Signal Processing 212 (2023) 109150

Contents lists available at ScienceDirect

Signal Processing

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

FastICENet: A real-time and accurate semantic segmentation model for aerial remote sensing river ice image

Xiuwei Zhang[a,b,c], Zixu Zhao[a,b,c], Lingyan Ran[a,b,c,*], Yinghui Xing[a,b,c], Wenna Wang[a,b,c], Zeze Lan[a,b,c], Hanlin Yin[a,b,c], Houjun He[e], Qixing Liu[d], Baosen Zhang[c,d], Yanning Zhang[a,b,c]

[a] National Engineering Laboratory for Integrated Aerospace-Ground-Ocean Big Data Application Technology, Xi'an 710072, China
[b] Shaanxi Provincial Key Laboratory of Speech & Image Information Processing, Xi'an 710072, China
[c] School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
[d] Yellow River Institute of Hydraulic Research, ZhengZhou 450003, China
[e] Information Center of Yellow River Conservancy Commission, ZhengZhou 450004, China

ARTICLE INFO

Article history:
Received 12 December 2022
Revised 26 April 2023
Accepted 19 May 2023
Available online 16 June 2023

Keywords:
River ice semantic segmentation
Deep learning
Ghost module
DUpsampling

ABSTRACT

River ice semantic segmentation is a crucial task, which can provide us with information for river monitoring, disaster forecasting, and transportation management. Previous works mainly focus on higher accuracy acquirement, while efficiency is also important for reality usage. In this paper, a real-time and accurate river ice semantic segmentation network is proposed, named FastICENet. The general architecture consists of two branches, i.e., a shallow high-resolution spatial branch and a deep context semantic branch, which are carefully designed for the scale diversity and irregular shape of river ice in remote sensing images. Then, a novel Downsampling module and a dense connection block based on a lightweight Ghost module are adopted in the context branch to reduce the computation cost. Furthermore, a learnable upsampling strategy DUpsampling is utilized to replace the commonly used bilinear interpolation to improve the segmentation accuracy. We deploy detailed experiments on three publicly available datasets, named NWPU_YRCC_EX, NWPU_YRCC2, and Alberta River Ice Segmentation Dataset. The experimental results demonstrate that our method achieves state-of-the-art performance with competing methods, on the NWPU_YRCC_EX dataset, we can achieve the segmentation speed as 90.84FPS and the segmentation accuracy as 90.770% mIoU, which also illustrates the good leverage between accuracy and speed. Our code is available at https://github.com/nwpulab113/FastICENet

1. Introduction

Image semantic segmentation remains a conventional challenging task in computer vision, which is widely used in geographic information systems [1,2], autonomous vehicles, medical image analysis, visual surveillance, disaster prediction [3] and so on. In the field of river ice semantic segmentation, accurate segmentation results can provide crucial information for river monitoring [4,5], disaster forecasting [6], and transportation management, especially for rivers with large latitude spans. Meanwhile, the segmentation model is often requested to be deployed to some hydrological stations without powerful computation resources. It is required to build time-efficient and accurate models in reality.

Trustful segmentation results are hard to get for river ices. In most cases, floating ice are of different sizes and in irregular shapes in the image, which requires the network to have the ability to extract multi-scale features. With the fast development of deep learning methods, many excellent semantic segmentation algorithms based on deep convolutional neural networks have been proposed. Most of them would adopt a backbone ResNet [7], DeepLab [8], Transformer [9] to extract meaningful features. Although these complex networks have improved the accuracy, their segmentation speeds are very slow. Generally, the capacity of some high-accuracy network models is very large. That makes it difficult to deploy on the devices in production. This urges us to design a more flexible network for our task.

Yet the pursue of time efficiency models is not easy to accomplish. Most real-time deep neural networks choose lightweight backbone networks and reduce the number of feature channels [10] and other methods. However, these ways will make the network's ability to capture fine information worse, so that further damages the accuracy.

We need to build a river ice semantic segmentation network with both accuracy and speed to meet the need for lightweight

[*] Corresponding author.
E-mail address: lran@nwpu.edu.cn (L. Ran).

https://doi.org/10.1016/j.sigpro.2023.109150
0165-1684/© 2023 Elsevier B.V. All rights reserved.

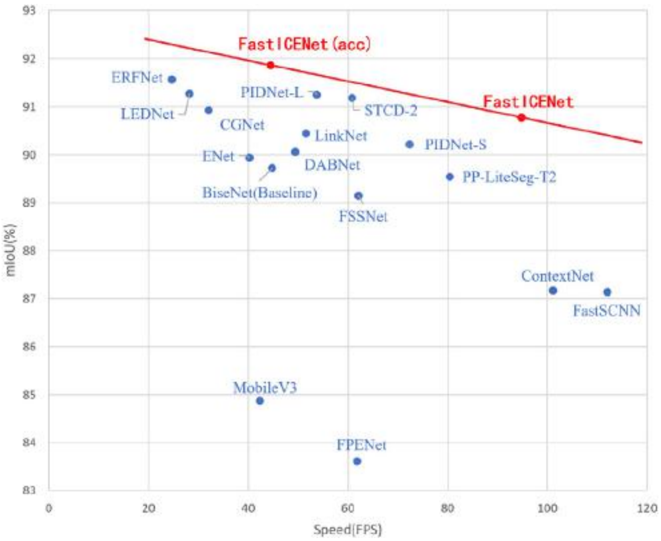


Fig. 1. Comparison with state-of-the-art methods in accuracy and speed on NWPU_YRCC_EX dataset. The horizontal axis represents the segmentation speed of the network, and the vertical axis represents the segmentation accuracy of the network. Our methods, FastICENet (acc) and FastICENet, are marked in red.

devices to process data quickly. There are two main problems for real-time and accurate semantic segmentation of river ice: (1) The segmentation accuracies of some semantic segmentation networks are very high, but due to their very complex model structures and some lightweight terminals often do not have the computing power of the laboratory server, which leads to their slow segmentation speed and difficult to process the image transmitted by the sensor in time; (2) Although some lightweight networks can meet the real-time requirements, their simple network structures result in low segmentation accuracy and are difficult to be applied in practice.

To this end, we propose a two-branch segmentation network, i.e. FastICENet, for accuracy and efficiency improvement. It consists of a spatial branch and a context branch, which is inspired by [11]. The spatial branch is a stack of convolutional and non-linear mapping layers to obtain the details of local region. The context branch is designed to provide deep semantic information for distinguishing different kinds of objects. That reaches a more complex branch and is also more time-consuming. Therefore, to speed up the segmentation, we adopt Downsampling and Ghost module strategies to reduce the analysis cost of the context branch. Furthermore, since there are many small ice blocks in the river ice images, a learnable upsampling strategy DUpsampling is utilized to replace the commonly used bilinear interpolation to increase the segmentation accuracy, especially for small ice blocks.

The main contributions are summarized as follows:

1) we adopt a two-branch structure and utilize a learnable upsampling strategy DUpsampling to replace the commonly used bilinear interpolation, named FastICENet Accuracy Version (FastICENet(acc)) to address the characteristics of scale diversity and manipulate lots of small ice blocks in river ice images. Compared with state-of-the-art methods, FastICENet(acc) performs excellently and achieves 91.86% mIoU on NWPU_YRCC_EX, the highest segmentation accuracy as shown in Fig. 1.

2) We develop a time and accuracy balanced model, FastICENet, which adopts a new Downsampling module and a dense connection block based on a lightweight Ghost module in the context branch. This design reform improves the speed significantly while maintaining high segmentation accuracy.

3) We enlarge and relabeled two river ice datasets, the NWPU_YRCC_EX and NWPU_YRCC2. Both of them are publicly accessible now. Along with Alberta River Ice Segmentation Dataset [12], we conduct our experiments on three public datasets. Compared with the most recent real-time semantic segmentation methods, FastICENet achieves the state-of-the-art trade-off between the accuracy and the speed, shown in Fig. 1.

The remainder of this paper is organized as follows. In Section 2, we summarize some related work of real-time semantic segmentation and ice segmentation. The proposed method is described in Section 3. Section 4 presents the experiment configuration and analyzes the results. Finally, the conclusion is given in Section 5.

2. Related work

2.1. Real-time semantic segmentation

Semantic segmentation is a time consuming task considering that it predicts pixel-wise labels, while one remote sensing image may contain millions of pixels. Various efforts have been made to tackle this problem. Some researchers clip or resize the input image to reduce the computational complexity, such as SNet [13] and ICNet [14]. Some approaches reduce the number of network channels to improve the reasoning speed, such as SegNet [15] and ENet [16]. The others mostly adopt lightweight classification network as their backbone and carefully design the network to improve segmentation speed and remedy the accuracy drop. For instance, EDANet [17] proposes a new network architecture with efficient dense modules using asymmetric convolution. FastSCNN [18] adopts the skip connection in the deep convolutional neural network and proposes a shallow learning module to downsample for fast and efficient multi-branch low-level feature extraction. STDC [19] proposes a novel and efficient structure named Short-Term Dense Concatenate network by removing structure redundancy. PIDNet [20] proposes a real-time semantic segmentation network inspired by PID Controller.

2.2. Ice segmentation

Many fields and applications have achieved rapid development and great progress due to the excellent performance of deep learning. Intelligent river ice monitoring is one of them. Wang et al. [21] use a basic deep convolutional neural network to estimate ice concentration using dual-pol SAR scenes collected during melting. Singh et al. [22] adopt some semantic segmentation models (e.g., UNet [23], SegNet [15], DeepLab [24] and DenseNet [25]) based on CNNs to segment river ice images into water and two distinct types of ice drift ice and anchor ice. ICENet [26] is a semantic segmentation deep convolution neural network for river ice segmentation, which uses the fusion of position and channel-wise attentive features. ICENetv2 [27] designs a multi-scale feature fusion framework for fine-grained river ice segmentation according to the characteristics of river ice. These models have constantly promoted the performance, however they rarely pay attention to the speed and cannot meet the real-time requirement. In this paper, we propose a real-time and accurate river ice semantic segmentation network, named FastICENet.

The appearance of river ice varies dramatically in scale, color, texture and shape. Especially, the size of river ice ranges from a few pixels to thousands of pixels in an image. ICENet [26] and ICENetv2 [27] show that the two-branch network structure like BiSeNet [11] with a finer spatial branch and a deep context semantic branch is very suitable to segment the river ice with scale diversity characteristic.

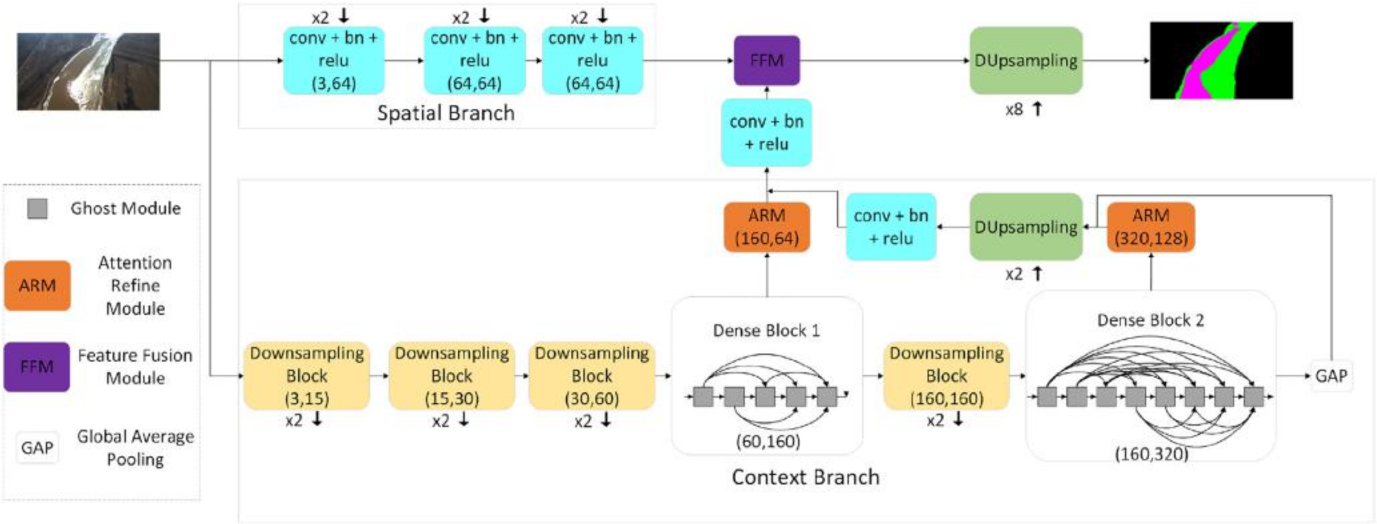


Fig. 2. The network architecture of FastICENet. The two numbers in bracket represent the number of input channels and output channels, respectively.

3. Method

3.1. The overall architecture

The architecture of the proposed FastICENet is shown in Fig. 2. It consists of two branches: the spatial branch and the context branch, which extract low level finer information and deep semantic information, respectively.

The spatial branch simply consists of three cascade convolution modules, as shown in Fig. 2. Each convolutional layer downsamples the feature map size to 1/2 of the previous one to obtain low level finer information. While the context branch are more carefully designed with scaling and refinement manipulations. Then, the output features of the two branches are fused by a feature fusion module FFM. Due to the feature maps of the two branches are not the same, it is not possible to simply add the feature maps of these two parts. The method of FFM is to concatenate two partial feature maps and then use convolution to calculate weighted features, multiply the weighted features with the original feature map to calculate channel attention, and finally perform residual connections with the original feature map. Finally, the fused feature maps are upsampled through a DUpsampling module to generate the final prediction results.

To reduce the time consumption of the context branch, we adopt the Downsampling module and a dense connection block based on a lightweight Ghost module. In details, the context branch contains three consecutive Downsampling modules and a dense connection block with five ghost modules (named Dense Block 1), followed by a Downsampling module and a dense connection block with eight ghost modules (named Dense Block 2). Then, the output of Dense Block 2 is directed to an attention refine module ARM and a global average pooling (GAP) layer in parallel. After ARM and the average pooling, their output feature maps are concatenated, upsampled by DUpsampling and recalculated by a convolution module. Finally, the obtained feature maps and the output of Dense Block 1 with ARM refined are concatenated, recalculated with another convolutional module to produce the final output of the context branch. The ARM module uses global pooling and 1 × 1 convolution to calculate the weight of the input feature map, and then multiplies it with the input feature map to calculate channel attention.

The details of the novel Downsampling module, the dense connection block based on a lightweight Ghost module, and the DUpsampling module are given in the following three subsections, respectively.

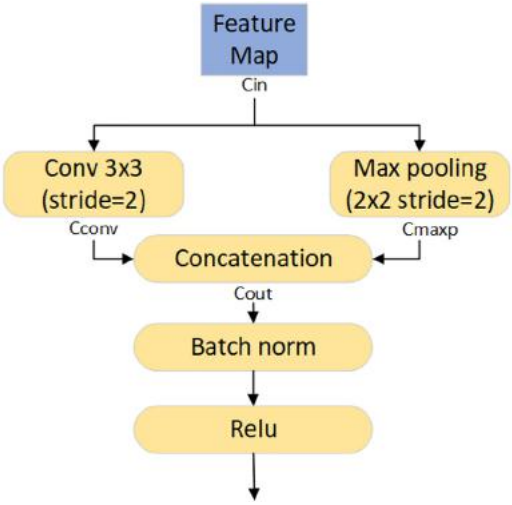


Fig. 3. Downsampling module. This module replaces partial convolution with maxpooling, which effectively reduces the amount of computation.

3.2. The time boosting strategy

3.2.1. Downsampling module

Inspired by the initial block in ENet [16], we use the Downsampling module in the context branch to obtain features. The partial convolution is replaced by maxpooling, which can reduce the computation cost. The module is illustrated in Fig. 3.

We denote the number of input, output channels as C_{in}, C_{out}, and the number of output channels of convolutional layers and maxpooling layers as C_{conv}, C_{maxp}. We have

$$C_{out} = C_{conv} + C_{maxp}, \tag{1}$$

where $C_{maxp} = \{0, C_{in}\}$.

The Downsampling module works as a feature map generator, who produces expanded features with lower dimensions. As we adopt C_{maxp} feature maps coming from maxpooling, the computation cost is lower than the traditional convolutional way. When

$C_{out} > C_{in}$, $C_{conv} = C_{out} - C_{in}$ and $C_{maxp} = C_{in}$, when $C_{out} \leq C_{in}$, all feature maps are generated through convolution.

Then, the outputs of the two layers are concatenated, batch normalized, and activated by ReLU in series. In this paper, we set the convolutional kernel as 3 × 3 and the maxpooling layer as 2 × 2, both walks with stride 2.

3.2.2. Dense block with ghost module

a. Dense Connection

DenseNet [25] puts forward the dense connection, which realizes feature reuse by connecting features across channels. EDANet [17] also utilizes the strategy of dense connection, and the difference is that EDANet adopts the dense connection between modules using asymmetric convolution. Each module connects the input features with the newly learned features to form the final output. Since each module is only responsible for obtaining relatively few new feature maps, the calculation cost can be greatly reduced and the prediction speed is improved.

Hence, in this paper, we propose the dense block, which succeeds the same dense connection method as EDANet. But instead of using asymmetric convolution, we adopt a more lightweight module, which names Ghost module.

b. Ghost Module

Given the input data $X \in R^{h \times w \times c}$, where c is the number of input channels, h and w are the height and width of input data respectively. The operation of generating n feature maps for convolution layer can be expressed as

$$Y = X * f + b, \tag{2}$$

where $*$ is the convolution operation, b is the bias term, $Y \in R^{h' \times w' \times n}$ is the output feature map of size h' and w' with n channels, and $f \in R^{c \times k \times k \times n}$ is the convolution filter with size $k \times k$. In this convolution process, since the number of convolution kernels n and channels c are usually very large, the number of floating-point operations required is as many as $n \cdot h' \cdot w' \cdot c \cdot k \cdot k$. This is where time consumes.

Reducing the number of convolutional channels is promising for our real-time request. The output feature maps of convolution layer usually contain a lot of redundancy, which does not meet our lightweight requirements. GhostNet [28] uses ordinary convolution filters to produce fewer feature maps $Y' \in R^{h' \times w' \times m}$. To further obtain the n desired feature maps, GhostNet applies a series of cheap linear operations on each original feature map in Y' to generate the ghost feature maps Y^{Ghost} according to:

$$y_{ij}^{Ghost} = \Phi_{i,j}\left(y_i'\right), \forall i = 1, \cdots, m, j = 1, \cdots, s. \tag{3}$$

Where y_i' is the ith original feature map in Y', and $\Phi_{i,j}$ is the jth linear operation on y_i', which is used to generate the jth Ghost feature map y_{ij}^{Ghost} of y_i'. Finally, the original feature maps Y' and the ghost feature maps Y^{Ghost} generated by Eq. 3 are concatenated to produce the final result as Y''.

3.2.3. DUpsampling module

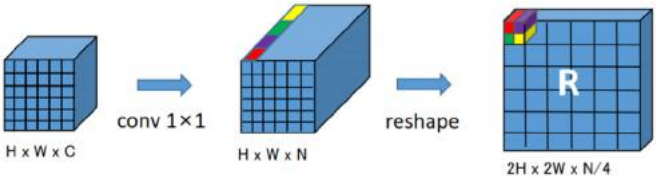


Fig. 4. DUpsampling module.

Bilinear interpolation is a commonly used method to upsample feature maps in decoder. However, this method is very simple and data independent, which may lead to sub-optimal results. In order to achieve better segmentation effect, we adopt DUpsampling [29] instead of bilinear interpolation for upsampling. The flowchart of DUpsampling is shown in Fig. 4. Given feature maps with size of $H \times W \times C$, N filters of 1×1 convolution are applied on the feature maps to produce new encoded feature maps with a size of $H \times W \times N$. Then the encoded feature maps are reshaped to the size $2H \times 2W \times N/4$, which is the output feature maps of the DUpsampling module. Generally, DUpsampling can upsample the feature map into any multiples along spatial dimensions. In this paper, the feature maps in the context branch and the fused feature map are enlarged 2 times and 8 times along the spatial dimension, respectively.

4. Experiments

In this section, we first introduce two new datasets for public usage, which are NWPU_YRCC_EX and NWPU_YRCC2. Then we give a brief description of the Alberta River Ice Segmentation Dataset. Furthermore, we conduct experiments on those three datasets for both FastICENet and competing methods. Our main evaluation indicators are chosen as mIoU and FPS. Lastly, we perform ablation experiments on NWPU_YRCC_EX to verify the effectiveness of our proposed model.

4.1. Dataset description

4.1.1. NWPU_YRCC_EX

NWPU_YRCC_EX is extended from NWPU_YRCC [26]. NWPU_YRCC dataset is composed of 814 images selected from the videos captured at the NingxiaInner Mongolia reach of the Yellow River from November 2015 to March 2019 using UAVs. The images of NWPU_YRCC are labeled pixel by pixel into three categories: ice, water and shore, as shown in Fig. 5(a). The size of the images is 1600 × 640. Considering the balance of data distribution, we further selected 73 images from the original videos and images as a supplement to the NWPU_YRCC. Totally, there are 887 fine labeled images. We name the dataset NWPU_YRCC_EX, and split it with 524 images for training, 180 images for validation and 183 images for testing. The NWPU_YRCC_EX can be downloaded on https://github.com/nwpulab113/NWPUYRCCEX.

4.1.2. NWPU_YRCC2

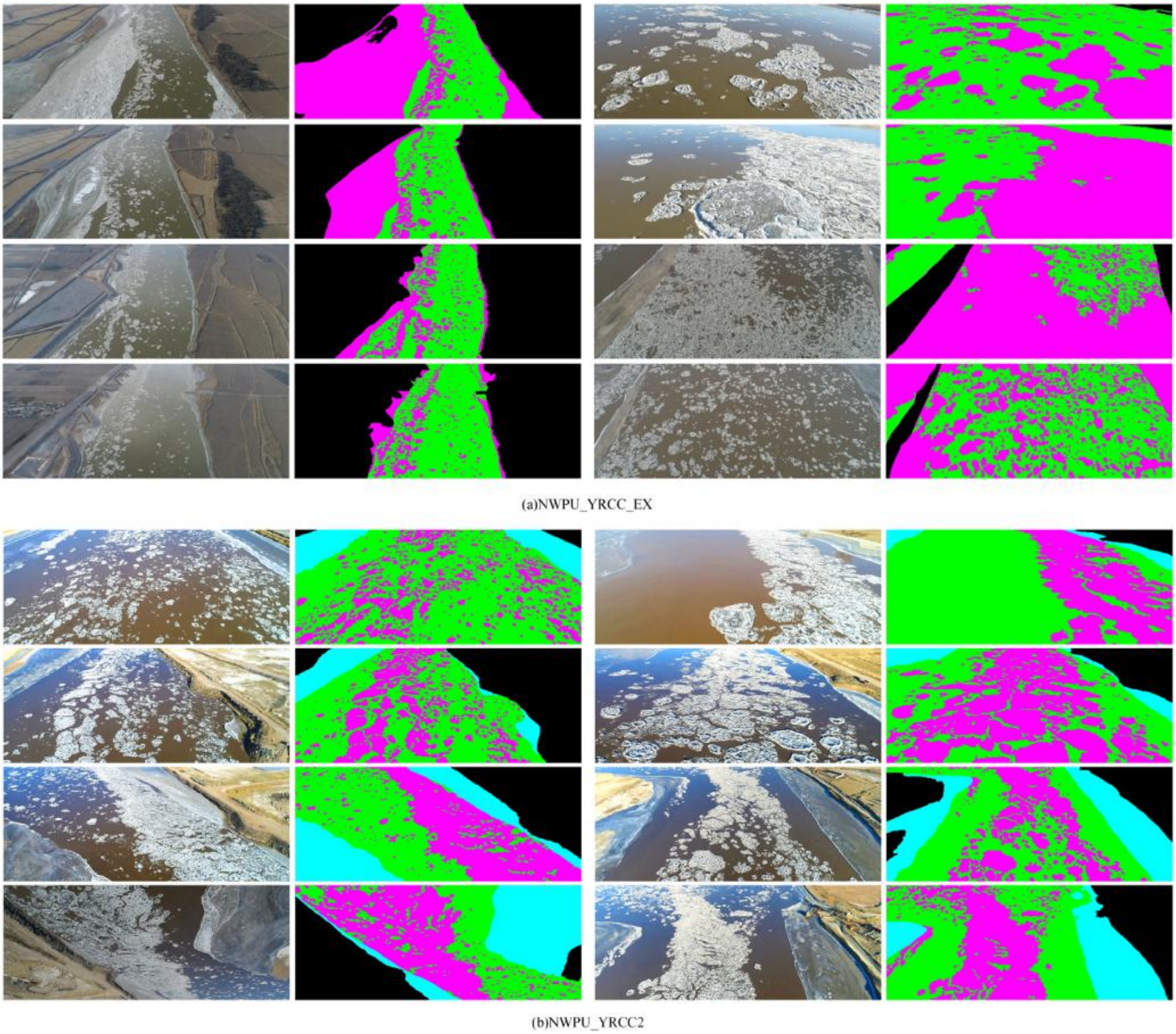
NWPU_YRCC2 [27] dataset is composed of 1525 images selected from the videos captured at the NingxiaInner Mongolia reach of the Yellow River from November 2015 to March 2019 using UAVs. The size of the images is 1600 × 640. The difference between NWPU_YRCC2 and NWPU_YRCC_EX is that we divide the ice into shore ice and drift ice according to the actual needs. Sample images and 4-class labels are shown in Fig. 5(b). And we divide the data into training set, validation set and test set according to the ratio of 3:1:1. The NWPU_YRCC2 can be downloaded on .

4.1.3. Alberta river ice segmentation dataset

The Alberta River Ice Segmentation Dataset [12] is captured by UAVs and bridge-mounted game cameras from two Alberta rivers during the winters of 2016 and 2017. The images of Alberta River Ice Segmentation Dataset are labeled into three categories: drift ice, anchor ice, and water, as shown in Fig. 6. Most of the images are of size 1281 × 1081. Since the pixel-wise labeling are time consuming, there are only 50 labeled images. We randomly crop these pictures into 800 × 320. Finally, we got 198 RGB sample images with fully annotated labels.

4.2. Training and optimization

The proposed network is implemented by Pytorch and run on the workstation with Intel Core CPU i5-7500 (3.40GHz) and


(a)NWPU_YRCC_EX

(b)NWPU_YRCC2

Fig. 5. (a) Visualization of images and labels of the Yellow River ice in NWPU_YRCC_EX. Black indicates shore, green indicates water, and purple indicates ice. (b) Visualization of images and labels of the Yellow River ice in NWPU_YRCC2. Black represents the shore, green represents the water, blue represents the shore ice, and purple represents drift ice.

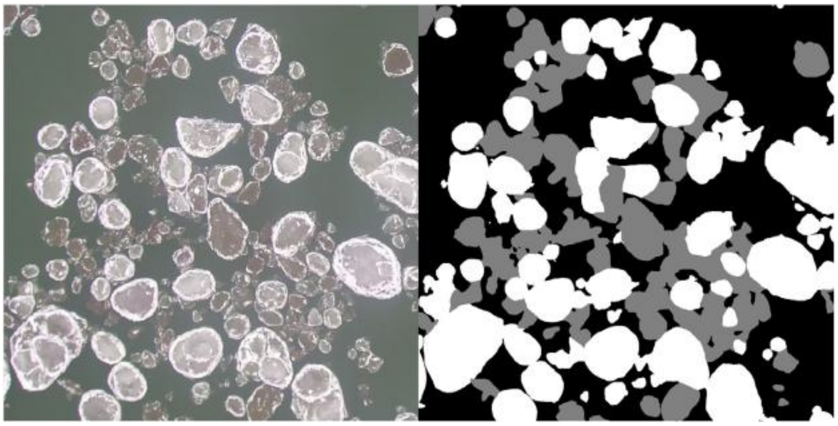


Fig. 6. The river ice image in Alberta River Ice Segmentation Dataset and its label where white, gray and black pixels respectively denote anchor ice, anchor ice, and water.

Table 1
Comparison with state-of-the-art methods on the NWPU_YRCC_EX dataset.

method	IoU(%)			mIoU(%)	speed(FPS)	parameters(k)
	ice	water	other			
ENet [16]	91.581	87.946	90.298	89.942	40.23	356
CGNet [31]	92.262	87.959	92.561	90.927	32.09	492
ICENet [26]	91.583	84.891	88.253	88.112	37.61	10694
DABNet [33]	91.652	87.140	91.370	90.054	49.48	752
MobileV3 [34]	87.575	80.448	86.580	84.868	42.31	3250
FPENet [35]	86.789	82.251	81.793	83.611	61.83	114
FSSNet [36]	91.234	86.800	89.408	89.147	62.13	173
LEDNet [37]	92.633	88.328	92.839	91.267	28.22	913
ContextNet [30]	92.633	87.121	85.006	87.163	101.11	874
FastSCNN [18]	89.339	87.077	84.976	87.131	112.04	1136
ERFNet [38]	92.871	89.330	92.507	91.569	24.64	2064
LinkNet [32]	91.930	87.861	91.543	90.445	51.61	11,534
Baseline(BiSeNet) [11]	91.221	87.610	90.337	89.723	44.75	14,090
STDC-2 [19]	92.790	88.507	92.257	91.183	60.82	16,073
PP-LiteSeg-T2 [39]	91.118	86.912	90.595	89.540	80.40	13240
PIDNet-S [20]	91.640	87.430	91.560	90.210	72.34	7717
PIDNet-L [20]	92.690	88.640	92.050	91.250	53.75	37,306
FastICENet(acc)	92.900	89.650	93.020	91.860	44.50	14,090
FastICENet	92.100	88.120	92.080	90.770	94.84	969

NVIDIA GTX 1080 Ti. We adopt the data augmentation methods of random_crop and random_mirror in the training process. The initial training rate is set to 5e-4, the batch size is 16, each model is trained 1000 epochs. The loss function is the CrossEntropy Loss, which is calculated as follows:

$$L = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic} log(p_{ic}) \tag{4}$$

Where N represents the number of pixels, i represents the index of pixels, M represents the number of categories, c represents the index of categories, y_{ic} is the label of pixel i, when the category of pixel i is c, y_{ic} is 1, otherwise it is 0. p_{ic} represents the prediction probability that the pixel i belongs to the category c. In the training process Adam optimizer is utilized to optimize the model.

4.3. Evaluation criteria

mIoU and FPS are two commonly used evaluation criteria for real-time semantic segmentation. mIoU is mean intersection over union, which represents the accuracy of segmentation. FPS is frame per second, which represents the speed of segmentation. In this paper, mIoU and FPS are as the main criteria to evaluate the performance of the proposed network and other comparison methods. In addition, we also record the number of parameters (in kilo bytes) of these networks.

4.4. Results and evaluation

To verify the effectiveness and superiority of FastICENet, we compare it with the most recent fast semantic segmentation networks on NWPU_YRCC_EX, NWPU_YRCC2, and Alberta River Ice Segmentation Dataset. Their performance is measured by the same machine under the same conditions. We take BiSeNet [11] as our baseline. FastICENet(acc) comes from BiSeNet, where the bilinear interpolation upsampling is replaced by DUpsampling. The comparison results are shown in Table 1, Table 2, and Table 3, the top three results are marked in red, green and blue respectively

4.4.1. NWPU_YRCC_EX

We can see that our method FastICENet(acc) achieves the highest segmentation accuracy (i.e., 91.860% mIoU), but its speed is slow. FastICENet performs much better with the speed 94.840 FPS, which is close to the fastest segmentation network, such as ContextNet [30] and FastSCNN [18]. Its accuracy achieves 90.770% mIoU, which is about 3% higher than that of the two networks.And FastICENet is close to CGNet [31], LinkNet [32] and PIDNet-S [20] in accuracy, but our segmentation speed is far ahead of these networks. At the same time, we observed that the speed of model segmentation is not negatively correlated with the number of parameters. We analyse that although some models have fewer parameters, their operation of these parameters is more complex, which will also cause the loss of segmentation speed. In the subsequent research, we can consider further accelerating the segmentation speed while maintaining the current segmentation accuracy.

We visualize the segmentation results of the three fastest networks FastSCNN, ContextNet and FastICENet on NWPU_YRCC_EX. The reason why we choose these three networks is that they are similar in segmentation speed and far ahead of other networks. The visualization results are shown in Fig. 7. The first image in each line represents the original image, the middle three images represent the segmentation results of FastSCNN, ContextNet and FastICENet respectively, and the last image represents the groundtruth. It can be seen that among the three networks with the fastest segmentation speed, our method accuracy is better than the other two networks. The most obvious parts have been marked with black boxes in the Fig. 7.

4.4.2. NWPU_YRCC2

Due to the need to divide ice into shore ice and drift ice, we also carried out experiments on NWPU_YRCC2 dataset. The results are shown in Table 2. This dataset needs to distinguish two different ice, which leads to the decline of the accuracy of all networks, but it can be seen that our FastICENet(acc) still achieves the second highest accuracy (i.e., 81.871% mIoU), but its segmentation speed is far faster than ICENetv2 [27]. ContextNet [30] and FastSCNN [18] have the fastest speed, but their accuracy is low. Our FastICENet achieves a trade-off between speed and accuracy, which can have the speed (i.e., 108.78 FPS) similar to the fastest method and achieve the accuracy of 80.790% mIoU at the same time. According to the specific task of NWPU_YRCC2, our model FastICENet(acc) can accurately distinguish shore ice and drift ice, it has the highest IoU in the class of drift ice, and also ranks high in the class of shore ice. At the same time, our model FastICENet can also accurately distinguish drift ice and shore ice, and greatly improve the segmentation speed.

Table 2
Comparison with state-of-the-art methods on the NWPU_YRCC2 dataset.

method	IoU(%)				mIoU	speed	parameters
	Drift Ice	Shore Ice	Water	Other			
ENet [16]	79.152	82.729	87.096	76.505	81.371	40.86	356
CGNet [31]	78.458	79.646	87.061	76.799	80.491	34.07	492
DABNet [33]	75.670	79.692	86.694	78.484	80.119	53.63	752
FPENet [35]	77.393	73.094	85.543	71.530	76.890	65.01	114
FSSNet [36]	77.595	77.953	87.520	75.463	79.633	63.96	173
LEDNet [37]	78.431	81.712	87.748	79.340	81.808	31.20	913
ContextNet [30]	74.352	74.930	83.894	78.150	77.832	116.63	874
FastSCNN [18]	74.116	77.083	82.959	75.299	77.364	127.63	1136
ERFNet [38]	80.813	81.772	89.412	74.323	81.580	26.69	2064
LinkNet [32]	81.561	80.562	89.774	73.254	81.288	56.07	11,534
ICENetv2 [27]	81.127	81.582	90.484	80.548	83.435	30.69	12803
Baseline(BiSeNet) [11]	72.623	84.026	87.282	76.441	80.093	59.85	14,090
STDC-2 [19]	80.323	81.807	89.015	79.234	81.460	69.33	16,073
PP-LiteSeg-T2 [39]	78.875	80.333	87.411	77.806	79.992	91.65	13240
PIDNet-S [20]	78.840	80.333	86.707	75.279	80.292	85.41	7717
PIDNet-L [20]	79.757	81.259	87.706	76.147	81.217	60.354	37,306
FastICENet(acc)	81.974	80.400	89.709	79.799	81.871	51.13	14,090
FastICENet	79.337	80.832	87.245	75.746	80.790	108.78	969

Table 3
Comparison with state-of-the-art methods on Alberta River Ice Segmentation Dataset.

method	IoU(%)			mIoU(%)	speed(FPS)	parameters(k)
	water	anchor ice	drift ice			
ENet [16]	95.64	70.88	75.73	80.75	46.15	356
CGNet [31]	95.52	72.28	77.05	81.62	68.07	492
DABNet [33]	95.61	72.45	78.31	82.13	103.00	752
FPENet [35]	95.01	69.14	75.54	79.89	88.18	114
FSSNet [36]	95.18	66.94	71.99	78.04	124.14	173
LEDNet [37]	95.13	71.90	77.64	81.56	60.14	913
ContextNet [30]	94.90	68.60	72.44	78.64	172.83	874
FastSCNN [18]	95.32	68.60	70.43	78.11	198.16	1136
ERFNet [38]	95.69	72.48	77.63	81.94	50.72	2064
LinkNet [32]	95.45	72.83	78.08	82.12	109.5	11,534
Baseline(BiSeNet) [11]	95.09	72.03	77.81	81.24	74.81	14,090
STDC-2 [19]	95.46	71.86	77.71	81.86	86.65	16,073
PP-LiteSeg-T2 [39]	94.75	70.17	75.71	80.39	114.55	13240
PIDNet-S [20]	94.98	71.83	76.98	81.27	121.90	7717
PIDNet-L [20]	95.87	72.66	77.86	82.10	90.58	37,306
FastICENet(acc)	95.96	73.88	78.33	82.34	73.50	14,090
FastICENet	95.57	72.28	77.46	81.77	159.82	969

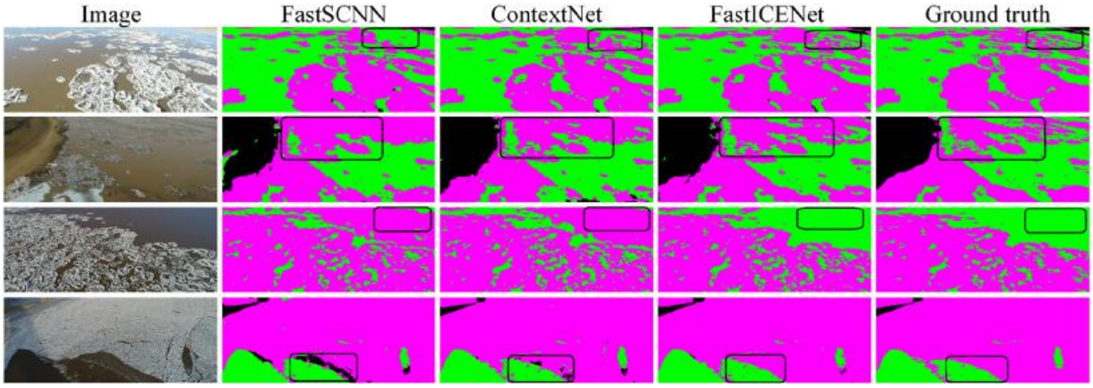


Fig. 7. Visual image comparison on NWPU_YRCC_EX with the three fastest methods.

On NWPU_YRCC2, we still visualize the segmentation results of the three fastest networks FastSCNN, ContextNet and FastICENet. The visualization result are shown in Fig. 8. The first image in each line represents the original image, the middle three images represent the segmentation results of FastSCNN, ContextNet and FastICENet respectively, and the last image represents the ground truth. It can be seen that among the three networks with the fastest segmentation speed, our method can better identify drift ice and shore ice. The most obvious parts have been marked with blackboxes in the Fig. 8.

4.4.3. Alberta river ice segmentation dataset

We compare our methods with the most recent fast semantic segmentation networks on the Alberta River Ice Segmentation Dataset. In the images of this dataset, the IoU of water is high due to the great difference in the spectral properties of water and the

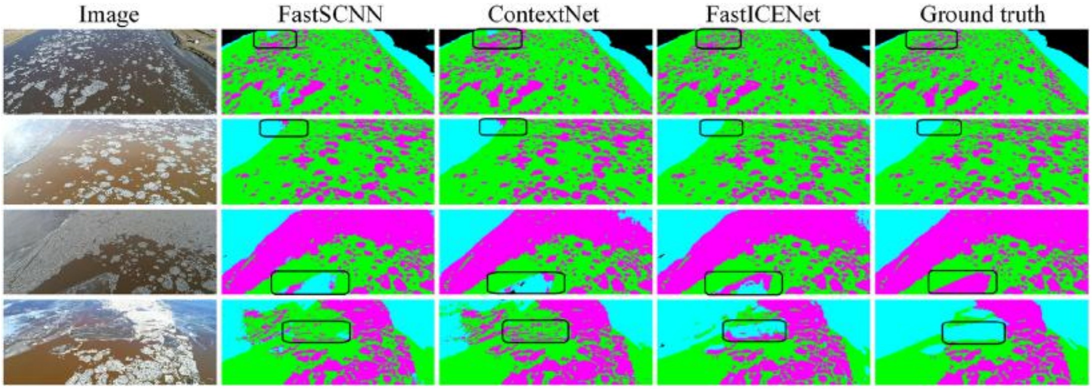


Fig. 8. Visual image comparison on NWPU_YRCC2 with the three fastest methods.

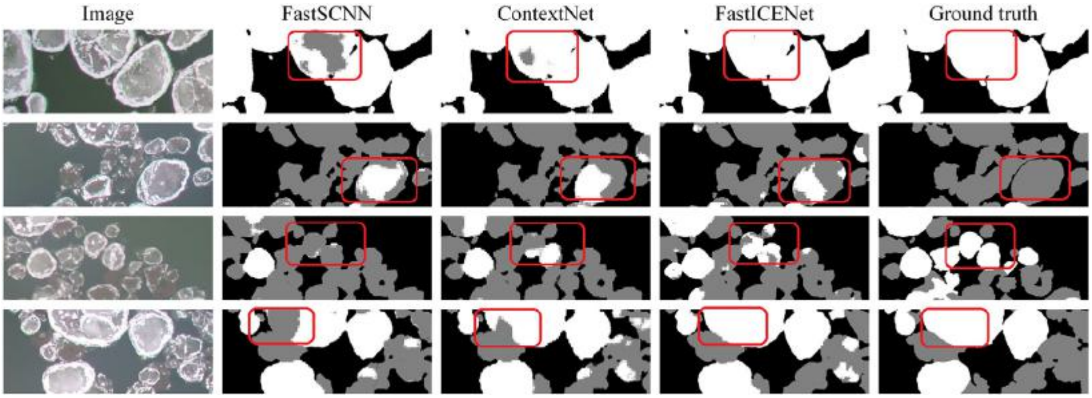


Fig. 9. Visual image comparison on Alberta River Ice Segmentation Dataset with the three fastest methods.

other two kinds of ice. Because the difference between drift ice and anchor ice is small, the IoU of all networks on these two kinds of ice is low. The comparison results are shown in Table 3. We can see that ContextNet [30] and FastSCNN [18] have the fastest speed. However, due to the small amount of data and the similar spectra of the two kinds of ice, the accuracy of ContextNet [30] and FastSCNN [18] are very low. But it can be seen that our FastICENet(acc) still achieves the highest accuracy (i.e., 82.34% mIoU), While FastICENet has greatly improved the speed to 159.82 FPS, it still maintains high accuracy (i.e., 81.77% mIoU).

On Alberta River Ice Segmentation Dataset, we visualize the segmentation results of the three fastest networks FastSCNN, ContextNet and FastICENet. The visualization results are shown in Fig. 9. The first image in each line represents the original image, the middle three images represent the segmentation results of FastSCNN, ContextNet and FastICENet respectively, and the last image represents the ground truth. It can be seen that among the three networks with the fastest segmentation speed, our method can better identify drift ice and anchor ice. The most obvious parts have been marked with red boxes in the Fig. 9.

4.5. Ablation study

To show the effectiveness of the submodules in our network, we conducted a set of comparative experiments on NWPU_YRCC_EX. We take BiSeNet [11] as our baseline. Baseline+DUp means that replacing the commonly used bilinear interpolation in the baseline with DUpsampling, which is our FastICENet(acc). Baseline+DUp+DS means that replacing the convolution downsampling in the Baseline+DUp with our Downsampling module. Baseline+DUp+DS+EDA means adding EDA module in [17], i.e. the dense connection block based on asymmetric convolution module to Baseline+DUp+DS. Baseline+DUp+DS+Ghost represents adding our dense connection block based on ghost module to Baseline+DUp+DS. Finally, FastICENet is obtained by halving the number of the channels in the context branch and the spatial branch in Baseline+DUp+DS+Ghost. The results of ablation experiment are shown in Table 4.

This experiment verifies the effectiveness of these modules. Under the joint action of these modules, the final model significantly improves the segmentation speed, but the accuracy de-

Table 4
Ablation experiment On NWPU_YRCC_EX.

method	IoU(%)			mIoU(%)	speed(FPS)	parameters(k)
	ice	water	other			
Baseline [11]	91.221	87.610	90.337	89.723	44.750	14,090
Baseline+DUp(FastICENet(acc))	92.900	89.650	93.020	91.860	44.500	14,090
Baseline+DUp+DS	91.980	87.670	91.300	90.320	76.370	1440
Baseline+DUp+DS+EDA	92.360	88.670	92.360	91.130	56.420	2620
Baseline+DUp+DS+Ghost	92.400	88.120	91.280	90.600	70.000	1820
FastICENet	92.100	88.120	92.080	90.770	94.840	969

creases slightly. DUpsampling improves the accuracy by 2% only with slight time attenuation. Then the Downsampling module greatly improves the segmentation speed, but its accuracy loss is large. So finally, EDA and Ghost modules are used to improve the segmentation accuracy on the premise of minimizing the speed loss. In all experimental settings, FastICENet achieved the best speed with a small precision reduction.

5. Conclusion

In this paper, an effective real-time river ice semantic segmentation network named FastICENet is proposed. To address the characteristics of scale diversity, FastICENet adopts a two-branch structure with a finer spatial branch and a deep multi-scale semantic context branch. While the context branch is time-consuming, to reduce the computation cost, we adopt a new Downsamping module and a very lightweight dense connection block with ghost module. These measures significantly reduce the computational complexity, meanwhile, keep a high segmentation accuracy. To improve the segmentation accuracy of small ice blocks, a learnable upsampling method called DUpsampling is used to restore the low-resolution feature image to the original size. The experimental results on three datasets all show that FastICENet can greatly improve the segmentation speed with a slightly decrease in accuracy, which meets the requirements of accurate and real-time. FastICENet has the best performance in recent real-time segmentation methods.

More explorations on other application scenarios need to be done. As the datasets for ice segmentation contains relatively fewer data than traditional datasets, such as Cityscapes [40] for scene understanding, the stability and performance of our model should be the subject of further investigation.

In future work, we will expand our dataset, which may improve our segmentation accuracy. At the same time, we will also study how to improve segmentation speed while better maintaining segmentation accuracy.

Credit Author Statement

The main idea was proposed by Xiuwei Zhang, Zixu Zhao and Yanning Zhang. The dataset was captured, labled and analyzed by Xiuwei Zhang, Zixu Zhao, Lingyan Ran, Wenna Wang, Zeze Lan, Houjun He, Qixing Liu and Baosen Zhang. The experiments were designed and carried out by Xiuwei Zhang, Zixu Zhao. The manuscript was written by Xiuwei Zhang, Zixu Zhao, Lingyan Ran and revised by Yinghui Xing and Hanlin Yin. All authors have read and agreed to the published version of the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This research was supported in part by the National Natural Science Foundation of China under Grant 61971356, Grant U19B2037 and Grant 62201467; in part by the National Natural Science Foundation of Shaanxi province under Grant 2021KWZ-03 and 2022JQ-686; in part by the Major Scientific and Technological Special Project of Henan Province under Grant 201400211000; and in part by the Science Foundation for Excellent Young Scholars of Henan Province under Grant 212300410059. (Corresponding author: Lingyan Ran).

References

[1] S. Jalayer, A. Sharifi, D. Abbasi-Moghadam, A. Tariq, S. Qin, Modeling and predicting land use land cover spatiotemporal changes: a case study in chalus watershed, iran, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 15 (2022) 5496–5513.
[2] A. Sharifi, H. Mahdipour, E. Moradi, A. Tariq, Agricultural field extraction with deep learning algorithm and satellite imagery, J. Indian Soc. Remote Sens. (2022) 1–7.
[3] A. Sharifi, Development of a method for flood detection based on sentinel-1 images and classifier algorithms, Water and Environment Journal 35 (3) (2021) 924–929.
[4] B. Altena, A. Kääb, Quantifying river ice movement through a combination of european satellite monitoring services, Int. J. Appl. Earth Obs. Geoinf. 98 (2021) 102315.
[5] K. Heinilä, O.-P. Mattila, S. Metsämäki, S. Väkevä, K. Luojus, G. Schwaizer, S. Koponen, A novel method for detecting lake ice cover using optical satellite data, Int. J. Appl. Earth Obs. Geoinf. 104 (2021) 102566.
[6] D. Davila, J. VanPelt, A. Lynch, A. Romlein, P. Webley, M.S. Brown, Adapt: an open-source suas payload for real-time disaster prediction and response with ai, arXiv preprint arXiv:2201.10366 (2022).
[7] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected crfs, arXiv preprint arXiv:1412.7062 (2014).
[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv Neural Inf Process Syst 30 (2017).
[10] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size, arXiv preprint arXiv:1602.07360 (2016).
[11] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, N. Sang, Bisenet: Bilateral segmentation network for real-time semantic segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 325–341.
[12] A. Singh, H. Kalke, N. Ray, M. Loewen, River ice segmentation with deep learning, 2019, 1901.04412.
[13] Z. Wu, C. Shen, A.v.d. Hengel, Real-time semantic image segmentation via spatial sparsity, arXiv preprint arXiv:1712.00213 (2017).
[14] H. Zhao, X. Qi, X. Shen, J. Shi, J. Jia, Icnet for real-time semantic segmentation on high-resolution images, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 405–420.
[15] 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.
[16] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, Enet: a deep neural network architecture for real-time semantic segmentation, arXiv preprint arXiv:1606.02147 (2016).
[17] S.-Y. Lo, H.-M. Hang, S.-W. Chan, J.-J. Lin, Efficient dense modules of asymmetric convolution for real-time semantic segmentation, in: Proceedings of the ACM Multimedia Asia, 2019, pp. 1–6.
[18] R.P. Poudel, S. Liwicki, R. Cipolla, Fast-scnn: fast semantic segmentation network, arXiv preprint arXiv:1902.04502 (2019).
[19] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, X. Wei, Rethinking bisenet for real-time semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9716–9725.
[20] J. Xu, Z. Xiong, S.P. Bhattacharyya, PIDNet: a real-time semantic segmentation network inspired from PID controller, arXiv preprint arXiv:2206.02066 (2022).
[21] L. Wang, K.A. Scott, L. Xu, D.A. Clausi, Sea ice concentration estimation during melt from dual-pol sar scenes using deep convolutional neural networks: a case study, IEEE Trans. Geosci. Remote Sens. 54 (8) (2016) 4524–4533.
[22] A. Singh, H. Kalke, M. Loewen, N. Ray, River ice segmentation with deep learning, IEEE Trans. Geosci. Remote Sens. 58 (11) (2020) 7570–7579.
[23] 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.
[24] 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.
[25] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700–4708.
[26] X. Zhang, J. Jin, Z. Lan, C. Li, M. Fan, Y. Wang, X. Yu, Y. Zhang, Icenet: a semantic segmentation deep network for river ice by fusing positional and channel-wise attentive features, Remote Sens (Basel) 12 (2020) 221, doi:10.3390/rs12020221.
[27] X. Zhang, Y. Zhou, J. Jin, Y. Wang, M. Fan, N. Wang, Y. Zhang, ICENETv2: a fine-grained river ice semantic segmentation network based on UAV images, Remote Sens (Basel) 13 (2021) 633, doi:10.3390/rs13040633.
[28] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, GhostNet: More features from cheap operations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1580–1589.

[29] Z. Tian, T. He, C. Shen, Y. Yan, Decoders matter for semantic segmentation: data-dependent decoding enables flexible feature aggregation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3126–3135.

[30] R.P. Poudel, U. Bonde, S. Liwicki, C. Zach, ContextNet: exploring context and detail for semantic segmentation in real-time, arXiv preprint arXiv:1805.04554 (2018).

[31] T. Wu, S. Tang, R. Zhang, J. Cao, Y. Zhang, CGNet: a light-weight context guided network for semantic segmentation, IEEE Trans. Image Process. 30 (2020) 1169–1179.

[32] A. Chaurasia, E. Culurciello, Linknet: Exploiting encoder representations for efficient semantic segmentation, in: 2017 IEEE Visual Communications and Image Processing (VCIP), IEEE, 2017, pp. 1–4.

[33] G. Li, I. Yun, J. Kim, J. Kim, Dabnet: depth-wise asymmetric bottleneck for real-time semantic segmentation, arXiv preprint arXiv:1907.11357 (2019).

[34] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for MobileNetV3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.

[35] M. Liu, H. Yin, Feature pyramid encoding network for real-time semantic segmentation, arXiv preprint arXiv:1909.08599 (2019).

[36] X. Zhang, Z. Chen, Q.J. Wu, L. Cai, D. Lu, X. Li, Fast semantic segmentation for scene perception, IEEE Trans. Ind. Inf. 15 (2) (2018) 1183–1192.

[37] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, L.J. Latecki, LEDNet: a lightweight encoder-decoder network for real-time semantic segmentation, in: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, pp. 1860–1864.

[38] E. Romera, J.M. Alvarez, L.M. Bergasa, R. Arroyo, ERFNet: efficient residual factorized convnet for real-time semantic segmentation, IEEE Trans. Intell. Transp. Syst. 19 (1) (2017) 263–272.

[39] J. Peng, Y. Liu, S. Tang, Y. Hao, L. Chu, G. Chen, Z. Wu, Z. Chen, Z. Yu, Y. Du, et al., PP-LiteSeg: a superior real-time semantic segmentation model, arXiv preprint arXiv:2204.02681 (2022).

[40] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.