Alexandria Engineering Journal 87 (2024) 17–30

Contents lists available at ScienceDirect

Alexandria Engineering Journal

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

Original Article

Advanced integrated segmentation approach for semi-supervised infrared ship target identification

Ting Zhang [a], Guanlun Jiang [a], Zhaoying Liu [a,*], Sadaqat ur Rehman [b], Yujian Li [c]

[a] Faculty of Information Technology, Beijing University of Technology, Beijing, China
[b] School of Sciences, Engineering and Environment, University of Salford, Manchester, UK
[c] School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin, China

ARTICLE INFO

Keywords:
Infrared ship target segmentation
Semi-supervised learning
Adaptive weight
Student network
Teacher network

ABSTRACT

Infrared ship target segmentation is the important basis of infrared guided weapon in the sea-air context. Typically, accurate infrared ship target segmentation relies on a large number of pixel-level labels. However, it is difficult to obtain them. To this end, we present a method of Semi-supervised Infrared Ship Target Segmentation with Dual Branch (SeISTS-DB), which utilizes a small amount of labeled data and a large amount of unlabeled data to train model and improve segmentation performance. There are three main contributions. First, we design a target segmentation branch to generate the pseudo labels for unlabeled data. It consists of a dual learning network and a segmentation network. The dual learning network generates pseudo labels with weights for unlabeled data. The segmentation network is trained using both labeled data and unlabeled data with pseudo labels to achieve target segmentation of infrared ship, obtaining the preliminary segmentation results. Secondly, we introduce an error segmentation pixel correction branch, which contains a student network and a teacher network, to modify the pixel category error of the preliminary segmentation map. Finally, the outputs of the two branches are combined to obtain the final segmentation result. The SeISTS-DB is compared with other fully-supervised and semi-supervised methods on the infrared ship images dataset. Experimental results demonstrate that when the labeled data accounts for 1/8 of the training data, the mean Intersection over Union (mIou) is respectively improved by 15.35% and 6.19% at most. Besides, it is also compared with other methods on the public IRSTD-1k dataset, when the proportion of labeled images is 1/8, the mIoU is respectively improved by 11.76% at most compared to the state-of-the-art semi-supervised methods, demonstrating its effectiveness.

1. Introduction

Infrared imaging is a technique that utilizes the infrared wavelength amplitude of an object for imaging. More specifically, it uses photoelectric technology to detect infrared specific band signals radiated by objects, and then converts the signals into images that can be distinguished by human vision. With the advantages of long working distance, stable imaging effect, all-weather work and strong anti-interference ability, infrared imaging technology is widely used in sea surface monitoring systems [1–5]. In the field of infrared ship images, there are two research directions, one is the task of small target detection and the other is the task of target segmentation. The goal of the small object detection task is to identify the target frame in the infrared image and the category of the target frame [6–8]. The goal of target segmentation is to classify correct category of pixels in the infrared ship image. This paper focuses on the task of target segmentation. By classifying the pixels in the infrared ship image and marking the target of the ship, infrared ship target segmentation is conducive to combating illegal ships and achieving accurate maritime rescue, which plays an important role in marine security, marine rights, and the safety of people's lives and property [9–12].

At present, most infrared ship segmentation methods are based on fully-supervised learning [13,14], which first designs a deep network model, and then uses all precisely labeled data for training to obtain segmentation results. Common segmentation networks include Fully Convolutional Network (FCN) [15], U-shape Network (UNet) [16], and Segmentation Network (SegNet) [17,18]. Although these networks achieve good segmentation results, they all require accurate pixel-level labels

* Corresponding author.
E-mail addresses: zhangting@bjut.edu.cn (T. Zhang), 1770819473@qq.com (G. Jiang), zhaoying.liu@bjut.edu.cn (Z. Liu), s.rehman15@salford.ac.uk (S. ur Rehman), liyujian@bjut.edu.cn (Y. Li).

https://doi.org/10.1016/j.aej.2023.12.016
Received 13 August 2023; Received in revised form 25 November 2023; Accepted 10 December 2023
Available online 15 December 2023

for training. While for infrared ship images, obtaining these tags is time-consuming and labor-intensive.

In this paper, we study semi-supervised learning method for infrared ship target segmentation, aiming to realize the training network based on a small amount of labeled data and a large amount of unlabeled data, and improve the segmentation accuracy [19–25]. Undoubtedly, the crux of the matter is how to effectively utilize unlabeled images. Self-training and consistent regularization are two of the more commonly used methods in semi-supervised learning. Self-training is initially developed in classification [26]. This method uses a segmentation network trained with labeled data to generate pseudo labels for unlabeled data, and then expands the labeled dataset, so that iteratively trains the segmentation network to achieve a better segmentation level. Consistency regularization makes the features of pseudo labels obtained by adding different perturbations to the same image through the network output close to each other [27]. Although the above two methods have achieved good performance in semi-supervised segmentation, they also have a prominent shortcoming: there is the problem of poor quality of pseudo labels generated for unlabeled data. In the process of repeated iterations, it is the network that generates confirmations bias. Most existing methods manually generate a threshold and filter out pseudo labels below the threshold to alleviate the problem of confirmation bias, but the impact on network performance largely depends on this manually set threshold [28–31].

Specifically, we propose a method of Semi-supervised Infrared Ship Target Segmentation with Dual Branch (SeISTS-DB). The two branches are the target segmentation branch and the error segmentation pixel correction branch, respectively.

The first branch is the target segmentation branch, which consists of dual learning network and a segmentation network. Among them, the dual learning network is used to generate pseudo labels for those unlabeled data, and the segmentation network is used to generate initial segmentation result. In order to prevent the pseudo labels generated by a network from misleading the segmentation network, the dual learning network contains two networks with the same structure but different parameters. These two networks generate two different segmentation results for each unlabeled sample. By combining these two segmentation results, a pseudo label of an unlabeled data is generated to augment the labeled data.

For the segmentation network, both the labeled data and the unlabeled data with pseudo labels are used for training to obtain initial segmentation results. Here, in order to reduce the impact of low-quality pseudo labels on updating segmentation network weights, we design an adaptive weight to reduce the proportion of such pseudo labels during training.

The second branch is the error segmentation pixel correction branch, which consists of a student network and a teacher network. During training, the two networks receive augmented images of the same image with different perturbations, and make the predictions of the student model as consistent as possible with those of the teacher model. Through semi-supervised learning, class labels for test images are generated. Finally, the category label is used to correct the initial segmentation results to improve the segmentation accuracy.

The main contributions of this paper are as follows:

1) We propose a semi-supervised method for infrared ship target segmentation, named as SeISTS-DB, to further improve the accuracy. It first generates a pseudo label for the unlabeled data, and then uses all the data to train the segmentation network to obtain the initial segmentation result. Finally, the initial segmentation result is modified to obtain the final segmentation result.
2) We design a target segmentation branch, including dual learning network and segmentation network, to achieve the initial segmentation result. It first generates high-quality pseudo labels for unlabeled data, and use pseudo labels to obtain initial segmentation results.
3) We introduce the error segmentation pixel correction branch to modify the initial segmentation result. It uses semi-supervised learning to obtain class labels for infrared ship images. Then, the category label is used to correct the initial segmentation result obtained in the target segmentation branch, thereby further improving the segmentation accuracy
4) We present a sample-level adaptive weight to assign different weights for different pseudo labels, thereby improving the segmentation performance of the segmentation network.
5) We verify the effectiveness of the proposed method on both homemade infrared ship dataset and public datasets.

The rest of the paper is organized as follows: Section 2 gives the related work. Then, Section 3 describes the proposed method and the learning algorithm of each branch. Later on, Section 4 provides the experimental results as well as their analysis, and finally, Section 5 concludes the paper.

2. Related work

The method of our work is mainly related to the segmentation methods based on fully-supervised learning and semi-supervised learning, which are described in detail below.

2.1. Segmentation methods based on fully-supervised learning

The segmentation method based on fully-supervised learning refers to train the network with a large amount of labeled data so that the network has the ability to predict the label of the pixels in the image [32–35]. Fully convolutional network (FCN) is the first deep learning model to achieve segmentation in an end-to-end manner [15,36–38]. It replaces the last fully-connected layer of the network with the convolutional layer in the convolutional network. However, the segmentation results of FCN are not fine enough, and the segmentation performance is poor [39–41].

Later on, researchers proposed some models to improve the segmentation performance. For example, Ronneberger et al. presented the encoder-decoder network [16], named UNet. By fusing the features with splicing, the network can extract more detailed features of the image, obtaining higher segmentation accuracy. Badrinarayanan et al. also designed a network based on the encoder-decoder structure [18], called SegNet. In the encoder part, the index of the maximum value located is recorded during maxpooling. In the decoder, the corresponding pooling index enables non-linear sampling during upsampling. By doing this, the learning process in the sampling phase is avoided. Chen et al. proposed the Deeplab series models [42–44]. They designed a pyramid hole pooling operation to capture different scales of objects, thus enhancing the accuracy of segmentation. Takikawa et al. introduced a dual-stream segmentation network [45]. It adds an additional shape stream to learn the edge information of the objects, improving the quality of the segmentation boundary area. Kirillov et al. used an iterative upsampling method to optimize the segmentation results of the object edge, which can improve the quality of the segmented boundary area [46].

The segmentation methods attempt to design different architectures to improve the segmentation accuracy. Overall, they all belong to the fully-supervised learning methods, and need accurately pixel-level labels to train the network. However, it is difficult to obtain the pixel-level labels. Therefore, it is necessary to explore a method which is less demanding on the pixel-level labeled data.

2.2. Segmentation methods based on semi-supervised learning

The semi-supervised method aims to explore how to use unlabeled data to improve network performance. At present, the common semi-supervised segmentation methods are mainly divided into two categories: self-training [47–55] and consistency training [56–61].

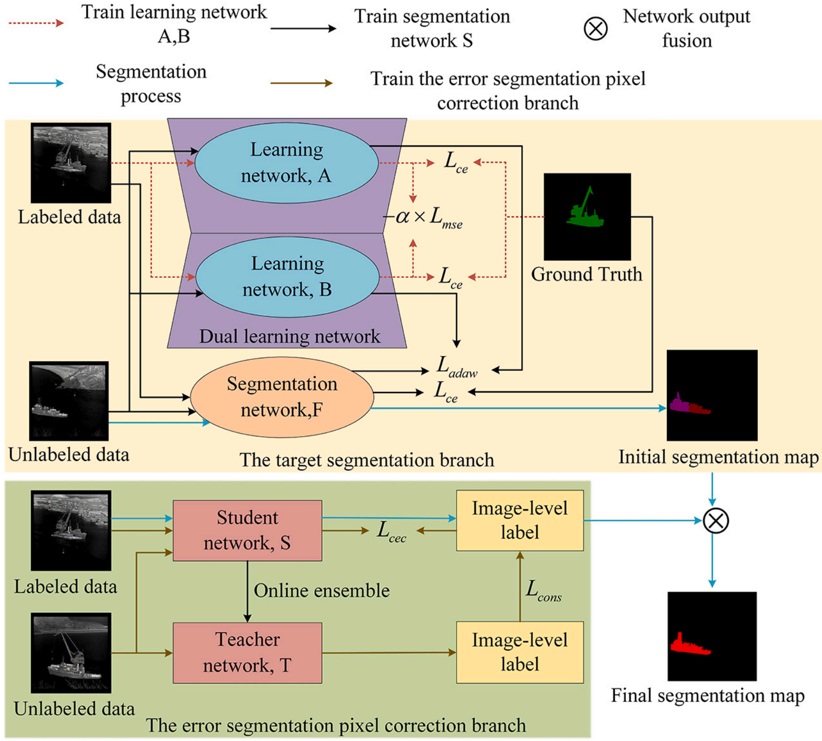


Fig. 1. The overall architecture of SeISTS-DB. The target segmentation branch is used to generate the initial segmentation map. The error segmentation pixel correction branch performs semi-supervised label classification, and uses class level information to modify the error segmentation pixels in the segmentation map.

The self-training method first generates pseudo labels for unlabeled data by pre-training the model, and then trains the model with two categories of labels. For example, Souly et al. proposed a semi-supervised segmentation network with generative adversarial network [62]. Among them, the discriminator is an FCN to classify each pixel in the image. However, this network will generate inaccurate label categories. Hung et al. used the segmentation network as the generator and designed a fully convolutional discriminator to distinguish pixel-level labels from segmentation results [63]. For the unlabeled data, the high confidence category in the discriminator is used as the pseudo label for training generator. This method significantly improves the segmentation performance by using a large amount of unlabeled data. However, there is still the problem of schema collapse. Mondal et al. applied the recurrent adversarial networks to semi-supervised segmentation and enforced cycle consistency to learn a bidirectional mapping between unpaired images and segmentation masks [64]. Additionally, they added an unsupervised regularization term in the loss function. The segmentation performance is improved when labeled data is limited. On the basis of the original generative adversarial network, Mittal et al. proposed a feature matching loss and a self-training loss [65]. Through these two networks, semi-supervised classification and semi-supervised segmentation are combined to reduce misclassified pixels in segmentation maps.

The method based on consistency training believes that adding perturbation to unlabeled data will improve the segmentation performance of the network [66]. Ouali et al. proposed a cross-consistency training method. It adds multiple auxiliary decoders based on the encoder-decoder structure [56]. During training, firstly, the labeled data is used to train the encoder and the main decoder; secondly, the unlabeled data is input into the network to add different perturbations to the encoder. The network is optimized by forcing the predictions of multiple decoders to be consistent. French et al. first used an adaptive variant of CutMix to augment the data, and then imposed a consistency constraint between the prediction of the augmented data and the original data [60]. Olsson et al. introduced a new segmentation augmentation strategy, ClassMix, into a unified framework. It leveraged the consistency regularization and pseudo-labeling for segmentation [61,67]. Zeng et al. performed network perturbation by two segmentation networks with the same structure but with different initializations, enhancing the consistency between perturbed network predictions [68].

The above semi-supervised segmentation methods can improve the segmentation accuracy by explicitly or implicitly extracting features for the unlabeled data. However, they all assume that the segmentation network's pseudo labels for unlabeled data are correct. When the prediction result is wrong, the error will be continuously amplified during training, thus affecting the performance of the network. Therefore, the fault tolerance of these methods is relatively poor.

Differently, the target segmentation branch proposed in this paper will generate a pseudo label for the unlabeled data according to the segmentation map of the two learning networks, and generate an adaptive weight for the generated pseudo label. In this way, the pseudo label with poor effect will have relatively little influence in the subsequent training. Additionally, to solve the pixel category error problem generated in the target segmentation branch, we design an error segmentation pixel correction branch based on the semi-supervised label classification method, which is used to correct the segmentation results of the target segmentation branch and improve the performance of the infrared ship target segmentation.

3. Methods

In this section, we first introduce the overall framework of the semi-supervised infrared ship target segmentation network with dual branch. Then, we describe the target segmentation branch, error segmentation pixel correction branch and two-branch fusion process one by one. Next, we give the loss functions of the two branches, respectively. Finally, we provide the learning algorithms of the two branches.

3.1. General framework

The overall architecture of SeISTS-DB is depicted in Fig. 1. As seen in Fig. 1, the model is mainly composed of the target segmentation branch and the error segmentation pixel correction branch. The target

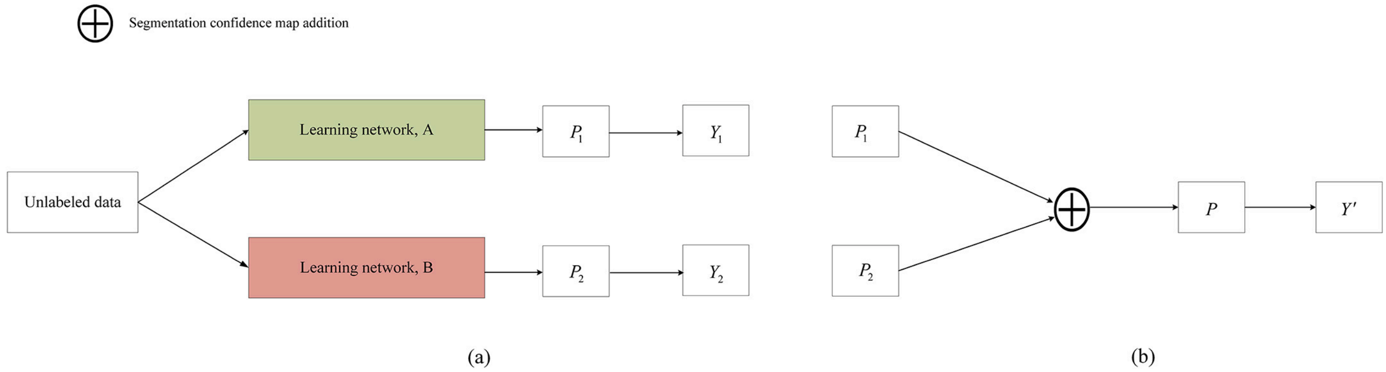


Fig. 2. Illustrating the architectures for (a) the process of unlabeled data passing through double learning network, (b) the process of generating pseudo label.

segmentation branch is used to generate the initial segmentation result, whereas, the error segmentation pixel correction branch is used to correct the wrong segmentation pixel category to improve the segmentation result.

Specifically, on the one hand, considering the low quality of the pseudo label generated by a single network, the target segmentation branch first uses the dual learning network to generate pseudo label for the unlabeled data, which prevents the pseudo label generated by a single network from causing confirmation bias in iterative training. Then, we use both the labeled data and unlabeled data to train the segmentation network to obtain the initial segmentation result. Here, the dual learning network and segmentation network have the same structure. We denote the two sub-networks of dual learning network as A and B, and the segmentation network as F. On the other hand, to correct the wrong segmentation of pixels in the target segmentation map generated by the target segmentation branch, the error segmentation pixel correction branch is adopted for correction. This branch is able to identify the categories contained in the segmented image, and use the image-level label to correct the missegmented pixel in the target segmentation branch. This branch consists of two networks with the same structure, namely student network S and teacher network T. This branch network is trained with both labeled data and unlabeled data, and the image-level label is obtained by the student network S. Finally, the image-level label is used to modify the initial segmentation map to obtain the final segmentation map (Tables 8 and 9).

3.2. Branch introduction

SeISTS-DB first uses the target segmentation branch to get the initial segmentation result, and then uses the error segmentation pixel correction branch to predict the image category. Finally, the results of the two branches are fused to obtain the final segmentation result. The following three parts are introduced sequentially.

3.2.1. Target segmentation branch

Target Segmentation Branch (TSB) consists of dual learning network A, B and segmentation network F. The dual learning networks A and B use the same segmentation network, but have different initialization weights. Firstly, labeled data is used to train dual learning networks A and B. Here, in order to avoid two networks producing segmentation maps with the same error, the segmentation maps output by networks A and B are not exactly the same during training. Then, the trained networks A and B generate segmentation confidence maps P_1 and P_2 of unlabeled data and corresponding pseudo labels Y_1 and Y_2, respectively, as shown in Fig. 2 (a) Among them, the segmentation confidence map is the network output after $softmax$ normalization. Finally, the two segmentation confidence maps are added and reduce the dimensionality to generate the final unlabeled data pseudo label. The adaptive weight of the pseudo label is calculated based on the outputs obtained by networks A and B.

The way to generate the pseudo label is showed in Fig. 2(b). By inputting unlabeled data into learning networks A and B, two three-dimensional segmentation confidence maps P_1 and P_2 are obtained, respectively. Then, the corresponding dimensions of the two confidence maps are added to obtain the overall segmentation confidence map P. Finally, dimensionality reduction is performed in the channel direction of the segmentation confidence map to obtain a one-dimensional pseudo label. Wherein, at each pixel point i, the value of Y_i' is the channel index where P_i obtains the maximum value in the channel direction.

The process of generating adaptive weight is as follows. First, the segmentation confidence maps P_1 and P_2 are dimensionally reduced into one-dimensional segmentation maps Y_1' and Y_2' according to the channel direction. Then, the Intersection over Union of the non-background pixels in the segmentation maps Y_1 and Y_2 are selected as the adaptive weight of the final pseudo label Y'.

After obtaining the pseudo labels and the adaptive weights, we train the segmentation network F. Here, labeled data and unlabeled data with pseudo label are used to train the segmentation network. For the unlabeled data, the gradient update is multiplied by the adaptive weight corresponding to the pseudo label (the weight value is between 0-1). Then, the trained segmentation network F is used to generate initial target segmentation map $S(x)_c$.

3.2.2. Error segmentation pixel correction branch

The error segmentation pixel correction branch is an image classification method based on semi-supervised learning [69–75], which consists of the student network S and the teacher network T. Both student network S and teacher network T use Resnet50 [76] pre-trained on the ImageNet dataset [77] as the backbone network. However, the weight of the teacher network (θ_0) is the exponential moving average of the weight of the student network (θ). Through online integration, the weights of the teacher network are updated [78–80]. In other words, updating the weight of the student network in each iteration will also update the weight of the teacher network. During the training process, the labeled data is first used to train the student network S, and the unlabeled data is then used to train the student network S simultaneously. At this time, the image with Gaussian noise added and the original image are respectively input into the student network and the teacher network to make their predictions as consistent as possible. In the testing phase, we use the trained student network to output the image-level label $G(x)_c$.

3.2.3. Fusion of results from two branches

After obtaining the two branch outputs, they are fused to obtain the final segmentation result. Here, according to the category label output by the error segmentation pixel correction branch, the segmentation map output by the target segmentation branch is corrected, which is expressed as:

$$S(x)_c = \begin{cases} 0, & \text{if } G(x)_c < \gamma \\ S(x)_c, & \text{if } G(x)_c \geq \gamma \end{cases} \tag{1}$$

where $S(x)_c$ is the segmentation result of the c-th category in the target segmentation map of the target segmentation branch, $G(x)_c$ represents the predicted probability of the c-th category of the error segmentation pixel correction branch. γ is a threshold, which is set to 0.1. When the value of $G(x)_c$ is less than the threshold, it indicates that there is no the c-th class pixel in the target segmentation map, thereby filtering the wrong c-th class pixel and increasing the probability that the pixel is segmented into the correct class pixel.

3.3. Loss function

3.3.1. Loss function of the target segmentation branch

The target segmentation branch consists of the dual learning network and the segmentation network. The dual learning network only uses labeled data for training. The segmentation network uses both labeled data and unlabeled data for training. The trained loss function is analyzed.

3.3.1.1. Dual learning network Learning network A and B need labeled data for training, and the training loss is cross-entropy loss [81]:

$$L_{ce}{}^{A,B} = -\frac{1}{|D_l|} \sum_{k\in\{A,B\}} \sum_{i\in|D_l|} \boldsymbol{y}_i \log P_{sm}(f_k(\boldsymbol{x}_i)), \tag{2}$$

where A, B represent learning networks A, B respectively. D_l denotes a set of labeled data. $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ respectively are the i-th sample in D_l and its corresponding pixel-level label. f_k means a single learning network $k \in \{A, B\}$. $p_s m(\cdot)$ indicates the softmax function [82]. $p_s m(\cdot)$ is calculated as:

$$p_{sm}(m_i) = \frac{m_i}{\sum_{n=1}^{N} e^{m_n}} \tag{3}$$

where N is the total number of categories, and m_i is the probability prediction value of the network for the i-th type of data.

After training, the two learning networks have a certain segmentation ability. Since the two learning networks have the same structure, almost the same segmentation map will be obtained when the same unlabeled image is input into learning networks A and B. In order to make the two networks produce not exactly the same results, we introduce the mean square error loss [83] as the dual learning loss of the dual learning network, so that the results of the two learning networks are not exactly the same. The dual learning loss is represented as:

$$L_{mse} = \frac{1}{|D_l|} \sum_{i\in|D_l|} ||P_{sm}(m_A) - P_{sm}(m_B)||^2 \tag{4}$$

where D_l represents the labeled dataset, $P_{sm}(\dot{)}$ is the $softmax$ function, $\boldsymbol{m}_A$ and $\boldsymbol{m}_B$ denotes the outputs of learning networks A and B, respectively.

Finally, the loss function when training learning networks A and B is:

$$L = L_{ce}{}^{A,B} - L_{mse} \tag{5}$$

It is worth noting that we choose the $softmax$ function to normalize the output of the network. The normalized results are used as the scores of the respective segmentation maps of the two learning networks, which are used as the basis for unlabeled data to train the adaptive weight of the segmentation network F.

After training, learning networks A and B are used to generate pseudo labels for unlabeled data. The segmentation map normalized by $softmax$ is called a segmentation confidence map. The segmentation confidence maps obtained by the two learning networks are added together as the total segmentation confidence map. It is calculated as:

$$\boldsymbol{I}_{score,j} = P_{sm}(\boldsymbol{m}_A)_j + P_{sm}(\boldsymbol{m}_B)_j \tag{6}$$

where $P_{sm}(\boldsymbol{m}_A)$ and $P_{sm}(\boldsymbol{m}_B)$ are the segmentation confidence maps generated by two learning networks for unlabeled data. $\boldsymbol{I}_{score,j}$ is a total segmentation confidence map generated for an unlabeled image, where j is the pixels in the segmentation map, and $score$ denotes the score for each pixel.

The total segmentation confidence map generated by two learning networks combines the segmentation results of the two networks, thus effectively avoiding the segmentation bias generated by a single network. The category with the highest score selected for each pixel is selected as the category of this pixel. The calculation method is as follows:

$$t_{pseudo,j} = \arg\max(\boldsymbol{I}_{score,j}) \tag{7}$$

where $\arg\max()$ is used to get the category with the highest score of each pixel, and $t_{pseudo,j}$ is the category of each pixel in segmentation map.

3.3.1.2. Segmentation network The segmentation network uses both labeled data and unlabeled data for training. On the one hand, using two networks to generate pseudo labels for unlabeled data can generate higher quality pseudo labels. On the other hand, this may also face misclassification of the two segmentation maps, resulting in poor quality pseudo labels.

In order to reduce the proportion of pseudo labels with poor quality in updating the weight of segmentation network F, we design a sample-level adaptive weight. By adjusting the proportion of different pseudo labels in the reverse gradient propagation, the weight of the segmentation network F is updated. The complete calculation process is as follows:

$$\boldsymbol{t}_{pseudo,A} = \arg\max\left(\boldsymbol{I}_{score,A}\right) \tag{8}$$

$$\boldsymbol{t}_{pseudo,B} = \arg\max\left(\boldsymbol{I}_{score,B}\right) \tag{9}$$

For the segmentation maps $\boldsymbol{t}_{pseudo,A}$ and $\boldsymbol{t}_{pseudo,B}$ generated by the two learning networks, the Intersection over Union of their non-background pixels is calculated as the weight of the pseudo label in the segmentation network F weight update. Its calculation formula is:

$$\omega = \frac{\boldsymbol{t}_{pseudo,A}{}' \bigcap \boldsymbol{t}_{pseudo,B}{}'}{\boldsymbol{t}_{pseudo,A}{}' \bigcup \boldsymbol{t}_{pseudo,B}{}'} \tag{10}$$

where $\boldsymbol{t}_{pseudo,A}{}'$ and $\boldsymbol{t}_{pseudo,B}{}'$ are the non-background pixels in the pseudo label, and ω is the adaptive weight of the pseudo label.

For the unlabeled data, an adaptive weighted cross-entropy loss is used for training:

$$L_{adaw} = -\frac{1}{|D_u|} \sum_{j\in D_u} \omega \times \boldsymbol{t}_{pseudo,j} \log P_{sm}(f_s(\boldsymbol{x}_j)) \tag{11}$$

where D_u is the set of unlabeled data, and ω is the sample-level adaptive weight. It is used to represent the influence of unlabeled data in adjusting the weights of the segmentation network F, as shown in formula (10).

For the labeled data, the cross entropy loss is used for training:

$$L_{ces} = -\frac{1}{|D_l|} \sum_{i\in D_l} \boldsymbol{y}_i \log P_{sm}(f_s(\boldsymbol{x}_i)) \tag{12}$$

where D_l is the labeled dataset, $\boldsymbol{x}_i$ is the labeled data, $\boldsymbol{y}_i$ is the label corresponding to $\boldsymbol{x}_i$.

Finally, the complete loss function of the segmentation network is:

$$L_S = L_{adaw} + L_{ces} \tag{13}$$

3.3.2. Loss function of the error segmentation pixel correction branch

During the training process, the input of the student network and the teacher network are the original image and the perturbed image with Gaussian noise added, respectively. The weight of the student network is updated through the gradient feedback of the network. The weight of the teacher network is the exponential moving average of the weight of the student network, which is adjusted by means of online integration.

The error segmentation pixel correction branch is a semi-supervised label classification network based on a consistency. Therefore, the predicted output of student network S and teacher network T should be close. Thus, the mean square error is used to measure the prediction results of these two networks. The training goal is to make the mean square error of the prediction results of the two networks as small as possible.

The goal of the error segmentation pixel correction branch is to make the student network capable of image label classification. The optimization method of the student network S is adjusted by classification cross entropy loss L_{cec} of the labeled data, and the consistency loss L_{cons} of the unlabeled data.

The cross-entropy loss is expressed as:

$$L_{cec} = -\frac{1}{|D_l|}\sum_{i\in D_l} \boldsymbol{y}_i \log P_{sm}(f_g(\boldsymbol{x}_l)) \tag{14}$$

where D_l is the labeled dataset, $\boldsymbol{x}_l$ represents the sample of the labeled data, $\boldsymbol{y}_l$ is the image-level label of $\boldsymbol{x}_l$, $f_g()$ denotes the student network.

The consistency loss is expressed as:

$$L_{cons} = ||f_g(\boldsymbol{x}_u) - f_h(\boldsymbol{x}_u)||^2 \tag{15}$$

where $f_h()$ represents the teacher network, $f_g()$ stands for the student network, $\boldsymbol{x}_u$ implies the unlabeled data.

Finally, the total loss function of this branch is:

$$L_p = L_{cec} + L_{cons} \tag{16}$$

3.4. Training process

3.4.0.1. Training of target segmentation branch The target segmentation branch consists of two training stages. In the first stage, the labeled data is used to train the learning networks A and B, to make the two learning networks generate the pseudo label of the unlabeled data. In the second stage, labeled data and unlabeled data are used to train the segmentation network F. For the labeled data, the cross-entropy loss is used to update the gradients. For the unlabeled data, pseudo labels are used to supervise information and update gradients. It should be noted that, the unlabeled data needs to use adaptive weight to adjust the proportion in the gradient update. See Algorithm 1 for the training process of target segmentation branch.

3.4.0.2. Training of error segmentation pixel correction branch This branch training needs to use labeled data x_l and the corresponding image-level label y_l, and unlabeled data x_u. For labeled data x_l, it is input to the student network S generate the predicted image-level label $f_g(\mathbf{x}_l)$. Then, the classification cross-entropy loss is used to tune the parameters of the student network S. For the unlabeled data $\mathbf{x}_u$, it is input to both the student network S and the teacher network T, and the mean square error is used to calculate the predicted values of the two networks. By adjusting the parameters of the student network, the mean square error moves in the direction of reduction. See Algorithm 2 for the training process of this branch.

3.5. Inference

For a new image, the segmentation process of our network is as follows: firstly, the network standardizes the size of the input image, and the size of the standardized image is 256*256; secondly, the standardized image is fed into the segmentation network F and the student network S simultaneously, and the preliminary segmentation map of the image and the category label of the image are obtained, respectively. Finally, the category label of the image is used to correct the preliminary segmentation map of the image to obtain the final segmentation map.

Algorithm 1 The target segmentation branch training process.

Input: labeled data $(\mathbf{x}_l, \mathbf{y}_l) \in D_l$, unlabeled data $\mathbf{x}_u \in$
D_u, coefficient of balance α, training times of double
learning modules E_{AB}, training times of segmentation
network E_F;
Output: Segmentation network F;
    Initialization Learning network A and B,segmentation
network F, current number of training rounds $E_{cur} = 0$;
    for $E_{cur} < E_{AB}$ do
        for $(\mathbf{x}_l, \mathbf{y}_l)$ in D_l do
        Use formula (5) to update the parameters of A and
        B;
        end for
        $E_{cur} = E_{cur} + 1$;
    end for
    for $E_{cur} < E_{AB} + E_s$ do
        for $\mathbf{x}_u$ in D_u do
            Use equation (7) to generate pseudo labels
            $t_{pseudo,j}$;
            Use A,B and equation (10) to calculate the weight
            $w_{iou,j}$ of x_u;
            Use equation (11) to update the parameters of F;
        end for
        for $(\mathbf{x}_l, \mathbf{y}_l)$ in D_l do
            Use equation (12) to update the parameters of F;
        end for
        $E_{cur} = E_{cur} + 1$;
    end for
return segmentation network F;

4. Experimental results

In order to verify the effectiveness of SeISTS-DB, we did some experimental results with the self-made infrared ship dataset and a public dataset. In this section, we first introduce the infrared ship dataset and evaluation metrics used in the experiments, and describe the experimental details. Then, we compare different baseline networks to explore the effectiveness of the target segmentation branch. Later, we analyze the adaptive weight used to train the segmentation network F. Finally, we compare the performance of SeISTS-DB with that of the fully-supervised methods and semi-supervised methods.

Algorithm 2 The error segmentation pixel correction branch training process.

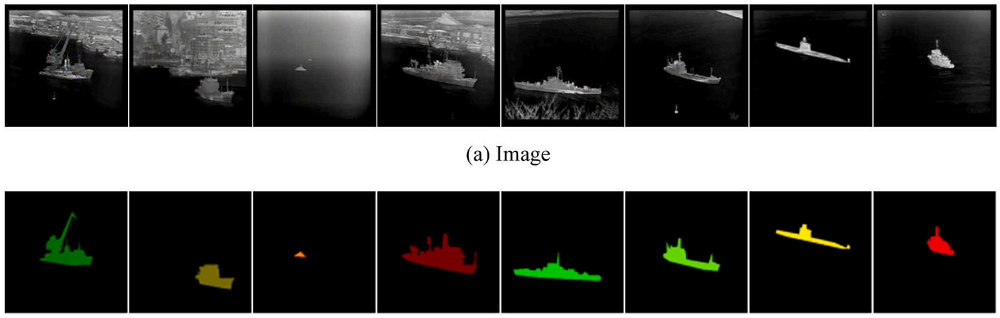
Input: labeled data $(\mathbf{x}_l, \mathbf{y}_l) \in D_l$, unlabeled data $\mathbf{x}_u \in$
D_u, training times E;
Output: Student network S for image classification;
    Initialization student network S and teacher network
T, current number of training rounds $E_c = 0$;
    for $E_c < E$ do
        for $(\mathbf{x}_l, \mathbf{y}_l), \mathbf{x}_u$ in D_l, D_u do
            Use (16) to update the parameters of S, and then
            the teacher network T is regularly changed accord-
            ing to the parameters of the student network S
            through online integration;
        end for
        $E_c = E_c + 1$;
    end for
return Student network S;

4.1. Infrared ship dataset

The images of the infrared ship dataset come from image frames extracted from real infrared ship videos. After manual deduplication, a total of 4,671 infrared ship images are included, and each image has a corresponding pixel-level label and image-level label. The dataset is divided into training set and testing set according to the ratio of 4:1. Finally, the training set contains 3,743 images and the testing set contains 928 images. In this dataset, the pixel category includes 10 foreground categories and 1 background category. Among them, the background category is removed from the image category. The foreground categories are *jyj, qwc, tc_qzc_sag, yyc, hwj, lqt, myyyc, qt, qzj* and *slj*. The

Table 1
Information of the infrared ship dataset.

Category	jyj	qwc	tc_qzc_sag	yyc	hwj	lqt	myyyc	qt	qzj	slj	Total
Quantity	504	161	600	629	614	350	600	210	537	466	4671



(a) Image

(b) Ground Truth

Fig. 3. Examples of Infrared Ships.

dataset information is shown in Table 1, whereas Fig. 3 shows some examples of images and their label images.

4.2. Evaluation indicators

We select the mean Intersection over Union (mIoU), network parameter and floating point operations per second as the evaluation indicators of the experimental results.

The Intersection over Union (IoU) of each class refers to the ratio of the intersection area and the merged area between the predicted area and the true value area of the class, with a range between 0 and 1. Assuming that the predicted result of class k is P_k, the actual value is T_k, the calculation formula is:

$$IoU_k = \frac{P_k \bigcap T_k}{P_k \bigcup T_k} \tag{17}$$

From formula (17), IoU can be expressed as $TP/(TP+FP+FN)$. Among them, TP is the actual category is a certain category and the predicted result is also this category. FP is the actual category is a certain category but predicted to be other categories. FN is that the actual category should not be of this type but is predicted to be of this type. Y_{ij} indicates that the actual class is the j-th class, and the predicted class is the i-th class. The IoU of the k-th category can be expressed as:

$$IoU_k = \frac{Y_{kk}}{\sum_{i=0}^{C} Y_{ik} + \sum_{j=0}^{C} Y_{kj} - Y_{kk}} \tag{18}$$

where C represents all the pixel categories (total of 10 categories), $k=0$ indicates that the class k is the background category, and $k \in [1,10]$ is the target category.

The mIoU of C classes can be expressed as:

$$mIoU = \frac{1}{C} \sum_{k=1}^{C} IoU_k \tag{19}$$

The network parameter (Parameters) [84] is the sum of the parameter quantities of all convolutional layers and fully-connected layers in the network, which represents the size of the network model.

Floating point operations per second (FLOPs) [85] represents the calculation amount of the network, which is mainly used to measure the complexity of the network.

4.3. Experimental details

The target segmentation branch contains two learning networks and one target segmentation network, with the same structure. In the experiments, the Adam optimizer [86–88] is chosen for branch optimization. The initial learning rate is 2e-4, the training epoches of both the dual learning network and the target segmentation network are 130, and the minibatch size is 12.

The error segmentation pixel correction branch contains a student network S and a teacher network T, with the same structure. This study uses ResNet50 pre-trained by ImageNet network as the backbone network. In this branch, stochastic gradient descent is used for network optimization. The base learning rate is 3e-2, the momentum term is 0.9, the weight decay is 1e-5, the epoch is 30, and the minibatch size is 8.

The input image size for both branches above is $256 \times 256 \times 1$. All experiments are run using the Pytorch [89] framework on an Intel(R) Xeon(R) E5-2603 v4 server with NVIDIA Tesla K40c 11G GPU.

4.4. Impact of target segmentation branch on the initial segmentation result

In order to explore the impact of target segmentation branch on the initial segmentation result, the following three sets of comparative experiments are designed:

(1) The segmentation network and the two learning networks are all SegNet, which are labeled as seg-SegNet, A-SegNet and B-SegNet, respectively. All three networks use a random initialization method to generate initialization parameters.
(2) The segmentation network and the two learning networks are both UNet, which are marked as seg-UNet, A-UNet and B-UNet, respectively. All three networks use a random initialization method to generate initialization parameters.
(3) The segmentation network and the two learning networks are both DeepLabV3+, which are marked as seg-DeepLabV3+, A-DeepLabV3+ and B-DeepLabV3+, respectively. The encoders of the three networks use ResNet101 and are all initialized with Kaiming [90].

Table 2 shows the mIoU of different segmentation networks and learning networks on the infrared ship dataset when the proportion of labeled data is 1/8, where the bold numbers indicate the best mIoU, and Fig. 4 shows the corresponding visualization results.

According to Table 2 and Fig. 4, we can find that:

Table 2
mIoU(%) of different segmentation networks and learning networks on the infrared ship dataset.

Class	seg-SegNet	A-SegNet	B-SegNet	seg-UNet	A-UNet	B-UNet	seg-DeepLabV3+	A-DeepLabV3+	B-DeepLabV3+
background	99.62	99.54	99.53	99.66	99.65	99.64	99.59	99.56	99.56
jyj	82.64	72.53	72.12	86.78	83.63	79.33	81.87	76.34	73.96
qwc	10.23	2.15	7.34	54.27	25.71	39.75	27.45	16.84	15.60
tc_qzc_sag	89.82	85.64	86.91	91.12	89.64	87.37	88.76	87.47	86.22
yyc	84.23	75.54	78.62	89.82	89.37	89.52	85.72	84.45	84.68
hwj	75.64	67.19	66.56	79.89	78.39	77.37	78.36	79.72	78.06
lqt	85.45	73.88	76.83	88.29	84.11	86.25	86.48	84.04	84.27
myyyc	85.46	79.00	79.58	88.78	88.93	88.12	86.01	85.26	86.76
qt	54.83	46.44	48.46	79.28	59.39	71.53	80.10	76.36	75.04
qzj	84.80	75.99	78.46	88.71	88.74	88.30	85.49	82.53	83.20
slj	70.92	63.14	68.06	82.31	75.33	85.19	77.56	80.73	78.29
mIoU	74.88	67.37	69.28	**84.45**	78.44	81.12	79.76	77.57	76.88

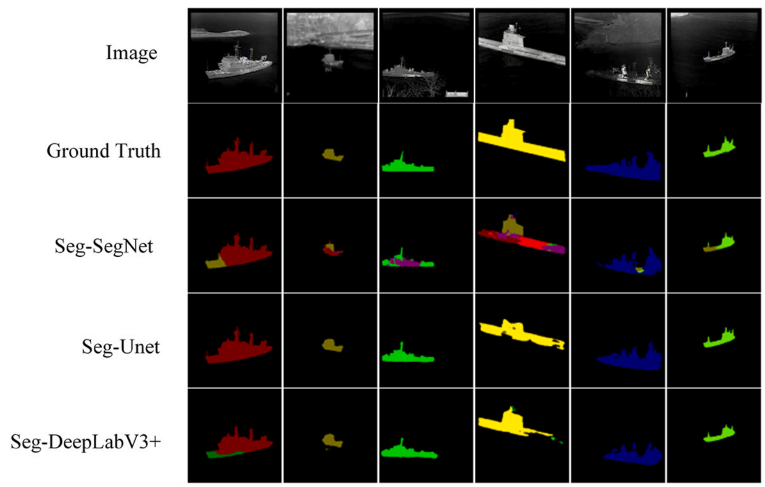


Fig. 4. Visualization results of different segmentation networks on the infrared ship dataset.

(1) When using different segmentation models, the three segmentation networks (seg-SegNet, seg-UNet, seg-DeepLabV3+) performed best on the tc_qzc_sag category, with mIoU of 89.82%, 91.12%, 88.76%, respectively, and performed worst on the qwc category, with mIoU are 10.23%, 54.27% and 27.45%, respectively. It indicates that all networks have poor segmentation effects on small-scale targets such as qwc, but better on large-scale targets such as tc_qzc_sag. The reason is that the information of small-scale targets is difficult to capture, but UNet has shown better performance in the three networks.
(2) When using different segmentation models, there are differences in the segmentation results produced by the two learning networks. For example, the mIoU of A-SegNet and B-SegNet are 67.37% and 69.28%, respectively. The mIoU of A-UNet and B-UNet are 78.44% and 81.12%, respectively. The mIoU of A-DeepLabV3+ and B-DeepLabV3+ are 77.57% and 76.88%, respectively. It demonstrates that the dual learning loss can make the segmentation results obtained by the two learning networks different, which helps to generate high-quality pseudo labels.
(3) The segmentation effect of seg-UNet and its corresponding dual learning network (A-UNet and B-UNet) is the best, with mIoU are 84.45%, 78.44%, 81.12%, respectively. It shows that the UNet network is more suitable for small sample dataset.

(4) Compared with the segmentation results of the corresponding two learning networks, the segmentation results of the three segmentation networks (seg-SegNet, seg-UNet, seg-DeepLabV3+) are improved, with the highest improvements of 5.6%, 3.33%, and 2.19%. The reason is that this method uses a dual learning loss in training to make the two learning networks produce different segmentation results as much as possible. The two networks are encouraged to identify different regions and capture more information. This can produce a better pseudo label for unlabeled data and improve the segmentation performance of the segmentation network.

4.5. Impact of adaptive weight on segmentation results

In order to explore the impact of adaptive weight on the segmentation network, we use three ways to set weights for unlabeled data:

(1) The weights of unlabeled data are all set to 1 and expressed as TSB only.
(2) The IoU of the non-background pixel in the two segmentation maps is taken as the power of the exponential function. The exponential function is used as the adaptive weight of the unlabeled data, which is expressed as TSB (with EX).
(3) The IoU of the non-background pixel in the two segmentation images is used as the adaptive weight of the unlabeled data, expressed as TSB (with AW).

Table 3 lists the mIoU obtained using these three weights, where bold numbers indicate the best results. Fig. 5 shows the segmentation results for different images.

According to Table 3 and Fig. 5, we can conclude that:

(1) Compared with the result of the method without adaptive weight (80.34%), the segmentation results of TSB (with EX) and TSB (with

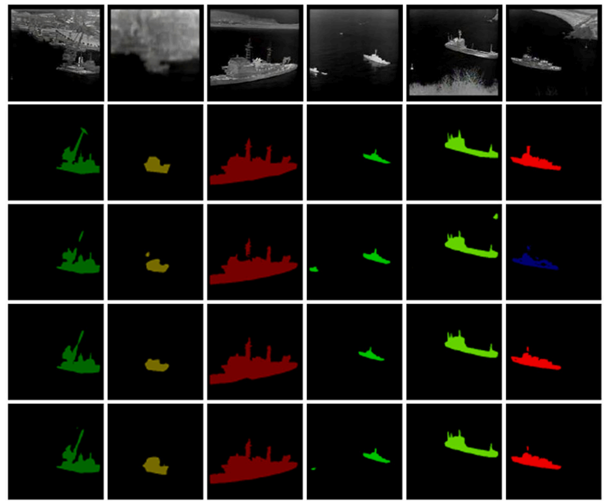


Fig. 5. Segmentation results of different adaptive weights on the infrared ship dataset.

Table 3
Performance of different target segmentation branches.

Method	mIoU(%)
TSB only	80.34
TSB (with EX)	83.28
TSB (with AW)	**84.45**

Table 4
Influence of error segmentation pixel correction branch on segmentation performance.

Method	mIoU(%)	Para(MB)	GFlops
SeISTS-DB(without MSP)	84.45	**197.56**	**120.36**
SeISTS-DB	**86.67**	521.98	140.84

AW) increase by 2.94% and 4.11%, respectively. This indicates that the use of adaptive weight can improve the segmentation performance of the network. This is because treating different quality pseudo labels differently can reduce the impact of poor-quality pseudo labels on network performance.

(2) Compared with the result of TSB (with EX) (83.28%), the segmentation result of TSB (with AW) improved 1.17%. This demonstrates that using the IoU of the non-background pixel in the two segmentation maps as the adaptive weight is more able to capture the small differences between the pseudo labels, which helps to improve the segmentation performance.

4.6. Impact of error segmentation pixel correction branch on segmentation results

In order to verify the influence of error segmentation pixel correction branch on the final segmentation result, models with and without this branch were trained separately to obtain segmentation results. Among them, labeled data accounts for 1/8 of the training set. Table 4 shows the segmentation results on the infrared ship dataset with or without error segmentation pixel correction branch, where the bold number indicates the best result, and SeISTS-DB (without MSP) indicates the model without error segmentation pixel correction branch. Fig. 6 shows the corresponding segmentation examples.

According to Table 4 and Fig. 6, we can get that:

(1) SeISTS-DB obtains the mIoU of 86.67%, which is 2.22% higher than the result without error segmentation pixel correction branch. This denotes the error segmentation pixel correction branch can correct the wrongly segmented pixels in the target segmentation branch and increase the accuracy of the image segmentation results.

(2) The parameter quantity and computation quantity of SeISTS-DB are 521.98MB and 140.84GFlops, which are respectively increased by 324.42MB and 20.48GFlops compared with the results without error segmentation pixel correction branch. It that shows that the increase of error segmentation pixel correction branch will increase parameter quantity and computation quantity.

(3) In short, the increase of error segmentation pixel correction branch can improve the segmentation effect of the model, and increase the parameter quantity and computation quantity to a certain extent.

4.7. Comparison SeISTS-DB with fully-supervised methods

The results of the proposed SeISTS-DB method and three fully-supervised methods (SegNet [18], UNet [16], DeepLabV3+ [42]) at different labeled data ratios were compared. Table 5 shows the segmentation results of SeISTS-DB and the fully-supervised method on the infrared ship dataset, where the bold data indicates the best results. Fig. 7 shows the segmentation effect of our method and the segmentation example of the fully-supervised network.

It can be seen from Table 5 and Fig. 7 that:

(1) When the proportion of labeled data is 1/8, 1/4, 1/2 and 1, SeISTS-DB obtains mIoU of 86.67%, 89.10%, 91.10% and 91.15% respectively, which is 15.35%, 4.37%, 1.9% and 0.12% higher than the results of the other three methods. It signifies that the SeISTS-DB method can better learn the characteristics of the data and achieve a good segmentation performance.

(2) The parameter quantity of SeISTS-DB is 521.98MB, which increases 409.64MB, 456.13MB, and 295.61MB respectively compared with the other three methods. The computation quantity of SeISTS-DB is 140.84Flops, which increases 100.36GFlops, 100.72GFlops, and 118.63GFlops respectively compared with the other three methods. It shows that SeISTS-DB has more parameter quantity and computation quantity than the other three networks. Except that SeISTS-DB has 3 identical segmentation networks, other networks have only one segmentation network. Thus, SeISTS-DB has more parameter quantity and computation quantity.

(3) In short, the SeISTS-DB method can learn the features of unlabeled data when there is less labeled data, so as to achieve a better segmentation performance. However, it should be noted that it

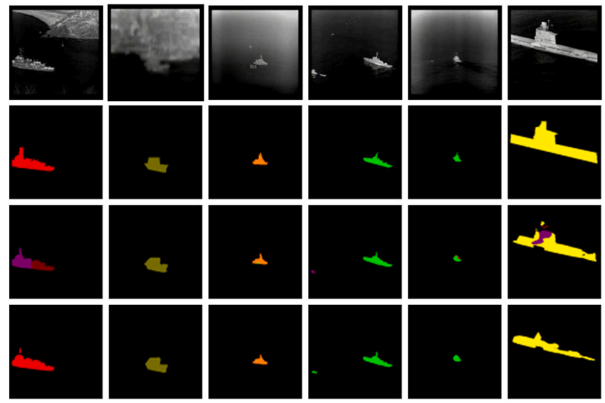


Fig. 6. Segmentation results with or without error segmentation pixel correction branch.

Table 5
Segmentation performance of different methods on the infrared ship dataset.

Methods	mIoU(%)				Para(MB)	GFlops
	1/8	1/4	1/2	1		
SegNet [18]	66.13	76.39	88.56	90.75	112.34	40.48
UNet [16]	71.32	84.73	89.20	91.03	**65.85**	40.12
DeepLabV3+ [42]	70.67	83.13	88.51	90.39	226.37	**22.21**
SeISTS-DB	**86.67**	**89.10**	**91.10**	**91.15**	521.98	140.84

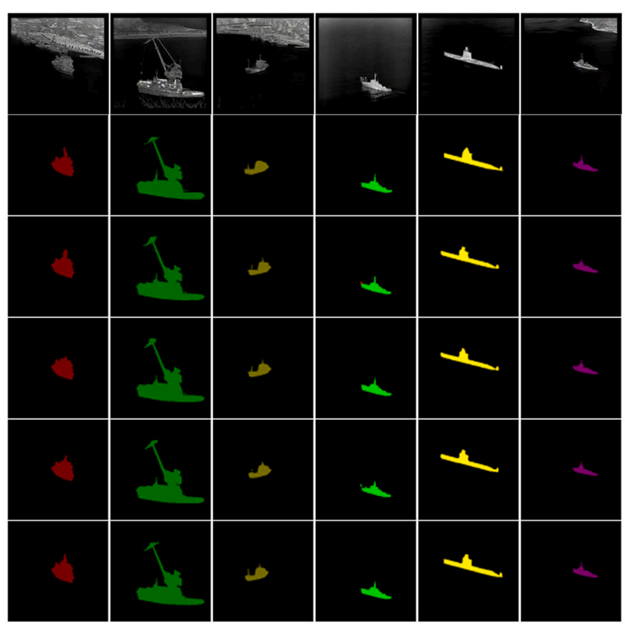


Fig. 7. Segmentation results with or without error segmentation pixel correction branch.

needs to consume more parameter quantity and computation quantity.

4.8. Comparison SeISTS-DB with semi-supervised methods

In the section, 1/8 of the training set is taken as labeled data, and the rest of the data is taken as unlabeled data. The segmentation performance of the proposed method is compared with other semi-supervised methods. Table 6 shows the segmentation results of different methods on the infrared ship dataset and the parameter quantity and floating point operations per second of the network, where the bold numbers indicate the best results. Fig. 8 shows examples of segmentation results of different networks on the infrared ship test set.

According to Table 6 and Fig. 8, we can conclude that:

(1) The SeISTS-DB method obtains an mIoU of 86.67%, which increases by 49.51%, 16.11%, 16.18%, and 6.19% compared with the results of AdvSemiSeg [63], S4GAN-MLMT [65], CCT [56], and Cycle-GAN [64], respectively. This shows that SeISTS-DB can achieve better segmentation performance.
(2) The parameter quantity of SeISTS-DB is 521.98MB, which increases by 420.18MB, 7.87MB, 343.66MB, and 107.58MB compared with

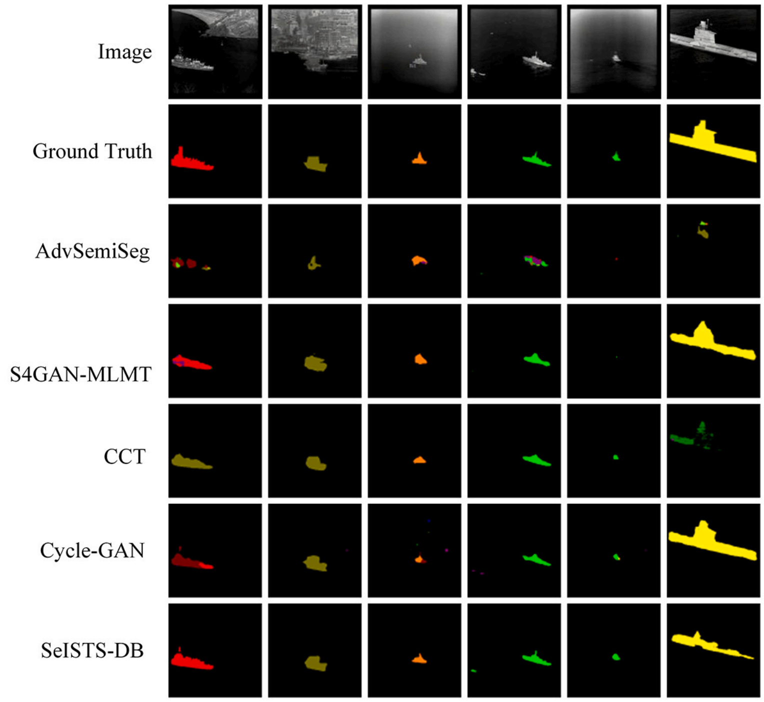


Fig. 8. Infrared ship segmentation results of different methods.

Table 6
The results of different semi-supervised object segmentation methods on the infrared ship dataset.

Method	mIoU(%)	Para(MB)	GFlops
AdvSemiSeg [63]	37.16	**101.80**	**28.69**
S4GAN-MLMT [65]	70.56	514.11	69.94
CCT [56]	70.49	178.32	46.16
Cycle-GAN [64]	80.48	413.4	194.67
SeISTS-DB	**86.67**	521.98	140.84

Table 7
Results of different semi-supervised object segmentation methods on the IRSTD-1k dataset.

Methods	mIoU(%)
AdvSemiSeg [63]	68.39
S4GAN-MLMT [65]	63.65
Cycle-GAN [64]	60.72
SeISTS-DB	**80.15**

the results of AdvSemiSeg [63], S4GAN-MLMT [65], CCT [56], and Cycle-GAN [64], respectively. The computation quantity of SeISTS-DB is 140.84 GFlops, which increases by 112.15GFlops, 70.9GFlops, 343.66GFlops, 94.65GFlops compared with the results of AdvSemiSeg [63], S4GAN-MLMT [65], and CCT [56], and decreases by 53.83GFlops compared with Cycle-GAN [64]. Therefore, it can be concluded that SeISTS-DB has more parameter quantity and computation quantity.

(3) With the correction of mis-segmented pixels, SeISTS-DB achieves higher mIoU. However, this requires more parameter quantity and computation quantity, which needs to be further improved.

4.9. Comparison of different methods on IRSTD-1k dataset

In order to further verify the performance of the method SeISTS-DB, we use the public infrared dataset IRSTD-1k [91] for experiments. The IRSTD-1k dataset consists of 1,160 infrared images, of which 928 images are training set data and 232 images are testing set data. The size of the image is 416×416. This dataset mainly segments ships in infrared images. The images include ships of varying sizes and numbers. Among them, the ship in the image is a category, and other environments are treated as the background.

In the experiment, 1/8 of the training set in the IRSTD-1k dataset is used as labeled data, and the rest of the data is used as unlabeled data. Table 7 shows the segmentation performance of different semi-supervised methods on the IRSTD-1k dataset, where the bold numbers indicate the best segmentation results. Fig. 9 shows examples of segmentation results of different networks on the IRSTD-1k testing set.

According to Table 7 and Fig. 9, it can be seen that the SeISTS-DB method obtained an mIoU of 80.15%, which increases by 11.76%, 16.40%, 16.18%, and 19.45% compared with the results of the AdvSemiSeg, S4GAN-MLMT, and Cycle-GAN methods, respectively. Therefore, it can be concluded that the SeISTS-DB method obtains a more accurate segmentation map by correcting wrongly segmented pixels.

5. Discussion

In this present study, our method has yielded better mIoUs due to the introduction of dual learning network and the error segmentation pixel correction branch.

First, the dual learning network can get pseudo label that contains more information. In previous work, the self-training method used single network approach to generate pseudo label for unlabeled data. This proves that it is effective to generate pseudo label on unlabeled data to enrich the dataset. However, the newly generated pseudo label have equal weights when used to reverse train the network, which makes the good quality pseudo label and the poor quality pseudo label play the same role in updating the network weights, thus causing the problem of confirmation bias of the network. We introduced the dual learning network combines the output of the two networks to generate pseudo label and adaptive weights. Our experiments show that the target segmen-

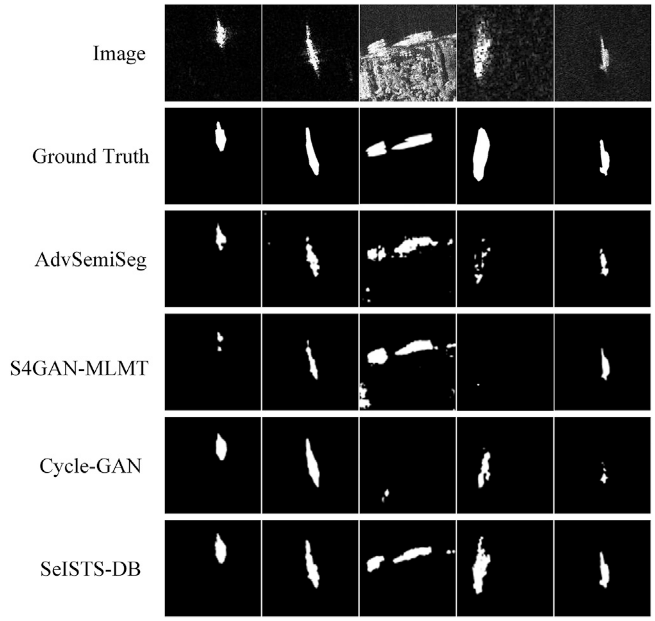


Fig. 9. Segmentation results of different segmentation methods on the IRSTD-1k dataset.

tation branch with dual learning network can generate more accurate segmentation result with location information.

Further, the error segmentation pixel correction branch can extract category information into image. Although the segmentation map obtained by the target segmentation branch contains accurate location information of the image, we find that there are misclassified pixels in the segmentation graph. To get the category information of the image, we designed this the error segmentation pixel correction branch. This branch is trained in a semi-supervised manner and is able to obtain category information in images. The results of these two branches are fused to produce more accurate segmentation maps of location and category information.

However, our method still has shortcomings. It has more parameter quantity and computation quantity. Besides, the two branches are trained separately and are unrelated during the training process, needing a relatively larger amount of GPU memory. Subsequent research will consider combining the two branches to make the training process more complete. We will explore to add a classification head to the segmentation network, so that the segmentation network can extract both semantic information and category information in the training process [92–94]. By this way, it can reduce the amount of parameters of the network and accelerate the iteration speed of the network.

6. Conclusion

In this paper, a semi-supervised infrared ship target segmentation model with dual branch is proposed, which obtains better segmentation results through the interaction of the two branches. The dual learning module in target segmentation branch can get high quality pseudo labels. Meanwhile, the balance of the weights of the pseudo labels can enable the network to accurately extract image features. The target segmentation branch can combine the segmentation results of the dual learning module to generate pseudo labels and corresponding weights, avoiding the problem of confirmation bias caused by single network. Additionally, the error segmentation pixel correction branch can correct the wrongly segmented pixels in the segmentation map, so that the final segmentation map is more accurate. Our method achieved higher accuracies on both the infrared ship image dataset and the public IRSTD-1k dataset.

However, our method has still limitations. It has a large number of parameters and consume more GPU memory than other models. In the future, we will explore to add a classification module at the tail of the segmentation network.

Declaration of competing interest

The authors declare no conflict of interest.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (61806013, 61876010, 61906005, 62166002, 62176009), General project of Science and Technology Plan of Beijing Municipal Education Commission (KM202110005028), Project of Interdisciplinary Research Institute of Beijing University of Technology (2021020101) and International Research Cooperation Seed Fund of Beijing University of Technology (2021A01).

Appendix A

Table 8
Parameter notation table.

Notation	Description
$c, c \in \{0,1,2,3,4,\ldots,9,10\}$	The number of categories in the infrared ship image, defined by Equation (1).
x	The infrared ship image, defined by Equation (1).
$S(x)_c$	The segmentation result of the c-th category in the target segmentation map of the target segmentation branch, defined by equation (1).
$G(x)_c$	$G(x)_c$ The predicted probability of the c-th category of the error segmentation pixel correction branch, defined by equation (1).
A, B	The learning networks A and learning networks B, defined by Equation (2).
${L_{ce}}^{A,B}$	The cross-entropy loss of learning network A and learning network B, defined by Equation (2).
D_l	The labeled dataset, defined by Equation (2).
$P_{sm}()$	Represent the softmax function, defined by Equation (3).
L_{mse}	The dual learning loss of learning network A and learning network B, defined by Equation (4).
L	The dual learning network loss, defined by Equation (5).
$\arg\max()$	A function that represents the index of the largest value, defined by Equation (7).
$t_{pseudo,A}$	The segmentation map generated by learning network A on unlabeled data, defined by Equation (8).
$t_{pseudo,B}$	The segmentation map generated by learning network B on unlabeled data, defined by Equation (9).
${t_{pseudo,A}}', {t_{pseudo,B}}'$	The segmentation map that is non-background pixels of unlabeled data, defined by Equation (10).
ω	The adaptive weight of the pseudo label, defined by Equation (10).
D_u	The unlabeled dataset, defined by Equation (11).
L_{adaw}	The cross-entropy loss with adaptive weights for unlabeled data in the segmentation network S, defined by Equation (11).
L_{ces}	The cross-entropy loss for labeled data in the segmentation network S, defined by Equation (12).
L_S	The complete loss of the segmentation network S, defined by Equation (13).
$f_g()$	The student network function, defined by Equation (14).
L_{cec}	The classification cross-entropy loss of the student network, defined by Equation (14).
L_{cons}	The consistency loss of the student network, defined by Equation (15).
$f_h()$	The teacher network function, defined by Equation (15).
L_p	The total loss of the student network, defined by Equation (16).

Table 9
Abbreviation notation table.

SeISTS-DB	A semi-supervised method for infrared ship target segmentation with dual branch.
FCN	Fully convolutional network.
UNet	U-shape fully convolutional Network.
SegNet	Segmentation Network.
CutMix	A data augmentation method based on cross-fusion of image blocks.
ClassMix	A data augmentation method based on pixel cross-fusion.
A	The learning network of the target segmentation branch.
B	The learning network of the target segmentation branch.
F	Segmentation network of the target segmentation branch.
S	Student network of the error segmentation pixel correction branch.
T	Teacher network of the error segmentation pixel correction branch.
TSB	Target segmentation branch.
P_i,i∈ {1,2}	Segmentation confidence map generated by learning network A and B.
Y_i,i∈ {1,2}	Pseudo label corresponds to segmentation confidence map P_i.
P	The overall segmentation confidence map that is the sum of the segmentation maps of learing network A and B.
Y'	Pseudo label corresponds to segmentation confidence map P.
mIou	The mean Intersection over Union.
FLOPs	Floating point operations per second.
TSB (with EX)	The adaptive weight of the unlabeled data.
TSB (with AW)	The two segmentation images is used as the adaptive weight of the unlabeled data.
SeISTS-DB (without MSP)	The SeISTS-DB model without error segmentation pixel correction branch.

References

[1] H. Tian, N. Huang, Z. Niu, Y. Qin, J. Pei, J. Wang, Mapping winter crops in China with multi-source satellite imagery and phenology-based algorithm, Remote Sens. 11 (7) (2019), https://doi.org/10.3390/rs11070820.
[2] H. Tian, J. Pie, J. Huang, X. Li, J. Wang, B. Zhou, Y. Qin, L. Wang, Garlic and winter wheat identification based on active and passive satellite imagery and the google earth engine in northern china, Remote Sens. 12 (21) (2020) 3539.
[3] H. Liu, Y. Xu, F. Chen, Sketch2Photo: synthesizing photo-realistic images from sketches via global contexts, Eng. Appl. Artif. Intell. 117 (2023) 105608.
[4] Y. Tao, J. Shi, W. Guo, J. Zheng, Convolutional neural network based defect recognition model for phased array ultrasonic testing images of electrofusion joints, J. Press. Vessel Technol. 145 (2) (2023), https://doi.org/10.1115/1.4056836.
[5] F. Hou, Y. Zhang, Y. Zhou, M. Zhang, B. Lv, J. Wu, Review on infrared imaging technology, Sustainability 14 (18) (2022) 11161.
[6] G. Zhou, W. Li, X. Zhou, Y. Tan, G. Lin, X. Li, R. Deng, An innovative echo detection system with STM32 gated and PMT adjustable gain for airborne LiDAR, Int. J. Remote Sens. 42 (24) (2021) 9187–9211.
[7] T. Zhang, H. Shen, S. ur Rehman, Z. Liu, Y. Li, O. ur Rehman, Two-stage domain adaptation for infrared ship target segmentation, IEEE Trans. Geosci. Remote Sens. (2023).
[8] Z. Song, J. Yang, D. Zhang, S. Wang, Z. Li, Semi-supervised dim and small infrared ship detection network based on haar wavelet, IEEE Access 9 (2021) 29686–29695.
[9] Z. Zhou, R. Deng, X. Zhou, S. Long, W. Li, G. Lin, X. Li, Gaussian inflection point selection for LiDAR hidden echo signal decomposition, IEEE Geosci. Remote Sens. Lett. 19 (2021) 1–5.
[10] X. Bai, Z. Chen, Y. Zhang, Z. Liu, Y. Lu, Infrared ship target segmentation based on spatial information improved fcm, IEEE Trans. Cybern. 46 (12) (2015) 3259–3271.
[11] F. Masood, W. Boulila, A. Alsaeedi, J.S. Khan, J. Ahmad, M.A. Khan, S.U. Rehman, A novel image encryption scheme based on Arnold cat map, Newton-Leipnik system and logistic Gaussian map, Multimed. Tools Appl. 81 (21) (2022) 30931–30959.
[12] D. Jin, X. Bai, Distribution information based intuitionistic fuzzy clustering for infrared ship segmentation, IEEE Trans. Fuzzy Syst. 28 (8) (2019) 1557–1571.
[13] M. Qi, S. Cui, X. Chang, Y. Xu, H. Meng, Y. Wang, T. Yin, Multi-region nonuniform brightness correction algorithm based on L-channel gamma transform, Secur. Commun. Netw. (2022), https://doi.org/10.1155/2022/2675950.
[14] X. Liu, Z. Deng, Y. Yang, Recent progress in semantic image segmentation, Artif. Intell. Rev. 52 (2) (2019) 1089–1106.
[15] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, 2015, pp. 3431–3440.
[16] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
[17] Y. Dong, B. Xu, T. Liao, C. Yin, Z. Tan, Application of local-feature-based 3D point cloud stitching method of low-overlap point cloud to aero-engine blade measurement, IEEE Trans. Instrum. Meas. 72 (2023), https://doi.org/10.1109/TIM.2023.3309384.
[18] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.
[19] Y. Shi, J. Xi, D. Hu, Z. Cai, K. Xu, RayMVSNet++: learning ray-based 1D implicit fields for accurate multi-view stereo, IEEE Trans. Pattern Anal. Mach. Intell. 45 (11) (2023) 13666–13682.
[20] Y. Di, R. Li, H. Tian, J. Guo, B. Shi, Z. Wang, K. Yan, Y. Liu, A maneuvering target tracking based on fastIMM-extended Viterbi algorithm, Neural Comput. Appl. (2023) 1–10.
[21] J.E. Van Engelen, H.H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2) (2020) 373–440.
[22] S. Hong, H. Noh, B. Han, Decoupled deep neural network for semi-supervised semantic segmentation, Adv. Neural Inf. Process. Syst. 28 (2015).
[23] W. Wang, C. Su, Semi-supervised semantic segmentation network for surface crack detection, Autom. Constr. 128 (2021) 103786.
[24] H. Tan, T. Ye, S. ur Rehman, O. ur Rehman, S. Tu, J. Ahmad, A novel routing optimization strategy based on reinforcement learning in perception layer networks, Comput. Netw. (2023) 110105.
[25] A. Chartsias, G. Papanastasiou, C. Wang, S. Semple, D.E. Newby, R. Dharmakumar, S.A. Tsaftaris, Disentangle, align and fuse for multimodal and semi-supervised image segmentation, IEEE Trans. Med. Imaging 40 (3) (2020) 781–792.
[26] M.F.A. Hady, F. Schwenker, Semi-supervised learning, in: Handbook on Neural Information Processing, 2013, pp. 215–239.
[27] H. Hu, F. Wei, H. Hu, Q. Ye, J. Cui, L. Wang, Semi-supervised semantic segmentation via adaptive equalization learning, Adv. Neural Inf. Process. Syst. 34 (2021) 22106–22118.
[28] I. Alonso, A. Sabater, D. Ferstl, L. Montesano, A.C. Murillo, Semi-Supervised Semantic Segmentation with Pixel-Level Contrastive Learning from a Class-Wise Memory Bank, 2021, pp. 8219–8228.
[29] R. He, J. Yang, X. Qi, Re-Distributing Biased Pseudo Labels for Semi-Supervised Semantic Segmentation: A Baseline Investigation, 2021, pp. 6930–6940.
[30] C. Mi, S. Huang, Y. Zhang, Z. Zhang, O. Postolache, Design and implementation of 3-D measurement method for container handling target, J. Mar. Sci. Eng. 10 (12) (2022) 1961.
[31] M. Yang, H. Wang, K. Hu, G. Yin, Z. Wei, IA-Net: an inception–attention-module-based network for classifying underwater images from others, IEEE J. Ocean. Eng. 47 (3) (2022) 704–717.
[32] S. Minaee, Y.Y. Boykov, F. Porikli, A.J. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: a survey, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[33] Y. Zheng, P. Liu, L. Qian, S. Qin, X. Liu, Y. Ma, G. Cheng, Recognition and depth estimation of ships based on binocular stereo vision, J. Mar. Sci. Eng. 10 (8) (2022) 1153.
[34] Z. Zhou, X. Liu, Orthorectification model for extra-length linear array imagery, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–10.
[35] K. Shang, L. Xu, X. Liu, Z. Yin, Z. Liu, X. Li, L. Yin, W. Zheng, Study of urban heat island effect in Hangzhou metropolitan area based on SW-TES algorithm and image dichotomous model, SAGE Open 13 (4) (2023) 21582440231208851.
[36] R. Zhang, L. Li, Q. Zhang, J. Zhang, L. Xu, B. Zhang, B. Wang, Differential feature awareness network within antagonistic learning for infrared-visible object detection, IEEE Trans. Circuits Syst. Video Technol. (2023), https://doi.org/10.1109/TCSVT.2023.3289142.
[37] W. Dong, J. Zhoa, J. Qu, S. Xiao, N. Li, S. Hou, Y. Li, Abundance matrix correlation analysis network based on hierarchical multihead self-cross-hybrid attention for hyperspectral change detection, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–13.
[38] W. Dong, Y. Yang, J. Qu, X. Xiao, Y. Li, Local information enhanced graph-transformer for hyperspectral image change detection with limited training samples, IEEE Trans. Geosci. Remote Sens. 61 (2023), https://doi.org/10.1109/TGRS.2023.3269892.
[39] Y. Wang, N. Xu, A. Liu, W. Li, Y. Zhang, High-order interaction learning for image captioning, IEEE Trans. Circuits Syst. Video Technol. 32 (7) (2021) 4417–4430.
[40] H. Liu, H. Yuan, Q. Liu, J. Hou, H. Zheng, S. Kwong, A hybrid compression framework for color attributes of static 3D point clouds, IEEE Trans. Circuits Syst. Video Technol. 32 (3) (2021) 1564–1577.
[41] Q. Liu, H. Yuan, R. Hamzaoui, H. Su, J. Hou, H. Yang, Reduced reference perceptual quality model with application to rate control for video-based point cloud compression, IEEE Trans. Image Process. 30 (2021) 6623–6636.

[42] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 801–818.
[43] 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 (4) (2017) 834–848.
[44] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587, 2017.
[45] T. Takikawa, D. Acuna, V. Jampani, S. Fidler, Gated-SCNN: gated shape CNNs for semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5229–5238.
[46] A. Kirillov, Y. Wu, K. He, R. Girshick, PointRend: image segmentation as rendering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9799–9808.
[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[48] S. Tu, S.U. Rehman, M. Waqas, O.U. Rehman, Z. Shah, Z. Yang, A. Koubaa, ModPSO-CNN: an evolutionary convolution neural network with application to visual recognition, Soft Comput. 25 (2021) 2165–2176.
[49] S. Tu, S. ur Rehman, M. Waqas, O. u. Rehman, Z. Yang, B. Ahmad, Z. Halim, W. Zhao, Optimisation-based training of evolutionary convolution neural network for visual classification applications, IET Comput. Vis. 14 (5) (2020) 259–267.
[50] S. ur Rehman, S. Tu, M. Waqas, Y. Huang, O. ur Rehman, B. Ahmad, S. Ahmad, Unsupervised pre-trained filter learning approach for efficient convolution neural network, Neurocomputing 365 (2019) 171–190.
[51] S. ur Rehman, Y. Huang, S. Tu, B. Ahmad, Learning a semantic space for modeling images, tags and feelings in cross-media search, in: Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2019 Workshops, BDM, DLKT, LDRC, PAISI, WeL, Revised Selected Papers 23, Macau, China, April 14–17, 2019, Springer, 2019, pp. 65–76.
[52] S.U. Rehman, S. Tu, O.U. Rehman, Y. Huang, C.M.S. Magurawalage, C.-C. Chang, Optimization of cnn through novel training strategy for visual classification problems, Entropy 20 (4) (2018) 290.
[53] S. Shim, J. Kim, G.-C. Cho, S.-W. Lee, Multiscale and adversarial learning-based semi-supervised semantic segmentation approach for crack detection in concrete structures, IEEE Access 8 (2020) 170939–170950.
[54] G. Li, J. Wan, S. He, Q. Liu, B. Ma, Semi-supervised semantic segmentation using adversarial learning for pavement crack detection, IEEE Access 8 (2020) 51446–51459.
[55] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: an overview, IEEE Signal Process. Mag. 35 (1) (2018) 53–65.
[56] Y. Ouali, C. Hudelot, M. Tami, Semi-supervised semantic segmentation with cross-consistency training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12674–12684.
[57] O.U. Rehman, S. Tu, S.U. Rehman, S. Khan, S. Yang, Design optimization of electromagnetic devices using an improved quantum inspired particle swarm optimizer, Appl. Comput. Electromagn. Soc. J. (2018) 951–956.
[58] S. Rehman, S. Tu, Y. Huang, G. Liu, et al., CSFL: a novel unsupervised convolution neural network approach for visual pattern classification, AI Commun. 30 (5) (2017) 311–324.
[59] S.U. Rehman, S. Tu, Y. Huang, Z. Yang, Face recognition: a novel un-supervised convolutional neural network method, in: 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS), IEEE, 2016, pp. 139–144.
[60] G. French, T. Aila, S. Laine, M. Mackiewicz, G. Finlayson, Consistency regularization and cutmix for semi-supervised semantic segmentation, arXiv preprint arXiv:1906.01916 2 (4) (2019) 5.
[61] V. Olsson, W. Tranheden, J. Pinto, L. Svensson, ClassMix: segmentation-based data augmentation for semi-supervised learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1369–1378.
[62] N. Souly, C. Spampinato, M. Shah, Semi supervised semantic segmentation using generative adversarial network, in: Proceedings of International Conference on Computer Vision, 2017, pp. 5688–5696.
[63] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, M.-H. Yang, Adversarial learning for semi-supervised semantic segmentation, arXiv preprint arXiv:1802.07934, 2018.
[64] A.K. Mondal, A. Agarwal, J. Dolz, C. Desrosiers, Revisiting cyclegan for semi-supervised segmentation, arXiv preprint arXiv:1908.11569, 2019.
[65] S. Mittal, M. Tatarchenko, T. Brox, Semi-supervised semantic segmentation with high-and low-level consistency, IEEE Trans. Pattern Anal. Mach. Intell. 43 (4) (2019) 1369–1379.
[66] Y. Chen, M. Mancini, X. Zhu, Z. Akata, Semi-supervised and unsupervised deep visual learning: a survey, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, 2022, [Online], available: https://api.semanticscholar.org/CorpusID:251765280.
[67] C. Zhao, C.F. Cheung, P. Xu, High-efficiency sub-microscale uncertainty measurement method using pattern recognition, ISA Trans. 101 (2020) 503–514.
[68] X. Chen, Y. Yuan, G. Zeng, J. Wang, Semi-supervised semantic segmentation with cross pseudo supervision, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2021, pp. 2613–2622.
[69] Z. Cui, H. Sheng, D. Yang, S. Wang, R. Chen, W. Ke, Light field depth estimation for non-lambertian objects via adaptive cross operator, IEEE Trans. Circuits Syst. Video Technol. (2023), https://doi.org/10.1109/TCSVT.2023.3292884.
[70] J. Li, N. Zhou, J. Sun, S. Zhou, Z. Bai, L. Lu, Q. Qen, C. Zuo, Transport of intensity diffraction tomography with non-interferometric synthetic aperture for three-dimensional label-free microscopy, Light: Sci. Appl. 11 (1) (2022) 154.
[71] Y. Li, J. Qian, S. Feng, Q. Chen, C. Zuo, Deep-learning-enabled dual-frequency composite fringe projection profilometry for single-shot absolute 3D shape measurement, Opto-Electron. Adv. 5 (5) (2022) 210021.
[72] H. Cevikalp, B. Benligiray, Ö.N. Gerek, H. Saribas, Semi-Supervised Robust Deep Neural Networks for Multi-Label Classification, 2019, pp. 9–17.
[73] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2018) 1979–1993.
[74] Q. Liu, L. Yu, L. Luo, Q. Dou, P.A. Heng, Semi-supervised medical image classification with relation-driven self-ensembling model, IEEE Trans. Med. Imaging 39 (11) (2020) 3429–3440.
[75] J. Enguehard, P. O'Halloran, A. Gholipour, Semi-supervised learning with deep embedded clustering for image classification and segmentation, IEEE Access 7 (2019) 11093–11104.
[76] R. Lan, L. Sun, Z. Liu, H. Lu, Z. Su, C. Pang, X. Luo, Cascading and enhanced residual networks for accurate single-image super-resolution, IEEE Trans. Cybern. 51 (1) (2020) 115–125.
[77] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[78] N. Bao, T. Zhang, R. Huang, S. Biswal, J. Su, Y. Wang, A deep transfer learning network for structural condition identification with limited real-world training data, Struct. Control Health Monit. 2023 (2023) 1–18.
[79] H. Sheng, S. Wang, H. Chen, D. Yang, Y. Huang, J. Shen, W. Ke, Discriminative feature learning with co-occurrence attention network for vehicle ReID, IEEE Trans. Circuits Syst. Video Technol. (2023), https://doi.org/10.1109/TCSVT.2023.3326375.
[80] A. Tarvainen, H. Valpola, Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results, Adv. Neural Inf. Process. Syst. 30 (2017).
[81] Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, Adv. Neural Inf. Process. Syst. 31 (2018).
[82] F. Wang, J. Cheng, W. Liu, H. Liu, Additive margin softmax for face verification, IEEE Signal Process. Lett. 25 (7) (2018) 926–930.
[83] O. Köksoy, Multiresponse robust design: mean square error (mse) criterion, Appl. Math. Comput. 175 (2) (2006) 1716–1729.
[84] H.-D. Nguyen, V.-D. Pham, Q.-H. Nguyen, V.-M. Pham, M.H. Pham, V.M. Vu, Q.-T. Bui, An optimal search for neural network parameters using the salp swarm optimization algorithm: a landslide application, Remote Sens. Lett. 11 (4) (2020) 353–362.
[85] Y. Li, Y. Chen, X. Dai, D. Chen, M. Liu, L. Yuan, Z. Liu, L. Zhang, N. Vasconcelos, MicroNet: improving image recognition with extremely low flops, in: Proceedings of the International Conference on Computer Vision, 2021, pp. 468–477.
[86] R. Cong, H. Sheng, D. Yang, Z. Cui, R. Chen, Exploiting spatial and angular correlations with deep efficient transformers for light field image super-resolution, IEEE Trans. Multimed. (2023), https://doi.org/10.1109/TMM.2023.3282465.
[87] G. Tian, Y. Hui, W. Lu, W. Tingting, Rate-distortion optimized quantization for geometry-based point cloud compression, J. Electron. Imaging 32 (1) (2023) 013047.
[88] L. Abualigah, Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications, Neural Comput. Appl. 32 (16) (2020) 12381–12401.
[89] Y. Liu, A. Chen, H. Shi, S. Huang, W. Zheng, Z. Liu, Q. Zhang, X. Yang, CT synthesis from mri using multi-cycle gan for head-and-neck radiation therapy, Comput. Med. Imaging Graph. 91 (2021) 101953.
[90] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of the International Conference on Computer Vision, 2015, pp. 1026–1034.
[91] M. Zhang, R. Zhang, Y. Yang, H. Bai, J. Zhang, J. Guo, ISNet: shape matters for infrared small target detection, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2022, pp. 877–886.
[92] J. Chen, M. Xu, W. Xu, D. Li, W. Peng, H. Xu, A flow feedback traffic prediction based on visual quantified features, IEEE Trans. Intell. Transp. Syst. 24 (9) (2023) 10067–10075.
[93] J. Chen, Q. Wang, H.H. Cheng, W. Peng, W. Xu, A review of vision-based traffic semantic understanding in ITSs, IEEE Trans. Intell. Transp. Syst. 23 (11) (2022) 19954–19979.
[94] X. Ma, Z. Dong, W. Quan, Y. Dong, Y. Tan, Real-time assessment of asphalt pavement moduli and traffic loads using monitoring data from Built-in Sensors: Optimal sensor placement and identification algorithm, Mech. Syst. Signal Process. 187 (2023) 109930.