 1–4. doi:https://doi.org/10.1109/WIAMIS.2013.6616124.

[9] Z. Liu, F. Zhou, X. Chen, X. Bai, C. Sun, Iterative infrared ship target segmentation based on multiple features, Pattern Recogn. 47 (2014) 2839–2852, https://doi.org/10.1016/j.patcog.2014.03.005.

[10] Z. Liu, X. Bai, C. Sun, F. Zhou, Y. Li, Infrared ship target segmentation through integration of multiple feature maps, Image Vis. Comput. 48–49 (2016) 14–25, https://doi.org/10.1016/j.imavis.2015.12.005.

[11] J. Shen, S. Liu, Y. Ma, Fast infrared image segmentation algorithm, J. Infrared Millim. Waves-Chinese Ed. 24 (2005) 224. http://en.cnki.com.cn/Article_en/CJFDTotal-HWYH200503014.htm.

[12] W. Tao, Unified mean shift segmentation and graph region merging algorithm for infrared ship target segmentation, Opt. Eng. 46 (2007), 127002. https://doi.org/10.1117/1.2823159.

[13] X. Wang, H. Xu, H. Wang, Adaptive recursive algorithm for infrared ship image segmentation based on gray-level histogram analysis. MIPPR 2007 Autom. Target Recognit. Image Anal, Multispectral Image Acquis. 67861U (2007) 6786, https://doi.org/10.1117/12.748937.

[14] T.X. Zhang, G.Z. Zhao, F. Wang, G.X. Zhu, Fast recursive algorithm for infrared ship image segmentation, Hongwai Yu Haomibo Xuebao/Journal Infrared Millim. Waves 25 (2006) 295–300.

[15] K. Ji, X. Xing, Z. Zhao, H. Zou, J. Sun, A refined ship segmentation method in SAR imagery. MIPPR 2013 Autom. Target Recognit, Navig. 89180R (2013) 8918, https://doi.org/10.1117/12.2031494.

[16] X. Zhang, B. Xiong, G. Dong, G. Kuang, Ship segmentation in SAR images by improved nonlocal active contour model, Sensors (Switzerland) 18 (2018) 1–16, https://doi.org/10.3390/s18124220.

[17] Liu, S. T., Lu, B., Wang, H. L., Yin, F. L., 2012. Ship visible image segmentation method based on combining coarse and precise interaction and iterative graph cut. Journal of Optoelectronics. Laser, (8), 34. http://en.cnki.com.cn/Article_en/CJFDTotal-GDZJ201208034.htm

[18] Rusch, O., Ruwwe, C., Udo, Z., 2005. Image segmentation in naval ship images. http://www.germancolorgroup.de/html/Vortr_05_pdf/b08_rusch

[19] D. Oliva, E. Cuevas, G. Pajares, D. Zaldivar, M. Perez-Cisneros, Multilevel thresholding segmentation based on harmony search optimization, J. Appl. Math. 2013 (2013) https://doi.org/10.1155/2013/575414.

[20] N. Senthilkumaran, R. Rajesh, Image segmentation - a survey of soft computing approaches. ARTCom 2009 - Int. Conf. Adv. Recent Technol, Commun. Comput. 1 (2009) 844–846, https://doi.org/10.1109/ARTCom.2009.219.

[21] S. Angelina, L.P. Suresh, S.H.K. Veni, Image segmentation based on genetic algorithm for region growth and region merging, 2012 Int. Conf. Comput. Electron. Electr. Technol. ICCEET 2012 2012, pp. 970–974, https://doi.org/10.1109/ICCEET.2012.6203833.

[22] J. Ning, L. Zhang, D. Zhang, C. Wu, Interactive image segmentation by maximal similarity based region merging, Pattern Recogn. 43 (2010) 445–456, https://doi.org/10.1016/j.patcog.2009.03.004.

[23] M. Van den Bergh, X. Boix, G. Roig, L. Van Gool, SEEDS: superpixels extracted via energy-driven sampling, Int. J. Comput. Vis. 111 (2015) 298–314, https://doi.org/10.1007/s11263-014-0744-2.

[24] O. Veksler, Y. Boykov, P. Mehrani, Superpixels and supervoxels in an energy optimization framework, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Berlin Heidelberg, Berlin, Heidelberg 2010, pp. 211–224, https://doi.org/10.1007/978-3-642-15555-0_16.

[25] A.M. Siddiqui, A. Ghafoor, M.R. Khokher, Image segmentation using multilevel graph cuts and graph development using fuzzy rule-based system, IET Image Process. 7 (2013) 201–211, https://doi.org/10.1049/iet-ipr.2012.0082.

[26] Z. Zaixin, C. Lizhi, C. Guangquan, Neighbourhood weighted fuzzy c-means clustering algorithm for image segmentation, IET Image Process. 8 (2014) 150–161, https://doi.org/10.1049/iet-ipr.2011.0128.

[27] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2015) 3431–3440, https://doi.org/10.1109/CVPR.2015.7298965.

[28] Liu, W., Rabinovich, A., Berg, A.C., 2015. ParseNet: looking wider to see better. arXiv Prepr. arXiv1506.04579. https://arxiv.org/pdf/1506.04579

[29] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, Proc. IEEE Int. Conf. Comput. Vis. 2015 International Conference on Computer Vision, ICCV 2015 2015, pp. 1520–1528, https://doi.org/10.1109/ICCV.2015.178.

[30] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9351, 2015, pp. 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.

[31] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-January, 936–944. doi:10.1109/CVPR.2017.106.

[32] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-January, 6230–6239. doi:10.1109/CVPR.2017.660.

[33] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062. https://arxiv.org/pdf/1412.7062v1

[34] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2017) 834–848, https://doi.org/10.1109/TPAMI.2017.2699184.

[35] Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017b. Rethinking atrous convolution for semantic image segmentation. https://arxiv.org/pdf/1706.05587

[36] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation. Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 11211, LNCS (2018) 833–851, https://doi.org/10.1007/978-3-030-01234-2_49.

[37] Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size 1–13. https://arxiv.org/pdf/1602.07360

[38] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-January, 2261–2269. doi:10.1109/CVPR.2017.243.

[39] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, Y. Bengio, The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation, IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. 2017-July (2017) 1175–1183, https://doi.org/10.1109/CVPRW.2017.156.

[40] D.K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, C. Quek, Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey, IEEE Trans. Intell. Transp. Syst. 18 (2017) 1993–2016. https://doi.org/10.1109/TITS.2016.2634580.

[41] Domenico D. Bloisi, Luca Iocchi, Andrea Pennisi, Luigi Tombolini, ARGOS-Venice boat classification, 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) 2015, pp. 1–6 , doi: 10.1109/AVSS.2015.7301727 http://www.dis.uniroma1.it/~labrococo/MAR/.

[42] J. Dong, K. Liu, J. Wang, Simulation of the foggy scene under outdoor natural scenes, Journal of Computer-Aided Design and Computer Graphics 25 (3) (2013) 397–409.

[43] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-January, 5967–5976. doi:10.1109/CVPR.2017.632.

[44] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, Proc. IEEE Int. Conf. Comput. Vis. 2015 International Conference on Computer Vision, ICCV 2015 2015, pp. 1026–1034, https://doi.org/10.1109/ICCV.2015.123.

[45] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2014 1–15. https://arxiv.org/pdf/1412.6980.

[46] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning efficient convolutional networks through network slimming, Proc. IEEE Int. Conf. Comput. Vis. 2017-October (2017) 2755–2763, https://doi.org/10.1109/ICCV.2017.298.