Incorporating Superpixel Context for Extracting Building From High-Resolution Remote Sensing Imagery

Fang Fang, *Member, IEEE*, Kang Zheng, Shengwen Li, *Member, IEEE*, Rui Xu, Qingyi Hao, Yuting Feng, and Shunping Zhou, *Member, IEEE*

***Abstract*—Extracting building from high-resolution (HR) remote sensing imagery (RSI) serves a variety of areas, such as smart city, environment management, and emergency disaster services. Previous building extraction methods primarily focus on pixel-level and superpixel-level features, which do not fully utilize the superpixel-level spatial context, leaving room for performance improvement. To bridge the gap, this study incorporates spatial context of both pixels and superpixels for building extraction of HR RSI. Specifically, the proposed method develops a trainable superpixel segmentation module to segment HR RSI into superpixels by fusing pixel features and pixel-level context. And a superpixel-level context aggregation module is devised to incorporate the multiple-scale spatial context of superpixels to extract buildings. Experiments on public challenging datasets show that our method is superior to the state-of-the-art baselines in accuracy, with better building boundaries and higher integrity. This study explores a new approach for HR RSI building extraction by introducing spatial context of superpixels, and a methodological reference for the HR RSI interpretation tasks.**

***Index Terms*—Building extraction, high-resolution (HR) remote sensing imagery (RSI), spatial context, superpixels.**

I. Introduction

As a fundamental task of remote sensing imagery (RSI) processing, building extraction serves for numerous applications, including urban planning [1], [2], [3], urban environmental change [4], geographic data updating [5], [6], and disaster emergency response [7], [8]. High-resolution (HR) RSIs are emerging as sensor technology continues to advance, providing new opportunities for extracting fine-grained buildings. Usually, building extraction can be performed by the binary semantic segmentation methods of computer vision, which aims to label pixels to building or nonbuilding. However, buildings in HR RSI are highly complex, as shown by varying building shapes, obstacle occlusions, and low contrast with the surrounding area. Extracting buildings from HR RSI are extremely challenging and highly valuable [9].

Recently, deep learning (DL) has emerged as a promising approach for extracting buildings from RSIs [10]. This progress is driven by the advantages of DL, including automatic feature learning, reduced human intervention. For example, fully convolutional networks (FCNs) [11] achieved great segmentation of images by capturing and predefining the spatial information of HR RSI and are widely employed for building extraction tasks. Maggiori et al. [12] developed a two-scale FCN, to improve the recognition and correct localization. Xu et al. [13] used an FCN-based deep convolutional neural network (DCNN) and guided filters to refine the building extraction from HR RSI. Wu et al. [14] developed an FCN-based model to extract buildings from aerial images with multiple constraint strategies. Zhang and Wang [15] designed a building extraction network by combining dilated convolution and dense connectivity. These models are pixel-based, in which the buildings are mainly extracted with pixel features and context, suffering from some limitations, such as salt-and-pepper noise [16], holes in the middle of connected changed components, and jagged boundaries.

Meanwhile, superpixel-based methods have shown its advantages in building extraction tasks, being the mainstream application of object-oriented approach in the task, which groups contiguous image pixels from homogeneous regions [17]. Superpixel-based methods commonly employ an iterative clustering technique to group similar pixels according to colors, textures, and brightness into superpixel clusters. Then, the obtained superpixels are utilized as basic units for various tasks. For example, a superpixel-aided CNN framework [18] is proposed to extract objects by utilizing scale-invariant features. A CNN-based superpixel model is developed to detect earthquake-induced damaged buildings [19]. Benchabana et al. [20] proposed an algorithm for building detection via deep feature extraction and adaptively superpixel classification. However, these superpixel-based methods do not fully utilize the superpixel-level spatial contextual semantics, which leads to imprecise building recognition and indistinct building boundary segmentation.

This study incorporates spatial context of both pixels and superpixels to improve building extraction, and proposes a spatial

Manuscript received 20 August 2023; revised 4 November 2023; accepted 23 November 2023. Date of publication 28 November 2023; date of current version 8 December 2023. This work was supported in part by the Open Research Project of The Hubei Key Laboratory of Intelligent Geo-Information Processing under Grant KLIGIP-2023-B11 and Grant KLIGIP-2023-B09, and in part by the National Natural Science Foundation of China under Grant 42071382. *(Corresponding author: Shengwen Li.)*

Fang Fang is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China, and also with the Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430078, China (e-mail: fangfang@cug.edu.cn).

Kang Zheng, Shengwen Li, Rui Xu, Qingyi Hao, Yuting Feng, and Shunping Zhou are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: zhengkang@cug.edu.cn; swli@cug.edu.cn; uiirux@cug.edu.cn; hqy@cug.edu.cn; fengyuting99@cug.edu.cn; zhoushunping@mapgis.com).

Digital Object Identifier 10.1109/JSTARS.2023.3337140

context-aware building extraction model that fuses superpixel features and superpixel-level spatial context. Specifically, we propose a spatial context-aware building extraction model that fuses superpixel features and superpixel-level spatial context. The proposed method designs a trainable superpixel segmentation (TSS) module that fuses the pixel features, pixel-level context, and the superpixel features to generate accurate superpixel boundaries. Then, a superpixel graph is constructed to capture the spatial context of the superpixels. Finally, a superpixel-level contextual aggregation component is developed to aggregate spatial contextual information at different scales to improve building extraction. To the best of our knowledge, this work represents the first attempt to fuse superpixels and superpixel-level spatial context for building fuses the pixel features and pixel-level context to extraction from HR RSIs. The primary contributions of this study are summarized as follows.

1) We highlight the effect of spatial context of superpixel in extracting ground objects from HR RSI, and propose to introduce superpixel-level spatial context to improve the building extraction of HR RSI.
2) A superpixel-level context aggregation (SLCA) module is developed, which provide an approach to adaptively integrate local and long-range superpixel features.
3) A spatial context-aware graph convolution network (GCN) is implemented. Extensive experiments on three public datasets show that the proposed method outperforms baselines, with notable improvements in boundary accuracy and preservation of fine-grained building details.

II. Related Work

A. Traditional Building Extraction Method

Traditional building extraction algorithms can be divided into traditional image processing-based and machine learning-based methods. The former can extract building information from RSI are designed based on handcrafted features [21], [22]. These methods typically designed handcrafted features to quantitatively describe the salient building features, including the building shape, size, color, texture, shadow, and roof material [23]. Wang et al. [24] employed distinctive image primitives to extracted buildings. Cui et al. [25] proposed a graph-based approach based on graph-based shape representation that utilizes a simple yet robust process involving Hough transformation and cycle detection to extract complex building descriptions from the HR RSI. Huang and Zhang [26] alleviated commission and omission of buildings by designing a morphological index of building and shadow for building extraction.

Many machine learning models have been used for building extraction. For instance, Karsli et al. [27] utilized the SVM model to extract buildings by leveraging spatial, spectral, and textural features from HR multispectral aerial images and LiDAR data. It effectively detected building boundaries by capitalizing on the complementary advantages of LiDAR data and HR optical imagery, resulting in reliable and accurate results. Li et al. [28] introduced conditional random field into rooftop segmentation. This method incorporated pixel-level color features, segment-level region consistency and shape features, achieving high accuracy in rooftop segmentation.

In practice, both traditional image processing-based and machine learning-based methods often rely on prior knowledge and parameter initialization, which can be time-consuming and require significant human resources [29]. In addition, these approaches suffer from some limitations, including ineffective use of many cues hidden in images [30], resulting in poor accuracy [31].

B. DL-Based Extraction Method

DL-based methods have been extensively promoted to interpret aerial and satellite RSIs, which typically exploit CNNs to capture deep pixel features. These methods regard building extraction from RSIs as a semantic segmentation task by labeling RSI pixels as a building or nonbuilding class [12], [31], [32], [33], [34], [35]. For example, Maggiori et al. [12] developed a pixel-wise classification framework of satellite imagery, in which CNNs are employed to label RSI pixels. Recent work has followed the idea of FCN [11], in which deconvolution layers are introduced to predict the classes of individual RSI pixels. Zuo and Juntao [36] used a hierarchical FCN to integrate information from multiscale receptive fields to improve building extraction. This novel architecture helps in addressing the challenges of complex scenarios, such as appearance variations, varying building sizes, and occlusions, resulting in a higher overall accuracy (OA). Wu et al. [37] introduced a comprehensive framework based on multiple constraints to extract buildings from aerial images. In addition, some pixel-level building extraction methods are evolved with advanced semantic segmentation methods, including SegNet [38], U-Net [39], and PSPNet [40]. For example, RFA-UNet [41] adopts reweighted attention to extract buildings from aerial imagery. SegNet was combined with multitask learning to improve building boundaries [42]. Yuan et al. [43] developed a PSPNet-based network for extracting buildings from RSIs by introducing a novel shift pooling technique. Although these pixel-level building extraction based on DL has gained promising results. In practice, these methods overlook the building-level semantics, resulting in some limitations of the model, such as salt-and-pepper noise [16], holes in the middle of connected changed components, and jagged boundaries.

Object-oriented methods have been widely used in RSI processing. Notably, superpixel-based methods, which treat superpixels as individual objects, achieve higher accuracy. For example, Li et al. [44] proposed an improved superpixel algorithm for RSI segmentation, which overcomes the limitation of the input feature dimension of pixels and improves performance by using more features. Liang et al. [45] introduced a differentiable superpixel branch to take advantage of the superpixel segmentation algorithm to accurately identify object. However, the object-oriented methods do not take full use the spatial relationships between superpixels.

III. Methodology

This study presents a spatial context-aware building extraction model that incorporates the trainable superpixel and the spatial context of superpixels to improve the extraction performance of HR RSI. As shown in Fig. 1, the approach comprises four

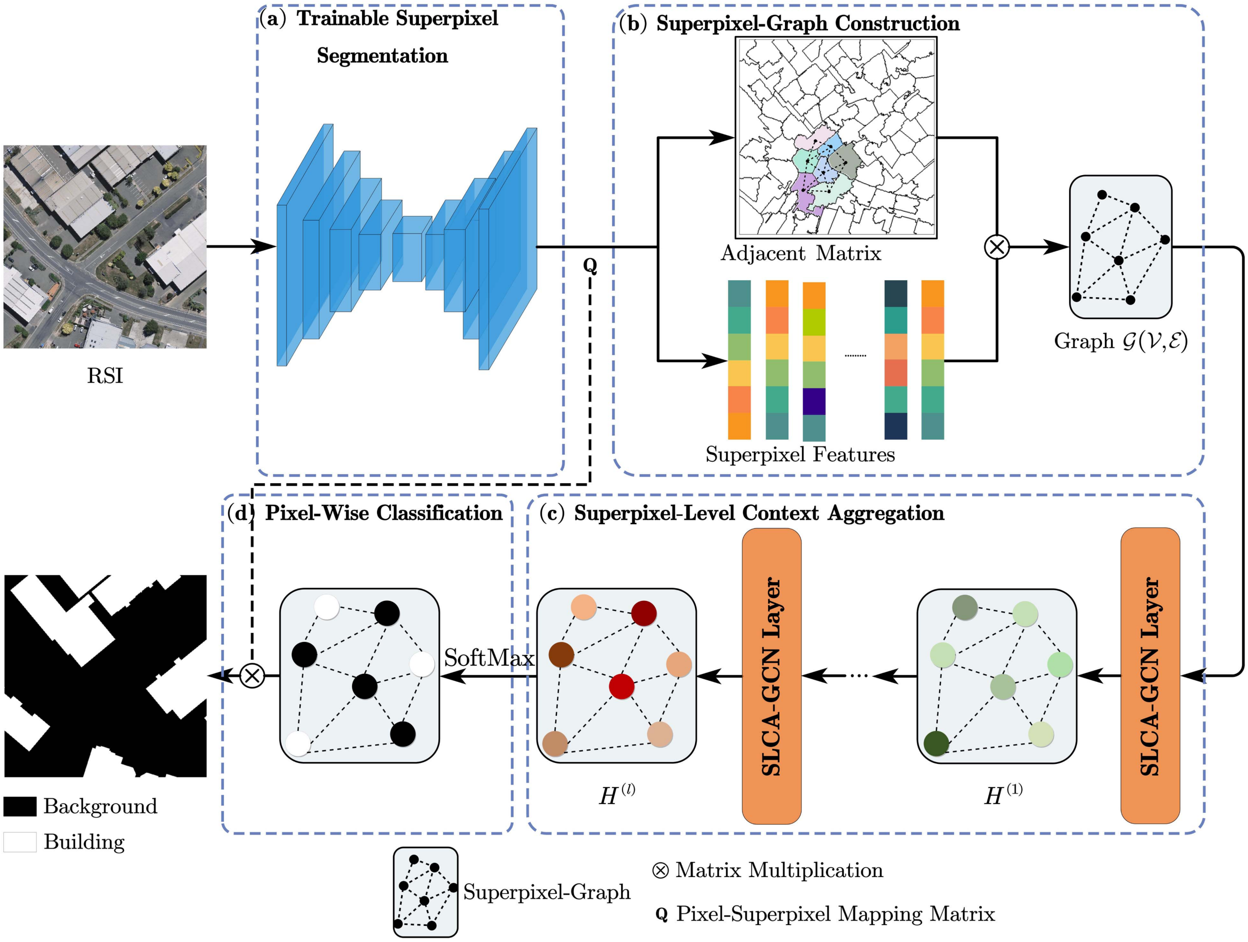

(a) Trainable Superpixel Segmentation
(b) Superpixel-Graph Construction
RSI
Q
Adjacent Matrix
Superpixel Features
Graph $\mathcal{G}(\mathcal{V},\mathcal{E})$
(d) Pixel-Wise Classification
(c) Superpixel-Level Context Aggregation
SoftMax
SLCA-GCN Layer
SLCA-GCN Layer
$H^{(l)}$
$H^{(1)}$
Background
Building
Superpixel-Graph
⊗ Matrix Multiplication
Q Pixel-Superpixel Mapping Matrix

Fig. 1. Framework of the proposed method. (a) TSS. (b) Superpixel-based graph construction. (c) SLCA. (d) Pixel-wise classification.

components: TSS, superpixel graph construction, SLCA, and pixel-wise classification. The TSS module employs a superpixel segmentation network to extract features and build a pixel-superpixel mapping from the input image. Then, a superpixel graph is constructed using an adjacency matrix and superpixel features. Subsequently, the SLCA module aggregates the features of the local and long-range superpixel node. Finally, the superpixel nodes are classified by the pixel-wise prediction module. The four components are elaborated in the following sections.

A. Trainable Superpixel Segmentation

This study designs a superpixel segmentation module, TSS, to generate the superpixels of buildings. Specifically, this module aims to segment the original image into superpixels, and map the pixel and positional features of the original image to its corresponding superpixels, which will output a pixel-superpixel mapping matrix, denoted as Q. TSS consists of an encoder and decoder networks. The encoder network encodes the images to high-level feature maps through multiple convolution operations. By sampling the feature maps, the decoder produces the pixel-to-superpixel mapping matrix Q. During the upsampling, the skip-connection mechanism is adopted to effectively reconstruct fine-grained detail. The decoder fuses feature map of early layers in the encoder which are rich in spatial detail, facilitating the optimization of superpixel edges. The networks of the module are trained with a reconstruction loss from the pixel-superpixel association matrix, and subsequently optimized via backpropagation and gradient descent. The TSS training process is illustrated in Fig. 2.

In the figure, the module feeds the feature map $\mathrm{f}(\mathbf{v}) \in \mathbb{R}^{H\times W\times(3+N)}$ composed of the HR RSIs and labeled images, where W and H are the width and height of the HR RSIs, respectively. N denotes the total number of classes. The initial superpixels are squares with side length r. That is, an RSI will generated around $\frac{H\times W}{r^2}$ superpixels. Given pixel-to-superpixel mapping matrix Q, a superpixel, s, can be presented as $c_s = (u_s, l_s)$, where u_s denotes attribute vector, l_s denotes location vector. u_s and l_s are defined as follows:

$$u_s = \frac{\sum_{\mathrm{v}:\mathrm{s}\in\mathcal{N}_{\mathrm{v}}} \mathrm{f}\left(\mathrm{v}\right)\cdot e_s\left(\mathrm{v}\right)}{\sum_{\mathrm{v}:\mathrm{s}\in\mathcal{N}_{\mathrm{v}}} e_s\left(\mathrm{v}\right)} \tag{1}$$

$$l_{\mathrm{s}} = \frac{\sum_{\mathrm{v}:\mathrm{s}\in\mathcal{N}_{\mathrm{v}}} \mathrm{v}\cdot e_s\left(\mathrm{v}\right)}{\sum_{\mathrm{v}:\mathrm{s}\in\mathcal{N}_{\mathrm{v}}} e_s\left(\mathrm{v}\right)} \tag{2}$$

where $\mathrm{v} = [x, y]^T$ represents the image coordinates of the pixel, and $e_s(\mathrm{v})$ signifies the probability that v is assigned to superpixel

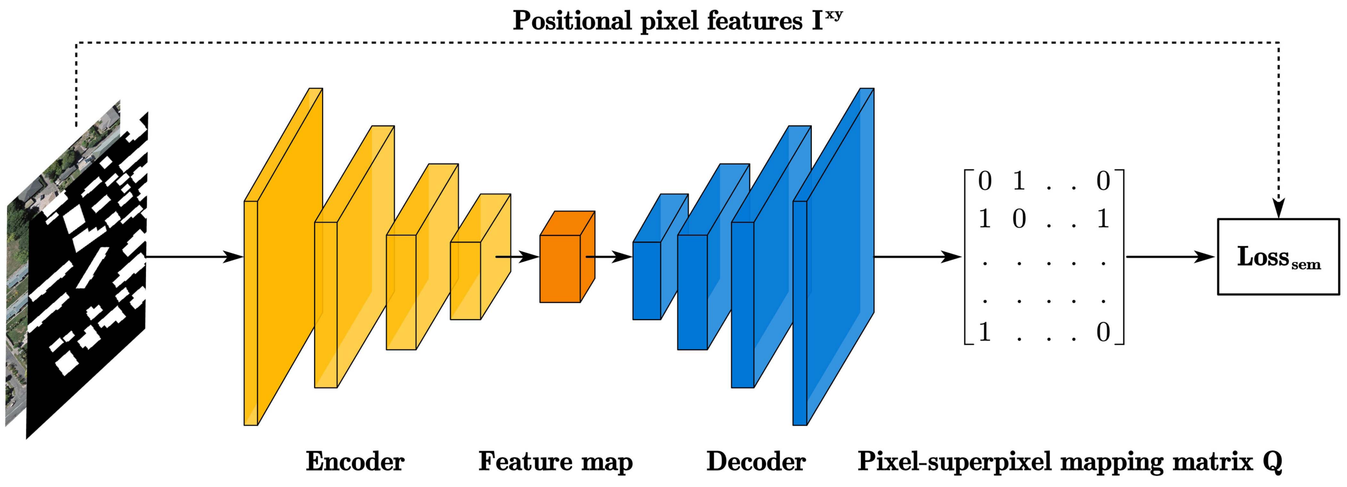

Positional pixel features $\mathbf{I}^{xy}$
Encoder
Feature map
Decoder
Pixel-superpixel mapping matrix Q
$\begin{bmatrix} 0 & 1 & . & . & 0 \\ 1 & 0 & . & . & 1 \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & . & . & . & 0 \end{bmatrix}$
Loss_{sem}

Fig. 2. Illustration of the TSS module.

s, $\mathcal{N}_{\text{v}}$ denotes the set of neighboring superpixels around pixel v. As shown in (1), this study will sum up all the pixels that are possibly clustered to the superpixel. And, the location and property of the pixel v are assigned by

$$\text{f}'(\text{v}) = \sum_{\text{s} \in \mathcal{N}_{\text{v}}} u_s \cdot e_s(\text{v}) \tag{3}$$

$$\text{v}' = \sum_{\text{s} \in \mathcal{N}_{\text{v}}} l_{\text{s}} \cdot e_s(\text{v}) \tag{4}$$

Equation (5) illustrates that the reconstruction loss, L_{sem}, of this module, which is composed of two parts. The first part to group pixels sharing similar semantic information. Meanwhile, the second part encourages that the superpixels remain spatially compact

$$L_{\text{sem}}(\mathbf{Q}) = \sum_{\text{v}} \text{E}(\text{f}(\text{v}), \text{f}'(\text{v})) + \frac{\text{m}}{S} \| \text{ v} - \text{v}' \|_2 \tag{5}$$

where $\text{E}(\cdot, \cdot)$ presents the cross-entropy (CE) function, $\boldsymbol{S}$ is the sampling interval of superpixels, and m is a hyperparameter.

Finally, the trained TSS network is used to infer the mapping between pixels and superpixels to extract superpixels from HR RSIs.

B. Superpixel Graph Construction

In this module, a superpixel graph is employed to characterize the spatial context of superpixels. The edges are presented by the adjacency matrix $A \in \mathbb{R}^{(m \times m)}$, where A_{ij} is the edge weight from node *i* to node *j*. The features of the superpixel graph are initialized with the superpixel representations, $S \in \mathbb{R}^{(m \times d)}$, where *m* is the number of superpixels and *d* denotes the dimensional size of the superpixel feature

$$\mathbf{S} = \mathbf{Q}^{\mathbf{T}}\mathbf{P}, \mathbf{P} \in \mathbb{R}^{N \times 3} \tag{6}$$

where $\mathbf{Q}$ and $\mathbf{P}$ denote pixel-superpixel mapping matrix and pixel feature matrix, respectively. As illustrated in Table I, the superpixel features consist of semantic and position features. The semantic features present the meta information from the pixels, including mean RGB, standard deviation of RGB values, and number of pixels. The position features are the relative location information in the image, including mean position and standard deviation of position values. Specifically, each superpixel's feature is an 11-dimensional vector that is concatenated by the individual vectors listed in Table I.

TABLE I
LIST OF SUPERPIXEL FEATURES

Category	Feature Name	Size	Description
Semantic feature	Mean RGB values	3	$\frac{\sum_{i=0}^{N}(r,g,b)}{N}$
	Standard deviation of RGB values	3	$\sqrt{\frac{\sum\left(r-\bar{r},g-\bar{g},b-\bar{b}\right)^2}{N}}$
	Number of pixels	1	N
Position feature	Mean Position values	2	$\frac{\sum_{i=0}^{N}(x_i,y_i)}{N}$
	Standard deviation of position values	2	$\sqrt{\frac{\sum(x_i-\bar{x}_i,y_i-\bar{y}_i)^2}{N}}$

Subsequently, the matrix $\mathbf{A}$ is used to presented the superpixel graph, where $\mathbf{A}_{\text{ij}}$ is assigned a value of 1 if the ith node shares an adjacency with jth node, and 0 otherwise. In practice, superpixels sharing common edges are deemed being adjacent. The values of A can be formulated as follows:

$$\mathbf{A}_{\text{ij}} = \begin{cases} 1, & \text{if } \text{S}_{\text{i}} \text{ and } \text{S}_{\text{j}} \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases}. \tag{7}$$

Fig. 3 shows the construction process of the graph.

C. Superpixel-Level Context Aggregation

The SLCA module is devised to update the feature representations of superpixels and characterize the spatial contexts between the superpixel nodes. In practice, the GCN models typically utilize a single convolutional filter of fixed size in each convolution layer, which may result in excessive dependence of the node features on the certain scale context information and neglect of information of other scales [46]. On the basis of GoogleNet [47], the module employs a set of convolution kernels of continuous size in all GCN layers. In this way, the nodes of

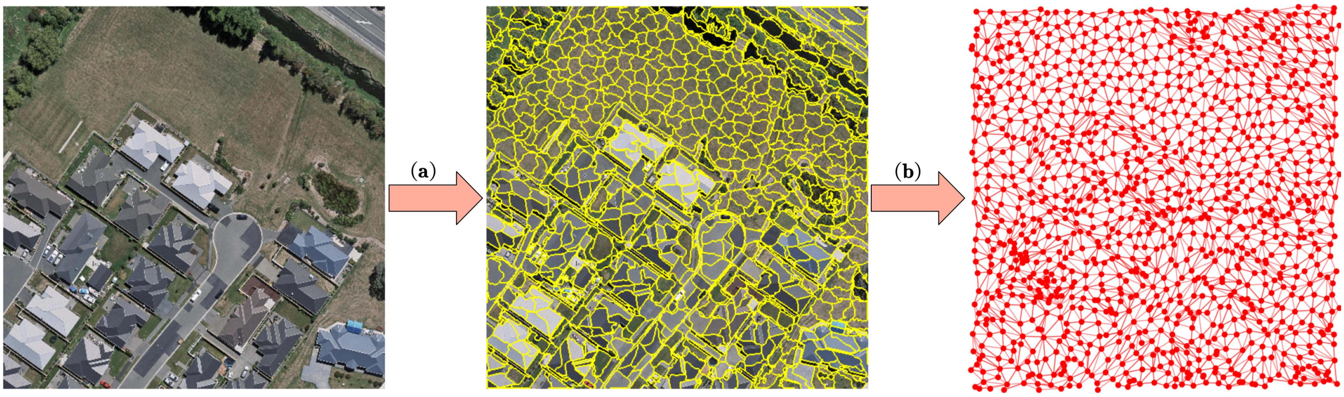


Fig. 3. Illustration of the construction process of the superpixel graph. (a) Superpixel segmentation. (b) Adjacent relationship construction. The red dot in the third image is the center point of the superpixel in the second image.

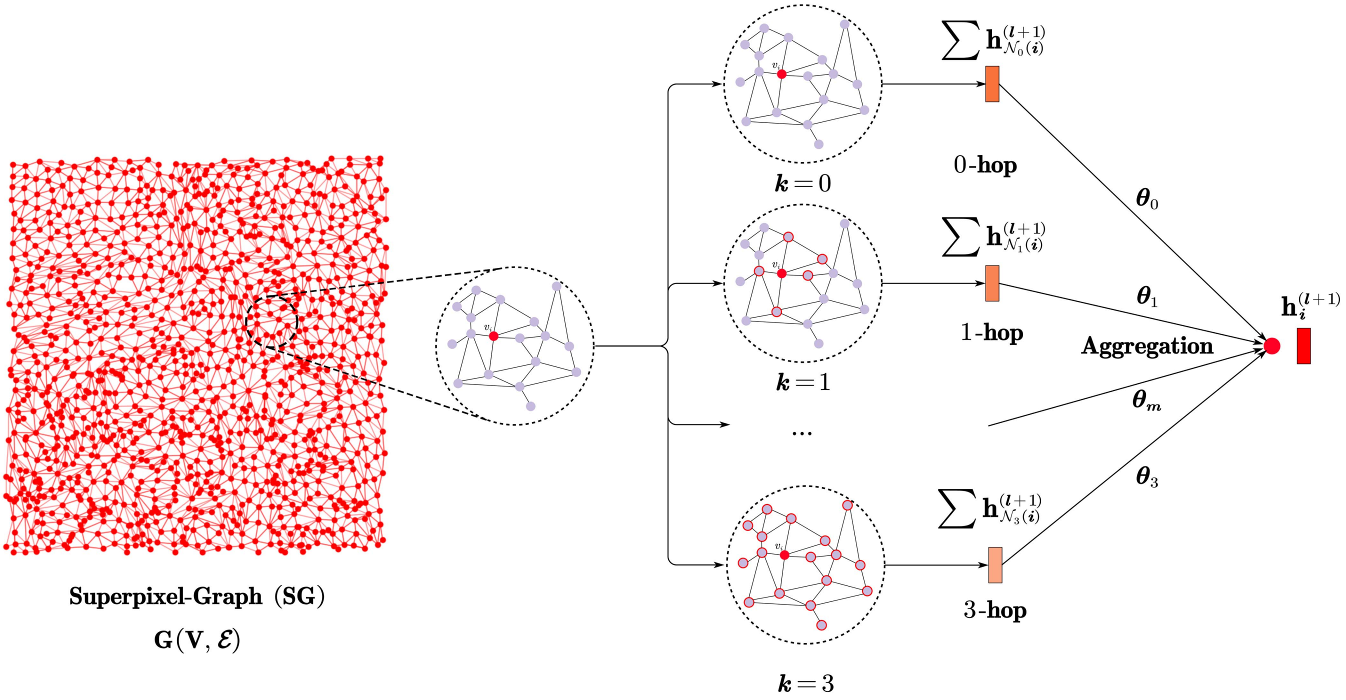

$\sum \mathbf{h}_{\mathcal{N}_0(i)}^{(l+1)}$
$k=0$
0-hop
θ_0
$\sum \mathbf{h}_{\mathcal{N}_1(i)}^{(l+1)}$
$k=1$
1-hop
θ_1
Aggregation
$\mathbf{h}_i^{(l+1)}$
...
θ_m
θ_3
$\sum \mathbf{h}_{\mathcal{N}_3(i)}^{(l+1)}$
3-hop
$k=3$
v_i
Superpixel-Graph (SG)
$\mathbf{G}(\mathbf{V}, \mathcal{E})$

Fig. 4. Illustration of the SLCA process. SLCA first aggregates features from K-hop neighbors, then fuses the different scale context vectors.

the superpixel graph are updated from their neighborhood nodes with superpixel-level context, as formulated as follows:

$$\mathbf{H}^K = \sigma\left(\sum_{k=0}^{K}\left(\mathbf{D}^{-\frac{1}{2}}A\mathbf{D}^{\frac{1}{2}}\right)^k \mathbf{X}\Theta_k\right) \tag{8}$$

where $\mathbf{D}$ represents the diagonal degree matrix, $\mathbf{D}_{ii} = \sum_{j=0}\mathbf{A}_{\mathrm{ij}}$; Θ_k denotes the linear weights that sum the aggregation results of different sizes; *K* represents a hyperparameter that indicating the maximum size of convolution kernels; *X* denotes the nodes features of the superpixel graph; $\Sigma(\cdot)$ denotes the activation function.

As shown in Fig. 4, the aggregation operation consists of two consecutive steps: *K*-hop context aggregation and SLCA. The *K*-hop context aggregation aims to aggregate the node representation from *K* graph convolution branches, where each branch captures the node representation of its neighbors in a unique number of hops. And the SLCA is used to combine different scale features. The SLCA module extracts node features by multilayer convolution operations rather than pooling operation, where the multilayer convolution operations allow aggregating the neighborhood features. The aggregation of node v_i from its neighbours is formulated as follows:

$$h_{\mathcal{N}_k(i)}^{(l+1)} = \sum_{k=0}^{K}\sum_{j\in\mathcal{N}_k(i)}\left(\theta_k \mathrm{x}_j^{(l)} + b\right) \tag{9}$$

where $\mathcal{N}_k(i)$ is the index set of nodes that are *k*-hop neighbor of node *i*. And θ_k is the corresponding aggregation weight, which is a learnable parameter. $x_j^{(l)}$ denotes the feature of the jth neighbor of node i after the lth convolutional layer. b denotes the bias. The SLCA operation involves 0-hop context features (i.e., features of the nodes). In this way, this operation facilitates the aggregation of both local and long-range features of the superpixels, thereby enabling the capture of superpixel-level contextual features. Meanwhile, the SLCA module uses adaptive aggregation weights to improve the discriminative feature representation of superpixels.

In practice, the generated superpixel graphs may have unbalanced node classes. To mitigate the problem, the module derives a loss function from the CE function, which adds the class balanced and superpixel weight. The loss is defined as follows:

$$\text{loss}_k = -w_k y_k \cdot \log \frac{\exp\left(\hat{y}_k, y_k\right)}{\sum_{c=1}^{C} \exp\left(\hat{y}_k, c\right)} \tag{10}$$

where $\hat{y}$ is prediction label, and y is ground truth label representing the truth category for each node. The pixel-superpixel mapping matrix is used to map from the ground truth label to graph nodes. Each superpixel, each node in the graph, is categorized into the dominant class of its pixels. w denotes class-balanced weight, and C represents the total number of classes. The w_k is calculated by the following:

$$w_k = \frac{N - n_k}{N} \tag{11}$$

where N presents the total samples and n is the number of superpixels in each class. Here, the class-balanced weight helps in penalizing misclassification in classes with fewer samples than the others. Based on the losses of class-balanced CE in each node, the superpixel penalty loss in each node k, SPL_k is defined as follows:

$$\text{SPL}_k = s_k \cdot \text{loss}_k \tag{12}$$

where s_k is a superpixel weight for each node k, defined as follows:

$$s_k = -\frac{1+\epsilon}{\log r_k + \epsilon}, \tag{13}$$

where r_k denotes the proportion of the pixels contained in the kth superpixel node relative to the pixels in an RSI image. ϵ is the constant value to avoid the zero division error, set as 10^{-5}. According to (13), the module imposes a greater penalty on superpixel nodes that contain more pixels than others, resulting in a greater effect on the prediction accuracy. The superpixel penalty loss is calculated by

$$SPL = \frac{1}{N}\left(\left[l_1, \ldots, l_N\right]^T \cdot \left[s_1, \ldots, s_N\right]\right) \tag{14}$$

where $l_1, \ldots, l_N$ denote the loss of superpixel 1 to superpixel n, respectively.

Although more GCN layers help in improving the representation and prediction capabilities of the GCN networks, many studies on GCNs employ shallow networks [48]. This phenomenon is mainly due to the GCN that aggregates the features of the neighboring nodes, increasing the number of layers that can potentially cause an oversmoothing problem. To overcome the oversmoothing problem of the deep GCN, this module takes into account losses from multiple GCN layers as follows:

$$\text{Loss}_{\text{overall}} = \text{SPL}_{l_2} + \text{SPL}_{l_4} + \text{SPL}_{l_6} \tag{15}$$

where SPL_{l_i} denotes the superpixel penalty loss at the ith hidden layer. In our experiment, we extract the intermediate loss from the 2nd, 4th, and 6th graph convolutional layers.

D. *Pixel-Wise Classification*

During the inference stage, the images to be predicted will be first put into the trained fully convolution superpixel network to obtain pixel-superpixel mapping $\tilde{\text{Q}}$. Subsequently, graph $\tilde{\mathcal{G}}(\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ is constructed in the superpixel-base graph construction module. Afterward, graph $\tilde{\mathcal{G}}$ will be fed into the trained SLCA to predict the category of each superpixel $\tilde{\text{S}}_{\text{gcn}}$ by the following:

$$\tilde{\text{S}}_{\text{gcn}} = \textbf{argmax}\left(\tilde{\mathbf{y}}_f^{(L)}\right) \tag{16}$$

where $\tilde{\mathbf{y}}_f^{(L)}$ denotes the node features through L-layer GCN convolution. In the end, the categories of the pixels are calculated by the following:

$$\tilde{\text{Z}} = \tilde{\text{Q}}\tilde{\text{S}}_{\text{gcn}} \tag{17}$$

where $\tilde{\text{Z}}$ represents the predicted pixel labels. In this study, the predicted labels of a pixel is building or nonbuilding.

IV. Experiments and Results

A. *Datasets*

In the experiments, we employed three publicly datasets to investigate the proposed method, including WHU dataset [31], CrowdAI dataset [49], and Typical Cities of China dataset (TCC) [50].

1) WHU Dataset: The WHU dataset covers buildings of various materials, structures, and distributions, making it challenging to extract buildings from the images. This dataset covers 450 km^2 regions in New Zealand, contains more than 2000 independent buildings, with a spatial resolution of 0.075 m. The dataset consists of 8189 nonoverlapping tiles. The training, validation and test sets consist of 4736, 1036, and 2416 images, respectively. Each tile is in TIF format, with 512 × 512 pixels. Some of the images are shown in Fig. 5(a).

2) CrowdAI Dataset: The training and validation set of the dataset includes 280 741 training and 60 317 validation images, respectively, with a spatial resolution of 0.3 m. These images are in 300 × 300-pixel JPEG format. Their ground-truth images are annotated in MS-COCO format. Buildings in the dataset are very diverse with varied shapes and sizes. Some of the images are shown in Fig. 5(b).

3) TCC Building Dataset: The dataset was collected from Google Earth, which are located in four Chinese cities, Shanghai, Beijing, Shenzhen, and Wuhan. It is published in [50] including many nonorthophoto images with a spatial resolution of 0.29 m. Each image is in 500 × 500 pixels with a spatial resolution of 0.29 m. The training and test set contains 5985 and 1275 images, respectively. Some images in the dataset are shown in Fig. 5(c).

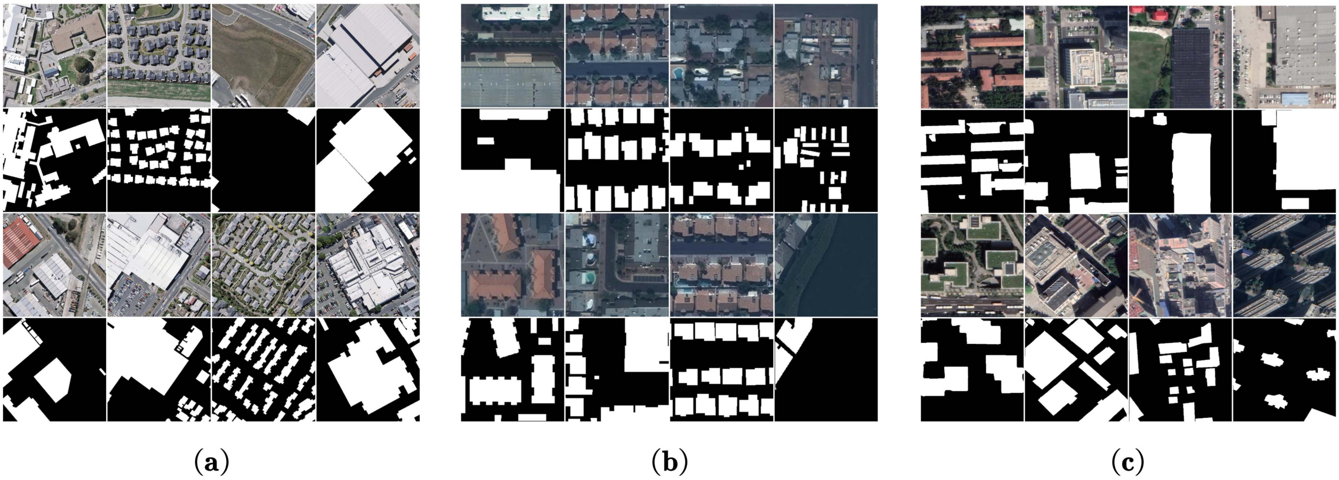


Fig. 5. Samples of the three datasets:(a) WHU. (b) CrowdAI. (c) TCC. The first and third rows are the original images, and the second and fourth rows are their ground truth images.

TABLE II
EXPERIMENTAL ENVIRONMENT AND PARAMETER SETTINGS

Environment Configuration		Parameter Settings	
Operating system	Ubuntu 20.04.5 LTS	Epoch	500
DL framework	Pytorch 1.10.1	Patience	50
Language	Python 3.9.13	Batch size	32
Memory	64GB	Optimizer	Adam
GPU	3090 24GB	Decay/Steps	0.6/10
CPU	i7-12700KF	Initial learning rate	0.001

B. Experimental Settings

The experiments were implemented with the PyTorch framework and CUDA11.1, and performed with a single NVIDIA GTX 3090. The parameter settings and experimental environment are reported in Table II.

Specifically, Adam [51] is employed to train model with an initial learning rate of 0.001 and decaying 0.6 times every 10 epochs. The LeakyReLU function was selected as the activation function of SLCA. The training was terminated when the loss of the proposed model does not decrease in 50 epochs. The batch size was set to 1 because an RSI formed a superpixel graph. When constructing the superpixel graph, the initial grid size r was set to 10. Following the previous work [52], the value of K is set to 4. Accordingly, each image consists of approximately 3000 images. In addition, we resize the images of the three datasets to 550×550. In the baselines, the FCN and PSPNet used HRNet_W18 [53] and ResNet50_vd [54] as the backbones, respectively.

C. Evaluation Metrics

Six widely used evaluation metrics, include OA, precision, recall, F1-Score, kappa coefficient, and mean intersection-over-union (mIoU) are chosen to examine the proposed model. For a class, recall represents the ratio of correctly predicted pixels to the total number of pixels. Precision is the proportion of correctly segmented pixels in a given class. OA indicates the rate of correctly predicted pixels to the number of all pixels. The F1-Score is defined based on precision and recall by

$$\text{F1-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{18}$$

The kappa coefficient is defined by

$$\text{Kappa} = \frac{p_0 - p_e}{1 - p_e} \tag{19}$$

where p_0 denotes the relative agreement between the segmentation; p_e represents the hypothetical probability of chance agreement. The definitions of the *IoU* and the *mIoU* are presented in the following equations:

$$\text{IoU} = \frac{\text{area}\,(M_p \cap M_{gt})}{\text{area}\,(M_p \cup M_{gt})} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \tag{20}$$

$$\text{mIoU} = \frac{1}{k+1} \sum_{i=0}^{k} \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \tag{21}$$

where M_p and M_{gt} denote the predicted labels and the corresponding ground truth labels, respectively; and k represents the number of classes. The *mIoU* stands for the mean intersection over union, which is the average of the intersection over union values for all categories. TP refers to the true positive rate, FN signifies the false negative rate, and FP represents the false positive rate

D. Baselines

Six classical semantic segmentation methods, namely FCN [11], U-Net [39], SegNet [38], PSPNet [40], SACANet [55], and BCE-Net [56] are selected as baselines to examine our method. Specifically, FCN reaches a milestone in semantic segmentation, which can accept input images and make a prediction for each pixel of the image. U-Net has found wide application in various fields for image semantic segmentation tasks due to its elegant network architecture. SegNet, one of the classic models of semantic segmentation, is known for its ability to preserve pixel-level detail while using fewer parameters. The encoder–decoder architecture of the model allows it to adapt

TABLE III
Comparison of the Different Extraction Methods on the WHU Dataset

Method	OA	Precision	Recall	F1-Score	Kappa	mIoU
FCN	94.85%	87.92%	90.26%	91.10%	90.19%	85.45%
U-Net	97.31%	91.87%	93.50%	91.69%	91.37%	88.27%
SegNet	95.23%	88.77%	95.22%	92.50%	90.99%	86.60%
PSPNet	96.85%	89.28%	92.85%	91.06%	90.12%	87.39%
SACANet	97.79%	91.01%	95.33%	91.10%	90.96%	88.17%
BCE-Net	97.18%	**92.98%**	93.34%	91.78%	89.93%	88.76%
SLIC-MLP	78.88%	63.97%	77.41%	68.25%	47.02%	56.88%
Proposed model	**98.55%**	92.01%	**96.95%**	**93.96%**	**91.59%**	**89.93%**

The best results are highlighted in bold and the second-best results are underlined.

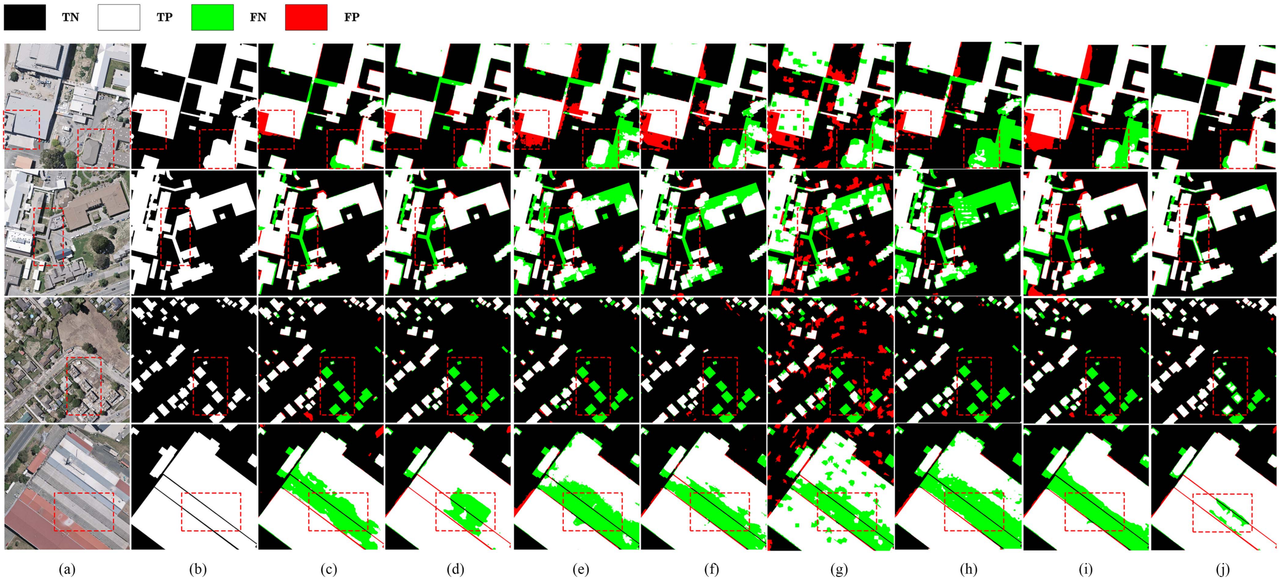


Fig. 6. Visualized results on the WHU dataset: (a) Original RSI. (b) Ground truth. (c) FCN. (d) PSPNet. (e) SegNet. (f) U-Net. (g) SLIC-MLP. (h) SACANet. (i) BCE-Net. (j) Proposed model. (Notation: white, black, green, and red pixels represent predictions of TP, TN, FN, and FP, respectively).

well to objects of varying scales with a more pronounced effect on the segmentation of small objects. PSPNet captures the global and local context information by introducing the pyramid pooling module and the dilated convolution, which can help the network achieve higher accuracy and better performance in complex scenarios. SACANet is a network integrating both scene-aware and class attentions for semantic segmentation of RSIs. BCE-Net is a multibranch building extraction method for HR RSIs, which uses a contrastive learning strategy to increase the distinguishability of buildings and nonbuildings.

In addition, a superpixel-based method, namely SLIC-MLP, is used as a baseline. SLIC-MLP employs the SLIC [57] algorithm to generate superpixels, and adopts an MLP network to predict the labels of the superpixels.

E. Results

1) Evaluation With the WHU Dataset: Its experimental results are presented in Table III. Notably, the proposed method outperforms all baselines. The proposed method improves OA, recall, F1-Score, and mIoU by 0.76%, 1.62%, 1.46%, and 1.17%, respectively, compared with the second-best results. This finding suggests that the proposed approach achieves advantages in building extraction by optimizing the superpixel generation and classification process. In addition, compared with SLIC-MLP, the proposed method achieves significant improvement in OA, recall, F1-Score, and mIoU by 19.67%, 19.54%, 25.71%, and 33.05%, respectively. The improvement is traced to two factors. First, the proposed method can achieve more precise boundary delineation of superpixels, which reduces the probability of different categories of pixels being segmented into the same superpixel. Second, the SLCA mechanism effectively captures the context of superpixels, including local and long-range features, resulting in improved classification accuracy.

To further observe the results, Fig. 6 visualizes the extracted buildings by these methods on the WHU dataset. The white (TP) and black areas (TN) areas denote the correct prediction, the red areas represent that the background areas are incorrectly

TABLE IV
COMPARISON OF DIFFERENT EXTRACTION METHODS ON THE CROWDAI DATASET

Method	OA	Precision	Recall	F1-Score	Kappa	mIoU
FCN	93.10%	92.27%	91.73%	92.42%	84.43%	85.58%
U-Net	94.80%	93.05%	92.01%	92.52%	85.03%	86.34%
SegNet	93.79%	91.72%	90.39%	91.03%	82.06%	83.92%
PSPNet	94.96%	93.89%	92.85%	93.56%	86.49%	87.14%
SACANet	94.99%	93.39%	94.77%	92.98%	86.33%	88.01%
BCE-Net	93.73%	92.55%	94.10%	93.13%	**87.89%**	87.94%
SLIC-MLP	72.64%	65.79%	70.30%	67.44%	45.13%	54.47%
Proposed model	**95.78%**	**94.11%**	**95.12%**	**94.61%**	87.77%	**88.86%**

The best results are highlighted in bold and the second-best results are underlined.

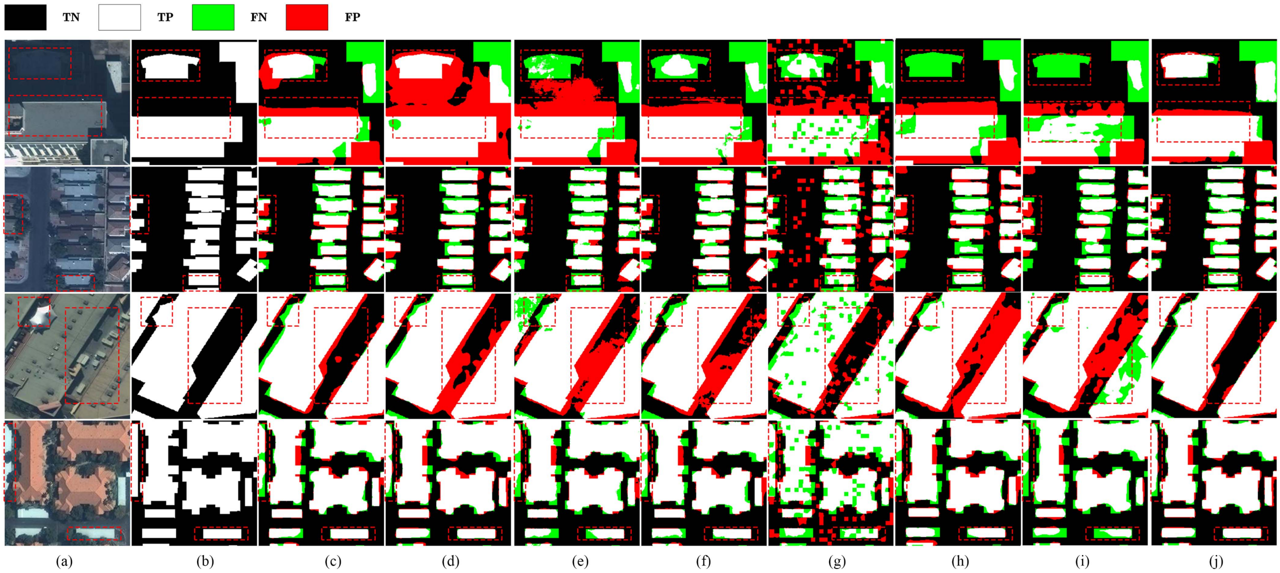


Fig. 7. Visualized results on the CrowdAI dataset: (a) Original RSI. (b) Ground truth. (c) FCN. (d) PSPNet. (e) SegNet. (f) U-Net. (g) SLIC-MLP. (h) SACANet. (i) BCE-Net. (j) Proposed model. (Notation: white, black, green, and red pixels represent predictions of TP, TN, FN, and FP, respectively).

predicted as building areas (FP), while the green areas indicate that the building areas are incorrectly predicted as the backgrounds (FN). The buildings in the proposed method demonstrate more integrity and exhibit more precise boundaries in these challenge scenarios compared with the baselines. For instance, the proposed method extracts buildings with clearer boundaries when the color and texture of the buildings are similar to the backgrounds, as presented in the first row of Fig. 6. As shown in the second and third rows of the figure, our method can more accurately extract buildings with different shapes and densely distributed small buildings. In the scenery of large buildings with different visual characteristics, our method can more correctly distinguish the buildings compared to the baselines, as shown in the last row of Fig. 6. The abovementioned results demonstrate that the proposed method helps in addressing the issue of "same objects with different spectrums, and same spectrums with different objects." In addition, the FN and FP errors of the proposed method are smaller than SLIC-MLP. TSS is trained with labeled samples, helping fit superpixels to building areas. Moreover, we argue that the SLCA module captures both local and long-range contextual features, thus improving the ability to discriminate building areas.

2) Evaluation With CrowdAI Dataset: Its experimental results are presented in Table IV, which are lower than those reported in Table III. This phenomenon is due to the dataset has lower resolution and more complex building scenes than the WHU dataset. Nevertheless, the proposed method performs best across all metrics. Furthermore, compared with the second-best results, the performance has increased by 0.79%, 1.05%, and 0.92% on OA, F1-Score, and mIoU, respectively.

The experimental results of these methods are visualized in Fig. 7. As shown the figure, compared to the baseline methods, the proposed approach yields buildings with more complete structures and relatively sharper boundaries. For example, in the first and second rows of the figure, the buildings are obscured or covered by trees. Most baseline methods misclassify them as background, while our method accurately extracts the buildings in these areas. As presented in the third row in Fig. 7, our method

TABLE V
COMPARISON OF NETWORK EXTRACTION METHODS ON THE TCC DATASET

Method	OA	Precision	Recall	F1-Score	Kappa	mIoU
FCN	92.69%	89.58%	89.54%	89.56%	81.12%	82.85%
U-Net	90.97%	87.40%	85.85%	86.60%	75.20%	78.25%
SegNet	90.47%	86.70%	84.87%	85.74%	73.49%	76.98%
PSPNet	92.55%	89.37%	89.30%	89.34%	80.68%	82.49%
SACANet	91.77%	88.98%	92.53%	88.22%	81.01%	81.57%
BCE-Net	92.81%	89.79%	91.65%	89.69%	81.31%	83.10%
SLIC-MLP	71.53%	58.82%	60.72%	59.35%	39.14%	45.85%
Proposed model	**93.77%**	**90.77%**	**93.01%**	**91.88%**	**82.06%**	**84.77%**

The best results are highlights in bold and the second-best results are underlined.

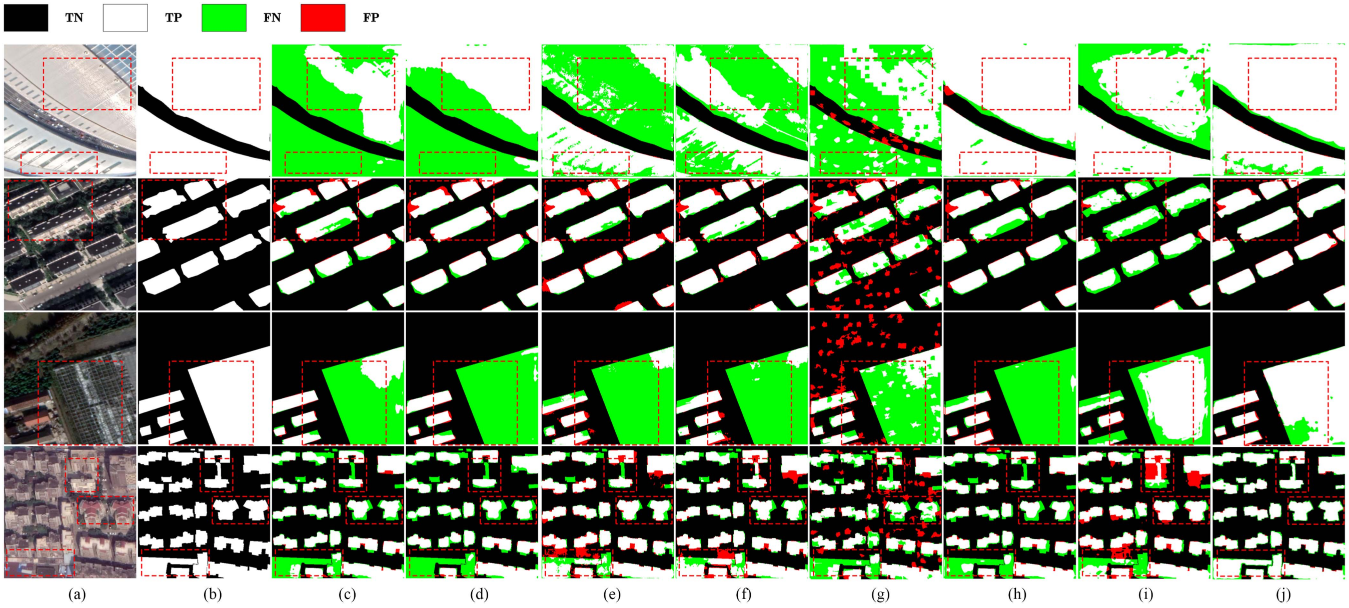


Fig. 8. Visualized results on the TCC dataset: (a) Original RSI. (b) Ground truth. (c) FCN. (d) PSPNet. (e) SegNet. (f) U-Net. (g) SLIC-MLP. (h) SACANet. (i) BCE-Net. (j) Proposed model. (Notation: white, black, green, and red pixels represent predictions of TP, TN, FN, and FP, respectively).

can recognize the buildings that resemble the ground more efficiently. In addition, our method can obtain more refined building boundaries in complex building shape scenarios, as shown in the last row of the figure. These visualization results further validate the outstanding performance of the proposed method.

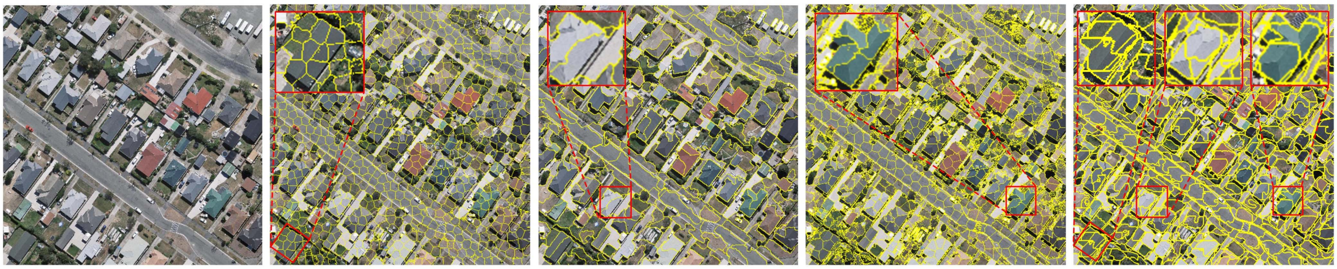
3) Evaluation With TCC Dataset: The dataset contains many nonorthophoto images, which make building extraction more difficult. The quantitative accuracy assessment results on this dataset are reported in Table V. As shown in the table, the dataset, the values of all the evaluation metrics are lower than those of the WHU and CrowdAI datasets. Among all the competitors, the proposed method achieves the highest OA among all the competitors with 93.77%. The recall, F1-Score and mIoU are 3.47%, 2.32%, and 1.92% higher than those of the second-best results, respectively. The OA, recall, F1-Score, and mIoU are 0.96%, 0.48%, 2.19%, and 1.67% higher than the second-best results, respectively.

Several samples are depicted in Fig. 8 to provide a more detailed observation of the results. As shown in the first and second rows, the buildings extracted by our method have less impulse noise, are smoother, and have more continuous boundaries than the buildings extracted by other methods. As presented in the third row of the figure, our method performs better in distinguishing buildings from nonbuildings. In addition, compared with the baselines, the extracted buildings by the proposed method shows more accurate, as shown in the fourth row of the figure. The visualized results further suggest that our method is promising.

V. DISCUSSION

A. Ablation Study

Ablation experiments are conducted to further evaluate the effect of key components. The OA values on the three datasets are listed in Table VI. Introducing the TSS module increases the OA values by 3.09%, 3.93%, and 4.36% on the three datasets, respectively, as shown in Table VI. This improvement is attributed to the fact that TSS learns building features from the training data integrating local and overall cues. And it benefits from the gains



(a) **Original Image** (b) **SLIC** (c) **SEED** (d) **LSC** (e) **TSS**

Fig. 9. Visualized results of different superpixel segmentation methods on the image of the WHU dataset.

TABLE VI
OA (%) Indices of the Combinations of Different Modules on Three Datasets

Baseline	TSS	SLCA	SPL	OA		
				WHU	CrowdAI	TCC
✓	-	-	-	78.88%	72.64%	71.53%
✓	✓	-	-	81.97% (+3.09%)	76.57% (+3.93%)	75.89% (+4.36%)
✓	-	✓	-	95.93% (+17.05%)	92.12% (+19.48%)	89.01% (+17.48%)
✓	-	-	✓	82.54% (+3.66%)	76.96% (+4.32%)	74.84% (+3.31%)
✓	✓	✓	✓	98.55% (+19.67%)	95.78% (+23.14%)	93.77% (+22.24%)

that the pixel category is consistent with the superpixel category. In addition, introducing SLCA significantly increases the OA values by 17.05%, 19.48%, and 17.48% on the three datasets, respectively. This phenomenon is due to the SLCA that captures both the local and the long-range neighborhood features, which helps in identifying obscure or confusing buildings. Finally, the introduction of SPL increases the OA values by 3.66%, 4.32%, 3.31% on the three datasets. The proposed method merges the TSS and SLCA modules, which increases the accuracies by 19.67%, 23.14%, and 22.24% in terms of OA on the three datasets, respectively. This results suggests both TSS and SLCA effectively contribute to the improvement of building extraction from HR RSI.

B. *Effect of the Superpixel Segmentation Algorithms*

Given that superpixel segmentation is a fundamental component of our method, the superpixel segmentation algorithm should affect the model performance. Experiments are performed on the WHU dataset with four algorithms, including SLIC [57], SEED [58], LSC [59], and our TSS, to investigate the effect of superpixel segmentation algorithms. SLIC is a widely used superpixel segmentation algorithm, which generates superpixels based on k-means clustering. SEED is an energy-driven sampling-based superpixel segmentation algorithm by continuously refining the boundaries of superpixels. The LSC algorithm represents image pixels as vectors in a low-dimensional space, and uses a spectral clustering algorithm to generate superpixels. In this article, the number of superpixel is set to 1000 to facilitate the observation and distinguishing the differences in the superpixel segmentation algorithms. In Table VII, the TSS achieves the best values in all metrics. The proposed TSS algorithm achieved higher scores in OA (1.42%), F1-Score (3.41%), and mIoU (3.37%) compared to the SLIC algorithm, which obtained the second-best average evaluation metrics.

To illustrate the segmentation results of these algorithms, an image from the dataset is taken as an example for further observation in Fig. 9. Three typical regions were selected, marked with red rectangular boxes and shown enlarged. As shown in Fig. 9(b), the SLIC algorithm results in blurring at the building boundaries and misclassification of the ground shadows as buildings. This is mainly due to the fact that the SLIC algorithm classifies the superpixels based only on the color and position of the pixels, resulting in insufficient feature information. As shown in Fig. 9(c) and (d), the superpixels generated by the SEED and LSC algorithms suffer from some limitations, with the former having rough superpixel boundaries and the latter having overly fragmented superpixels. While TSS aggregates pixels of the same type more efficiently and generates superpixels with clearer boundaries compared with the untrainable algorithms, including SLIC, SEED, and LSC. The experimental results suggest that TSS is an effective superpixel segmentation method and can promote to better extract buildings.

C. *Effect of Aggregation Strategies*

In this study, the SLCA module aims to aggregate superpixel-level contextual information. Experiments are conducted with five types of aggregation strategies, including vanilla GCN [46], graph sample and aggregate network (GraphSage [60]), graph attention network (GAT [61]), graph isomorphism network (GIN [62]), Mixhop [63], and our SLCA, to examine the effect of aggregation strategies. In Table VIII, SLCA shows a large advantage in all metrics. This phenomenon is due to the SLCA that aggregates local and long-range context information through a SLCA mechanism. Moreover, compared with Mixhop, SLCA adopts adaptive weights to aggregate features from neighbors, which enhances the discriminability of superpixels, thereby improving the performance of the model.

D. *Parameter Sensitivity Analysis*

Two hyperparameters, including the initial grid size r in superpixel segmentation (see Section III-A) and the size of the convolution kernel in SLCA (seeSection III-C), have the potential to impact the model's performance. The sensitivity analysis of these hyperparameters was conducted in this section.

1) Effect of R: The initial grid size r in Section III-B determines the number of superpixels generated from each image. To be more specific, the number of superpixels is equal to the area

TABLE VII
EVALUATION RESULTS OF THE DIFFERENT SUPERPIXEL SEGMENTATION ALGORITHMS ON THE WHU DATASET

Method	OA	Precision	Recall	F1-Score	Kappa	mIoU
SLIC	97.13%	89.73%	94.97%	90.55%	90.33%	86.56%
SEED	96.69%	88.31%	93.33%	88.77%	88.79%	84.61%
LSC	94.89%	85.54%	91.17%	87.32%	84.43%	82.27%
TSS	**98.55%**	**92.01%**	**96.95%**	**93.96%**	**91.59%**	**89.93%**

The best results are highlighted in bold and the second-best results are underlined.

TABLE VIII
EVALUATION RESULTS OF THE AGGREGATION STRATEGIES ON THE WHU DATASET

Algorithm	OA	Precision	Recall	F1-Score	Kappa	mIoU
+GCN	96.02%	88.68%	94.89%	90.41%	89.89%	85.36%
+GAT	96.27%	88.96%	93.68%	91.13%	90.11%	84.41%
+GraphSage	96.53%	89.28%	95.05%	91.89%	90.70%	85.59%
+Gin	93.58%	83.83%	83.57%	83.70%	84.41%	74.04%
+Mixhop	97.12%	89.52%	91.10%	91.53%	89.17%	88.11%
+SLCA	**98.55%**	**92.01%**	**96.95%**	**93.96%**	**91.59%**	**89.93%**

The best results are highlighted in bold and the second-best results are underlined.

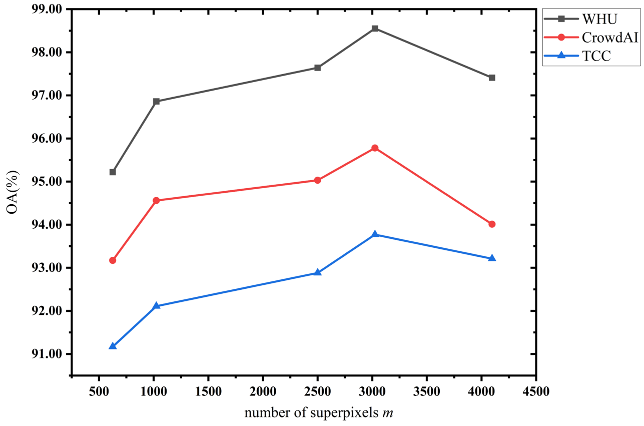


Fig. 10. OA (%) indices with different segmentation numbers *m* on three datasets.

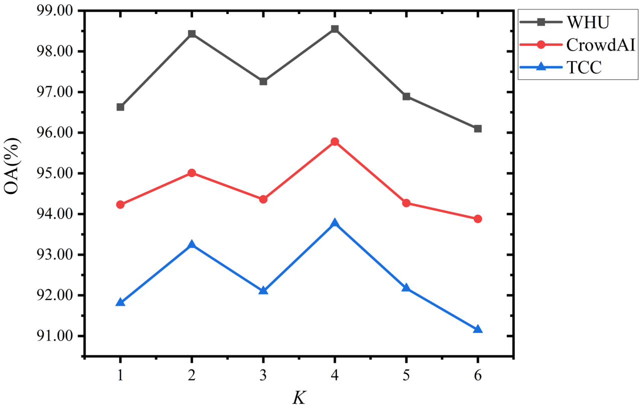


Fig. 11. OA (%) indices with different filter sizes *K* of SLCA on three datasets.

of the image divided by the area of the square with r as the side length. Specifically, to investigate the effects of r, we conducted experiments by setting the size of the initial grid r is in the range of 8, 10, 16, 20, which produces the number of superpixels m in the range of 625, 1024, 2500, 3025, 4096, respectively. The OA values of the datasets are plotted in Fig. 10. The OA values gradually increases as the number of superpixels increments. We argue that fewer superpixels may result in that some superpixels contain pixels from both nonbuilding and building areas, thus decreases extraction accuracy. And in this phase, increasing the number of superpixels will improve the model accuracy.

The OA values on all three datasets reach the highest values when the number of superpixels is 3000. Thereafter, the OA values first gradually decrease. This can be explained that the excessive number of superpixels reduces the information contained in each superpixel, resulting in fragmented segmentation results. In addition, the greater number of superpixels, the more complexity of adjacency matrix, superpixel-graph and the model. This added complexity can make training more challenging and hinder the achievement of high accuracy.

2) Effect of K*:* The hyperparameter, K, in Section III-C represents the maximized size of the convolution kernels. The values of K may affect the performance of our model. In the section, K is set to values from 1 to 6. The experimental results are illustrated in Fig. 11. The proposed method performs best when the filter size is $K = 4$, and the values are quite close when $K = 2$ and $K = 4$. As K increases beyond four, the OA values exhibits a decreasing trend as the scale decreases gradually, which should be due to the occurrence of the "overfitting phenomenon."

E. Model Efficiency

To investigate the efficiency of our method, comparative experiments were conducted on the WHU dataset. We reported

TABLE IX
COMPARISONS EXPERIMENT ON TRAINABLE PARAMETERS, FLOPS, AND TEST TIME OF DIFFERENT METHODS

Method	Params (M)	FLOPs (G)	Test.(second)
FCN	9.67	18.51	0.062
PSPNet	67.90	265.59	0.055
SegNet	29.61	170.25	0.034
U-Net	13.40	124.30	0.030
SLIC-MLP	**0.09**	**0.19**	**0.001**
SACANet	2.7	51.2	0.022
BCE-Net	31.2	15.01	0.006
Ours	5.78	32.91	0.017

The best results are highlights in bold.

the number of trainable parameters, floating point of operations (FLOPs), and the time costs (seconds) of their inference are reported in Table IX. SLIC-MLP requires minimal trainable parameters and spends least inference time compared with pixel-level methods. The proposed method takes relatively less computation time than most other methods. The experimental results suggest that the proposed method makes a better tradeoff between accuracy and efficiency.

VI. CONCLUSION

In this study, a spatial context-aware approach is developed to enhance building extraction for HR RSI. The method develops a TSS module and a SLCA module to improve the ability to discriminate building areas. As the TSS module is trained, it helps in generating superpixels with building features, improving the accuracy of building extraction. SLCA employs more convolutional layers with different filter sizes to enhance feature extraction, which aggregating superpixel-level contextual information and consequently improving the building extraction. The experimental result on the three public datasets illustrate that the proposed method is superior to baselines. This work explores a new approach of superpixel-based building extraction of HR RSI, and provides a methodological reference for the various segmentation of HR RSI images.

A further study can focus on introducing the spatial relationship of buildings and more pixel features to improve the superpixel segmentation. In addition, this model can be optimized with the semisupervised paradigm to alleviate the lack of building image labels.

REFERENCES

[1] R. Alshehhi, P. R. Marpu, W. L. Woon, and M. D. Mura, “Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 130, pp. 139–149, 2017.

[2] X. Gao, M. Wang, Y. Yang, and C. Li, “Building extraction from RGB VHR images using shifted shadow algorithm,” *IEEE Access*, vol. 6, pp. 22034–22045, 2018.

[3] Q. Zhu et al., “Knowledge-guided land pattern depiction for urban land use mapping: A case study of chinese cities,” *Remote Sens. Environ.*, vol. 272, 2022, Art. no. 112916. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003442572200030X

[4] G. Tepanosyan, V. Muradyan, A. Hovsepyan, G. Pinigin, A. Medvedev, and S. Asmaryan, “Studying spatial-temporal changes and relationship of land cover and surface urban heat island derived through remote sensing in Yerevan, Armenia,” *Building Environ.*, vol. 187, 2021, Art. no. 107390.

[5] Q. Bi, K. Qin, H. Zhang, Y. Zhang, Z. Li, and K. Xu, “A multi-scale filtering building index for building extraction in very high-resolution satellite imagery,” *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 482.

[6] W. Li, C. He, J. Fang, J. Zheng, H. Fu, and L. Yu, “Semantic segmentation-based building footprint extraction using very high-resolution satellite images and multi-source GIS data,” *Remote Sens.*, vol. 11, 2019, Art. no. 403.

[7] C. Xiong, Q. Li, and X. Lu, “Automated regional seismic damage assessment of buildings using an unmanned aerial vehicle and a convolutional neural network,” *Automat. Construction*, vol. 109, 2020, Art. no. 102994.

[8] A. J. Cooner, Y. Shao, and J. B. Campbell, “Detection of urban damage using remote sensing and machine learning algorithms: Revisiting the 2010 Haiti earthquake,” *Remote Sens.*, vol. 8, no. 10, 2016, Art. no. 868.

[9] Q. Hu, L. Zhen, Y. Mao, X. Zhou, and G. Zhou, “Automated building extraction using satellite remote sensing imagery,” *Automat. Construction*, vol. 123, 2021, Art. no. 103509.

[10] X. Yuan, J. Shi, and L. Gu, “A review of deep learning methods for semantic segmentation of remote sensing imagery,” *Expert Syst. Appl.*, vol. 169, 2021, Art. no. 114417.

[11] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[12] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “Convolutional neural networks for large-scale remote-sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[13] Y. Xu, L. Wu, Z. Xie, and Z. Chen, “Building extraction in very high resolution remote sensing imagery using deep learning and guided filters,” *Remote Sens.*, vol. 10, no. 1, 2018, Art. no. 144.

[14] G. Wu et al., “Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks,” *Remote Sens.*, vol. 10, no. 3, 2018, Art. no. 407.

[15] Z. Zhang and Y. Wang, “JointNet: A common neural network for road and building extraction,” *Remote Sens.*, vol. 11, no. 6, 2019, Art. no. 696.

[16] Y. Sun and W. Zheng, “HRNet- and PSPNet-based multiband semantic segmentation of remote sensing images,” *Neural Comput. Appl.*, vol. 35, no. 12, pp. 8667–8675, 2022.

[17] X. Jin and Y. Gu, “Superpixel-based intrinsic image decomposition of hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4285–4295, Aug. 2017.

[18] T. K. Behera, S. Bakshi, M. Nappi, and P. K. Sa, “Superpixel-based multiscale CNN approach toward multiclass object segmentation from UAV-captured aerial images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1771–1784, Jan. 2023.

[19] Y. Qing et al., “Operational earthquake-induced building damage assessment using CNN-based direct remote sensing change detection on superpixel level,” *Int. J. Appl. Earth Observation Geoinf.*, vol. 112, 2022, Art. no. 102899. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843222001017

[20] A. Benchabana, M.-K. Kholladi, R. Bensaci, and B. Khaldi, “Building detection in high-resolution remote sensing images by enhancing superpixel segmentation and classification using deep learning approaches,” *Buildings*, vol. 13, no. 7, 2023, Art. no. 1649. [Online]. Available: https://www.mdpi.com/2075-5309/13/7/1649

[21] M. Ghanea, P. Moallem, and M. Momeni, “Building extraction from high-resolution satellite images in urban areas: Recent methods and strategies against significant challenges,” *Int. J. Remote Sens.*, vol. 37, no. 21, pp. 5234–5248, 2016.

[22] G. Sohn and I. Dowman, “Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 62, no. 1, pp. 43–63, 2007.

[23] M. Dixit, K. Chaurasia, and V. K. Mishra, “Dilated-ResUnet: A novel deep learning architecture for building extraction from medium resolution multi-spectral satellite imagery,” *Expert Syst. Appl.*, vol. 184, 2021, Art. no. 115530.

[24] J. Wang, X. Yang, X. Qin, X. Ye, and Q. Qin, “An efficient approach for automatic rectangular building extraction from very high resolution optical satellite imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 3, pp. 487–491, Mar. 2015.

[25] S. Cui, Q. Yan, and P. Reinartz, “Complex building description and extraction based on hough transformation and cycle detection,” *Remote Sens. Lett.*, vol. 3, no. 2, pp. 151–159, 2012.

[26] X. Huang and L. Zhang, "Morphological building/shadow index for building extraction from high-resolution imagery over urban areas," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 161–152, Feb. 2012.

[27] F. Karsli, M. Dihkan, H. Acar, and A. Ozturk, "Automatic building extraction from very high-resolution image and LiDAR data with SVM algorithm," *Arabian J. Geosci.*, vol. 9, 2016, Art. no. 635.

[28] E. Li, J. Femiani, S. Xu, X. Zhang, and P. Wonka, "Robust rooftop extraction from visible band images using higher order CRF," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4483–4495, Aug. 2015.

[29] Y. Liu et al., "ARC-net: An efficient network for building extraction from high-resolution aerial images," *IEEE Access*, vol. 8, pp. 154997–155010, 2020.

[30] Y. Li, X. Huang, and H. Liu, "Unsupervised deep feature learning for urban village detection from high-resolution remote sensing images," *Photogrammetric Eng. Remote Sens.*, vol. 83, no. 8, pp. 567–579, 2017.

[31] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 574–586, Jan. 2019.

[32] W. Feng, H. Sui, L. Hua, C. Xu, G. Ma, and W. Huang, "Building extraction from VHR remote sensing imagery by combining an improved deep convolutional encoder-decoder architecture and historical land use vector map," *Int. J. Remote Sens.*, vol. 41, no. 17, pp. 6595–6617, 2020.

[33] H. Hosseinpoor and F. Samadzadegan, "Convolutional neural network for building extraction from high-resolution remote sensing images," in *Proc. Int. Conf. Mach. Vis. Image Process.*, 2020, pp. 1–5.

[34] J. Ma, L. Wu, X. Tang, F. Liu, X. Zhang, and L. Jiao, "Building extraction of aerial images by a global and multi-scale encoder-decoder network," *Remote Sens.*, vol. 12, no. 15, 2020, Art. no. 2350.

[35] X. Pan et al., "Building extraction from high-resolution aerial imagery using a generative adversarial network with spatial and channel attention mechanisms," *Remote Sens.*, vol. 11, 2019, Art. no. 917.

[36] T. Zuo and F. Juntao, "HF-FCN: Hierarchically fused fully convolutional network for robust building extraction," in *Proc. Asian Conf. Comput. Vis.*, 2017, pp. 291–302.

[37] G. Wu et al., "Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks," *Remote Sens.*, vol. 10, 2018, Art. no. 407.

[38] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239.

[41] Z. Ye, Y. Fu, M. Gan, J. Deng, A. Comber, and K. Wang, "Building extraction from very high resolution aerial imagery using joint attention deep neural network," 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/24/2970

[42] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1480–1484.

[43] W. Yuan, J. Wang, and W. Xu, "Shift pooling PSPNet: Rethinking PSPNet for building extraction in remote sensing images from entire local feature pooling," 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/19/4889

[44] Z. Li, E. Li, A. Samat, T. Xu, W. Liu, and Y. Zhu, "An object-oriented CNN model based on improved superpixel segmentation for high-resolution remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4782–4796, Jan. 2022.

[45] Z. H. S. Liang and J. Li, "Hybrid transformer-CNN networks using superpixel segmentation for remote sensing building change detection," *Int. J. Remote Sens.*, vol. 44, no. 8, pp. 2754–2780, 2023. [Online]. Available: https://doi.org/10.1080/01431161.2023.2208711

[46] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[47] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[48] C. Yang, R. Wang, S. Yao, S. Liu, and T. F. Abdelzaher, "Revisiting "oversmoothing" in deep GCNS," 2020, *arXiv:2003.13663*.

[49] S. P. Mohanty, "CrowdAI mapping challenge 2018 : Baseline with mask RCNN," *GitHub Repository*, GitHub, 2018. [Online]. Available: https://github.com/crowdai/crowdai-mapping-challenge-mask-rcnn

[50] K. Wu et al., "A dataset of building instances of typical cities in China," *Chin. Sci. Data*, vol. 6, pp. 191–199, 2021.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[52] J. Du, S. Zhang, G. Wu, J. E. M. Moura, and S. Kar, "Topology adaptive graph convolutional networks," 2017, *arXiv:1710.10370*.

[53] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[54] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 558–567.

[55] X. Ma et al., "SACANet: Scene-aware class attention network for semantic segmentation of remote sensing images," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2023, pp. 828–833.

[56] C. Liao et al., "BCE-Net: Reliable building footprints change extraction based on historical map and up-to-date images using contrastive learning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 201, pp. 138–152, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271623001284

[57] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[58] M. V. D. Bergh, X. Boix, G. Roig, B. D. Capitani, and L. V. Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 13–26.

[59] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3317–3330, Jul. 2017.

[60] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[61] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[62] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," 2018, *arXiv:1810.00826*.

[63] S. Abu-El-Haija et al., "MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 21–29.

Fang Fang (Member, IEEE) received the B.S. degree in computer science and technology and the Ph.D. degree in management science and engineering from the China University of Geosciences, Wuhan, China, in 1998 and 2012, respectively.

She is currently an Associate Professor with the School of Computer Science, China University of Geosciences. Her research interests include intelligent interpretation of remote sensing imagery and urban visual intelligence.

Kang Zheng received the B.S. degree in software engineering from the China University of Geosciences, Wuhan, China, in 2021. He is currently working toward the M.S. degree in software engineering with the China University of Geosciences, Wuhan, China.

His research interests include semantic segmentation of remote sensing.

Shengwen Li (Member, IEEE) received the B.S. degree in computer science and technology and the Ph.D. degree in cartography and geographic information engineering from the China University of Geosciences, Wuhan, China, in 2000 and 2010, respectively.

He is currently an Associate Professor with the School of Computer Science, China University of Geosciences. His research interests include deep learning for remote sensing, spatialtemporal data mining, and knowledge graph.

Rui Xu is currently working toward the M.S. degree in software engineering with the China University of Geosciences, Wuhan, China.

His research interests include semisupervised learning-based image segmentation, and remote sensing image processing.

Qingyi Hao is currently working toward the M.S. degree in software engineering with the China University of Geosciences, Wuhan, China.

Her research interests include lightweight semantic segmentation.

Yuting Feng is working toward the M.S. degree in software engineering with the China University of Geosciences, Wuhan, China.

Her research interests include remote sensing image processing and unsupervised domain adaptive semantic segmentation.

Shunping Zhou (Member, IEEE) received the B.S. degree in computer science and technology and the Ph.D. degree in cartography and geographic information engineering from the China University of Geosciences, Wuhan, China, in 1991 and 2003, respectively.

His research interests include spatial database technology, geospatial artificial intelligence, and computer vision.