Applied Intelligence (2023) 53:6826–6843
https://doi.org/10.1007/s10489-022-03932-3

Class-aware edge-assisted lightweight semantic segmentation network for power transmission line inspection

Qingkai Zhou[1] · Qingwu Li[1,2] · Chang Xu[1] · Qiuyu Lu[1] · Yaqin Zhou[1]

Accepted: 23 June 2022 / Published online: 11 July 2022

Abstract

The demand for real-time efficient scene comprehension has been increasing rapidly in the drone-based automatic inspection of power transmission lines (PTL). The extensive application of semantic segmentation in urban scenes proves that it can meet the requirements for scene understanding. However, existing methods have difficulty adapting to changes in the scene, which leads to problems of performance degradation and fuzzy contours of segmented objects. To overcome the existing problems, a class-aware edge-assisted lightweight semantic segmentation network is proposed in this paper. Class-aware edge detection is introduced as an auxiliary task, and a two-branch network is designed to locate instances and refine contours. Specifically, hybrid graph learning uses task-specific graph-based structures to reason attention information of region and edge features. Based on the complementary characteristic of region and edge features, cascaded shared decoders adopt specific interaction functions to enhance the ability of region features to locate targets and the ability of edge features to improve contour details. In addition, to verify the effectiveness of the proposed method, we construct two datasets named the transmission tower component recognition dataset (TTCRD) and the transmission line regional classification dataset (TLRCD). Comprehensive experiments on TTCRD and TLRCD prove that the proposed method can accurately refine the contour of objects and overcome the challenges in the two datasets. Comparison experiments and ablation experiments also demonstrate the superior performance of the proposed method and the effectiveness of each component in our architecture.

Keywords Power transmission line · Drone · Semantic segmentation · Class-aware edge detection · Hybrid graph learning

1 Introduction

Inspecting power transmission lines (PTL) is significant to ensure power grid safety [1]. The drone-based inspection method is an efficient and intelligent method. Compared with high spatial-resolution remote sensing images [2], aerial images captured by drones at a low altitude have the advantage of showing the scene details. Therefore, the distribution of the environment around PTL can be well identified, and equipment faults on PTL can be detected more economically and conveniently [3–5]. The conventional PTL drone-based inspection methods are usually completed by manual control [6]. However, it is often difficult to accurately understand the surrounding environment around drones and make corresponding adjustments. In addition, the conventional methods only collect images for postprocessing with poor real-time performance. Automatic

✉ Qingwu Li
li_qingwu@163.com

Chang Xu
xuchang@hhu.edu.cn

Qiuyu Lu
gdddgll@hhu.edu.cn

Yaqin Zhou
hhu_zyq@163.com

Qingkai Zhou
zhouqingkai@hhu.edu.cn

[1] College of Internet of Things Engineering, Hohai University, Changzhou, Jiangsu, 213022, China

[2] Changzhou Key Laboratory of Sensor Networks and Environmental Sensing, Changzhou, Jiangsu, China

drone-based inspection methods can navigate autonomously according to the PTL distribution [7], but the ability to deal with complex backgrounds and real-time performance of existing methods are still limited. To analyze complex backgrounds more accurately and quickly in automatic drone-based inspection, a real-time scene understanding method suitable for practical application is needed.

Among many drone-captured image analysis methods [8], semantic segmentation is one of the methods that can meet the above requirements. Semantic segmentation aims to predict pixel-level labels in images, which is helpful for understanding different scenes [9]. However, many popular semantic segmentation methods designed for drone-captured images are generally not suitable for the PTL inspection task [10–12]. The first reason is that most of these methods focus on urban scenes and analyze targets in street, intersection, or block view more meticulously by cropping high-resolution images. However, there are great differences between PTL scenes and urban scenes [13, 14], and the optimization tricks of these methods for urban scenes may not be appropriate for PTL scenes, resulting in poor performance. Another defect of popular semantic segmentation methods is that more layers and modules are applied to form a network with better performance but more complexity. Such a large-scale network has a large number of parameters and a low inference speed, and its training depends on a large number of annotated images [15]. Even if it is deployed on a drone platform, it is difficult to meet the requirements of stability and real-time performance.

Accordingly, two semantic segmentation datasets named the transmission tower component recognition dataset (TTCRD) and transmission line regional classification dataset (TLRCD) were constructed for the PTL automatic inspection task by drones. TTCRD is for segmenting components in the close-in transmission tower, and TLRCD is for the regional classification of PTL scenes. In two datasets, there is a problem of size differences between objects caused by different classes, distances, or flight altitudes. In addition, overlapping transmission towers and electrical fittings or high-density geographical elements may make segmentation more difficult.

Figure 1 shows the mIoU and the parameters of different methods on TTCRD. Methods with a large number of parameters are marked as blue points, and green points represent lightweight methods. Our method is represented by the red point. The performance of these networks with a high number of parameters is poor on our dataset. The overall performance of lightweight networks is better, such as UNet++ [16] of 9.2 M, SwiftNet [17] of 11.8 M, and ShelfNet [18] of 14.5 Their parameters do not exceed the deployment limit on drones, but their mIoU is less than 60%, and the IoU of small tower components, such as vertical insulators and dampers, is low. Our method has optimal overall performance. It achieves 68.93% mIoU at the parameter of 15.3 M. We also observed that when applied to our datasets, many existing methods segment objects with very fuzzy contours and cannot obtain the exact object shapes. Studies [19–21] show that the edge feature plays an important role in realizing accurate and intact segmentation of a target. In the semantic segmentation task, class-insensitive edge guidance does not distinguish edge pixels between classes, which leads to the inability to play a significant auxiliary role.

Aiming at all of the above problems and the imperfections of existing methods, a lightweight semantic segmentation network with a reasonable framework is proposed. We treat class-aware edge detection (CAED) as an auxiliary task and merge it with region reasoning. The motivation of CAED is to capture the real edges of objects and assist the semantic segmentation task to refine the contour of instances. Distinguishing strategies are applied to the early feature extraction stage of two tasks to reduce the number of parameters in our network and meet the multiscale requirement of semantic segmentation. In the feature attention stage, graph-based structures combine the collaborative learning of attribute information and structure information with long-term dependence knowledge. The attribute information is the category knowledge learned by the nodes, and the structure information is the connection between the edge nodes representing the target contour. In addition, the node similarity matrix of the region feature is processed as the adjacency matrix to guide the construction of the attentive edge feature. On the premise of avoiding redundant feature interaction, the feature representations are enhanced, and the close relationship between region and edge is established. In the decoding stage, cascaded shared decoders equip region features and edge features with the capability to improve each other and restore spatial details from hierarchical features. Therefore, shared decoders are helpful for accurately locating the edge of an object and improving its contour. To summarize, the contributions of this paper are listed as follows.

1. A novel lightweight semantic segmentation network with better performance is proposed for PTL drone-based inspection, which introduces class-aware edge detection as an auxiliary task to improve object integrity.
2. Hybrid graph learning is proposed to capture attribute information and structure information based on distinct elaborate graph structures in two tasks. In the interaction between two branches, the attribute and structure information of the region feature are extracted as guidance information to help the attention of the edge feature.

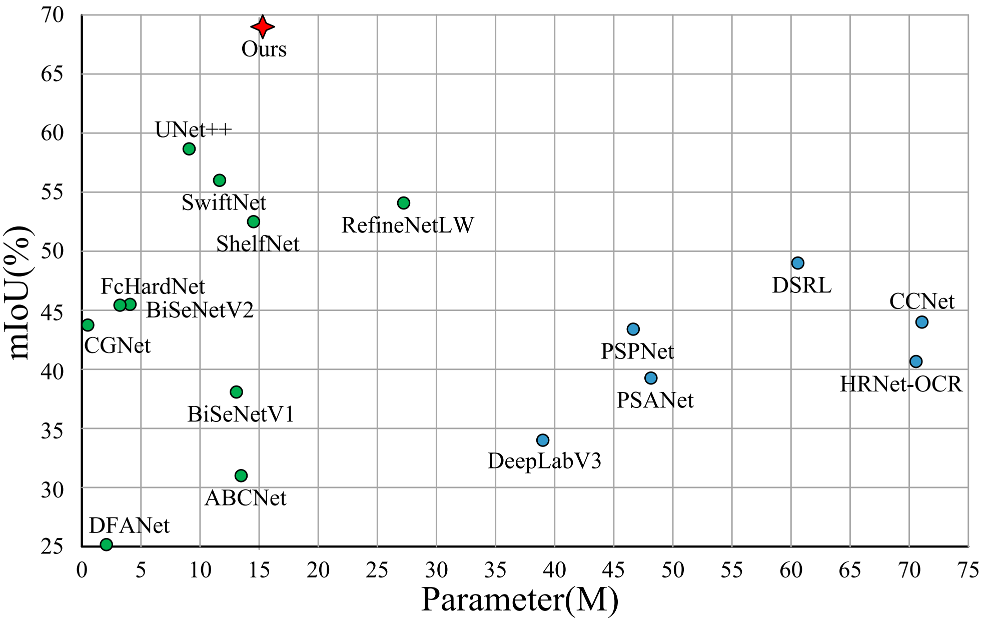


Fig. 1 Accuracy vs. parameters of popular semantic segmentation methods on TTCRD

3. Cascaded shared decoders are proposed to process the region and edge features at the same time and use hierarchical features step by step to restore the spatial details. By reasoning about the complementary relationship between the region and edge features, useful information from two tasks is sufficiently utilized to enhance interactive collaboration and constantly refine the edge of each instance.
4. Two semantic segmentation datasets have been constructed. One is for the recognition of electrical components on transmission towers, and another is for the regional classification of power transmission line scenes. Comprehensive experiments on two datasets demonstrate the effectiveness and superior performance of the proposed method.

The rest of this paper is organized as follows. Section 2 introduces related work on PTL inspection and semantic segmentation. The proposed network is described in Section 3. Section 4 introduces the constructed datasets first and presents the detailed experimental results and ablation studies. Finally, Section 5 summarizes this paper.

2 Related work

2.1 Analysis methods in the PTL inspection scene

Compared with traditional inspection methods, using drones to detect the status of power transmission lines is less expensive, faster and safer for transmission lines [22, 23]. In the PTL inspection scene, the regular status detection for transmission towers and electrical components is very important to better provide power service delivery [24].

For small-size defective component detection under a complex tower background, Jiao et al. [25] proposed context information and a multiscale pyramid network to detect defects in bolts. Liu et al. [26] improved RetinaNet anchor frame extraction mechanism based on a modified K-means++ algorithm, which achieved high-precision defect detection of towers, fittings, insulators and ground wires. In [24], an automatic inspection system was proposed for detecting component defects on transmission lines and transmission towers. To detect the key components of transmission lines in the image, Li et al. [27] constructed a scene classification dataset and proposed a novel network named TL-Net. Ma et al. [28] proposed an RGB-D saliency detection and skeleton structure search algorithm to detect insulators. Furthermore, binocular stereo vision and GPS were integrated to locate the detected insulators in real time. Considering the insulator defect inspection problem as a two-level object detection problem, Tao et al. [29] proposed a novel cascaded CNN network to localize insulators and detect defects simultaneously.

However, most of these methods focus on object-level prediction to make human eyes more intuitively understand the detection results. In contrast, our approach is designed as a lightweight semantic segmentation network for pixel-level prediction, which is more beneficial for computers embedded in drones to understand and analyze power transmission line scenes. Our method is appropriate for both identifying transmission tower components and region classification, which means that it has fewer limitations than other methods.

2.2 Semantic segmentation

Semantic segmentation is a basic task in the computer vision field. It has been greatly developed and has shown great application in scene understanding [30], autonomous driving [31], etc.

2.2.1 Generic semantic segmentation

With the great breakthrough of the fully convolutional network [32] in semantic segmentation, an increasing number of methods adopt deep learning models. PSPNet [33] and DeeplabV3+ [34] both exploited multiscale global context information and made great progress on multiple semantic segmentation datasets. The difference is that PSPNet applies a pyramid pooling module to generate various subregion representations, while DeeplabV3+ applies an atrous spatial pyramid pooling module with filters at multiple sampling rates and receptive fields. In PSANet [35], long-range contextual information was bidirectionally aggregated to form attention maps. RefineNet-LW [36] reduced part of the network parameters by modifying redundant building blocks. To better distinguish categories and improve details, a hierarchical multiscale attention mechanism was designed to combine the segmentation results of multiple inference scales in HRNet-OCR [37]. DSRL [38] added a single-image superresolution branch to enhance the high-resolution representation of low-resolution inputs by recovering detailed structural information. In CCNet [39], a recurrent 2D crisscross attention module was proposed to capture intensive and global context information both horizontally and vertically, and a discriminant loss function was used to guide the learning of category consistency features.

2.2.2 Lightweight semantic segmentation

To meet the demands of scene understanding under limited calculation, lightweight semantic segmentation models have become a research hotspot. UNet++ [16] innovatively reduced the semantic differences between the encoding and decoding feature maps by redesigning skip pathways and deep supervision. To strike a balance between the reasoning speed and performance of the network, the spatial path and context path were introduced in BiSeNetV1 [40]. The former preserved the spatial details of the original features, and the latter expanded the receptive field. To increase the receptive field, DFANet [41] incorporated multiple interconnected encoding streams, while SwiftNet [17] utilized a parameter-shared resolution pyramid. They both achieved a better tradeoff between efficiency and accuracy. Based on a novel shelf-shaped structure, ShelfNet [18] learned information flow from multiple paths. In addition, it greatly reduced the parameter number by decreasing the number of channels and using the shared weight strategy in the residual blocks. Aiming at the problem that memory tracking generated in the progress of accessing intermediate feature mapping will cause reasoning delay, HarDNet [42] was proposed to achieve high accuracy and efficiency by increasing the density of computation. Similar to BiSeNetV1, BiSeNetV2 [43] has a detail branch and a semantic branch and further utilizes a guided aggregation layer to strengthen the interaction between low-level details and high-level semantic information. To balance the effective utilization of global context information and a large amount of computational complexity, ABCNet [12] added an attention enhancement module on the basis of bilateral paths. CGNet [44] achieves good performance with extremely small parameter numbers by utilizing convolution layers of small channels and context-guided modules.

However, the above semantic segmentation methods mostly focus on multiscale information and context features while ignoring edge features. In contrast, considering that the contour of the segmented object is accurate the key to semantic segmentation, our approach is proposed to capture the intact edges of objects. In addition, since labels of various classes are different in the semantic segmentation task, we construct class-aware edge ground truth to assist supervision and establish two tasks, namely, semantic segmentation and class-aware edge detection. Different from other methods that directly collect a single high-level feature in the encoding phase, our approach extracts a multiscale region feature and an elaborate edge feature in differentiated ways. Moreover, two graph structures are proposed. The first structure combines the collaborative learning of attribute information and structure information with long-term dependence knowledge. The second structure constructs the attentive information of edge features. Finally, on the premise of ensuring accuracy, we use a lightweight backbone and channel compression operation to restrict the parameter of the proposed network.

3 Proposed approach

3.1 Overview

As shown in Fig. 2, class-aware edge detection is introduced to assist semantic segmentation, and a lightweight two-branched network is designed for joint learning, which is divided into three parts. Different from other encoders that directly collect a single high-level region feature, the two-task feature extraction (TTFE) module extracts a multiscale region feature and an elaborate edge feature in differentiated ways. Hybrid graph learning includes two modules named region attention (RA) and region-guided edge attention (RGEA). The RA module modifies

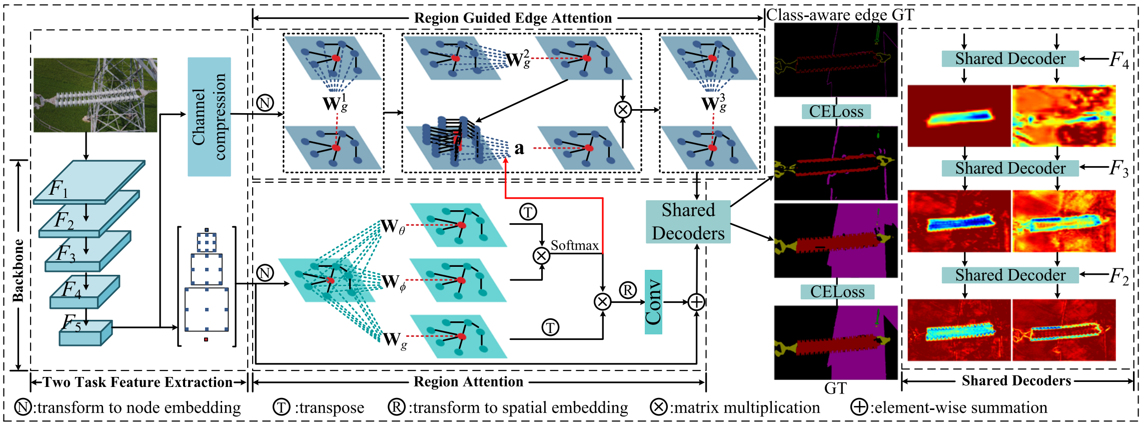


Fig. 2 The overview of the proposed framework

the nonlocal module [45] by replacing the embedding weights with graph convolution layers and finishes the collaborative learning of attribute information and structure information based on long-term dependence knowledge. The RGEA module is designed in the form of a graph convolution layer-graph attention layer-graph convolution layer. It constructs the adjacency matrix with the help of the similarity matrix introduced from the RA module and forms the attentive information of edge features. In contrast to the split decoding design, cascaded shared decoders (SDs) jointly deal with region and edge features and explore their complementary relationship reasonably. In the rightmost part of Fig. 2, we show the heatmaps of the region features and edge features output by three shared decoders. They are generated with Grad-CAM [46, 47] and illustrate the attention information of the horizontal insulator. Finally, the segmentation results of the edge and region are generated in the last step, and they are supervised. In the rest of this section, we describe each part of the network framework in detail and briefly introduce the loss function.

3.2 Two-task feature extraction

TTFE aims to extract abundant semantic information and take distinguishing strategies to generate a multiscale region feature and an elaborate edge feature. Taking an image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ as input, TTFE deploys a lightweight ResNet18 network as the backbone to extract deep features, including five-stage hierarchical features $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$, and $\mathbf{F}_5$. Through layer-by-layer downsampling operations, the heights and widths of hierarchical features are reduced to 1/2, 1/4, 1/8, 1/16 and 1/32.

Then, a channel compression operation and an atrous spatial pyramid pooling (ASPP) module [34] are applied to process $\mathbf{F}_5 \in \mathbb{R}^{C\times h\times w}$ to generate two task-specific features named $\mathbf{F}_E \in \mathbb{R}^{c\times h\times w}$ and $\mathbf{F}_R \in \mathbb{R}^{c\times h\times w}$, where h and w denote the height and weight of the feature map, and c denotes the compressed channel number. The channel compression operation is realized by a convolution layer:

$$\mathbf{F}_E = f_{conv}\left(\mathbf{F}_5; \mathbf{W}_E\right), \tag{1}$$

where $\mathbf{W}_E \in {}^{C\times c}$ is the weight parameter of the convolution layer with a 1×1 kernel size.

Atrous convolution is the core of the ASPP module. This means the convolution layer with different strides, which is denoted as $f_{conv}(; \mathbf{W}, s)$, and s represents the stride. In this module, four atrous convolution layers with strides of [1,3,5,7] are used to process $\mathbf{F}_5$,

$$\mathbf{R}_i = f_{conv}\left(\mathbf{F}_5; \mathbf{W}_i, s_i\right), i = 1, ..., 4, s_i \in [1, 3, 5, 7], \tag{2}$$

To maintain global semantic information, an adaptive average pooling layer and a bilinear upsample operation are used to handle $\mathbf{F}_5$,

$$\mathbf{R}_5 = \text{Upsample}\left(f_{avgpool}\left(\mathbf{F}_5\right)\right). \tag{3}$$

Finally, we concatenate the feature maps from five branches and use a convolution layer to export the multiscale region feature:

$$\mathbf{F}_R = f_{conv}\left([\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4, \mathbf{R}_5]; \mathbf{W}_R\right) \tag{4}$$

$\mathbf{F}_E$ is used for class-aware edge detection (CAED), and $\mathbf{F}_R$ is used for semantic segmentation (SS). The attenuation of the channel number from C to c contributes to removing noise information, reducing redundant features and lightening the network. ASPP captures multiscale information through a larger receptive field, which makes our method more robust in the challenge of size differences. Notably, ASPP is not utilized in task CAED because CAED emphasizes the continuity of edges rather than the size differences of regions.

3.3 Hybrid graph learning

Hybrid graph learning aims at constructing the global attention information of region features and utilizes it as guidance to reason the graph representation of edge features. Then, the coherence of edges is enhanced during this process.

The graph convolution layer [48, 49] learns structure information and attribute information simultaneously, without splitting and deconstructing. In the feature map, the attribute information describes the inherent properties of the nodes in the graph, that is, the feature description in all channels of the node. The structure information describes the correlation and the similarity measurement between nodes. In the CAED task, the attribute information is the category knowledge learned by the nodes, and the structure information is the connection between the edge nodes representing the contour of objects. Generally, attribute information and structure information have a good complementary relationship. For some graphs with sparse structures, attribute information can improve the learning quality of the model for nodes. In addition, the knowledge contained in the structure information plays a very important role in describing nodes in the attribute information. The graph convolution layer puts structure information and attribute information into a network layer for simultaneous learning so that two pieces of information can synergistically influence the representation of the final node. The graph convolution layer is described with the following formula:

$$f_{g-conv}(\mathbf{F};\mathbf{W}) = \left(\mathbf{A} \times \mathbf{F}^{\mathrm{T}} \times \mathbf{W}\right)^{\mathrm{T}}, \mathbf{A} = \mathrm{softmax}\left(\mathbf{F}^{\mathrm{T}} \times \mathbf{F}\right), \tag{5}$$

where $\mathbf{F}^{\mathrm{T}} \times \mathbf{W}$ is the affine transformation of attribute information, which learns the interaction between attribute features. The adjacency matrix $\mathbf{A}$ is constructed by computing the similarity between nodes. Therefore, from the perspective of the spatial domain, $\left(\mathbf{A} \times \mathbf{F}^{\mathrm{T}} \times \mathbf{W}\right)^{\mathrm{T}}$ is the process of aggregating neighboring nodes, which represents the coding of structure information. The contour of an object is continuous, and its structure is clear. Therefore, by learning the attribute information and structure information, we can build a close connection between edge nodes and effectively distinguish the contours between different instances.

3.3.1 Region attention

The effectiveness of nonlocal blocks for segmentation is showcased in reference [45]. To combine the collaborative learning of attribute information and structure information with long-term dependence knowledge, three embedding weights of the nonlocal block are replaced by graph convolution layers. Given the region feature $\mathbf{F}_R \in \mathbb{R}^{c\times h\times w}$, the graph nonlocal block first transforms it to graph node embedding $\mathcal{V}_R \in \mathbb{R}^{c\times k}$, in which $k = h \times w$. Then, three graph convolution layers are exploited to process $\mathcal{V}_R$,

$$\begin{aligned}\mathcal{V}_\theta &= f_{g-conv}(\mathcal{V}_R;\mathbf{W}_\theta), \mathcal{V}_\phi = f_{g-conv}(\mathcal{V}_R;\mathbf{W}_\phi),\\ \mathcal{V}_g &= f_{g-conv}(\mathcal{V}_R;\mathbf{W}_g),\end{aligned} \tag{6}$$

where $\mathbf{W}_\theta$, $\mathbf{W}_\phi$ and $\mathbf{W}_g \in \mathbb{R}^{c\times c}$ are the weight parameters of the three layers. $\mathcal{V}_\theta$, $\mathcal{V}_\phi$ and $\mathcal{V}_g$ represent query, key and value maps, respectively. Then, the similarity matrix between the query and key maps is calculated as

$$\mathbf{A}_R = \mathrm{softmax}\left(\mathcal{V}_\theta^{\mathrm{T}} \times \mathcal{V}_\phi\right), \tag{7}$$

where $\mathbf{A}_R \in \mathbb{R}^{k\times k}$. $\mathbf{A}_R\,(i, j)$ values the affinity between the query node i and the key node j. The attention mechanism is achieved by aggregating the similarity matrix and the value map,

$$\mathbf{Y}_R = \mathbf{A}_R \times \mathcal{V}_g^{\mathrm{T}} \tag{8}$$

where $\mathbf{Y}_R \in \mathbb{R}^{k\times c}$ and it is reshaped and flattened to $\mathbf{Y}'_R \in \mathbb{R}^{c\times h\times w}$. Finally, the graph nonlocal block learns the attentive feature and residual representation synchronously as:

$$\mathbf{F}'_R = \mathbf{F}_R + f_{conv}(\mathbf{Y}_R;\mathbf{W}_z), \tag{9}$$

where $f_{conv}()$ denotes a convolution layer with a kernel size of 1×1, and $\mathbf{W}_z$ is the corresponding weight.

3.3.2 Region-guided edge attention

The purpose of RGEA is to learn the attribute information and structure information of objects with the guidance of attentive region characteristics to construct continuous and accurate edge features. In this module, $\mathbf{F}_E$ is first transformed to graph node embedding $\mathcal{V}_E \in \mathbb{R}^{c\times k}$, where $k = h \times w$ denotes the number of nodes. Then, a graph convolution layer is used to enhance the graph representation as:

$$\mathbf{E}_1 = f_{g-conv}(\mathcal{V}_E;\mathbf{W}_g^1), \tag{10}$$

where $\mathbf{W}_g^1 \in \mathbb{R}^{c\times c}$ is a learnable parameter of the graph convolution layer, and $\mathbf{E}_1 \in \mathbb{R}^{c\times k}$. The convolutional kernel of this layer acts on all nodes in the whole graph, and the weight parameters are shared in the calculation at each node. With such a treatment, the number of parameters in our network is greatly reduced, and the occurrence of the overfitting phenomenon is effectively avoided.

After that, a graph attention layer establishes contact between edge nodes with similar characteristics. As an initial step, $\mathbf{E}_1$ is regarded as a set of nodes, namely, $\mathbf{E}_1 = \{\upsilon_1, \upsilon_2, ..., \upsilon_k\}$. Then, a linear transformation is applied to every node $\upsilon_i \in \mathbb{R}^c$ for learning attribute information, and the formula is:

$$\upsilon_i{}' = \mathbf{W}_g^2 \times \upsilon_i, \mathbf{W}_g^2 \in \mathbb{R}^{c \times c}. \tag{11}$$

The attention mechanism is then performed on the nodes, and the weight coefficient between a node and other nodes in the neighborhood is calculated by:

$$\omega_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^{\mathrm{T}}[\upsilon_i{}' \parallel \upsilon_j{}']))}{\sum_{t \in \mathbb{N}_i} \exp(\text{LeakyReLU}(\mathbf{a}^{\mathrm{T}}[\upsilon_i{}' \parallel \upsilon_t{}']))}, \tag{12}$$

where ω_{ij} denotes the importance of node j to node i, and $\mathbf{a} \in \mathbb{R}^{2c \times 1}$ is a learnable parameter. LeakyReLU is selected as the activation function. $t \in \mathbb{N}_i$ denotes nodes belonging to the neighborhood of node i. To make the layer learn more robust structure information, we innovatively introduce the similarity matrix of region features and set a threshold to filter more similar nodes,

$$\mathbf{A}_E\,(m, n) = \begin{cases} 1, & if \quad \mathbf{A}_R\,(m, n) > th \\ 0, & if \quad \mathbf{A}_R\,(m, n) \le th \end{cases}, \tag{13}$$

$\mathbf{A}_E$ denotes the adjacency matrix, which determines the neighborhood of each node m by selecting all nodes that satisfy $\mathbf{A}_E\,(m, n) = 1$. Then, $\mathbb{N}_i$ can be generated from $\mathbf{A}_E$. Experiments show that the best effect is achieved when th=0.8. The induction of the region information is helpful for constructing the connection of continuous edge nodes of the same instance. Once the weight coefficient is obtained, the new feature vector of node i is calculated as:

$$\upsilon_i{}'' = \sigma\left(\sum\nolimits_{j \in \mathbb{N}_i} \omega_{ij} \upsilon_j{}'\right), \tag{14}$$

and feature $\mathbf{E}_2 = \left\{\upsilon_1'', \upsilon_2'', ..., \upsilon_k''\right\} \in \mathbb{R}^{c \times k}$ is obtained. Finally, feature learning is carried out through a graph convolution layer:

$$\mathbf{E}_3 = f_{g-conv}(\mathbf{E}_2; \mathbf{W}_g^3). \tag{15}$$

$\mathbf{E}_3 \in \mathbb{R}^{c \times k}$ is then transformed back to the original coordinate space, which can be represented as $\mathbf{F}_E{}' \in \mathbb{R}^{c \times h \times w}$.

3.4 Shared decoders

The two tasks of SS and CAED have complementary characteristics and can be refined by each other, so the joint calculation is essential in shared decoders. Integrating hierarchical features is helpful for restoring details in the decoding stage. Based on this principle, a cascaded structure formed by three shared decoders is designed and shown in the rightmost part of Fig. 2. The goal of SDs is to equip region features and edge features with the capability to improve each other and restore spatial details from hierarchical features to accurately locate the edge of an object and improve its contour. In addition, to visualize the attention information of horizontal insulators, we adopt Grad-CAM [46, 47] to generate heatmaps of the region feature and edge feature processed by three decoders. Obviously, the edge information becomes continuous and accurate, and the contour in the region feature is gradually improved. This proves the rationality and effectiveness of our shared decoders. The architecture of a single shared decoder is shown in Fig. 3.

As mentioned above, we take the first SD as an example; it takes three feature maps as input, including $\mathbf{F}_E{}' \in \mathbb{R}^{c \times h \times w}$, $\mathbf{F}_R{}' \in \mathbb{R}^{c \times h \times w}$ and $\mathbf{F}_4$. $\mathbf{F}_E{}'$ and $\mathbf{F}_R{}'$ are first upsampled to the resolution of $\mathbf{F}_4$. Then, three feature maps are embedded with different weights:

$$\begin{aligned} \mathbf{D}_{R1} &= f_{conv}\left(\mathbf{F}_R{}'; \mathbf{W}_d^1\right), \mathbf{D}_{E1}{=}f_{conv}\left(\mathbf{F}_E{}'; \mathbf{W}_d^2\right), \\ \mathbf{D}_{H1} &= f_{conv}\left(\mathbf{F}_4; \mathbf{W}_d^3\right), \end{aligned} \tag{16}$$

where $\mathbf{W}_d^1$, $\mathbf{W}_d^2 \in \mathbb{R}^{c \times c}$ and $\mathbf{W}_d^3 \in \mathbb{R}^{256 \times c}$ are weight parameters of convolution layers with a 3×3 kernel size. Subsequently, the detailed information of hierarchical features is integrated into the region feature and edge feature by pixel-level multiplication. We argue that benefiting from the correlation between region features and edge features, the multiplication between them can effectively enlarge the variance between edge nodes and nonedge nodes, while the summation can enhance the edge node representation and refine the instance contours. In this way, the edge nodes in the region feature contribute to object contour improvement, and the edge nodes in the edge feature are more prominent. More specifically,

$$\begin{aligned} \mathbf{D}_{R1}{}' &= (\mathbf{D}_{R1} * \mathbf{D}_{H1}) + (\mathbf{D}_{E1} * \mathbf{D}_{H1}), \\ \mathbf{D}_{E1}{}' &= (\mathbf{D}_{R1} * \mathbf{D}_{H1}) * (\mathbf{D}_{E1} * \mathbf{D}_{H1}), \end{aligned} \tag{17}$$

where $*$ and $+$ denote pixelwise multiplication and summation. Finally, two convolution layers are leveraged to two feature maps:

$$\begin{aligned} \mathbf{F}_R^{d1} &= f_{conv}\left(\mathbf{D}_{R1}{}' + \text{Upsample}\left(\mathbf{F}_R{}'\right); \mathbf{W}_d^4\right), \\ \mathbf{F}_E^{d1} &= f_{conv}\left(\mathbf{D}_{E1}{}'; \mathbf{W}_d^5\right), \end{aligned} \tag{18}$$

where $\mathbf{W}_d^4$ and $\mathbf{W}_d^5 \in \mathbb{R}^{c \times 256}$. The latter two SDs differ only in the inputs and variation in the number of channels. After cascaded SDs, two prediction heads are utilized to generate the final label map P_{SS} and P_{CAED}. Each prediction head contains two convolution layers. Their weights can be represented by $\mathbf{W}_P^1 \in \mathbb{R}^{64 \times 32}$ and $\mathbf{W}_P^2 \in \mathbb{R}^{32 \times n}$, where n denotes the number of classes.

3.5 Loss function

SS and CAED are supervised simultaneously in the training stage. The class-aware edge detection result is not generated in the testing stage. The ground-truth label G_{SS} is processed to obtain the class-aware edge ground truth. First, the binary maps of all classes are generated from G_{SS}. Then, the

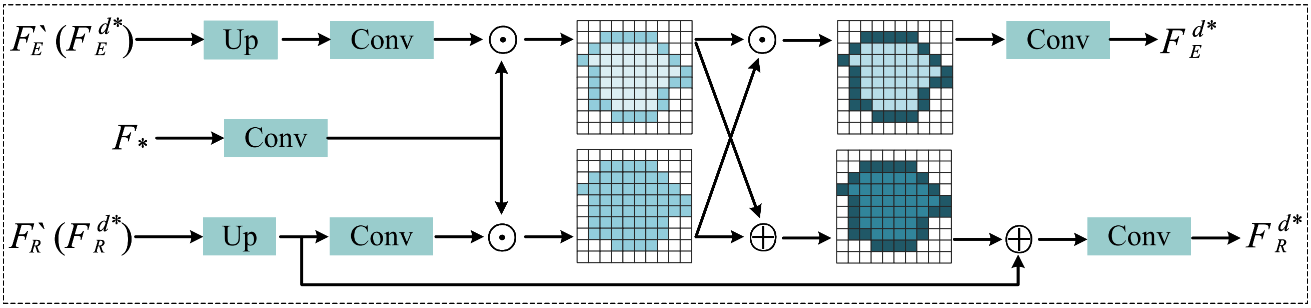


Fig. 3 The structure of a single shared decoder. ⊙ and ⊕ denote pixelwise multiplication and summation

edge of each object is extracted by calculating the gradients of the binary map in the X and Y directions. Finally, the class-aware edge ground truth G_{CAED} is obtained by merging all edge maps, and each pixel is labeled with its original class id. The loss function is designed based on joint cross-entropy loss [50]:

$$L = L_{CE}^{SS}\left(P_{SS}, G_{SS}\right) + w * L_{CE}^{CAED}\left(P_{CAED}, G_{CAED}, \right) \quad (19)$$

where w=0.5 is the combination weight. Moreover, the class weighting technique is applied to deal with the pixel number imbalance between classes. Specifically, the weight of each class is formulated as:

$$\omega_{class} = \frac{1}{\ln\left(1.2 + P_{class}\right)}, \quad (20)$$

where P_{class} represents the pixel ratio of each class.

4 Experiments

In this part, the preparatory work of the experiment is first introduced, and then the proposed method is compared with the most advanced models quantitatively and qualitatively, which shows its superiority in accuracy and parameter cost. Finally, the contribution of each component was evaluated by an ablation study.

4.1 Dataset

To meet the requirements of drone-based PTL inspection and verify the effectiveness of the proposed method, a transmission tower component recognition dataset (TTCRD) and a transmission line regional classification dataset (TLRCD) are proposed for training and evaluation.

4.1.1 Data acquisition and annotation

Several aspects, such as the data collection method, drone flight strategy, and class selection, are considered during dataset design. To ensure that drones can observe tower components from multiple angles and understand power transmission lines from multiple directions during inspection, we collected images at different observation points and altitudes.

Additionally, because of the high shooting frame rate of the camera on a drone, there are many repeated and similar images. Therefore, the problem of data redundancy is prevented by screening and eliminating images of high similarity. Finally, we obtain 312 images for TTCRD and 264 images for TLRCD. The image resolution is either 5280×2970 or 2448×2048, which contains clear image details and enables the accurate segmentation of tiny tower components and long-distance scene contents.

It is difficult for human eyes to define some objects and distant scenes from high-resolution and high-complexity images. Therefore, only the most common and representative object categories are considered when labeling. We adopt the LabelMe [51] annotation tool to label high-resolution aerial images at the pixel level. Some annotation examples and class definitions of the two datasets are shown in Fig. 4(a) and (b). The first row of Fig. 4(a) or (b) shows the aerial images, and the second row shows annotated labels. The corresponding colors and the specific kinds of all classes are shown in the right part.

In addition, TTCRD is divided into training and testing splits, which contain 240 and 72 images, respectively. The TLRCD is divided into 200 training images and 64 testing images. The resolution of all images is resized to 512 × 512 during training and testing.

4.1.2 Challenges

There are some difficulties and challenges in the two datasets due to the inherent characteristics of the PTL scenarios. The following is a detailed introduction to these challenges.

Size Difference and Pixel Number Imbalance It can be observed in Fig. 4 that the size of different classes varies

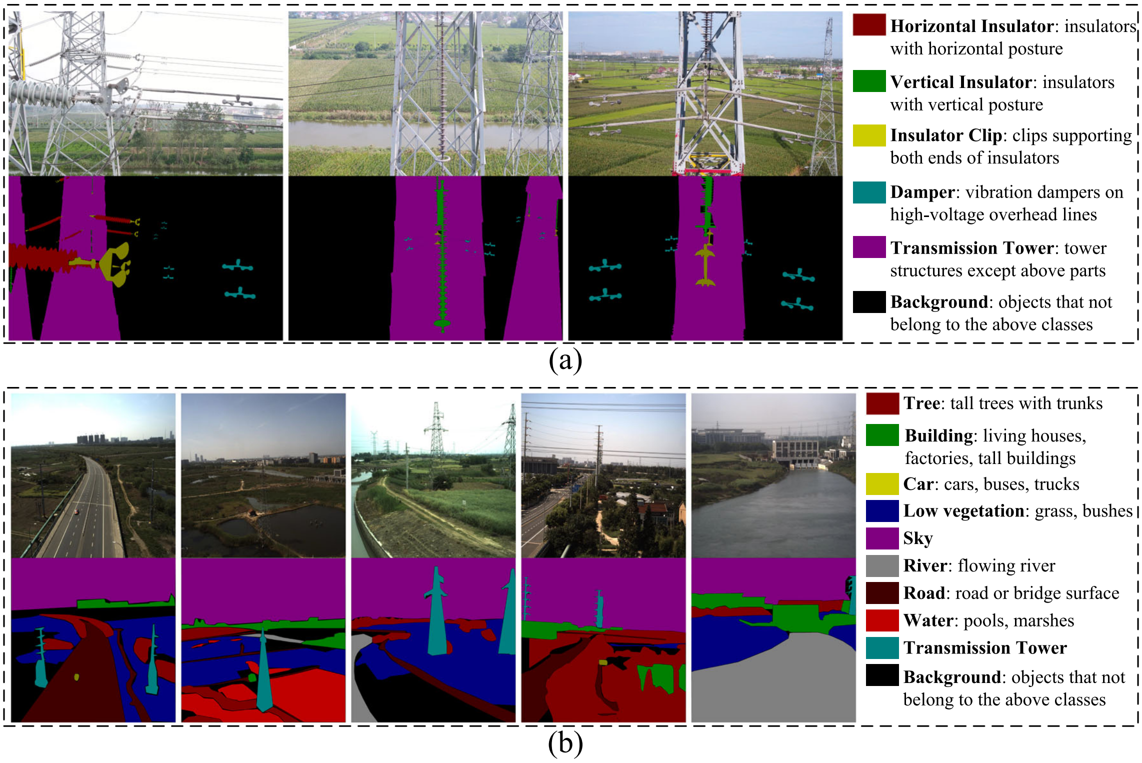

Horizontal Insulator: insulators with horizontal posture
Vertical Insulator: insulators with vertical posture
Insulator Clip: clips supporting both ends of insulators
Damper: vibration dampers on high-voltage overhead lines
Transmission Tower: tower structures except above parts
Background: objects that not belong to the above classes
(a)
Tree: tall trees with trunks
Building: living houses, factories, tall buildings
Car: cars, buses, trucks
Low vegetation: grass, bushes
Sky
River: flowing river
Road: road or bridge surface
Water: pools, marshes
Transmission Tower
Background: objects that not belong to the above classes
(b)

Fig. 4 Example aerial images, labels and class definitions. (a) TTCRD, (b) TLRCD. The first row shows the images captured by the drone. The second row shows the corresponding annotated labels. Right color bars indicate the annotation colors detailed definitions corresponding to all classes

greatly, and even objects of the same class have large size differences due to the diverse distances from shooting points.

We calculated statistics on the pixel numbers of each class in the two datasets, which are shown in Fig. 5. It can be seen in both Fig. 5(a) and (b) that the pixel distribution of different classes is extremely unbalanced. In TTCRD, most pixels come from backgrounds and transmission towers, and few pixels come from horizontal insulators. However, the pixel ratios of insulator clips, vertical insulators and dampers are all less than 1%. In TLRCD, most pixels come from the sky, backgrounds, trees and low vegetation,

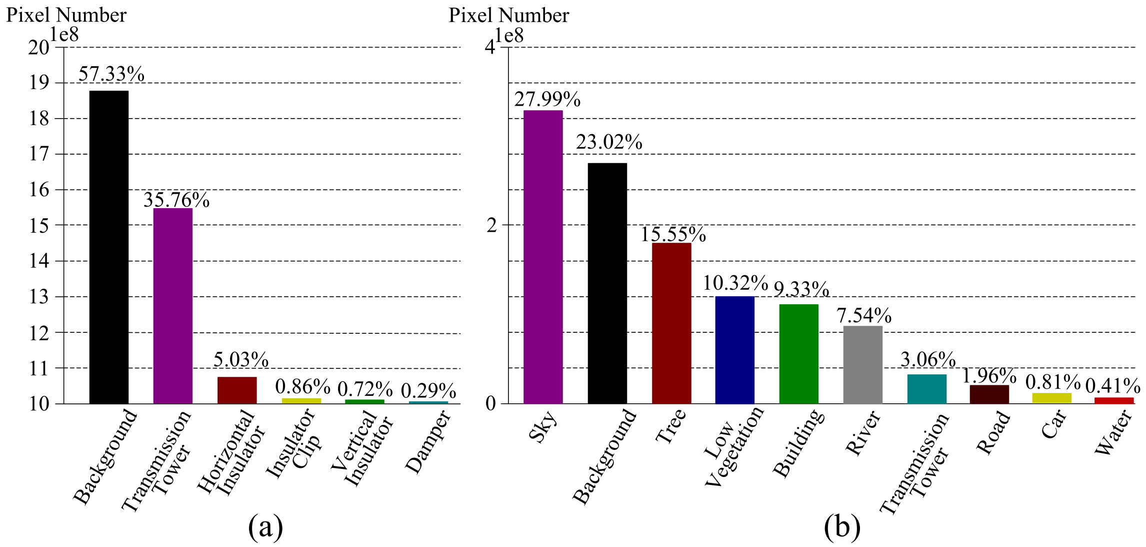


Fig. 5 Pixel number histogram. (a) TTCRD, (b) TLRCD

and few pixels come from buildings, rivers, transmission towers and roads, while pixel ratios of car and water are less than 1%. Few instances and small sizes of car and water result in low proportions of these two classes in TLRCD. However, TTCRD has many instances of insulator clips, vertical insulators and dampers, but compared with transmission towers, their sizes are much smaller, which leads to an imbalance in the number of pixels.

Overlap and occlusion In TTCRD, the structure and layout of transmission towers are complicated. Most of the vertical insulators are in the tower body, and horizontal insulators extend from the tower body. The insulator clip is generally on both sides of an insulator. Therefore, the problem of overlapping and occlusion is very serious.

Similarly, in TLRCD, the distribution of high-density geographical elements is irregular and complicated. The oblique view will also worsen the problem of overlapping and occlusion.

4.2 Performance metric and implementation details

The commonly used metric, mean intersection over union (mIoU), is applied to evaluate the performance of semantic segmentation on both datasets. The IoU of a certain class and the mIoU of all classes can be calculated as:

$$IoU = \frac{TP}{TP + FP + FN},$$
$$mIoU = \frac{1}{N}\sum\nolimits_{n=1}^{N} \frac{TP_n}{TP_n + FP_n + FN_n} \tag{21}$$

The experiments are performed on a platform with an NVIDIA RTX 3060 GPU. The initial learning rate and weight decay are set to 0.01 and 0.0001, respectively. The total number of training epochs is set to 500. The model is optimized by stochastic gradient descent (SGD), and the "poly" learning rate strategy [52] with a power of 0.9 is applied during the training process. Random horizontal flipping and random Gaussian blurring are applied in the data loading step for data augmentation.

4.3 Comparison with SOTA methods

The proposed method is compared with state-of-the-art methods on two datasets to demonstrate its effectiveness and superiority. All methods are retrained on two datasets to make a fair comparison between different networks.

4.3.1 Efficiency analysis

Figure 1 shows an intuitive comparison of the mIoU and parameters between our network and other methods. Table 1 provides a more detailed comparison, including FLOPs (floating-point operations), memory footprint, running time and parameters of different models. Compared to deep semantic segmentation networks with a large number of parameters, the FLOPs of our method are 5 times and 10 times less than those of HRNet-OCR [37] and CCNet [39], respectively, and the parameter is 3.6 times smaller. Our mIoU on TTCRD is 28% and 24.8% higher than the latter two methods. Furthermore, compared with UNet++ [16], SwiftNet [17], and RefineNetLW [36] with high mIoU performance, the FLOPs and memory footprint of the proposed network are second only to SwiftNet, and the parameters are 6.1 M and 3.5 M higher than those of UNet++ and SwiftNet, respectively. However, the mIoU of our network is improved by approximately 10% and 12.86%. Although the parameters of some networks are particularly low, such as CGNet [44], DFANet [41], and BiSeNetV2 [43], the highest performance of mIoU among them is only 45.20%, which is 23% lower than our method. With fewer parameters and the highest mIoU, the proposed method is very suitable for deployment on the platform of a drone and application in a PTL scene. For verification, we tested our method on the computer NVIDIA Jetson AGX Xavier, which is embedded in our refitted DJI M300 RTK Drone. First, our method is lightweight enough to be deployed on our platform. Second, during the actual flight, its running process is very stable, and the detection speed can reach 12 FPS, which proves that its FLOPs and memory footprint meet the requirements. Most importantly, the segmentation results are accurate, and the contours of objects are clear. It is helpful to judge the defects of tower components and understand the PTL scene during inspections.

4.3.2 Accuracy Analysis

Performance on TTCRD To qualitatively validate the effectiveness of our architecture, we first visualize the segmentation maps generated by our method and four comparative methods in Fig. 6. Notably, the performance of these four methods is the best among the 16 SOTA methods. As shown in the box areas of the first column, because of the high similarities between tower components and background, other methods segment dampers at close range incompletely or even detect nothing, not to speak of locating long-distance vertical insulators. Our method is more accurate and complete for segmenting components at different distances and sizes. In the box areas of the second column, the insulator clip at close range is very clear compared with the background, but only our method segments it with intact contour. In the box areas of the third column, three groups of tower components with the same structure are displayed in different sizes, which reflects the challenge of size difference due to steadily increasing distances. It can be observed that our

Table 1 Efficiency comparison with SOTA methods

Model	Year	FLOPs(G)	Memory(M)	Parameter(M)	Runtime(s)	mIoU(%)
PSPNet [33]	2017	190.07	1482.55	46.7M	0.052	43.63
DeepLabV3 [34]	2017	50.61	678.61	39.0M	0.018	34.17
PSANet [35]	2018	202.97	1578.71	48.3M	0.040	39.14
UNet++ [16]	2018	138.46	1361.00	9.2M	0.015	58.92
RefineNetLW [36]	2018	31.57	717.12	27.3M	0.062	54.12
BiSeNetV1 [40]	2018	14.84	244.97	13.3M	0.019	38.19
DFANet [41]	2019	1.83	221.47	2.2M	0.095	25.03
SwiftNet [17]	2019	12.98	208.22	11.8M	0.015	56.07
ShelfNet [18]	2019	11.71	144.50	14.5M	0.020	52.42
FcHardNet [42]	2019	4.43	143.72	4.1M	0.025	45.64
BiSeNetV2 [43]	2020	12.32	329.85	3.3M	0.021	45.20
HRNet-OCR [37]	2020	162.15	1854.97	70.4M	0.113	40.97
DSRL [38]	2020	121.05	1850.58	60.5M	0.111	49.00
CCNet [39]	2020	310.22	2824.58	71.3M	0.252	44.13
ABCNet [12]	2021	15.61	257.28	13.4M	0.025	31.12
CGNet [44]	2021	3.48	391.7	0.5M	0.034	43.71
Ours	–	27.32	301.19	15.3M	0.066	68.93

method achieved better results in this challenge. Likewise, the problems of overlapping and occlusion are extremely serious in the image of the fourth column. All methods have difficulty segmenting the tower components under serious occlusion, but our method can deal with slight occlusion. The above comparisons prove that our method performs better for the challenges in TTCRD. Additionally, we segment the contour of tower components more accurately, which proves the effectiveness of our method.

The quantitative performance comparison between our method and 16 SOTA methods on TTCRD is shown in Table 2. Our method achieves the best performance across the IoU of all classes and the mIoU. Compared with the most competitive methods, UNet++ [16] and SwiftNet [17], the mIoU of our method is improved by 10% and 12.8%, respectively. From the detailed comparison of class IoU, for large-scale classes such as horizontal insulators and towers, our method has only a small improvement of 2.6% and 2.4% compared to the suboptimal methods. For small-scale classes, our method has great advantages because the network can better deal with the challenge of scale differences. More specifically, our IoU on the vertical insulator is 8.7% higher than UNet++. In particular, our IoU values of the insulator clip and damper are 18.4% and 16.2% greater than those of SwiftNet and UNet++. The above comparison shows the superiority of our method and its great effectiveness for the semantic segmentation of tower components.

Performance on TLRCD Figure 7 shows a visual comparison of TLRCD between our method and the other four competitive methods. In the first row, even if GT annotates the distant houses as a whole piece, our method can accurately judge the sky background between buildings and distinguish the sky and buildings with fine contours. From the second row to the fourth row, although other methods can locate and segment areas of the transmission tower, our method is more accurate in terms of details and integral contours. In addition, the error rate of our method is lower in the segmentation of some distant region classes. In the fifth line, the contour of targets segmented by other methods is too smooth, and the pixel classification at the boundary between the background and the target is obscure. The contour of targets segmented by our method is sharper and more accurate.

As shown in Table 3, the proposed method outperforms the other 16 state-of-the-art methods on TLRCD. In the three classes of sky, river and road, the IoU of our method is slightly inferior to those of ShelfNet [18], DSRL [38] and PSPNet [33] by 1.28%, 0.77% and 1.79%, respectively. This is because our method tends to maintain the contour of the instance at some boundaries between the sky and other instances, which leads to a classification error of the sky. Our method has insufficient learning about the bridges on rivers, which leads to the slightly poor performance of rivers and bridges. However, in the other six classes, our method achieves the best performance. It is outstanding in the IoU of water, which is approximately 8.6% higher than that of the suboptimal method. Overall, the most competitive methods are DSRL and CGNet [44]. The former relies on the superresolution branch to enhance feature representation and improve the prediction of dense labels, which achieves 60.0% mIoU at the parameter number of 60.5M. Under

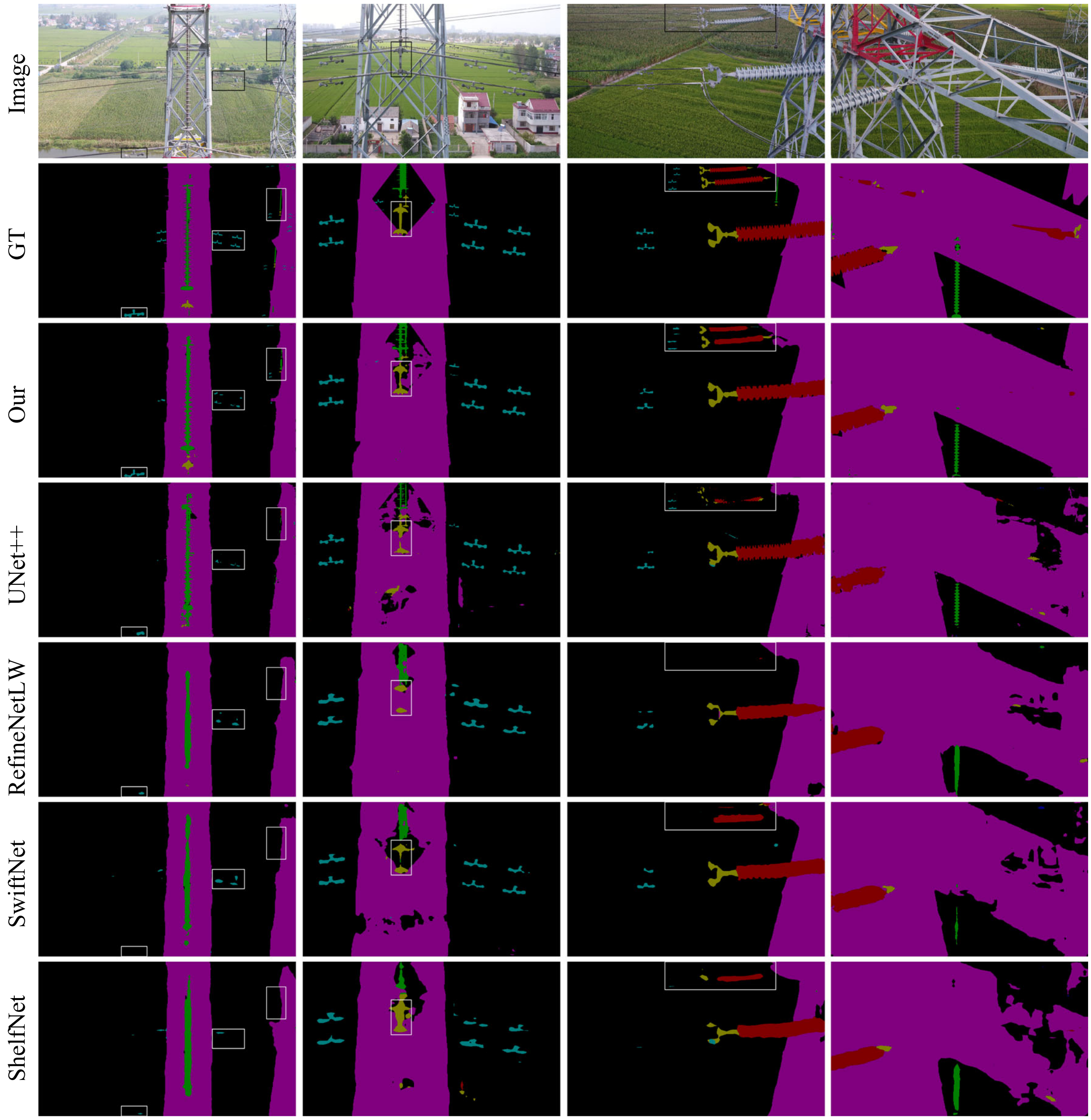


Fig. 6 Comparison of segmentation results on TTCRD

Table 2 Comparison with SOTA methods on TTCRD. The best results are shown in bold type

Model	Class IoU(%)					mIoU (%)	Param
	Horizontal Insulator	Vertical Insulator	Insulator Clip	Damper	Transmission Tower		
PSPNet [33]	68.30	25.50	31.72	7.84	84.80	43.63	46.7M
DeepLabV3 [34]	61.96	11.29	18.33	0.00	79.26	34.17	39.0M
PSANet [35]	65.87	15.20	28.44	3.28	82.89	39.14	48.3M
UNet++ [16]	81.48	40.19	48.05	36.88	88.06	58.92	9.2M
RefineNetLW [36]	74.83	34.34	41.27	33.35	86.79	54.12	27.3M
BiSeNetV1 [40]	67.03	20.52	18.10	3.05	82.23	38.19	13.3M
DFANet [41]	50.41	0.19	0.05	1.9	72.55	25.03	2.2M
SwiftNet [17]	77.63	34.89	48.61	31.14	88.05	56.07	11.8M

Table 2 (continued)

ShelfNet [18]	74.40	32.05	40.67	27.47	87.53	52.42	14.5M
FcHardNet [42]	67.91	23.01	32.48	21.59	83.21	45.64	4.1M
BiSeNetV2 [43]	71.33	26.56	24.38	18.77	84.96	45.20	3.3M
HRNet-OCR [37]	65.20	12.77	27.55	17.21	82.12	40.97	70.4M
DSRL [38]	75.77	30.95	40.64	8.60	89.03	49.00	60.5M
CCNet [39]	70.11	19.55	26.87	19.34	84.77	44.13	71.3M
ABCNet [12]	68.68	0.00	0.00	0.00	86.87	31.12	13.4M
CGNet [44]	68.41	23.20	26.91	14.74	85.26	43.71	0.5M
Ours	**84.15**	**48.90**	**67.00**	**53.13**	**91.44**	**68.93**	15.3M

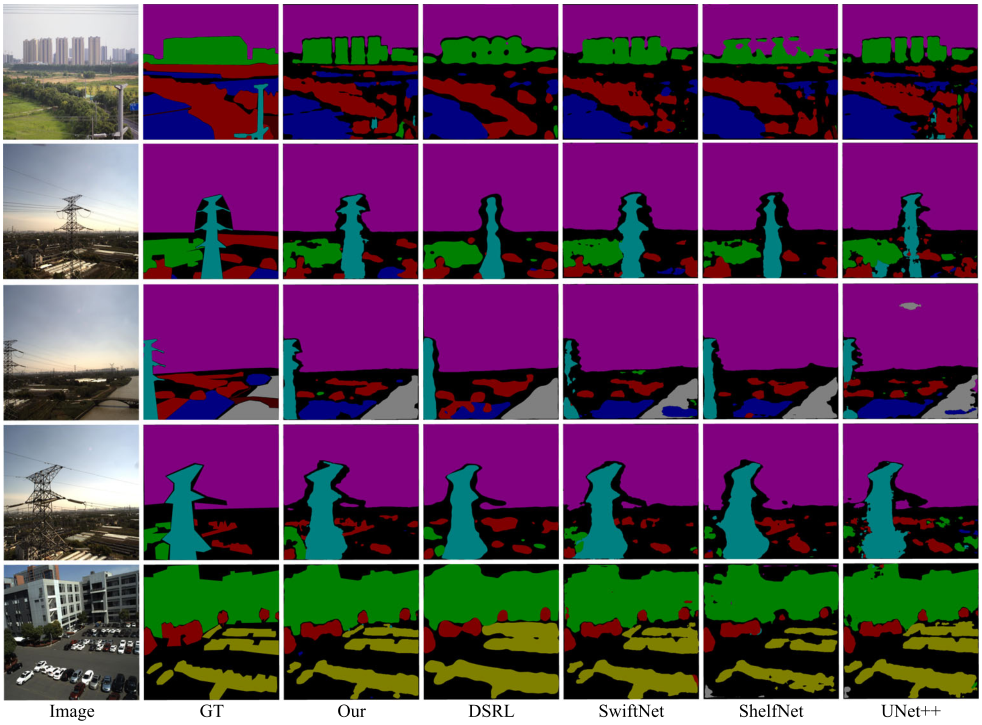


Fig. 7 Comparison of segmentation results on TLRCD

Table 3 Comparison with SOTA methods on TLRCD. The best results are shown in bold type

Model	Class IoU(%)									mIoU (%)	Param
	Tree	Building	Car	Low Veg.	Sky	Trans. Tower	River	Road	Water		
PSPNet [33]	53.69	52.43	63.44	44.52	92.28	47.49	68.69	**43.24**	0.83	51.85	46.7M
DeepLabV3 [34]	43.26	51.28	59.82	31.94	92.46	48.90	70.57	36.32	0.00	48.29	39.0M
PSANet [35]	50.17	49.80	70.24	38.50	92.29	45.63	75.29	35.45	0.00	50.82	48.3M
UNet++ [16]	56.29	60.18	68.75	39.36	91.74	53.29	76.32	39.70	18.94	56.06	9.2M
RefineNetLW [36]	56.86	56.81	71.41	36.13	91.23	46.79	74.52	38.71	7.26	53.30	27.3M
BiSeNetv1 [40]	49.41	53.55	59.47	35.30	93.02	51.23	76.92	39.42	26.36	53.85	13.3M
DFANet [41]	43.67	22.53	9.91	10.59	82.73	19.23	44.13	0.23	0.00	25.89	2.2M
SwiftNet [17]	56.39	64.83	72.99	42.24	91.67	66.42	77.35	35.46	14.26	57.96	11.8M
ShelfNet [18]	51.31	60.80	61.27	42.72	**93.44**	58.06	78.86	40.51	18.36	56.15	14.5M

Table 3 (continued)

Model	Class IoU(%)									mIoU (%)	Param
	Tree	Building	Car	Low Veg.	Sky	Trans. Tower	River	Road	Water		
FcHardNet [42]	42.28	50.21	68.86	31.92	92.66	51.74	76.93	35.24	4.71	50.50	4.1M
BiSeNetv2 [43]	51.85	51.42	64.09	35.14	90.15	41.67	71.24	37.66	1.32	49.40	3.3M
HRNet-OCR [37]	47.03	53.52	68.54	33.08	93.24	47.64	73.55	35.32	5.84	50.86	70.4M
DSRL [38]	59.15	64.27	62.50	43.35	93.23	65.24	**83.42**	41.66	27.20	60.00	60.5M
CCNet [39]	47.70	49.46	68.98	34.47	92.77	36.57	69.78	29.86	1.10	47.85	71.3M
ABCNet [12]	53.85	55.26	28.27	33.64	89.70	55.30	78.02	31.56	0.00	47.29	13.4M
CGNet [44]	53.48	54.72	60.41	44.32	91.70	56.91	74.82	35.71	29.50	55.73	0.5M
Ours	**59.96**	**66.52**	**75.48**	**45.25**	92.16	**66.88**	82.65	41.45	**38.14**	**63.16**	15.3M

the condition that the parameter number is only 0.5 M, CGNet reaches 55.73% mIoU. Our network achieves the best performance with a number of parameters of 15.3M.

4.4 Ablation study

To further verify the contribution of each innovation in our method, we conduct three ablation experiments on TTCRD and TLRCD.

4.4.1 Effectiveness of components in architecture

Table 4 shows the quantitative comparisons of all components. First, when the basic component, ResNet18, is only used, we directly predict and upsample its final high-level feature, resulting in missing details and poor effect. The mIoU on TTCRD and TLRCD are only 42.02% and 45.67%. Although TTFE uses ASPP to learn multiscale information, there is no subsequent processing to make it effective. Once SDs are incorporated into the network, the region features, edge features and hierarchical features are processed simultaneously, which not only restores the spatial details layer by layer but also makes the region and edge features complement each other. Thus, the performance has been greatly improved. The mIoU on TTCRD is improved by 24% and the mIoU on TLRCD is improved by 9%. RA captures the global attention information of region features and uses the information as guidance to help RGEA form the attention information of edge features. Therefore, the addition of both modules improves the performance.

4.4.2 Effectiveness of threshold in RGEA

To further verify the effect of various thresholds in the RGEA module, we set the threshold as 0, 0.2, 0.5 and 0.8 for comparison. As shown in Fig. 8, the blue and yellow marks represent the mIoU of our method on TTCRD and TLRCD, respectively. The performance on both datasets will be improved with the increase in the threshold. When the threshold is raised from 0 to 0.8, the mIoU on both datasets is improved by approximately 2%. This is because the similarity of the screened node pairs is stronger when the threshold increases. Therefore, the relationship between each node and its neighbors in the adjacency matrix is closer, which is very useful for continuous and accurate edge construction.

4.4.3 Effectiveness of CAED

The auxiliary task CAED is the most novel and essential part of the proposed method, and we show its superiority more specifically and intuitively. First, as a comparison, we not only remove the edge supervision in the loss function but also modify the part that involves edge features in all

Table 4 Effect of individual component

					mIoU(%)		
ResNet18	TTFE	SDs	RA	RGEA	TTCRD	TLRCD	Param
√					42.02	45.67	11.2M
√	√				42.62	50.15	12.1M
√	√	√			66.30	59.00	15.2M
√	√	√	√		67.32	60.32	15.3M
√	√	√	√	√	68.93	63.17	15.3M

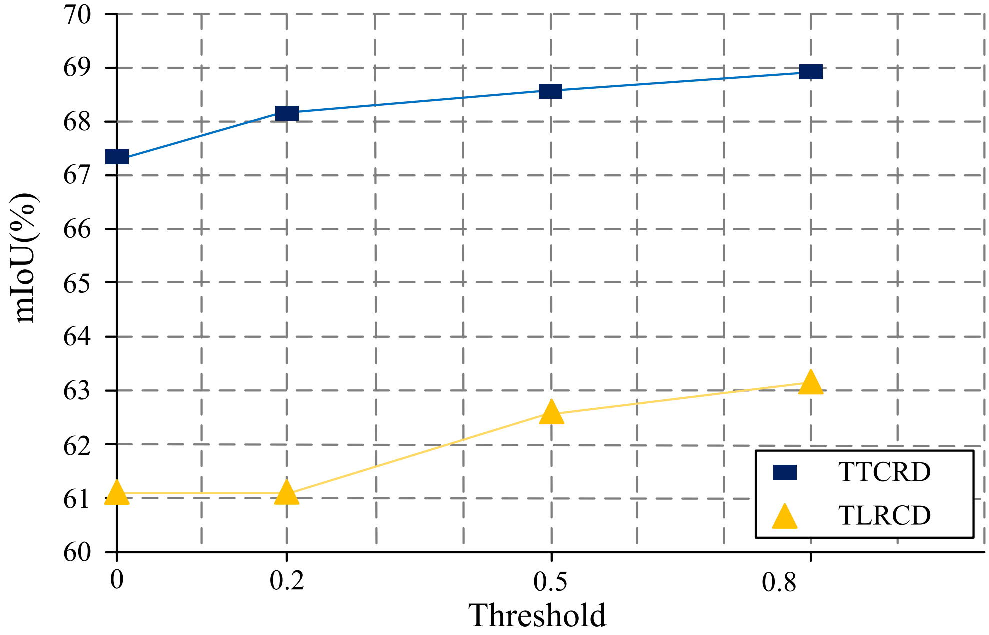


Fig. 8 Performance comparison of two datasets under different thresholds

modules. In this way, only the region features are processed. Then, Grad-CAM [46, 47] is used to visualize the attention information of horizontal insulators in the region feature outputted by the last SD. As shown in Fig. 9, without the assistance of CAED, the feature information on the contour is coarser, and there are features on the insulator clip and transmission tower, which lowers the localization performance. Once CAED is introduced, the localization of the horizontal insulator is more accurate, and its contour is more meticulous. From the enlarged boxed area in the corresponding segmentation results, the segmented contour with CAED is more accurate.

5 Conclusion

In this paper, a class-aware edge-assisted lightweight semantic segmentation network is proposed, which contains a two-task feature extraction module, a hybrid graph learning module and cascaded shared decoders. Distinguishing

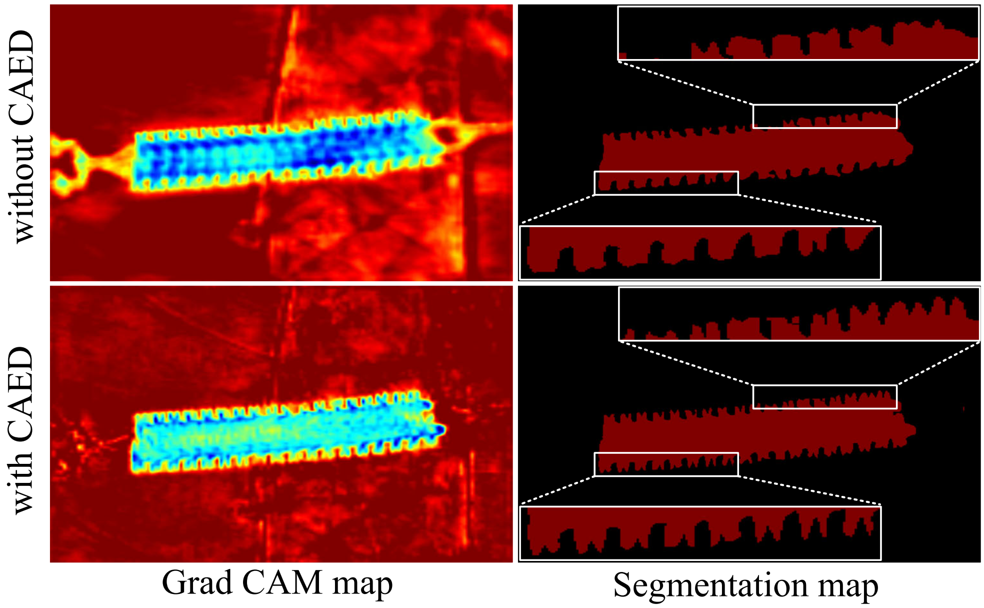


Fig. 9 Visualization of the Grad-CAM map and segmentation map

strategies were applied to two tasks, namely, semantic segmentation and class-aware edge detection. By jointly exploring valuable complementary information on region and edge features, the targets were better located, and their contours were improved. To verify the effectiveness and the practical value of our method in the inspection task of power transmission lines, we constructed two new datasets named TTCRD and TLRCD from drone-captured images. Two challenges were identified in TTCRD and TLRCD. One challenge is the size difference between objects caused by different classes, distances or flight altitudes. The other is overlapping and occlusion between a transmission tower and electrical fittings or between high-density geographical elements. Comprehensive experiments and ablation studies on TTCRD and TLRCD demonstrate the effectiveness and superior performance of our method. We also deploy and test our method on our refitted drone. The experimental results in the actual inspection scene show the value of our method in practical application. In the future, we will investigate the fault detection of tower components and real-time regional analysis in the drone-based autonomous inspection mission of power transmission lines.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (grant number 62001156) and in part by the Key Research and Development Plan of Jiangsu Province (grant numbers BE2019036, BE2020092 and BE2020649)

Data Availability Statement The data are not publicly available due to the confidentiality of the research projects.

Declarations

Conflict of Interests The authors declare that they have no conflicts of interest.

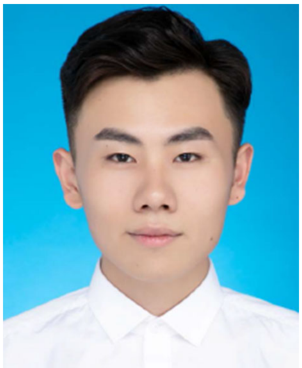
References

1. Alhassan AB, Zhang X, Shen H, Xu H (2020) Power transmission line inspection robots: a review, trends and challenges for future research. Int J Electr Power Energy Sys 118:105862. https://doi.org/10.1016/j.ijepes.2020.105862
2. Yu B, Yang L, Chen F (2018) Semantic segmentation for high spatial resolution remote sensing images based on convolution neural network and pyramid pooling module. IEEE J Sel Top Appl Earth Obs Remote Sens 11(9):3252–3261. https://doi.org/10.1109/JSTARS.2018.2860989
3. Niu W, Ning B, Zhou H (2019) Design of data transmission system of human-autonomous devices for UAV inspection of transmission line status. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-019-01504-x
4. Chen W, Li Y, Zhao Z (2021) InsulatorGAN: A transmission line insulator detection model using multi-granularity conditional generative adversarial nets for UAV inspection. Remote Sens 13(19):3971. https://doi.org/10.3390/rs13193971
5. Wu Y, Zhao G, Hu J, Ouyang Y, Wang SX, He J, Gao F, Wang S (2019) Overhead transmission line parameter reconstruction for UAV inspection based on tunneling magnetoresistive sensors and inverse models. IEEE Trans Power Deliv 34(3):819–827. https://doi.org/10.1109/tpwrd.2019.2891119
6. Alhassan AB, Zhang X, Shen H, Xu H (2020) Power transmission line inspection robots: a review, trends and challenges for future research. Int J Electr Power Energy Syst 118:105862. https://doi.org/10.1016/j.ijepes.2020.105862
7. Lopez RL, Sanchez MJB, Jimenez MP, Arrue BC, Ollero A (2021) Autonomous UAV system for cleaning insulators in power line inspection and maintenance. Sensors 21(24):8488. https://doi.org/10.3390/s21248488
8. Yao H, Qin R, Chen X (2019) Unmanned aerial vehicle for remote sensing applications—a review. Remote Sensing 11(12). https://doi.org/10.3390/rs11121443
9. Xiao R, Wang Y, Tao C (2022) Fine-grained road scene understanding from aerial images based on semisupervised semantic segmentation networks. IEEE Geosci Remote Sens Lett 19:1–5. https://doi.org/10.1109/lgrs.2021.3059708
10. Lyu Y, Vosselman G, Xia G-S, Yang MY (2021) Bidirectional multi-scale attention networks for semantic segmentation of oblique uav imagery. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-2-2021:75–82. https://doi.org/10.5194/isprs-annals-v-2-2021-75-2021
11. Liu S, Cheng J, Liang L, Bai H, Dang W (2021) Light-weight semantic segmentation network for uav remote sensing images. IEEE J Sel Top Appl Earth Obs Remote Sens 14:8287–8296. https://doi.org/10.1109/JSTARS.2021.3104382
12. Li R, Zheng S, Zhang C, Duan C, Wang L, Atkinson PM (2021) Abcnet: Attentive bilateral contextual network for efficient semantic segmentation of fine-resolution remotely sensed imagery. ISPRS J Photogramm Remote Sens 181:84–98. https://doi.org/10.1016/j.isprsjprs.2021.09.005
13. Wu Q, Yang H, Wei M, Remil O, Wang B, Wang J (2018) Automatic 3d reconstruction of electrical substation scene from lidar point cloud. ISPRS J Photogramm Remote Sens 143:57–71. https://doi.org/10.1016/j.isprsjprs.2018.04.024
14. Wang Y, Chen Q, Liu L, Li K (2019) A hierarchical unsupervised method for power line classification from airborne lidar data. Int J Digit Earth 12(12):1406–1422. https://doi.org/10.1080/17538947.2018.1503740
15. Lo S-Y, Hang H-M, Chan S-W, Lin J-J (2019) Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In: Proceedings of the ACM multimedia asia. https://doi.org/10.1145/3338533.3366558
16. Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J (2018) Unet++: A nested u-net architecture for medical image segmentation. In: Deep learning in medical image analysis and multimodal learning for clinical decision support, pp 3–11. https://doi.org/10.1007/978-3-030-00889-5_1
17. Oršic M, Krešo I, Bevandic P, Šegvic S (2019) In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 12599–12608. https://doi.org/10.1109/CVPR.2019.01289
18. Zhuang J, Yang J, Gu L, Dvornek N (2019) Shelfnet for fast semantic segmentation. In: 2019 IEEE/CVF International conference on computer vision workshop (ICCVW), pp 847–856. https://doi.org/10.1109/ICCVW.2019.00113
19. Han H-Y, Chen Y-C, Hsiao P-Y, Fu L-C (2021) Using channel-wise attention for deep cnn based real-time semantic segmentation with class-aware edge information. IEEE Trans Intell Transp Syst 22(2):1041–1051. https://doi.org/10.1109/TITS.2019.2962094
20. Chen Y, Dapogny A, Cord M (2020) SEMEDA: Enhancing segmentation precision with semantic edge aware loss. Pattern Recogn 108:107557. https://doi.org/10.1016/j.patcog.2020.107557

21. Yu Z, Feng C, Liu M-Y, Ramalingam S (2017) Casenet: Deep category-aware semantic edge detection. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 5964–5973. https://doi.org/10.1109/cvpr.2017.191
22. Zhao W, Dong Q, Zuo Z (2022) A method combining line detection and semantic segmentation for power line extraction from unmanned aerial vehicle images. 6 14:1367. https://doi.org/10.3390/rs14061367
23. Meng L, Peng Z, Zhou J, Zhang J, Lu Z, Baumann A, Du Y (2020) Real-time detection of ground objects based on unmanned aerial vehicle remote sensing with deep learning: Application in excavator detection for pipeline safety. Remote Sensing 12(1). https://doi.org/10.3390/rs12010182
24. Siddiqui ZA, Park U (2020) A drone based transmission line components inspection system with deep learning technique. Energies 13(13). https://doi.org/10.3390/en13133348
25. Jiao R, Liu Y, He H, Xuehai M, Li Z (2021) A deep learning model for small-size defective components detection in power transmission tower. IEEE Transactions on Power Delivery, p 1–1. https://doi.org/10.1109/TPWRD.2021.3112285
26. Liu J, Jia R, Li W, Ma F, Abdullah HM, Ma H, Mohamed MA (2020) High precision detection algorithm based on improved retinanet for defect recognition of transmission lines. Energy Reports 6:2430–2440. https://doi.org/10.1016/j.egyr.2020.09.002
27. Li H, Yang Z, Han J, Lai S, Zhang Q, Zhang C, Fang Q, Hu G (2020) Tl-net: A novel network for transmission line scenes classification. Energies 13(15). https://doi.org/10.3390/en13153910
28. Ma Y, Li Q, Chu L, Zhou Y, Xu C (2021) Real-time detection and spatial localization of insulators for uav inspection based on binocular stereo vision. Remote Sensing 13(2). https://doi.org/10.3390/rs13020230
29. Tao X, Zhang D, Wang Z, Liu X, Zhang H, Xu D (2020) Detection of power line insulator defects using aerial images analyzed with convolutional neural networks. IEEE Trans Syst Man Cybern Syst 50(4):1486–1498. https://doi.org/10.1109/TSMC.2018.2871750
30. Zhou B, Zhao H, Puig X, Xiao T, Fidler S, Barriuso A, Torralba A (2019) Semantic understanding of scenes through the ade20k dataset. Int J Comput Vis 127(3):302–321. https://doi.org/10.1007/s11263-018-1140-0
31. Wang X, Ma H, You S (2020) Deep clustering for weakly-supervised semantic segmentation in autonomous driving scenes. Neurocomputing 381:20–28. https://doi.org/10.1016/j.neucom.2019.11.019
32. 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, pp 3431–3440. https://doi.org/10.1109/cvpr.2015.7298965
33. 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 (CVPR). https://doi.org/10.1109/cvpr.2017.660
34. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the european conference on computer vision (ECCV), pp 833–851. https://doi.org/10.1007/978-3-030-01234-2_49
35. Zhao H, Zhang Y, Liu S, Shi J, Loy CC, Lin D, Jia J (2018) Psanet: Point-wise spatial attention network for scene parsing. In: Proceedings of the european conference on computer vision (ECCV), pp 270–286. https://doi.org/10.1007/978-3-030-01240-3_17
36. Nekrasov V, Shen C, Reid I (2018) Light-weight refinenet for real-time semantic segmentation. In: 2018 British machine vision conference (BMVC)
37. Yuan Y, Chen X, Wang J (2020) Object-contextual representations for semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 173–190. https://doi.org/10.1007/978-3-030-58539-6_11
38. Wang L, Li D, Zhu Y, Tian L, Shan Y (2020) Dual super-resolution learning for semantic segmentation. In: 2020 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 3773–3782. https://doi.org/10.1109/CVPR42600.2020.00383
39. Huang Z, Wang X, Wei Y, Huang L, Shi H, Liu W, Huang TS (2020) Ccnet: Criss-cross attention for semantic segmentation. IEEE Trans Pattern Anal Mach Intell, 1–1. https://doi.org/10.1109/TPAMI.2020.3007032
40. 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), pp 325–341. https://doi.org/10.1007/978-3-030-01261-8_20
41. Li H, Xiong P, Fan H, Sun J (2019) Dfanet: Deep feature aggregation for real-time semantic segmentation. In: 2019 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 9514–9523. https://doi.org/10.1109/CVPR.2019.00975
42. Chao P, Kao C-Y, Ruan Y, Huang C-H, Lin Y-L (2019) Hardnet: A low memory traffic network. In: 2019 IEEE/CVF International conference on computer vision (ICCV), pp 3551–3560. https://doi.org/10.1109/ICCV.2019.00365
43. Yu C, Gao C, Wang J, Yu G, Shen C, Sang N (2021) Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. Int J Comput Vis 129(11):3051–3068. https://doi.org/10.1007/s11263-021-01515-2
44. Wu T, Tang S, Zhang R, Cao J, Zhang Y (2021) Cgnet: A light-weight context guided network for semantic segmentation. IEEE Trans Image Process 30:1169–1179. https://doi.org/10.1109/TIP.2020.3042065
45. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 7794–7803. https://doi.org/10.1109/cvpr.2018.00813
46. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the ieee international conference on computer vision, pp 618–626. https://doi.org/10.1109/iccv.2017.74
47. Chattopadhay A, Sarkar A, Howlader P, Balasubramanian VN (2018) Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). https://doi.org/10.1109/wacv.2018.00097
48. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International conference on learning representations (ICLR)
49. Li K, Ye W (2022) Semi-supervised node classification via graph learning convolutional neural network. Applied Intelligence. https://doi.org/10.1007/s10489-022-03233-9
50. Jamin A, Humeau-Heurtier A (2019) (Multiscale) cross-entropy methods: a review. Entropy 22(1):45. https://doi.org/10.3390/e22010045
51. Russell BC, Torralba A, Murphy KP, Freeman WT (2008) LabelMe: a database and web-based tool for image annotation. Int J Comput Vis 77(1–3):157–173. https://doi.org/10.1007/s11263-007-0090-8
52. He J-Y, Liang S-H, Wu X, Zhao B, Zhang L (2021) Mgseg: Multiple granularity-based real-time semantic segmentation network. IEEE Trans Image Process 30:7200–7214. https://doi.org/10.1109/tip.2021.3102509

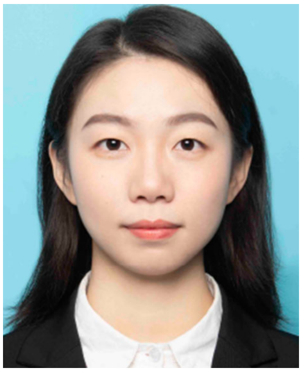
Qingkai Zhou received the B.Sc. degree in Communication Engineering from Hohai University, Changzhou, China, in 2020, where he is currently pursuing the M.Sc. degree in Communication and Information System. His current research interests include computer vision and deep learning.

Qingwu Li received the B.Sc. degree in Radio Engineering from Zhengzhou University, Zhengzhou, China, in 1985, the M.Sc. degree in Signal, Circuit and System from Xidian University, Xi'an, China, in 1990, and the Ph.D. degree in Water Information Science from Hohai University, Nanjing, China, in 2010. He is currently a professor and a doctoral supervisor with the College of Internet of Things Engineering, Hohai University, Changzhou, China. He also serves as the director of the Changzhou Key Laboratory of Sensor Networks and Environmental Sensing, director of China Geophysical Society and director of Jiangsu Artificial Intelligence Society. His current research interests include visual perception, artificial intelligence, underwater environment imaging detection, sensor networks and their applications.



Chang Xu received the B.Sc. degree in Computer Science and Technology from Hohai University, Changzhou, China, in 2019, where he is currently pursuing the Ph.D. degree in Internet of Things Technology and Application. His current research interests include computer vision, bionic vision, and deep learning.

Qiuyu Lu received the B.Sc. degree in Communication Engineering from Hohai University, Changzhou, China in 2021, where he is pursuing the M.Sc. degree in Communication and Information System. His current research interests include computer vision.



Yaqin Zhou received the B.Sc. degree in Communication Engineering from Hohai University, Changzhou, China, in 2017, and the Ph.D. degree in Internet of Things Technology and Application from Hohai University, Nanjing, China, in 2021. She is currently a postdoctoral of Computer and Information Technology in Hohai University, Changzhou, China. Her current research interests include computer vision and stereoscopic visual perception.