Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

PMNet: a multi-branch and multi-scale semantic segmentation approach to water extraction from high-resolution remote sensing images with edge-cloud computing

Ziwen Zhang[1*†], Qi Liu[1†], Xiaodong Liu[2], Yonghong Zhang[3], Zihao Du[4] and Xuefei Cao[5]

Abstract

In the field of remote sensing image interpretation, automatically extracting water body information from high-resolution images is a key task. However, facing the complex multi-scale features in high-resolution remote sensing images, traditional methods and basic deep convolutional neural networks are difficult to effectively capture the global spatial relationship of the target objects, resulting in incomplete, rough shape and blurred edges of the extracted water body information. Meanwhile, massive image data processing usually leads to computational resource overload and inefficiency. Fortunately, the local data processing capability of edge computing combined with the powerful computational resources of cloud centres can provide timely and efficient computation and storage for high-resolution remote sensing image segmentation. In this regard, this paper proposes PMNet, a lightweight deep learning network for edge-cloud collaboration, which utilises a pipelined multi-step aggregation method to capture image information at different scales and understand the relationships between remote pixels through horizontal and vertical spatial dimensions. Also, it adopts a combination of multiple decoding branches in the decoding stage instead of the traditional single decoding branch. The accuracy of the results is improved while reducing the consumption of system resources. The model obtained F1-score of 90.22 and 88.57 on Landsat-8 and GID remote sensing image datasets with low model complexity, which is better than other semantic segmentation models, highlighting the potential of mobile edge computing in processing massive high-resolution remote sensing image data.

Keywords Mobile edge computing, Deep learning, Light-weight computing, Image semantic segmentation

Introduction

The study of extracting water bodies from high-resolution remote sensing images is garnering increased attention, particularly in light of the global rise in extreme climate events [1]. For instance, the unusual drought in China's Yangtze River Basin and the appearance of "hungry rocks" in European rivers have underscored the urgent need for monitoring and analyzing water body dynamics [2]. These events not only demonstrate the profound impacts of climate extremes on water resources but also emphasize the importance of precise monitoring

[†] Ziwen Zhang and Qi Liu contributed equally to this work.

*Correspondence:
Ziwen Zhang
20211249476@nuist.edu.cn
[1] School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2] School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK
[3] School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China
[4] McMaster University Software Engineering faculty, 1280 Main St West, Hamilton L8S 4L8, ON, Canada
[5] School of Cyber and Information Security, Xidian University, Xi'an 710071, China

and analysis of water body distribution for disaster prevention and resource management [3].

The primary task of image water body segmentation is to extract water bodies from complete remote sensing images and separate them from other backgrounds, which is essential for understanding and analyzing regional hydrological conditions [4]. With the rapid development of satellite technology, it has become easier to acquire high-resolution remote sensing images, which provide a rich data resource for researchers. These high-resolution images can provide more detailed and specific information about water bodies, thus making water body monitoring and analysis more accurate and comprehensive. However, the complexity of high-resolution images creates new challenges for water body extraction [5, 6]. These images contain a large amount of non-water body surface object information and often present the problem of homoscedastic and heteroscedastic objects, which may lead to misclassification and omission of water bodies. In particular, some narrow water bodies surrounded by objects with similar colours and textures are more likely to be misidentified. Therefore, it becomes particularly important to develop accurate and efficient water body extraction algorithms. In addition, to ensure that the model can be deployed on some low-cost, low-configuration edge devices, the new model should be lightweight and have low complexity. In this context, the application of deep learning-based methods, such as full convolutional networks (FCNs) [7], has become a hot topic in water body extraction research. However, these methods still face limitations when processing high-resolution images, such as insufficient consideration of multi-scale properties and high storage and computational resource requirements. Mobile Edge Computing (MEC) relocates part of the computation to the edge devices, which reduces the consumption of core system resources to a certain extent and reduces the computation latency and avoids network congestion problems [8]. It provides near real-time data processing capability for remote sensing image processing that requires fast response, reducing the dependence on remote data centres and the need for large-scale data transmission [9, 10]. Therefore, in order to further improve the accuracy of the model, reduce the complexity of the model and avoid excessive resource overheads, this paper proposes a lightweight PMNet network driven by edge-cloud assistance.

The contributions of this paper are as follows:

- An edge-cloud collaborative approach for water segmentation in high-resolution remote sensing images is proposed, which reduces the consumption of system resources and improves the accuracy of water segmentation.
- A lightweight network PMNet with low complexity is proposed in this paper. Comparison of several deep learning networks on Landsat-8 dataset and GID dataset shows that the network is suitable for extracting water bodies from high resolution remote sensing images.
- A pipeline deep learning encoder based on equal channel number division is proposed to aggregate different levels of feature information. this method can capture multi-scale and deep-level semantic information with fewer parameters.
- This paper proposes an Object Enhancement Module (OEM) to strengthen the water information during the information fusion between the encoding and decoding phases of the network.

The rest of the paper is organized as follows. "Related work" section focuses on related work. "Methodology" section describes the proposed method of this paper in detail. "Experiment" section discusses the experimental results in detail. "Conclusion" section is the conclusion and outlook.

Related work

Traditional water extraction methods

The early water body extraction method mainly used the threshold segmentation method to determine a better gray threshold through continuous experimental analysis and then compared each pixel with this to distinguish the water object from the background information. However, the error of this method is large, and it is easy to mistake the darker shaded information in the surrounding area or the wetland and farmland as a water body. Therefore, subsequent water body extraction methods mainly rely on spectral information in remote sensing images, such as near-infrared (NIR) and short-wave infrared (SWIR). The more widely known are various water index methods [11], such as automatic water extraction (AWEI) [12], normalized difference water index (NDWI) , pixel region index (PRI) [13], and so on. NDWI mainly considers the correlation between different bands of sensed images and extracts water bodies using NIR and green bands. But this method does not perform well in cloudy and densely built-up areas. Therefore, Xu[14] further proposed MNDWI, using mid-infrared band instead of near-infrared band to further suppress surface noise. For further improvement of accuracy, Guo [15] proposed weighted WNDWI. However, the robustness and generalization of the above methods are poor and they all demand manual rectification, leading to a tendency to fall into local optimum solutions.

Deep learning based semantic segmentation approach

Different from traditional water body extraction methods, deep learning-based image semantic segmentation methods can free researchers from the tedious manual tuning and automatically learn to extract the optimal solution for water bodies. Most current segmentation networks are based on the Unet architecture [16]. Unlike traditional FCN networks, Unet uses a progressive up-sampling structure that incorporates the output of each stage of the encoder at each up-sampling stage. The shallow detail information is taken into account along with the semantic messages, which is important for pixel-level classification tasks such as image segmentation. In addition, extending the perceptual field of model allows it to precisely identify the target and the background. For example, PSPNet [17] aggregates different contextual information according to different scale sizes in the last stage of the network. DeepLab [18] extends the perceptual domain of the network with three convolutional kernels with different dilated rates to extract richer multi-scale contextual information, but this design makes the model perform less well with small objects and details than with large objects, as the dilated convolution may result in the information of small objects being diluted in the process of expanding the perceptual domain. OCRNet [19] converts pixel classification to object-area classification, explicitly enhancing the global information of the object, however this approach may not work well with edge blurring or small objects as it focuses on large-scale regional information and may ignore local details. The attention mechanism also has an impact on the network results. The self-attentive mechanism [20] calculates the relevance of each pixel point by using three matrices Q, K, and V. It expands the perceptual space of each pixel point, which can better highlight the target object and suppress the background information. However, its number of parameters is too large to apply this module on some lightweight devices. The CCNet [21] uses a cross-crossing approach based on the self-attentive mechanism, allowing each pixel to consider only the relationship between itself and the pixels in its row and column, further reducing the number of parameters. GCNet [22] further proposes that the global context is not location dependent and there is no need to compute a global context for each location. ECANet [23] notes that channel degradation in SENet [24] will weaken the network's learning of different channel weights, and proposes a partial cross-channel interaction strategy without degradation which can reduce the complexity of the model as well as maintain the performance [25] proposed a new image segmentation network MobileUNet-FPN. it designed a multilevel edge computing system with distributed edge nodes deployed in servers at different locations. The MobileUNet-FPN model is trained in parallel on each edge node, which effectively reduces the network communication overhead [26] proposed a segmentation method based on convolutional neural network (CNN), which used the distributed computing architecture provided by edge cloud computing in the image acquisition and processing stages to improve the efficiency of data processing. This approach can reduce network latency, improve real-time performance, and perform calculations close to the data source, thereby reducing the need for data transmission and storage [27] optimized the target algorithm of the edge computing platform for GPU, only uploading the algorithm output to the cloud, and the segmentation task was performed on the local edge device. This method reduces the dependence on infrastructure and the consumption of network resources, and is more robust to connection interruptions, thus achieving efficient real-time processing.

Overall, there is great potential in applying mobile edge computing technology to the segmentation task of high-resolution remote sensing images. It helps spread the computational load, reduce system overhead, and improve processing efficiency. This article aims to introduce the concept of mobile edge computing and improve the performance of high-resolution remote sensing image segmentation by implementing the collaborative working method of edge and cloud computing.

Methodology

System model

The architecture of the entire system is shown in Fig. 1. This system uses edge computing to improve the efficiency and reliability of the semantic segmentation task of remote sensing images. By performing data preprocessing on the edge device, model training on the cloud center server, and model inference on the edge server, the respective advantages are fully utilized to achieve high-quality and efficient ground object segmentation tasks. First, the ground receiving station will serve as a mobile edge node device. It is mainly responsible for receiving a variety of remote sensing satellite image data from satellites and performing various image preprocessing operations, including data correction, radiation correction, and atmospheric correction operations to improve data quality, reduce noise, and ensure accuracy. This helps reduce model errors during training and inference and improves the reliability of results. And this operation distributes some computing operations to various edge computing devices, reducing the pressure on servers in the cloud. Subsequently, the processed data will be transferred to the cloud center for inference training of the deep learning model to adjust model parameters to continuously optimize performance and improve accuracy and

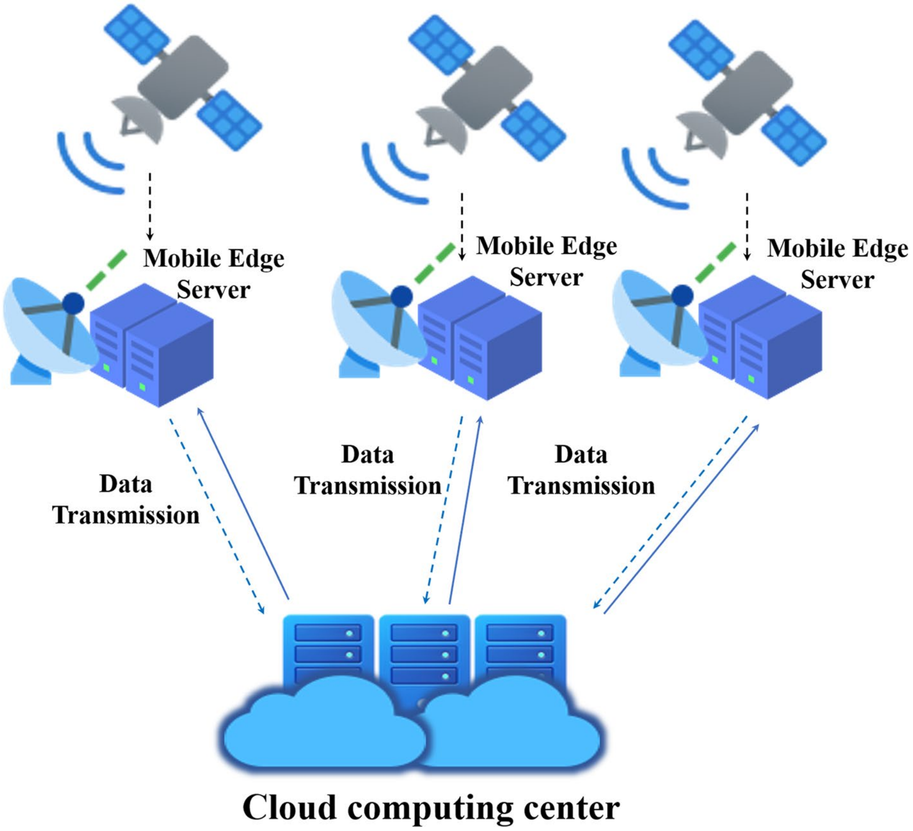

Fig. 1 System model

generalization capabilities. The benefit of cloud servers is that they have high-performance computing resources and can support complex model training and large-scale data processing. Finally, the trained model will be deployed to the edge server for inference. Migrating the result output calculation to edge computing devices avoids problems such as network delay and congestion, and enables fast decision-making with low latency.

Overall semantic segmentation model structure

Figure 2 shows the general structure of PMNet. The training process of this model will be carried out in the cloud center, and the pre-processed images of each mobile edge node will be transmitted to there as the training data set of the model. The whole model can be divided into encoding phase and decoding phase. In the encoding stage, the image is first down-sampled to one-half of the original image using the basic convolution, followed by four successive P-Block encoding modules proposed in this paper. The P-Block module can build a narrower and deeper network encoding model with fewer parameters while learning the multi-scale nature of the target object at different stages. At the end of the encoding stage, an attention module is used to enhance the texture information of the target object. In the decoder part, a multi-branch decoder architecture is designed. Each decoder stage combines the high-level semantic information of the final output of the encoder with the shallow detail information of the output of each stage of the encoder. However, unlike the conventional decoder stage which only has a single branch to up-sample the high-level semantic information to the original image size, the approach in this paper adds an additional branch to each decoding stage to up-sample each layer of the decoding stage directly to the original image size before fusion. This can effectively learn the multi-scale information in the image, and also indirectly increase the perceptual field of the shallow information. In addition, an OEM module is added in the middle of each encoding and decoding stage to enhance the water part of the detailed information. The details of the implementation of P-Block, multi-branch decoder, OEM, and CBAM are given in the subsequent subsections.

Design of P-Block in encoder

A very important point in the field of image segmentation is that the encoder stage of the neural network needs the network to be able to learn the high-level features of

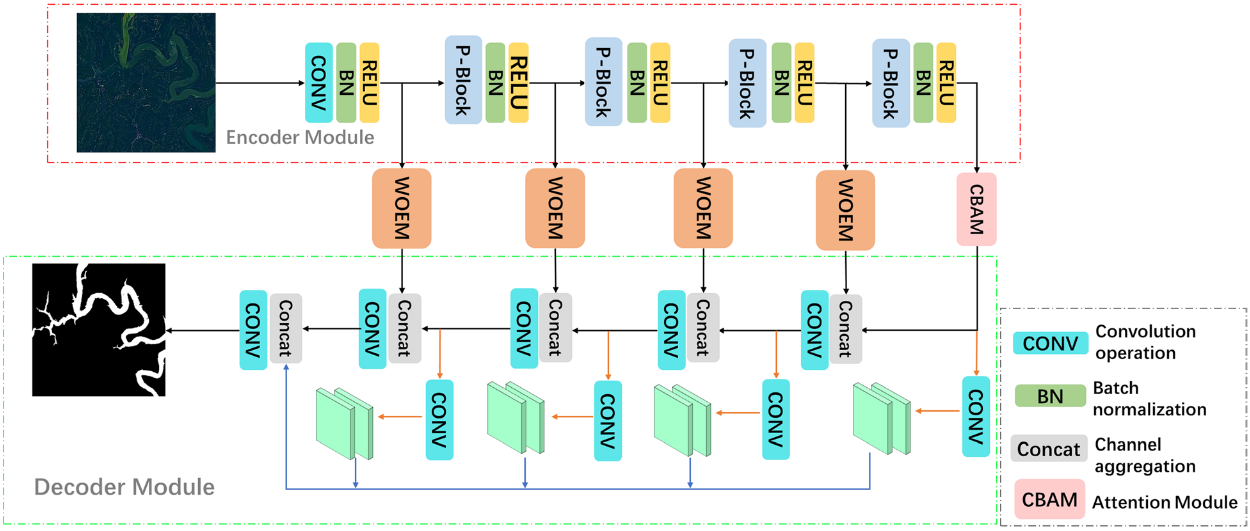

Fig. 2 The overall Structure of the Proposed PMNet

the image sufficiently to obtain richer semantic information. This requires the network to have a large enough perceptual field. The main and basic way to expand the field of perception is to increase the depth of the network. The deeper the network depth is, the richer the feature level is, and the richer the semantic information can be extracted. Although this is theoretically true, recent studies have shown that the theoretical field size is much smaller than the actual field size. In addition, as the network deepens the model will also have problems such as gradient disappearance. The main manifestation is that the accuracy of the results gradually increases to the highest value as the depth of the network increases, but at a certain point, there will be a sudden cliff-like decline. To solve this problem, the ResNet [28] network was proposed. It solves the problem of network degradation by skip connection. However, the second problem arises. The deeper the network, the larger the number of parameters, which leads to an increase in model complexity and memory consumption. Thus, this paper designs a new feature encoding module, P-Block, to train narrower and deeper networks with less number of parameters while containing rich multiscale information.

Unlike STDCNet, which considers shallow information to be more important than high-level semantic information in semantic segmentation. This paper considers that shallow and deep information should be equally important for semantic segmentation. The learning of high-level semantic feature information is the key to accurately identifying the object in the segmentation result while the information of shallow details is the key to refining the target object segmentation result. So the shallow and deep information in the network proposed in this paper has an equal number of channels. Compared with Res2Net [29], Res2Net is unfair to the information contained in the channel. The number of channels in the latter part of the input of each layer aggregates the information of the former part of the channel, resulting in a greater weight for the latter part of the channel information. It is explicitly assumed that the number of channels in the latter part of each layer of information is more important than the former part before the model is trained. In contrast, the design of this paper lies in the fairness that both deep and shallow information is equally important. The general architecture of P-Block is shown in Fig. 3.

First, an 1×1 convolution is used to expand the number of feature channels to ensure that enough information can be learned in the subsequent dimensionality reduction operations. Four successive convolution operations are then used, and it should be noted that the convolutions used in the middle are all dilated convolutions with different dilated rates. The purpose is to extend the perceptual field without losing the resolution of the remote sensing image while capturing the multiscale information of the image with different convolution sizes. In addition, the inflated convolution itself does not require the introduction of any additional parameters, which is what the model in this paper seeks. Unlike traditional ResNet, the number of output channels of each intermediate convolution layer is only one-fourth of the original map, which is intended to obtain a deeper network with fewer parameters. However, the reduction of the number of output channels in each layer will lead to insufficient information extraction, so the output of each intermediate convolution is finally stitched together to enrich the

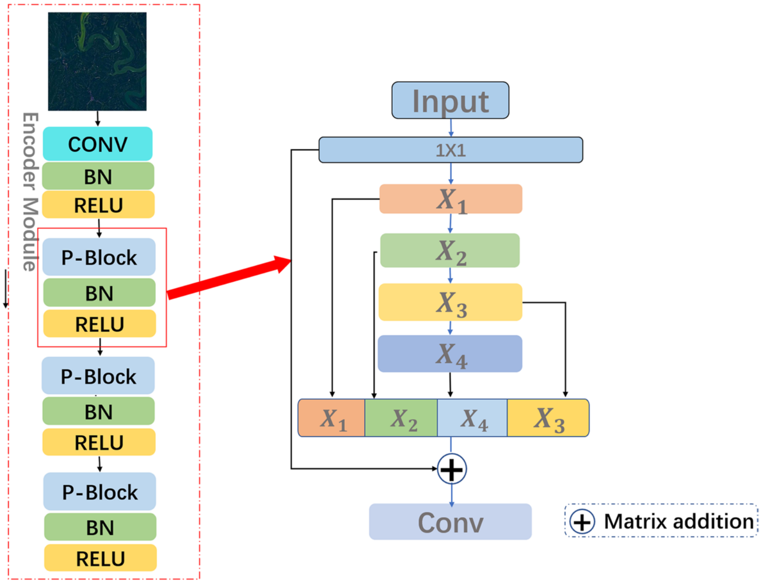

Fig. 3 The framework of P-Block in the encoder

information between channels. This operation also allows the information at different scales to be superimposed, further expanding the perceptual field of the model and allowing the model to learn information about the target object at different scales during the encoding stage. Finally, an 1×1 convolution is used again for feature fusion, allowing the network to adaptively learn different scale features at each convolution stage. To prevent network degradation and other problems, this module also adopts the skip connection operation proposed in Res2Net. The overall formula is as follows:

$$x_i = f_{3\times3}(x_{i-1}) \quad 0 < i \leq 4. \tag{1}$$

$$y = f_{1\times1}(x_0 + \mathrm{concat}(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \mathrm{x}_4)) \tag{2}$$

where $f_{3\times3}$ and $f_{1\times1}$ represent the convolution operation of the size of 3×3 and 1×1, respectively. x_0 represents the input after the dimension upgrade, x_i represents the output of each intermediate convolution, and $concat(\cdot)$ represents the concatenation operation of output feature graphs at each stage.

Design of WOEM between encoder and decoder

Figure 4 shows the general structure of the WOEM. First of all, the first stage of this module adopts a channel selection structure, which allows the network to strengthen the relatively important channels of each stage and suppress irrelevant information channels by combining average pooling and maximum pooling. In the second stage of the module, the spatial relationship of each pixel is mainly considered. Referring to the cross-attention idea of CCNet, the rectangular pooled kernel is adopted to establish the long-dependence relationship between discrete regions rather than the traditional square pooled kernel. However, since the cross-cross attention of CCNet still requires a large number of parameters, this paper makes a further simplification by using average pooling to collect contexts with long dependencies in both horizontal and vertical dimensions and then aggregating the two dimensions for an 1×1 convolution operation as a way to learn the weight information between each direction. Finally, a normalization operation using the Sigmoid function is performed to obtain the final output by dot product with the output of the first stage. Compared with the global average pooling, this rectangular pooling operation considers a long and narrow range instead of a whole feature map, avoiding some irrelevant calculations between remote locations. And due to the continuity and internal consistency of most water bodies, most of them present a slender and narrow shape that is suitable for pooling kernel learning in this rectangular shape. In addition, the proposed module in this paper is more lightweight compared to the self-attention module which requires a lot of computation of position relationships between each pixel.

Design of the multi-branch in decoder

In the decoding part, a multi-branch decoder is designed in this paper as shown in Fig. 5. The traditional single-branch decoder may gradually weaken the importance of the deep semantic information learned by the encoder during the up-sampling process, so this paper extends

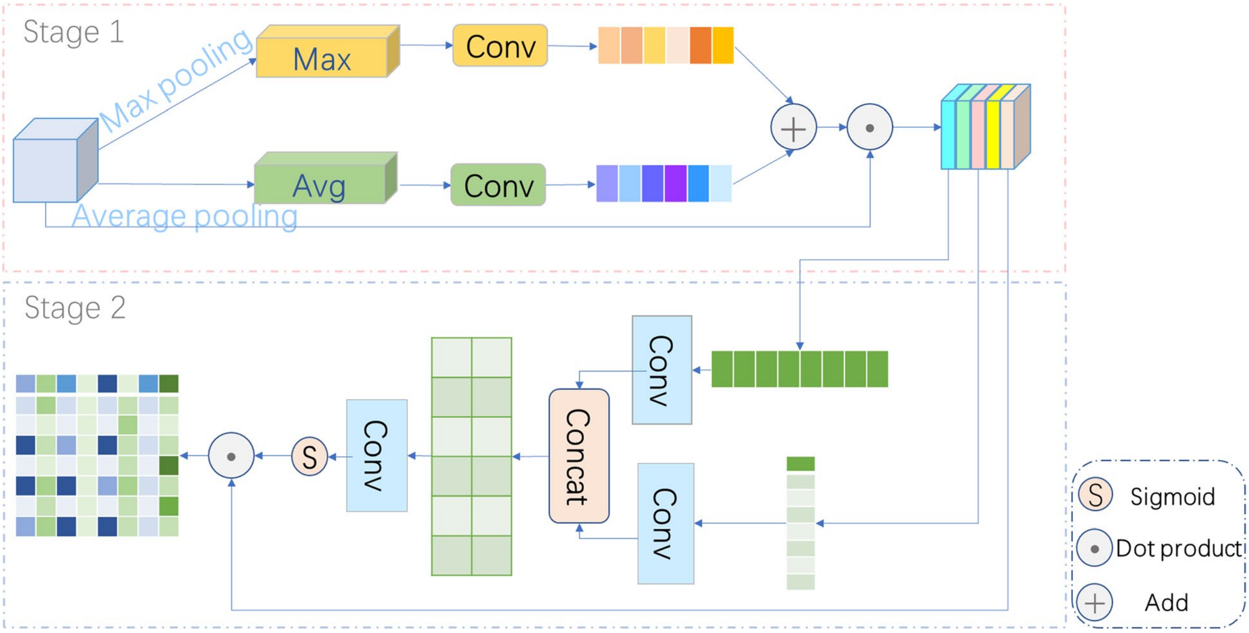

Fig. 4 The framework of WOEM

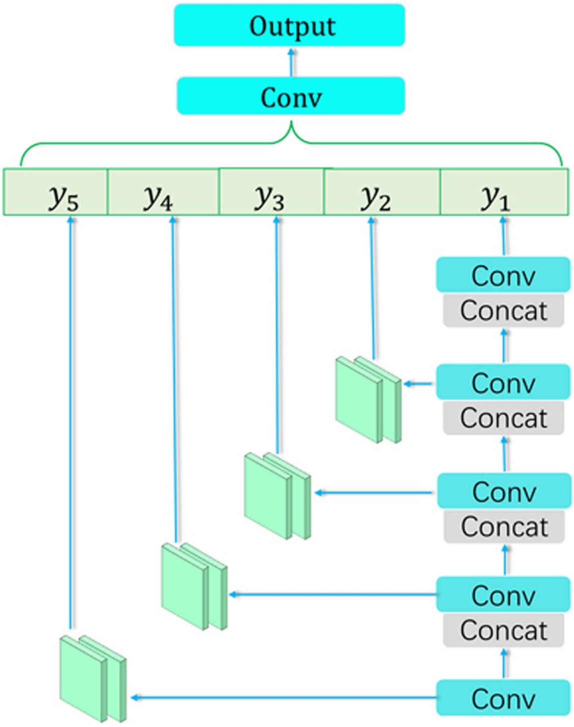

Fig. 5 The framework of the multi-branch decoder

an additional branch at each stage of the decoder for the enhancement of the deep semantic information. The feature maps of each stage of the decoder are directly up-sampled to the original map size, this up-sampling operation at different stages helps the decoder stage to learn multi-scale features. Finally, the decoded outputs of each stage are stitched together and fused, allowing the network to learn the important features of the water target at different scales adaptively. Specifically, at each stage of the decoder, the feature map is first directly up-sampled to the original map size using a bilinear interpolation method in order to reduce the information loss. Then the number of channels is transformed to 2 using a convolution kernel of 1×1 size, and finally, a batch normalization operation is added to prevent the network gradient from disappearing. The whole process can be described as follows:

$$D_i = BN(C_{1\times 1}(F(X_i)))\ i = 1, 2, 3, 4, 5 \tag{3}$$

where x_i represents the input at each stage of up-sampling, F represents the bilinear interpolation function, BN represents the batch normalization operation, $C_{1\times 1}$ represents a point convolution with a kernel size of 1×1. Its function is to fuse the multi-level feature information after splicing and change the dimension of the feature information, and D_i represents the final output of each branch.

The whole decoder is divided into 5 paths, one main path, and four branches. Each of the branched paths is collocated before the result is output. With multiple scales and multiple different sensory fields for fusion, the information about the water body at different levels can be learned more fully. The final fusion process is as follows:

$$Y_{out} = C_{1\times1}(concat(D_1, D_2, D_3, D_4, D_5)) \tag{4}$$

Where D_i is the output of each stage of the decoder, $concat(\cdot)$ indicates the superposition of channel dimensions.

Design of CBAM

Considering the overall computational and parametric size of the model, the convolutional block attention module (CBAM) is used in the last stage of the encoder. The main focus of the CBAM is on the texture features of the image. In high-resolution remote sensing images, the texture of the water object is very clear, so this module is very effective in distinguishing the water body from the background information. The CBAM module generates the respective attention maps along two different dimensions, which gives better results than the previous SENet which only focuses on the channels. The overall structure of the CBAM is shown in Fig. 6 , where the output of the encoder is first weighted by a channel attention module, followed by a spatial attention module to obtain the final output.

$$\begin{aligned} F_c &= \sigma(MLP(\text{Avg}(\text{X})) + \text{MLP}(\text{Max}(\text{X}))) \\ &= \sigma\left(W_1\left(W_0\left(X^c_{\text{avg}}\right)\right) + W_1\left(W_0\left(X^c_{\text{max}}\right)\right)\right) \end{aligned} \tag{5}$$

Equation 5 gives the basic process of channel attention calculation, σ represents the Sigmoid activation function, MLP represents the nonlinear fitting operation and W_1 represents the weight of attention. The whole channel attention mechanism aggregates features by averaging pooling and maximum pooling, compresses the feature map into a one-dimensional vector, and then sends it to a shared network for learning and element-by-element summation. The final generated attention graph is then multiplied with the original input to get the final output.

$$\begin{aligned} M_s(X') &= \sigma\left(f^{7\times7}\left(\left[\text{Avg}(\text{X}'); \text{Max}(\text{X}')\right]\right)\right) \\ &= \sigma\left(f^{7\times7}\left[X'^s_{\text{avg}}; X'^s_{\text{max}}\right]\right) \end{aligned} \tag{6}$$

Equation 6 gives the basic formulation of the spatial attention mechanism. X' represents the output of the channel attention mechanism, $f_{7\times7}$ represents the convolution operation of a 7×7 size convolution kernel. The spatial attention mechanism is a compression of the channel, where the average pooling and maximum pooling of each pixel point are performed at the channel level, thus highlighting the texture features of each pixel point.

Experiment

Dataset

The dataset used in this paper is Landsat-8 remote sensing image dataset. The data imaging time is the whole year of 2019 in the lower reaches of the Yangtze River, January to April 2021 in the middle reaches of the Yangtze River, and July to September 2020 in the upper reaches of the Yangtze River. The total number of samples used in the experiment is 2870, of which 2009 (70%) are used for network training, 574 (20%) for network validation, and 287 (10%) for testing. In order to further enhance the information content of the raw data, the image data were also processed by translation, rotation, mirroring and edge flipping to avoid overfitting.

To further validate the generalisation capability, the GID high-resolution image dataset released by the State Key Laboratory of Wuhan University, China, is also used in this paper. The GID dataset contains a total of 150 complete land-covered images with labels, including five different categories. The size of each image is 6800×7200, and the coverage area is large enough to include nearly 60 cities in China covering an area of 506 square kilometres. Since this paper extracts information about water bodies, labels other than water bodies are eliminated. In addition, in order to reduce the training cost, this paper crops the image to 512×512 size and excludes the samples with too large or too small proportion of water bodies. Other operations are the same as Landsat-8 dataset.

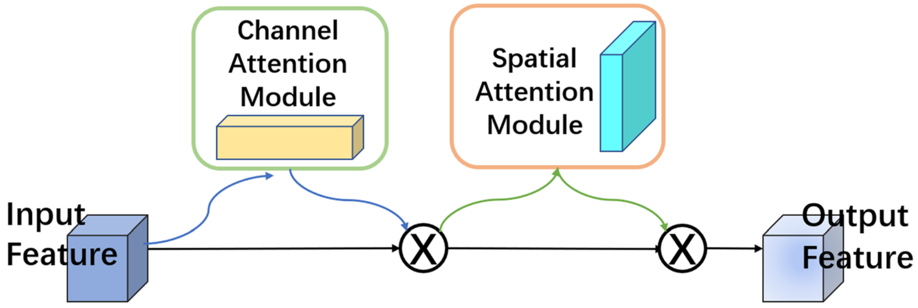

Fig. 6 The framework of CBAM

Evaluation metrics

Since the study is based on the binary classification task of water body extraction, two metrics are chosen in this paper to measure the accuracy of the results: the Recall and the F1-score. Recall measures how many targets the model is able to find correctly when recognising a target object. It is the ratio of the number of samples correctly detected as positive examples to the total number of true positive examples. F1 Score combines recall and precision and is used to evaluate the comprehensive performance of the model in image segmentation tasks to avoid the situation where the recall is high but the proportion of samples that are mistakenly identified as positive is also high. The formula for both is as follows:

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

where TP refers to the case where the model correctly predicts a positive example sample as a positive example. FN refers to the case where the model incorrectly predicts a positive example sample as a negative example. In addition to the above metrics, params, GFLOPs, and Memory metrics are used to measure the complexity and lightness of the model. The params represent the number of parameters of the model, GFLOPS represent the gigabit floating point operations per second of the model, and Memory represents the memory usage of the graphics card.

Experimental setup

The environment of the cloud center in this experiment is Windows 10 Professional Edition, and the processor is i9-10980XE. The GPU model is GeForce RTX™ 3060, and the memory is 32 GB. The optimizer used in lab is Adam, and the cosine annealing algorithm is used to dynamically adjust the learning rate size to avoid the network from falling into local optima. Considering that the experiment in this paper is a pixel-based classification task, the loss function used in this experiment is Dice Loss + Binary Cross Entropy Loss. BCE focuses on the pixel-by-pixel classification accuracy and strengthens the model's ability to determine whether each pixel point belongs to the target category, which is particularly suitable for dealing with category imbalance, while DCE measures the similarity between the predicted and the real segmented regions to optimise the overall quality of segmentation. This combination strategy ensures that the model captures the shape and size of the target region while focusing on individual pixel accuracy, thus striking a balance between ensuring local detail and overall consistency, which is crucial for improving the comprehensive performance of image segmentation. The formula for Binary Cross Entropy Loss is as below:

$$L_{BCE} = 1 - \frac{2\sum_i^N p_i g_i + soomth}{\sum_i^N p_i^2 + \sum_i^N g_i^2 + smooth} \tag{9}$$

where g_i represents the value of the true label, p_i on behalf of the prediction result, and N delegate total sample size.

$$L_{DCE} = 1 - \frac{2\sum_i^N p_i g_i + soomth}{\sum_i^N p_i^2 + \sum_i^N g_i^2 + smooth} \tag{10}$$

Equation 10 is the calculation process of Dce Loss, where p_i represents the predicted value of each pixel, g_i represents the label value of each pixel. The *smooth* is a small constant added to the denominator and numerator for stabilising the loss function. It is usually a very small positive number, such as 1e. It helps to mitigate the over-penalisation of the model for boundary pixels, thus improving the robustness of the model to some extent. The loss function used for the experiments in this paper is shown in Eq. 11.

$$L = L_{DCE} + L_{BCE} \tag{11}$$

Comparisons and analysis

In this paper, some classical network models and some advanced network models are selected to compare with the model proposed in this paper. The details are presented as follows:

- FCN was the first deep-learning method used for the semantic segmentation task. This model simply used the CNN backbone network to extract image features and substitute fully connected layer with a convolutional layer, then finally performed a direct up-sampling operation to restore to original size.
- DeepLabV3 [30] utilized cascading multiple convolutional modules with different hole rates to catch multi-scale contextual information to tackle the issue of multi-scale size of target objects. The ASPP structure proposed in V2 was extended by using four cascaded convolutions with different sampling rates and Batch Normalization in parallel at the top layer of the feature mapping.
- SPNET [31] first proposed the method of band-shaped pooling to capture the long-range relationship between isolated areas. In addition, a hybrid pooling module was designed to further model the

advanced semantic information. It used different shapes of pooling cores to detect complex scenarios to collect richer context information.

- BiseNetV1 [32] designed a two-way network in order to take into account the lightweight nature of the model and the fact that high-level semantic information was as important as shallow detailed information for semantic segmentation.
- DDRNet [33] proposed deep dual-resolution networks to learn richer contextual information with bilateral fusion and designed a new pyramid pooling module DAPPM to collect richer multi-scale information.

Among them, BISENETV1 and DDRNet are relatively advanced lightweight networks, and the lightest DDRNET-23-Slim is selected in DDRNET as the comparison object of the experiment. SPNET is a relatively complicated and heavy network.

Table 1 provides a comprehensive comparison of various models on the GID dataset, with a specific focus on their various performance metrics. It is clear from the table that PMNet performs well on two key evaluation metrics: recall and F1-score. PMNet's Recall, at 0.8963, is about 1.1 percentage points higher than its closest competitor, SPNet, which stands at 0.8760. In addition, the F1-score of PMNet is about 1.4 percentage points higher than that of SPNet. Although these differences may seem small in numerical terms, they demonstrate PMNet's excellent ability to accurately identify target object while minimising false positives and omissions. This balance of high recall and F1 scores demonstrates PMNet's robust and reliable performance in identifying true positives from non-target elements, thus greatly reducing the possibility of misclassification. In addition, PMNet's memory usage is further highlighted by its memory usage of only 338.2MB, which is much lower than SPNet's 1703.97MB. This lower memory requirement makes PMNet a more viable option for deployment in systems with limited memory resources. Although PMNet exhibits a slight disadvantage in terms of memory, parameters, and GFLOPs when compared to the more advanced lightweight network DDRNet, the difference is not significant and remains relatively small overall. Importantly, PMNet achieves a significant improvement in target recognition capability, with F1-scores about 5 percentage points higher and Recall about 7 percentage points higher compared to DDRNet. The significant improvement in performance metrics highlights PMNet's increased ability to accurately recognise target objects. Such results are acceptable under realistic conditions.

Table 1 Comparison of different models on Landsat-8 Dataset

Model Name	Recall↑	F1-score↑	Params↓	GFLOPs↓	Memory↓
FCN	0.8269	0.8527	15.3M	80.5	571.01MB
DeepLabV3	0.7755	0.8092	39.6M	25.5	578.01MB
BiseNetV1	0.7902	0.8347	<u>12.7M</u>	<u>13.02</u>	<u>228.42MB</u>
DDRNet	0.8113	0.8502	**5.6M**	**4.6**	**115.51MB**
SPNet	<u>0.8760</u>	<u>0.8941</u>	39.2M	158.9	1703.97MB
PMNet	**0.8963**	**0.9022**	14M	14.6	338.2MB

Bold is optimal, underline is suboptimal

Figure 7 in the study provides insightful visual comparative analysis. The recognition capabilities of the different models in terms of object extraction are shown in detail. These result images highlight the significant advantage of PMNet in this task, especially in its ability to detect smaller or finer objects. Compared to other models, PMNet is able to detect smaller or finer objects, while other models tend to ignore or misclassify this information. In addition, PMNet also shows higher accuracy in object edge delineation. Smaller objects or objects with inconspicuous boundaries are very important issues for accurate detection and classification tasks, and PMNet is able to accurately outline the edge details of objects.

Table 2 shows the result metrics of each model on the GID dataset. It is evident that the Recall and F1-score evaluation metrics of PMNet are the highest. Compared to SPNet, which has the smallest difference, PMNet is about 1.1 percentage points higher in the Recall metric and about 1.4 percentage points higher in the F1-score. In a word, the generalization ability of the PMNet model is still good. Figure 8 shows some prediction figures for each model on the GID dataset. Overall, the prediction results obtained by PMNet are still the best in terms of edge details and missed judgments of water bodies compared with other models.

Ablation analysis

The PMNet proposed in this article is composed of different modules. To confirm the validation of the modules for final results, this section conducts an ablation experiment.

Since the F1-score is a comprehensive evaluation index, it was chosen only as the judging index for this ablation experiment. As shown in Table 3, when the CBAM module and WOEM module are removed, the F1 score drops by 0.46 and 0.69 percentage points, respectively. When using the P-Block module and the multi-branch decoder module, the F1-score increased by 1.47 and 1.12 percentage points respectively, which are slightly larger. This shows that the learning of multi-scale features is more effective.

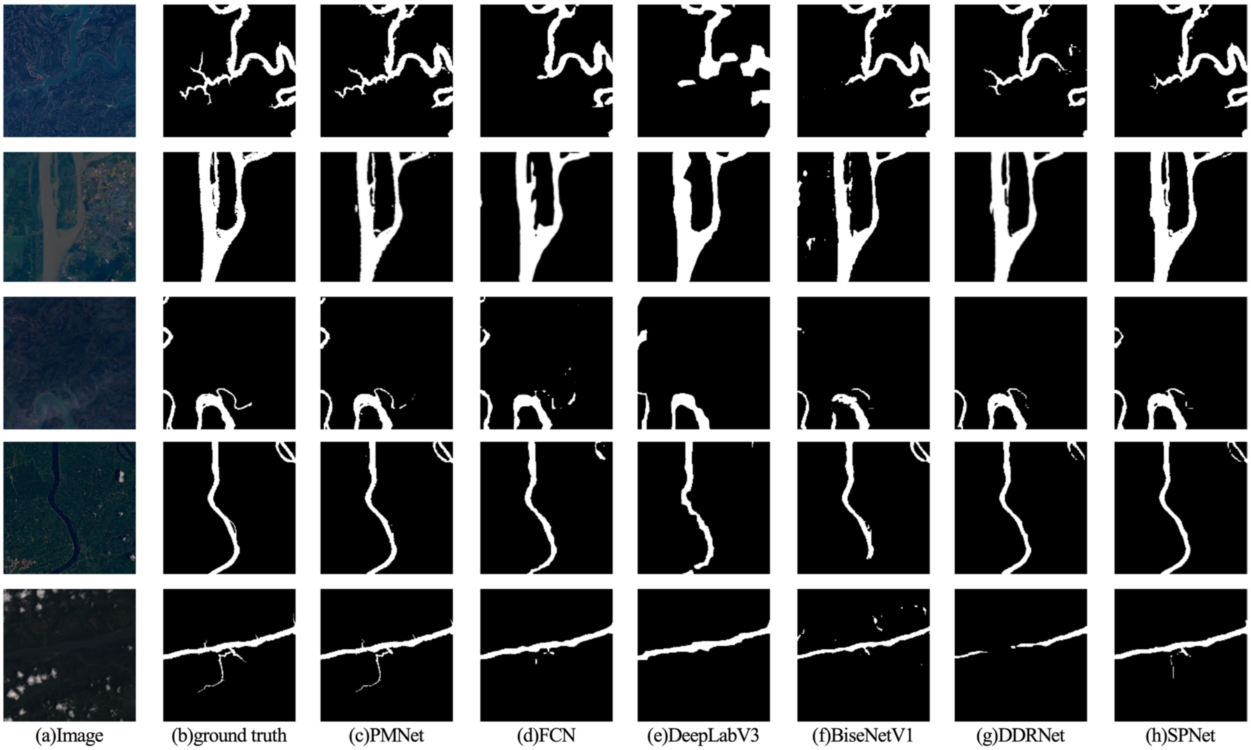

Fig. 7 Prediction of different models on Landsat-8 Dataset

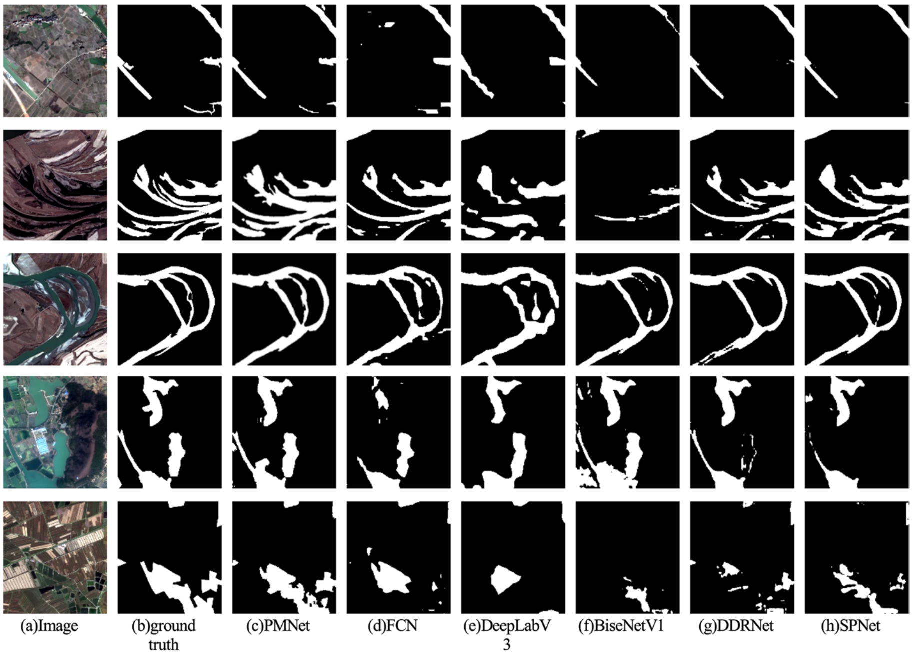

Fig. 8 Prediction of different models on GID Dataset

Table 2 Comparison of different models on GID Dataset

Model Name	Recall	F1-score
FCN	0.8269	0.8526
DeepLabV3	0.8789	0.8567
BiseNetV1	0.8298	0.8198
DDRNet	0.8555	0.8518
SPNet	0.8882	0.8714
PMNet	0.8993	0.8857

Table 3 Ablation study of our PMNet

Module	F1-score
PMNet-without-(P-Block)	0.8875
PMNet-without-(Multi-branch)	0.8910
PMNet-without-WOEM	0.8953
PMNet-without-CBAM	0.8976
PMNet	0.9022

Conclusion

This paper proposes an edge-cloud collaboratively driven network named PMNet to handle image segmentation tasks. Since large amounts of data need to be processed in mobile edge computing scenarios, efficient and lightweight networks are required, and PMNet is designed for this. By introducing the P-Block module, we successfully constructed a network with fewer parameters, narrower structure, and deeper structure, taking full advantage of the multi-scale characteristics of the image. In the decoding stage, in order to maintain deep semantic information and prevent the loss of deep semantic information during the gradual upsampling process, this paper designs a multi-branch decoder to enhance the detection of water objects by combining upsampling information at different scales. In addition, to further enhance the detailed information of the water body, we added a WOEM module between the encoder and decoder. Experiments show that the model proposed in this article achieved F1-score indicators of up to 90.22 and 88.57 on the two data, respectively, and minimized the number of parameters, floating point operations and memory usage. However, it should be noted that although this article proposes a lightweight method to deploy AI applications on mobile edge devices, it does not fully consider the real-time nature of model operations, such as the number of images that can be processed per second. Therefore, the next task will focus on implementing AI applications on mobile edge devices from a lightweight and real-time perspective for rapid data processing and analysis.

Acknowledgements

This work has received funding from the Key Laboratory Foundation of National Defence Technology under Grant 61424010208, National Natural Science Foundation of China (No. 41911530242 and 41975142), 5150 Spring Specialists (05492018012 and 05762018039), Major Program of the National Social Science Fund of China (Grant No. 17ZDA092), 333 High-Level Talent Cultivation Project of Jiangsu Province (BRA2018332), Royal Society of Edinburgh, UK and China Natural Science Foundation Council (RSE Reference: 62967_Liu_2018_2) under their Joint International Projects funding scheme and basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20191398 and BK20180794).

Authors' contributions

Design the study: Ziwen Zhang and Qi Liu. Collected the data from different sources: Yonghong Zhang, Zihao Du and Xuefei Cao. Analysis and interpretation of data: Ziwen Zhang,Qi Liu and Xiaodong Liu. Drafting of Manuscript: Ziwen Zhang and Qi Liu. All authors read and approved the final manuscript.

Funding

This work has received funding from the Key Laboratory Foundation of National Defence Technology under Grant 61424010208, National Natural Science Foundation of China (No. 41911530242 and 41975142), 5150 Spring Specialists (05492018012 and 05762018039), Major Program of the National Social Science Fund of China (Grant No. 17ZDA092), 333 High-Level Talent Cultivation Project of Jiangsu Province (BRA2018332), Royal Society of Edinburgh, UK and China Natural Science Foundation Council (RSE Reference: 62967_Liu_2018_2) under their Joint International Projects funding scheme and basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20191398 and BK20180794).

Availability of data and materials

No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 21 December 2023 Accepted: 15 March 2024
Published online: 27 March 2024

References

1. Rahman MR, Thakur PK (2018) Detecting, mapping and analysing of flood water propagation using synthetic aperture radar (sar) satellite data and gis: A case study from the kendrapara district of orissa state of india. Egypt J Remote Sens Space Sci 21:S37–S41
2. Holgerson MA, Raymond PA (2016) Large contribution to inland water co2 and ch4 emissions from very small ponds. Nat Geosci 9(3):222–226
3. Li W, Du Z, Ling F, Zhou D, Wang H, Gui Y, Sun B, Zhang X (2013) A comparison of land surface water mapping using the normalized difference water index from tm, etm+ and ali. Remote Sens 5(11):5530–5549
4. Hafizi H, Kalkan K (2020) Evaluation of object-based water body extraction approaches using landsat-8 imagery. J Aeronaut Space Technol 13(1):81–89
5. Qin P, Cai Y, Wang X (2021) Small waterbody extraction with improved u-net using zhuhai-1 hyperspectral remote sensing images. IEEE Geosci Remote Sens Lett 19:1–5
6. Feng W, Sui H, Huang W, Xu C, An K (2018) Water body extraction from very high-resolution remote sensing imagery using deep u-net and a

superpixel-based conditional random field model. IEEE Geosci Remote Sens Lett 16(4):618–622

7. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, p 3431–3440
8. Ahamad A, Sun CC, Kuo WK (2022) Quantized semantic segmentation deep architecture for deployment on an edge computing device for image segmentation. Electronics 11(21):3561
9. Yang Y, Ko YC (2022) Application of mobile edge computing combined with convolutional neural network deep learning in image analysis. Int J Syst Assur Eng Manag 13(Suppl 3):1186–1195
10. De Lucia G, Lapegna M, Romano D (2022) Towards explainable ai for hyperspectral image classification in edge computing environments. Comput Electr Eng 103:108381
11. Feyisa GL, Meilby H, Fensholt R, Proud SR (2014) Automated water extraction index: A new technique for surface water mapping using landsat imagery. Remote Sens Environ 140:23–35
12. McFeeters SK (1996) The use of the normalized difference water index (ndwi) in the delineation of open water features. Int J Remote Sens 17(7):1425–1432
13. Zhang Y, Liu X, Zhang Y, Ling X, Huang X (2018) Automatic and unsupervised water body extraction based on spectral-spatial features using gf-1 satellite imagery. IEEE Geosci Remote Sens Lett 16(6):927–931
14. Xu H (2006) Modification of normalised difference water index (ndwi) to enhance open water features in remotely sensed imagery. Int J Remote Sens 27(14):3025–3033
15. Guo Q, Pu R, Li J, Cheng J (2017) A weighted normalized difference water index for water extraction using landsat imagery. Int J Remote Sens 38(19):5430–5445
16. Milletari F, Navab N, Ahmadi SA (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV), IEEE, p 565–571
17. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, p 2881–2890
18. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intel 40(4):834–848
19. Yuan Y, Chen X, Wang J (2020) Object-contextual representations for semantic segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16, Springer, p 173–190
20. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, p 6000–6010
21. Huang Z, Wang X, Huang L, Huang C, Wei Y, Liu W (2019) Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, Seoul, Korea (South), p 603–612
22. Cao Y, Xu J, Lin S, Wei F, Hu H (2019) Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE/CVF international conference on computer vision workshops, Seoul, Korea (South), p 1971–1980
23. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q (2020) Eca-net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, p 11534–11542
24. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, p 7132–7141
25. Pu B, Lu Y, Chen J, Li S, Zhu N, Wei W, Li K (2022) Mobileunet-fpn: A semantic segmentation model for fetal ultrasound four-chamber segmentation in edge computing environments. IEEE J Biomed Health Inform 26(11):5540–5550
26. Wang W, Lin H, Wang J (2020) Cnn based lane detection with instance segmentation in edge-cloud computing. J Cloud Comput 9:1–10
27. Hernández D, Cecilia JM, Cano JC, Calafate CT (2022) Flood detection using real-time image segmentation from unmanned aerial vehicles on edge-computing platform. Remote Sens 14(1):223
28. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, p 770–778
29. Gao SH, Cheng MM, Zhao K, Zhang XY, Yang MH, Torr P (2019) Res2net: a new multi-scale backbone architecture. IEEE Trans Pattern Anal Mach Intell 43(2):652–662
30. Chen LC, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587
31. Al Arif SMR, Knapp K, Slabaugh G (2018) Spnet: Shape prediction using a fully convolutional neural network. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I, Springer, p 430–439
32. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV), p 325–341
33. Pan H, Hong Y, Sun W, Jia Y (2022) Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. IEEE Trans Intell Transp Syst 24(3):3448–3460

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.