Multiscale U-Shaped CNN Building Instance Extraction Framework With Edge Constraint for High-Spatial-Resolution Remote Sensing Imagery

Yuanyuan Liu, *Member, IEEE*, Dingyuan Chen, *Graduate Student Member, IEEE*, Ailong Ma, *Member, IEEE*, Yanfei Zhong, *Senior Member, IEEE*, Fang Fang, *Member, IEEE*, and Kai Xu

Abstract **Building extraction based on high-resolution remote sensing imagery has been widely used in automatic surveying and mapping. However, few methods have been developed for building instance extraction, i.e., extracting each building s footprint separately, which is required in a number of applications, such as the smallest unit of a cadastral database. In building instance extraction, there are two challenges: 1) buildings with various scales exist in the imagery and 2) precise building footprints are dif cult to extract due to the blurry boundaries. In this article, to solve these problems, a multiscale U-shaped convolutional neural network building instance extraction framework with edge constraint (EMU-CNN) for high-spatial-resolution remote sensing imagery is proposed. The proposed framework consists of three components: 1) a multiscale fusion U-shaped network (MFUN); 2) a region proposal network (RPN); and 3) an edge-constrained multitask network (ECMN). First, in the proposed method, the MFUN includes three parallel branches to learn multiple building features with different scales. The RPN then detects the positions of the building instances, even for buildings that are connected with each other. Moreover, according to the instance positions, the ECMN is proposed to extract a precise mask and suppress over tting. The experiments conducted on a self-annotated data set and two public data sets (the ISPRS Vaihingen semantic labeling contest data set and the WHU aerial image data set) show that the EMU-CNN method can achieve excellent performance and shows great robustness at different scales.**

Index Terms **Deep learning, edge constraint loss (ECL), high-resolution imagery, instance segmentation, multiscale building extraction.**

Manuscript received February 5, 2020; revised May 25, 2020 and August 4, 2020; accepted August 28, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0504202, in part by the China Aerospace Science and Industry Joint Foundation and the Wuhan Applied Fundamental Frontier Project under Grant 2020010601012166, and in part by the National Natural Science Foundation of China under Grant 41771385 and Grant 41801267. *(Corresponding authors: Ailong Ma and Dingyuan Chen.)*

Yuanyuan Liu, Fang Fang, and Kai Xu are with the College of Information Engineering, China University of Geosciences, Wuhan 430074, China (e-mail: liuyy@cug.edu.cn; ffang1014@163.com; xukai_cug@163.com).

Dingyuan Chen, Ailong Ma, and Yanfei Zhong are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: dingyuanchen_2018@163.com; maailong007@whu.edu.cn; zhongyanfei@whu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2020.3022410

I. Introduction

BUILDING extraction based on high-resolution remote sensing imagery is important for urban management planning, land-cover change detection, and cadastration. The process of the building extraction is aimed at extracting a building footprint using pixel- or object-based algorithms [1]. Among the building extraction methods, the mainstream methods are the traditional handcrafted feature-based methods [2]–[5], including the edge detection methods, the Hough transform-based methods, and the object-oriented methods. These methods perform well, but the handcrafted features can only process low-level or midlevel information, resulting in a poor generalization ability. More recently, deep learning has been proposed to automatically learn abstract features and has been shown to be a state-of-the-art approach. As such, a number of deep learning-based methods [6]–[16] have been developed for building extraction.

For most deep learning-based methods, the mainstream methods are the multiscale building extraction methods. They can be divided into three categories: image pyramid-based methods [6], combined multiscale images, and parallel deep networks to extract multiscale buildings, but parallel modules increase the number of parameters in the network that leads to a longer training time; filter pyramid-based methods [13] utilized multiple filters to aggregate multiscale features, which is a simple and flexible approach. However, the features derived from this method are similar; feature pyramid-based methods [10] cascaded multilayer features to extract multiscale buildings. Although these methods all take into account the scale-variance problem, they extract imprecise building boundaries. As such, other mainstream methods are the precise building boundary extraction methods. Knowledge-based methods [17] utilized geometric information to extract buildings, but this method is reliant on additional information. Postprocessing-based methods [18] adopted a boundary improvement module to refine the building boundaries.

Although the abovementioned methods perform well, they cannot distinguish the connected buildings. The instance segmentation method [19] has recently been proposed to extract instance masks. As such, connected objects can be distinguished by making use of each object boundary. There are two different types of instance segmentation methods:

1) proposal-based methods and 2) proposal-free methods. The proposal-based methods build a multistage pipeline to simultaneously extract objects and produce a classification map. For example, Dai *et al.* [19] won the 2015 MS-COCO instance segmentation challenge by adopting a multitask cascade framework, but they did not consider the occlusion problem. Li *et al.* [20] reduced the impact of occlusion by other instances but obtained a blurry mask; He *et al.* [21] simply added a new branch for a semantic mask in Faster R-CNN [22], obtaining a more precise mask. More recently, a new approach called the path aggregation network for instance segmentation [23] modifies the feature pyramid network (FPN) [24] architecture to achieve state-of-the-art performance. Other proposal-free methods [25], [26] transform an image into an instance mask through pixel embedding, which can successfully distinguish connected instances, but not in an end-to-end trainable model. Bai and Urtasun [27] proposed deep watershed transform for instance segmentation, but the method is unable to extract object instances bisected by occlusions; Tighe *et al.* [28], Yang *et al.* [29], Zhang *et al.* [30], and Mou and Zhu [31] utilized the depth information to improve the performance of instance extraction.

Recently, the instance segmentation methods [16], [32] are adopted to extract buildings, which can extract each building footprint and distinguish connected buildings. However, these methods do not consider the problems of the scale-variance and the blurry boundaries in building extraction.

Hence, in this article, to address these limitations, we propose the novel EMU-CNN method to obtain precise building masks in remote sensing images. The multiscale features, which are derived from three parallel branches, are fused to detect each building position and extract its precise mask (see Fig. 1).

The main contributions of this article are as follows.

1) The end-to-end trainable EMU-CNN method is proposed for building instance extraction with three components: a) a multiscale fusion U-shaped network (MFUN); b) a region proposal network (RPN); and c) an edge-constrained multitask network (ECMN). The evaluation of three typical building data sets confirms the advantages of EMU-CNN over the other state-of-the-art methods.
2) The MFUN uses three parallel sub-CNNs to learn multiscale building features, which can suppress the multiscale variations, especially very small building objects.
3) The ECMN is proposed to extract a more precise building mask for each building instance, even when the buildings are connected with each other. The edge constraint loss (ECL) in the ECMN can accelerate the network convergence and avoid overfitting.
4) A new building data set is labeled for building instance segmentation evaluation and analysis, which we refer to as the "self-annotated building instance segmentation data set."

II. Related Work

In this section, we discuss the methods related to instance segmentation and semantic segmentation. These two kinds of methods segment images in different ways.

Fig. 1. Visualization examples for the proposed EMU-CNN. (a) Original input image. (b) Multiscale feature maps. (Left to right) "2X" branch, "1X" branch, and "0.5X" branch. (c) Edge maps extracted by the Sobel filters, left to right: X-dimension, Y-dimension. (d) Building instance results.

Instance segmentation can be divided into two categories: proposal-based methods and proposal-free methods. In the following, the main proposal-based methods are introduced in detail, which were adopted as the comparison methods in the experiments conducted in this study. The proposal-based methods can be regarded as the multitask frameworks, which detect objects and simultaneously acquire pixelwise masks. In the following, as the two most advanced methods [20], [21], fully convolutional instance-aware semantic segmentation (FCIS) and Mask R-CNN are introduced in detail.

1) FCIS consists of three steps: a) the feature extraction architecture; b) the RPN; and c) an additional convolutional layer for segmentation. First, the feature extraction architecture is adopted to extract abstract features. The features are then fed into the RPN to propose candidate regions with different scales and aspect ratios. For efficient computation, RoIPooling is adopted to regularize the feature maps. Finally, a convolutional layer is utilized to achieve instance segmentation by using position-sensitive inside/outside score maps, which can reduce the impact of occlusion by other instances.

   FCIS considers the relationship between instance and background but does obtain blurry boundaries for very small scale objects. By contrast, EMU-CNN obtains a more precise mask by adopting the ECMN.
2) Mask R-CNN consists of an FPN [24] for feature extraction, an RPN for precise region proposal, and a semantic mask branch. First, the imagery is fed into the FPN. The FPN combines top–down and bottom–up

Fig. 2. Overview of the proposed framework, including (Left) MFUN, (Middle) RPN, and (Right) ECMN. The MFUN fuses the features with different scales to overcome the problem of scale variance in a single remote sensing image. The RPN utilizes the attention mechanism to extract the building instances. The ECMN adopts the ECL to segment a precise mask. Finally, EMU-CNN outputs a mask corresponding to the different building instances.

layers to construct a U-shaped architecture. It integrates the multistage features with lateral connections and adds an alternative classifier after each stage. The RPN is then adopted for the region proposal. Different from FCIS, RoIAlign is used to replace RoIPooling. There are four sampling points in each bin, and RoIAlign computes the value of each sampling point by bilinear interpolation on the feature map. To obtain precise locations, no quantization is performed on any computation. Finally, a semantic mask branch is added after the region proposal, which adopts a deconvolution layer to remap back to the original size.

Mask R-CNN can obtain a more precise instance mask and solve the scale-variance problem, to a certain extent. However, its performance is far from adequate for building instance extraction in remote sensing imagery. Thus, the proposed method extracts features from different-resolution images, as well as different stages to solve the scale-variance problem, and utilizes the building boundary information to obtain a more precise mask.

The semantic segmentation model consists of an encoder, a decoder, and a prediction head, and it classifies each pixel into a certain class.

The fully convolutional network (FCN) [33] is introduced as an example. At the encoder stage, FCN utilizes VGGNet [34] as an encoder to extract semantic information. At the decoder stage, a deconvolution layer is adopted to bilinearly upsample the encoder's outputs to pixelwise outputs. After this, the semantic information from the deep layer and the appearance information from the shallow layer are combined by a summation operation. At the prediction head, during testing, SoftMax is adopted as a linear classifier to determine each pixel's class. During training, the logistic loss is calculated on each pixel and summed. A final loss is then used for backward and parameter updating.

However, semantic segmentation methods cannot extract building instances. In addition, both the instance segmentation and semantic segmentation methods cannot solve the long-standing problems of scale variance and blurry boundaries.

III. Multiscale U-Shaped CNN Building Instance Extraction Framework With Edge Constraint

In this work, the EMU-CNN instance segmentation approach is proposed to extract a building mask from high-resolution remote sensing imagery. The proposed method is an end-to-end trainable approach, with convolutional features shared in both the building detection and semantic segmentation. The network architecture of EMU-CNN consists of three components, i.e., MFUN, RPN, and ECMN, as illustrated in Fig. 2. First, the remote sensing images are fed into the MFUN for multiscale feature learning. The MFUN includes three parallel pretrained ResNet sub-CNNs, which are followed by a fusion operation and a U-shaped deconvolution network to learn building features with different scales. Second, RPN is adopted to extract building instance positions. The RPN introduces an attention mechanism to extract building bounding boxes (bboxes), which can eliminate the influence of occlusion between overlapping buildings. Finally, according to the building positions, the ECMN is proposed to extract a precise mask and suppress overfitting through the ECL.

A. MFUN for Building Feature Fusion

In order to learn robust features with various scales, each input image is preprocessed into three streams of different sizes: a $2\times$ stream (by $2\times$ interpolation), a $1\times$ stream (original size), and a $0.5\times$ stream (by downsampling). Instead of the "one-size-fits-all" feature extractor in the traditional CNN, the MFUN trains three parallel feature branches tuned for different-scale streams. Especially, the MFUN architecture consists of three parallel feature branches, two fusion operators, and four U-shaped deconvolution blocks, as shown in Fig. 3. Each branch is composed of the input stream and a pretrained ResNet model. The first branch includes the $2\times$ stream and a three-block ResNet model, the second branch includes the $1\times$ stream and a two-block ResNet model, and the

Fig. 3. Architecture of the MFUN. The MFUN consists of three parallel feature branches. The first branch includes the 2× stream and a three-block ResNet model, the second branch includes the 1× stream and a two-block ResNet model, and the third branch includes the 0.5× stream and a one-block ResNet model. The two fusion operators integrate the multiscale convolutional features in a hierarchical way. After this, four U-shaped deconvolution blocks are used to extract the fused features with different resolutions (which is simply depicted by the parabola), followed by multiple classifiers.

third branch includes the 0.5× stream and a one-block ResNet model. The two fusion operations (generated by summation) then jointly integrate the multiscale feature maps from the three branches. To improve both the efficiency and accuracy, the dimensions of "Fusion 1" and "Fusion 2" are modified to 64 and 256, respectively. The feature maps then go through a chained residual pooling (CRP) block (as defined in [35]) to capture the contextual information. Finally, the output is fed into ResNet Block-4 and Block-5, parameterized by the pretrained ResNet model.

After ResNet Block-5, four U-shaped deconvolution blocks (UDNs) are used for discriminative feature extraction via integrating the midlevel and high-level representations. As shown in Fig. 3, each UDN block consists of an upsampling and deconvolution operation (followed by a batch-normalization layer and a nonlinear activation layer, ReLU). Each block is followed by an auxiliary classifier. In the process of forward propagation, all the outputs of the classifiers are integrated, while, in the backward propagation, a total loss is calculated to contribute to each classifier.

Note that the multiscale inputs may result in a high computational cost. To address this issue, the residual convolution unit (RCU) block (as defined in [35]) is modified by adding a 1×1 dimension reduction layer and a 3×3 convolutional layer, which is followed by the fusion operator. Considering the problem of representational bottlenecks, the 5×5 max-pooling layer in the CRP block is followed by a 1×1 convolutional layer. The current model results in a speed increase of nearly 40% on the original basis, as well as avoiding overfitting, due to the reduced parameters.

B. RPN for Building Instance Detection

After obtaining the multiscale building features, the RPN is used to detect single building instance positions, which is an approach that was first proposed by Ren *et al.* [22]. It is worth noting that the multiscale inputs are fed into several parallel RPNs, and then, multiple proposals are aggregated together. In order to explain the structure of the RPN more clearly, only one branch is introduced.

Fig. 4. Architecture of the RPN. The RPN consists of an intermediate convolutional layer and two sibling convolutional layers, which are adopted for the classification and bbox regression. Anchor: box with different scales and aspect ratios.

As demonstrated in Fig. 4, the RPN consists of an intermediate convolutional layer, two sibling convolutional layers, and an anchor. Each pixel of the aforementioned intermediate convolutional layer is projected back to multiple candidate regions of the original image. This is realized by introducing an anchor, i.e., a box with many kinds of scales and aspect

Fig. 5. Architecture of the ECMN. In this architecture, a hybrid loss function is introduced into multitask learning to obtain a more precise mask. First, the Sobel filters are adopted in both the predicted mask and ground truth. The ECL is then computed over these two edge maps. Finally, a hybrid multitask loss function considering classification, bbox regression, semantic segmentation, and ECL is adopted to optimize the whole architecture.

ratios, the number of which is k. By default, $k = 12$, i.e., there are 12 anchors with four scales (4, 8, 16, and 32) and three aspect ratios (1:1, 1:2, and 2:1). These k candidate regions are then fed into two sibling convolutional layers (for classification and bbox regression). For the classification branch, the 2k SoftMax output is calculated to divide the candidate region into positive (object) and negative (no object) samples that are normalized by the minibatch size number (i.e., $N_{\text{cls}} = 256$). For the bbox regression branch, the bbox is refined by smooth L1 loss, which is based on the building's central coordinates. This branch is utilized to refine the building's position. The box regression loss is then normalized by the number of anchor locations, i.e., there are $W \bullet H \bullet k$ anchors. W and H, respectively, represent the width and height of the feature map. Finally, the bbox corresponding to a single building instance is detected.

C. ECMN for Precise Mask Extraction

According to the detected instance positions, the ECMN is proposed to extract each precise mask and suppress overfitting by using a hybrid loss function, as shown in Fig. 5. The multiple losses, which consist of the classification loss, the bbox loss, the segmentation loss, and the ECL, are calculated based on the region of interest (RoI, a candidate region that may be an object). During training, an RoI is positive if its intersection over union (IoU) is related to the nearest ground-truth object that is larger than 0.5; otherwise, it is regarded as negative. The hybrid loss function is defined as follows:

$$L = L_{\text{cls}} + {}_1 p L_{\text{bbox}} + {}_2 p L_{\text{seg}} + {}_3 p L_{\text{edge}} \tag{1}$$

where $p \in \{0, 1\}$ corresponds to the negative and positive RoI. L_{cls}, L_{bbox}, L_{seg}, and L_{edge} represent the four terms of object classification loss, bbox regression loss, segmentation loss, and ECL, respectively. Among the four terms, L_{cls} and L_{seg} are computed by a SoftMax function. L_{bbox} is computed utilizing smooth L_1 loss; ${}_1$, ${}_2$, and ${}_3$ are weighting factors used to balance the contributions of the four terms. By default, ${}_1 = 1$, ${}_2 = 1$, and ${}_3 = 1$, which means that these four terms are of equal importance. The first two values are the same as those defined in [21]. ${}_3$ is identical to the term defined in [36]. In the following, these loss terms are introduced in detail.

1) *Classification Loss* $\boldsymbol{L_{\text{cls}}}$*:* The negative log of the SoftMax output. This term is based on the pixels within the RoI. x_i represents the predicted value of the ith object, and y_i represents the ground truth of the ith object. The classification loss is defined as

$$L_{\text{cls}}(x) = \sum_i y_i \log \frac{exp(x_i)}{\sum_j \exp(x_j)} . \tag{2}$$

2) *Bbox Regression Loss* $\boldsymbol{L_{\text{bbox}}}$*:* Smooth L1 loss. This term is based on the pixels within the RoI. t_i predicted the ith box's center coordinates and its width and height (x, y, w, h). t_i predicts the ground-truth center coordinates and its width and height. The bbox regression loss is defined as

$$L_{\text{bbox}}(t_i, t_i) = \begin{cases} 0.5\,(t_i - t_i)^2, & \text{if } |t_i - t_i| < 1 \\ t_i - t_i - 0.5, & \text{otherwise.} \end{cases} \tag{3}$$

3) *Segmentation Loss* $\boldsymbol{L_{\text{seg}}}$*:* Negative log of the SoftMax output, based on the pixels within the RoI. x_p represents the predicted value of the pth pixel, and y represents the ground truth. The segmentation loss is defined as

$$L_{\text{seg}}(x_p) = \sum_p y_p \log \frac{\exp(x_p)}{\sum_j \exp(x_j)} . \tag{4}$$

4) *ECL:* This term is proposed to constrain the differences between the boundaries of the predicted instance mask and the ground truth. First, the Sobel filters (first-order edge filter) in both horizontal and vertical directions are adopted on both the predicted mask and the ground truth. It is worth noting that this pair of filters are adopted on both the foreground (building region) and background (nonbuilding region), which differs from the approach adopted in [36], where only edge detection on the foreground is considered. Edge detection on both the background and foreground can eliminate false alarms, which is useful for building extraction. After this, the score map is computed by the ECL function. y represents the predicted value, and y represents the ground truth. Mathematically speaking, the ECL can be defined as

$$L(y, y)) = \frac{(|y - y|)^2}{2}. \tag{5}$$

D. Discussion

In this section, we give a brief discussion of the differences between the proposed method and some of the existing end-to-end methods.

1) Faster R-CNN [22]: Both EMU-CNN and Faster R-CNN adopt RPN to extract the object instances. However, these methods have three differences.

1) EMU-CNN can extract each building's mask, while Faster R-CNN can only locate each building's position using a bbox.

(a) (b)

Fig. 6. Distribution of the building scales in (a) ISPRS Vaihingen semantic labeling contest data set and (b) self-annotated data set.

2) EMU-CNN utilizes the MFUN to extract multiresolution and multiscale features, which is more suitable for multiscale building extraction, while Faster R-CNN extracts single-scale features from an image, with a fixed size.
3) EMU-CNN utilizes both ECL and segmentation loss. ECL and segmentation loss can acquire precise boundaries, while Faster R-CNN can only extract the object's box.

2) Mask R-CNN [21]: The proposed method differs from Mask R-CNN in two aspects.

1) EMU-CNN adopts the MFUN to extract multiresolution/multiscale features that are specially designed for remote sensing imagery, while Mask R-CNN does not take the characteristics of remote sensing imagery into account.
2) EMU-CNN introduces geometric information into the deep learning network to enhance the performance of the building extraction, while Mask R-CNN does not consider the building's geometric information.

3) Method Proposed in [36]: Both EMU-CNN and the method proposed in [36] can segment each building footprint. However, they have two differences.

1) EMU-CNN adopts the MFUN to extract multiresolution/multiscale features and is, thus, more robust for remote sensing imagery.
2) EMU-CNN utilizes ECL on both the foreground (building) and background (nonbuilding), which is useful for eliminating false alarms, while the method in [36] does not consider the background.

IV. Experiments

A. Data Sets

In the experiments, the proposed approach was applied to three challenging building segmentation data sets, i.e., the ISPRS Vaihingen semantic labeling contest data set, the WHU aerial image data set, and a self-annotated building instance segmentation data set. Fig. 6 shows the distributions of the building scales in the ISPRS data set and the self-annotated data set, where it can be seen that most of the object scales of the data sets are small. Compared with the ISPRS data set, the object scales of the self-annotated data set are smaller.

1) ISPRS Vaihingen Semantic Labeling Contest Data Set: The ISPRS Vaihingen semantic labeling contest data set [37] consists of near-infrared, red, and green orthorectified imagery. These three channels were transformed into RGB color imagery during training. The data set is made up of 33 large image patches of 2500 × 2500 pixels, the ground sample distance (GSD) of which is 9 cm. Sixteen tiles were labeled with the ground truth and were randomly cropped into fixed-size images of 600 × 600 pixels. All the cropped images were then sliced into training, evaluation, and test data sets, with the proportion of 5:3:2. Samples of the cropped images are shown in Fig. 7.

Fig. 7. Samples of the ISPRS Vaihingen semantic labeling contest data set.

2) WHU Aerial Image Data Set: The WHU data set [32] consists of an aerial image data set and two satellite image data sets, among which the aerial image data set was utilized to evaluate the proposed method's generalization ability. The spatial resolution of the WHU aerial image data set is 0.3 m, and it is similar to the self-annotated building instance segmentation data set (introduced in Section IV-A3). The WHU aerial image data set covers 450 km^2 in Christchurch, New Zealand. In this data set, there are 2416 tile images with 512 × 512 pixels in the test set (a total of 42 000 buildings). This aerial image data set was utilized to evaluate EMU-CNN's generalization ability in different lighting and atmospheric conditions, sensor qualities, scales, and building architectures, as demonstrated in Fig. 8.

3) Self-Annotated Building Instance Segmentation Data Set: This data set was annotated and built independently by the authors. The remote sensing images of this building instance data set were collected from the UC Merced data set, the Aerial Image Data set (AID) data set, and some other remote sensing images. The original images of the UC Merced data set [38] were acquired from the United States Geological Survey (USGS) National Map program, with a spatial resolution of one foot (nearly 0.3048 m), cropped into fixed sizes of 256 × 256 pixels. The AID data set [39] is a large-scale aerial imagery data set, the images of which were downloaded from Google Earth imagery. It is worth noting that the other remote sensing images in the building instance data set are all high-resolution remote sensing images. These three data sets were combined and cropped into tiles of 256 × 256 pixels. The self-annotated building instance segmentation data set

Fig. 8. Samples of the WHU aerial image data set.

Fig. 9. Samples of the self-annotated building instance segmentation data set.

consists of 1488 tiles with 256 × 256 pixels and a total of 13 108 buildings. For annotation, the building instance mask was labeled by class ID and instance ID (the order number in an image). Some typical buildings were chosen as examples and annotated by remote sensing experts, and the boundaries of the buildings were marked. The remaining buildings were then annotated according to the examples. Finally, all the annotated labels were checked again by the remote sensing experts. Buildings under trees were annotated by directly connecting a straight line. The self-annotated data set consists of multiresolution, multisensor images that were captured from different areas in different weather conditions. As such, the data set is challenging, as well as valuable. Some samples from the self-annotated building instance segmentation data set are shown in Fig. 9.

B. Experimental Setup and Results

To increase the diversity, rotate, flip, and brightness operations were adopted on each remote sensing image, which enlarged the volume of the data sets by 12 times.

TABLE I

EXPERIMENTS ON THE SELF-ANNOTATED DATA SET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Model	IoU=0.5			IoU=0.7		
	mAP	pre	rec	mAP	pre	rec
Mask R-CNN [21]	86.8	94.2	88	71	80.2	75.7
Mask+edge [36]	86.42	93.8	88.3	69.49	78.8	74.9
FCIS [20]	79.63	82.47	80.12	67.43	70.2	68.76
Modified DeepLab v3+	55.64	58.93	56.0	33.96	37.04	34.86
EMU-CNN(our)	**89.2**	**95.0**	**90.0**	**74.1**	**81.7**	**78.2**

The proposed approach was compared with the state-of-the-art methods of FCIS [20], Mask R-CNN [21], Mask R-CNN + edge filter [36], and a popular proposal-free method [40], on all three data sets. FCIS was proposed by Li *et al.* [20] and is the first end-to-end fully convolutional instance segmentation network. FCIS is based on R-FCN [41] and Instance-FCN [19]. Mask R-CNN has achieved state-of-the-art performances in instance segmentation. It simply adds a branch of semantic segmentation to Faster R-CNN [22] and changes RoIPooling into RoIAlign. Mask R-CNN + edge filter [36] adds an edge filter to Mask R-CNN to extract a precise mask. The proposal-free method consists of DeepLab v3+ [40] (for semantic segmentation) and connected component processing (for postprocessing). The latter architecture is used to transform the semantic mask into an instance mask.

The metric of mean average precision (mAP) based on each pixel is used to assess the quantitative performance. The mAP represents the relationship between the precision and recall, indicating the arithmetic mean of C + 1 categories. The average precision of each category can be calculated by integrating the area under the precision–recall (P-R) curve. Precision and recall can be defined as follows:

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, \quad \text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{6}$$

where TP_i is the number of true positives of the ith class. FP_i and FN_i represent the number of false positives and false negatives, respectively. These metrics are calculated by using the IoU threshold between the predicted score maps and the ground truth, based on pixels.

The proposed model was trained and fine-tuned based on the MXNet [19] platform. The shared convolutional layers were initialized with the released pretrained ResNet-50 model. The initial learning rate was set to 0.004 at the beginning, with a weight decay rate of 0.0001 and a momentum value of 0.9. Stochastic gradient descent (SGD) optimization was used, and the model was trained on a single Tesla K40C or Titan X GPU. The source code of the proposed model will be released on GitHub in the future.

1) Evaluation With the Self-Annotated Building Instance Segmentation Data Set: Table I lists the results of the experiment with the proposed EMU-CNN, FCIS [20], Mask R-CNN [21], Mask R-CNN + edge filter [36], and modified DeepLab v3+ [40]. "mAP," "pre," and "rec" in the table represent mAP, precision, and recall, respectively. As can be seen, the mAP of the proposed method exceeds that of Mask R-CNN by 2.4% and 3.1% with IoU = 0.5 and IoU = 0.7,

Fig. 10. mAP versus IoU over the self-annotated data set.

TABLE II
EXPERIMENT ON THE ISPRS VAIHINGEN SEMANTIC LABELING CONTEST DATA SET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Model	mAP@0.3	mAP@0.4	mAP@0.5
Mask R-CNN [21]	97.15	96.78	96.46
Mask+edge [36]	96.12	95.60	94.87
FCIS [20]	90.77	90.77	90.76
Modified DeepLab v3+	88.7	87.36	85.7
EMU-CNN(ours)	**97.41**	**97.04**	**96.63**

respectively. By comparing the precision and recall among the different methods, it can be seen that the proposed EMU-CNN can distinguish positive and negative samples more effectively. Moreover, the relationship between the mAP and IoU is depicted in Fig. 10. As shown in this figure, the proposed method performs better than the state-of-art Mask R-CNN model. The proposal-free method performs poorly since it cannot distinguish connected building instances well in this data set.

2) Evaluation With the ISPRS Vaihingen Semantic Labeling Contest Data Set: The experiment with the ISPRS Vaihingen semantic labeling contest data set shows that EMU-CNN is also robust to large-scale buildings. As shown in Table II, the mAP of the proposed method exceeds that of Mask R-CNN by 0.2% when the IoU is 0.5. However, when the IoU is larger, Mask R-CNN performs slightly better. This can be explained by the increased number of parameters in EMU-CNN causing an overfitting problem.

To analyze the robustness of the proposed EMU-CNN for different scales, the mAP versus IoU curve is shown in Fig. 11. As shown in the figure, when the IoU is smaller than 0.6, EMU-CNN performs better than Mask R-CNN, and vice versa. This shows that the EMU-CNN method is robust for use with smaller scale buildings.

3) Evaluation on Cross-Data Sets: To verify the generalization ability of the proposed method, Table III compares the performance of EMU-CNN systems with different models and training data. Trained self-annotated models were used to evaluate the very challenging WHU aerial image data set

Fig. 11. mAP versus IoU over the ISPRS Vaihingen semantic labeling contest data set.

TABLE III
CROSS-DATA SET EXPERIMENT ON THE TRAINED SELF-ANNOTATED MODEL. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Model	IoU=0.5			IoU=0.7		
	mAP	pre	rec	mAP	pre	rec
Mask R-CNN [21]	65.0	**87.8**	68.1	47.6	**69.5**	55.0
EMU-CNN	**70.4**	87.0	**73.2**	**50.0**	65.8	**57.0**

for robustness testing. As shown in Table III, EMU-CNN performs better than Mask R-CNN in generalization ability, and the mAP of the proposed method exceeds that of Mask R-CNN by 5.4% and 2.4% with an IoU of 0.5 and 0.7, respectively. The analysis of both the precision and recall confirms that EMU-CNN performs better as it recalls more positive instances. In addition, for the trained ISPRS models using EMU-CNN and Mask R-CNN, they are all invalid on the WHU aerial image data set because of the resolution difference. Thus, the results are not exhibited.

Some visualization results for the WHU aerial image data set are shown in Fig. 12. The first and third rows represent the results of EMU-CNN without ECL (i.e., with only the MFUN architecture), while the second and fourth rows show the results of Mask R-CNN. Typically, the results of Mask R-CNN include large regions that are not buildings. Overall, it can be seen that the proposed method can effectively extract buildings with different scales.

C. Ablation Study and Discussion

1) Quantitative Analysis of the Different Branches of the MFUN: To demonstrate the effectiveness of each branch in the MFUN, an ablation experiment was conducted. This was done by setting a certain branch's weight factor to 0 (which means that this branch's feature maps were not concatenated into the fused output). The results are listed in Table IV.

All the methods were evaluated on the self-annotated data set. As is shown, the first row shows the results of the proposed model, which performs the best on the self-annotated data set. Interestingly, the model with the "2×" branch is more effective.

Fig. 12. Visualization results of the evaluation on the cross-data sets: trained on the self-annotated data set, tested on the WHU aerial image data set. (a) First row: EMU-CNN and second row: Mask R-CNN. (b) First row: EMU-CNN and second row: Mask R-CNN.

Apart from this, by comparing the results of the "2 × +1×" model and the "2 × +0.5×" model, it is demonstrated that the "1×" branch is more important than the "0.5×" branch. By visualizing the final mask of the "0.5×" model, it is found that there are no buildings extracted, due to the coarse feature map. The "0.5×" model can be regarded as a building detector, but not as a mask extractor. Hence, it is unsuitable for buildings with a small size, as in the self-annotated data set. This also leads to the combination of "1× + 0.5×" performing worse than the "1×" branch.

We also evaluated the fusion operations in the MFUN. The "Fusion 2" model in this table represents only utilizing the feature map from the Fusion 2 operation. This experiment shows that the Fusion 2 operation can extract more important features than the Fusion 1 operation.

This experiment reveals that the MFUN can learn effective multiscale building features due to the three parallel convolutional branches and the two hierarchical fusion operations.

2) Analysis of the Multiscale Feature Maps in the MFUN: To explain why the MFUN is effective, feature map outputs from streams with different resolutions are depicted in Fig. 13(a) and (b).

Fig. 13(a) represents feature maps fed into the first fusion operation, while Fig. 13(b) represents the second fusion operation. As indicated, local features can be extracted from the higher resolution branch (the first row exhibits more details), while global features can be extracted from the lower resolution branch (the second and third rows reveal the spatial relationship between the road in the middle and the buildings). Visualization of the feature maps from the fusion operation indicates that the second fusion operation can extract more complete information, while the first fusion operation tends to obtain local information [see Fig. 13(c)]. Features from the UDNs are exhibited in Fig. 13(d). "Deconv_1" (the first row) represents the features from the final deconvolution layer. Clearly, these features focus more on spatial information

TABLE IV

ABLATION EXPERIMENT FOR THE EFFECTS OF THE DIFFERENT BRANCHES. EACH COLUMN IN "MAP@IOU" REPRESENTS THE MAP UNDER DIFFERENT IOUS (I.E., 50%, 60%, AND 70%)

Model	mAP@IoU				
	50	60	70	80	90
2X+1X+0.5X	**89.2**	**84.7**	**74.1**	**46.1**	**4.1**
0.5X	0.1	0.0	0.0	0.0	0.0
1X	29.5	22	13.1	0.41	0.1
2X	86.3	80.8	68.2	38.5	2.4
2X+1X	89.0	84.1	73.2	44.5	3.5
2X+0.5X	87.1	82.3	70.8	41.6	2.8
1X+0.5X	27.8	20.1	11.01	3.25	0.08
Fusion 1	31.1	24.0	15.4	1.7	0.2
Fusion 2	88.9	84.4	74.0	46.0	4.0

(a)

(b)

(c)

(d)

Fig. 13. Visualization of the feature maps. (a) Input branch in Fusion 1 operation. (b) Input branch in Fusion 2 operation. (c) Outputs of Fusion operations. (d) Outputs of four deconvolution layers.

and tend to locate where buildings are located. By contrast, "Deconv_4" can extract more abstract features while focusing less on spatial information.

3) Analysis of the Instance Proposal in the RPN Based on Different CNNs: To evaluate the performance of the instance proposal based on the MFUN, an ablation experiment comparing the proposed MFUN and a popular CNN was conducted. The performance of the instance proposal is quite relevant to the building extraction since it determines whether the building instances are extracted properly. The experiment was implemented on the self-annotated data set (as shown in Table V). As indicated, the proposed method recalls more positive samples, especially when the IoU is higher. This can be explained by the fact that the proposed method extracts features with multiple scales and is suitable for small-scale building extraction.

Fig. 14. Total training loss versus iteration on Mask R-CNN and EMU-CNN without and with ECL.

TABLE V

EXPERIMENTS WITH REGARD TO THE PERFORMANCE OF INSTANCE PROPOSAL IN THE RPN BASED ON DIFFERENT CNNS. THE BEST PERFORMANCE IS HIGHLIGHTED IN BOLD

Model	recall@0.5	recall@0.6	recall@0.7	recall@0.8	recall@0.9
Mask R-CNN+ResNet-50	98.3	97.4	94.1	62.7	8.6
Mask R-CNN+MFUN	**98.5**	**97.8**	**95.7**	**80.3**	**24.4**

TABLE VI

ABLATION EXPERIMENT WITH THE SELF-ANNOTATED DATA SET

Model	IoU=0.5			IoU=0.7		
	mAP	pre	rec	mAP	pre	rec
Mask R-CNN	86.8	94.2	88	71	80.2	75.7
Mask+edge [36]	86.42	93.8	88.3	69.49	78.8	74.9
Mask R-CNN+ECL	**87.75**	**94.42**	**89.1**	**73.56**	**80.24**	**77.83**
EMU-CNN w/o ECL	87.1	93.3	88.1	70.7	78.2	75.1
EMU-CNN	**89.2**	**95.0**	**90.0**	**74.1**	**81.7**	**78.2**

4) Analysis of the ECL in the ECMN: In Table VI, an ablation experiment comparing EMU-CNN with (w) and without (w/o) ECL is shown. This illustrates that EMU-CNN with ECL obtains a better performance. As exhibited in the table, both the precision and the recall of EMU-CNN are higher than those of the compared method. This means that the building mask extracted by the proposed method is more precise. It is worth noting that the performance of the proposed ECL is better than that of the method proposed in [36], which indicates that both the foreground (building region) and background (nonbuilding region) are supervised by the edge filters.

Fig. 15. Visualization results of the proposed method (with only MFUN architecture) and Mask R-CNN. The first and third rows represent the results of EMU-CNN without ECL (with only the MFUN architecture), while the second and fourth rows exhibit the results of Mask R-CNN (the differences between the two methods are pointed out with the red arrows). (a) Sparse building. (b) Dense building.

As demonstrated, the ECL also helps the network to converge to a lower loss. Fig. 14 describes the training iteration of Mask R-CNN and EMU-CNN, with and without ECL. The total training loss consists of the object classification loss, the bbox regression loss, and the segmentation loss. The blue, green, and red lines represent the total loss of Mask R-CNN, EMU-CNN without ECL, and EMU-CNN with ECL, respectively. It can be seen that EMU-CNN both with and without the ECL results in a lower loss than Mask R-CNN. In addition, at the end of the training, EMU-CNN with ECL results in a lower loss than EMU-CNN without ECL. This proves that the ECMN speeds up the convergence of the whole model.

5) Building Extraction Examples and Analysis: To evaluate the proposed MFUN architecture intuitively, visualization of the predicted results for the self-annotated data set is provided in Fig. 15. The first and third rows represent the results of EMU-CNN without the ECL (i.e., with only the MFUN architecture), while the second and fourth rows exhibit the results of Mask R-CNN. By comparing the results, it can be observed that some nonbuilding regions that are wrongly extracted by Mask R-CNN are ignored by the MFUN. Some buildings with a very large or small scale are neglected by Mask R-CNN but are detected successfully by the MFUN. What is more, some cars with rectangular shapes are detected as buildings by Mask R-CNN but are successfully distinguished by the MFUN. The proposed MFUN network extracts features from different resolutions and fuses them, which extracts more local and global information and helps to detect instances at different scales.

Fig. 16 shows a visualization of the predicted results of EMU-CNN and Mask R-CNN. As shown in Fig. 16(a), the two rows represent the results of EMU-CNN and Mask R-CNN, respectively, in order, while, in Fig. 16(b), a certain region is magnified, and more details are exhibited. It can be observed

Fig. 16. Visualization of the ECL. (a) First row represents the results of EMU-CNN, while the second row represents the results of Mask R-CNN. To show the details, certain regions are magnified and exhibited in (b).

that, with the auxiliary of the ECL, the instances' boundaries are smoother and better fit the outlines of the buildings.

To evaluate the performance of EMU-CNN intuitively, Fig. 17 shows the predicted results of EMU-CNN with the three data sets.

6) Analysis of Building Extraction in Complex Scenarios: In this section, we describe the thorough evaluation of the proposed method conducted under different challenging circumstances. As pointed out by Khosravi *et al.* [1] and Khosravi and Momeni [42], building extraction in urban areas is a complex problem, in some cases, which can be summarized as follows: 1) when the shadows or vegetation are in the proximity of (or even intrusive of) the buildings; 2) when the buildings are diverse in terms of height and the images are oblique; 3) when there is low contrast (high similarity) between the buildings and nonbuilding regions; and 4) when there is an irregular alignment and building blocks are present. As shown in Fig. 18, the visualization results for some typical buildings in these complex scenarios are exhibited to evaluate the performances of the proposed method and Mask R-CNN. Overall, it can be seen that the proposed method performs better than Mask R-CNN in the abovementioned complex scenarios.

From the analysis of the visualization results, we can make the following conclusions.

1) As shown in Fig. 18(a), buildings can be distinguished from shadows and vegetation well by our method. This can be explained by the fact that the proposed MFUN can extract robust features and spectra, as well as texture information, which can be better utilized by the proposed EMU-CNN.
2) As shown in Fig. 18(b), the geometric information of the roof is regular. In this way, even if the side view of the buildings can be observed in addition to the building

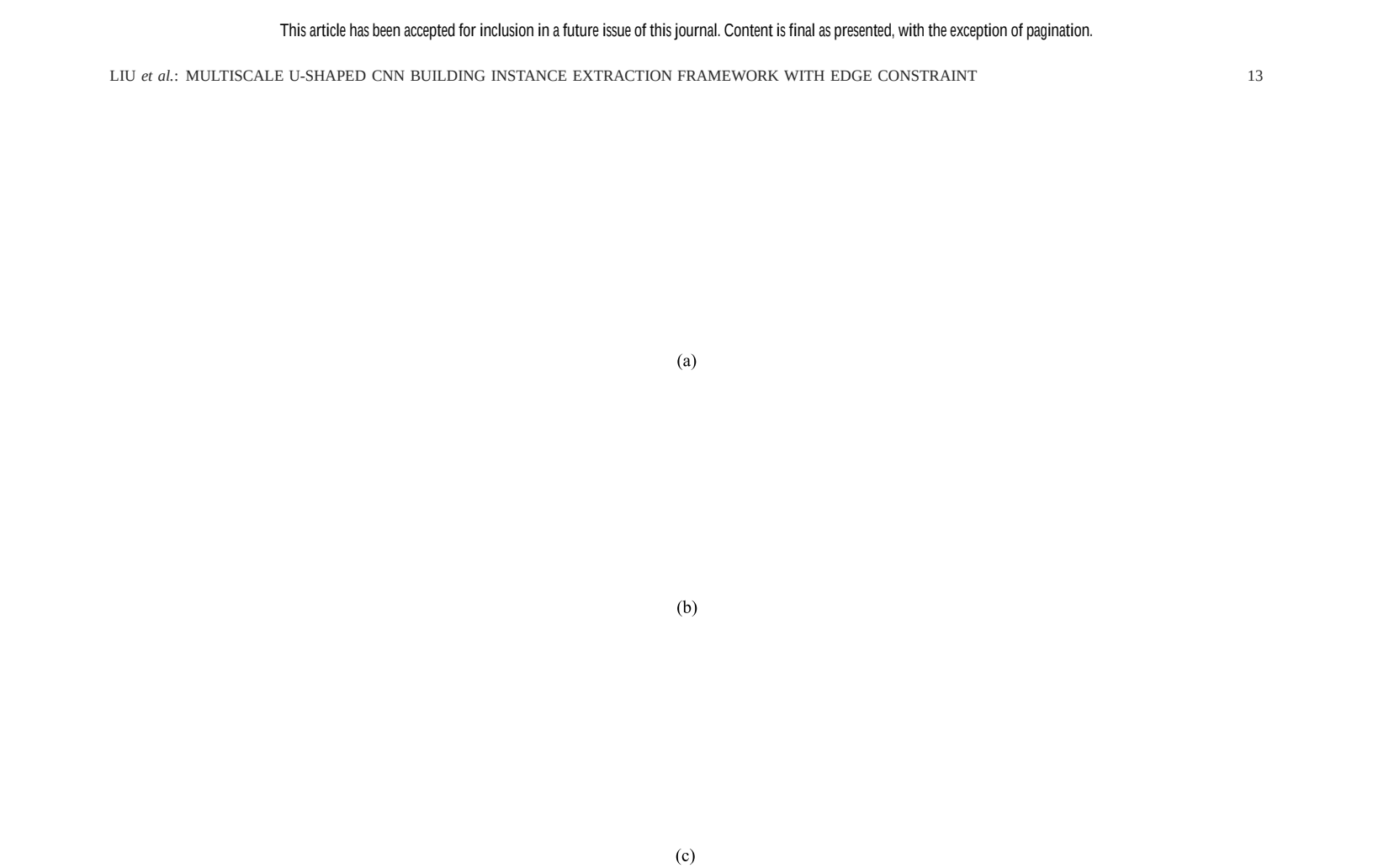

Fig. 17. Evaluation with (a) ISPRS Vaihingen semantic labeling contest data set, (b) WHU aerial image data set, and (c) self-annotated building instance segmentation data set.

Fig. 18. Visualization results for buildings in complex scenarios. (a) When the shadows or vegetation are in the proximity of the buildings. (b) When the buildings are diverse in terms of height and the images are oblique. (c) When there is low contrast (high similarity) between buildings and nonbuilding regions. (d) When there are an irregular alignment and building blocks. First column: original image with labels. Second column: visualization results of the proposed EMU-CNN. Third column: Mask R-CNN results.

roofs, the roofs of the buildings can be extracted well. This makes sense since only building roofs are annotated artificially, and the proposed EMU-CNN learns features extracted from the building roofs.

3) In Fig. 18(c), buildings with low contrast can also be extracted from nonbuilding regions by the proposed method. Due to the proposed ECMN, which introduces geometric information into the deep learning model, this