Building extraction from remote sensing images using deep residual U-Net

Haiying Wang and Fang Miao

Abstract
Building extraction is a fundamental area of research in the field of remote sensing. In this paper, we propose an efficient model called residual U-Net (RU-Net) to extract buildings. It combines the advantages of U-Net, residual learning, atrous spatial pyramid pooling, and focal loss. The U-Net model, based on modified residual learning, can reduce the parameters and degradation of the network; atrous spatial pyramid pooling can acquire multiscale features and context information of the sensing images; and focal loss can solve the problem of unbalanced categories in classification. We implemented it on the WHU aerial image dataset and the Inria aerial image labeling dataset. The results of RU-Net were compared with the results of U-Net, FastFCN, DeepLabV3+, Web-Net, and SegNet. Experimental results show that the proposed RU-Net is superior to the others in all aspects of the WHU dataset. For the Inria dataset, most evaluation metrics of RU-Net are better than the others, as well as the sharp, boundary, and multiscale information. Compared with FastFCN and DeepLabV3+, our method increases the efficiency by three to four times.

Introduction
Automatic detection and building extraction from remote-sensing images is important for population density estimation, urban planning, and the production and updatation of topographic maps. Building extraction has received considerable attention, but it is still a challenging task due to the noise, occlusion, and complexity of the background in original remote sensing images. In recent years, a variety of methods have been proposed to extract buildings from remote sensing images. Most of these methods focus on pixel (Sirmacek & Unsalan, 2008), spectrum (Zhang, 1999; Zhong et al., 2008), length, edge (Ferraioli, 2010; Li & Wu, 2008), shape (Dunaeva & Kornilov, 2017), texture (Awrangjeb et al., 2013; C.Y. He et al., 2004; Zhang, 1999), and shadow (Sirmacek & Unsalan, 2008; D.Y. Chen et al., 2014). Principal component analysis (PCA) was performed to effectively combine texture and structure information and extract urban building information directly from Landsat 7 ETM+ panchromatic data (C.Y. He et al., 2004). Linear discriminant analysis (LDA) color features and support vector machine (SVM) classifiers were used to identify the shadow patches, and the adaptive region-growing algorithm was used to build rough segmentation (D.Y. Chen et al., 2014). A local Gaussian Markov random field (GMRF) was proposed to detect the building edge (Ferraioli, 2010). In the process of texture filtering, the directional dependence of the co-occurrence matrix is utilized. It combines multispectral classification and texture filtering to realize building detection (Zhang, 1999). The pixels, spectrum, length, edge, shape, texture, and shadow of remote sensing images change according to season, illumination, atmospheric conditions, sensor quality, and scale. (Ji & Wei, 2019) pointed out such methods are processed by hand and depend on the experience of the specialist; they can only work with specific data. Currently, the resolution of remote sensing images has reached the decimeter level, and the structure, texture, and spectral information of buildings are becoming more refined. Meanwhile, the intraclass variance of very-high-resolution (VHR) images increases, while the interclass variance decreases, it is very difficult to manually design classification features (Pan et al., 2019; Xu et al., 2018). Therefore, traditional recognition methods based on manual annotation characteristics are not suitable for the extraction of buildings from VHR images (Chen et al., 2018; Huang et al., 2017; Ok et al., 2013; Senaras et al., 2013).

Traditional methods must undergo a two-step method for feature extraction and classification. In Step 1, the spatial and texture features of images are extracted from mathematical feature descriptors. In Step 2, according to the extracted features, each pixel is predicted by classifiers such as PCA (C.Y. He et al., 2004), SVM (D.Y. Chen et al., 2014), and GMRF (Ferraioli, 2010). Due to the complexity of building structures and the similarity with other categories (such as cement floor and roads), the prediction...
results depend on the artificial feature design and adjustment, which lead to bias and poor generalization. With the development of algorithms, computational capabilities, and the availability of big data, deep learning has attracted more and more attention in the field of remote sensing image processing.

The concept of deep neural networks (DNNs) was proposed in (Hinton et al., 2006). It refers to the machine learning process of obtaining the multi-layer network structure through some training methods based on the sample data (Bengio, 2009). The basic structure of DNN is consistent with that of neural network (Psaltis et al., 1988). After the initial low-level feature representation is gradually transformed into a high-level feature representation through multi-layer processing, complex learning tasks such as classification can be completed automatically with simple models (Buduma & Locascio, 2017). Thus, deep learning can be understood as feature learning or representation learning. DNNs include feed-forward deep networks, feedback deep networks, and bidirectional deep networks. The convolutional neural networks (CNNs) proposed by (LeCun et al., 1990, 1998) are typical feed-forward deep networks. A CNN is a trainable multi-layer network composed of multiple single-layer networks. Each single-layer network has three stages: convolution, nonlinear transformation, and down sampling (LeCun et al., 2010).

Deep learning methods can automatically learn effective classification features and perform more detailed feature mapping via sequential CNNs, and the results of introducing CNNs into remote-sensing image processing have indicated great practical potency in image retrieval, image classification, and object detection (Guo et al., 2017; Yuan et al., 2017). Different from the traditional methods, CNNs can automatically extract features and make classifications through sequential convolution with fully connected layers. Hence, the CNN method can be regarded as a one-step method combining feature extraction and classification in a single model. Because the feature information is extracted from it, CNN usually possesses good generalization ability. Patch-based CNN models have made remarkable achievements in building extraction, but because they rely only on a small patch around the target to predict labels and ignore the internal relationship between patches, they cannot guarantee the spatial continuity and integrity of building structures (Lam et al., 2015; Vakalopoulou et al., 2015). In addition, patch-based CNNs are time-consuming. To overcome the shortcomings of them (Long et al., 2015) proposed the use of fully convolutional networks (FCNs), in which the fully connected layer in traditional CNNs is replaced by a convolutional layer, and an upsampling layer is incorporated into the model. FCNs have become an effective technique for semantic segmentation, like DilatedFCN and Encoder-Decoder, and have multiple variants including SegNet (Badrinarayanan et al., 2017), DeconvNet (Noh et al., 2015), and U-Net (Ronneberger et al., 2015).

There are two issues in the semantic segmentation of buildings. One is to get the building boundary as accurate as possible. The other is to deal with buildings with different morphological and scale characteristics. To further enhance the multiscale feature representation of the model G. M. Wu et al., (2018) used multi-constraint FCNs to optimize the parameters of the intermediate layer and enhance the multiscale feature representation of the model. (Zhou et al., 2019) used Mask R-CNN to detect buildings of different scales and obtain better results for the edge region in building segmentation. (Kang et al., 2019) proposed EU-Net, which adopts dense spatial pyramid pooling to obtain better details and various scales of building structures. (P. H. Liu et al., 2019) proposed SRI-Net, which uses a spatial residual inception module in FCN to accurately detect large buildings. (Shrestha & Vanneschi, 2018) presented a FCN with conditional random fields to obtain higher accuracy. To identify building boundaries (Zhu et al., 2021) proposed an E-D-NET model, in which E-Net is used to obtain and preserve boundary information, and D-Net is used to improve the results of E-Net and get more precise details.

Different from the above methods, which use only RGB images, some studies have introduced additional geographic information (digital elevation model, digital surface model, etc.) to improve the extraction accuracy of building boundaries. (Gamal et al., 2020) proposed an automatic building segmentation method that uses LiDAR data directly based on DGCNN and Euclidean clustering. The accuracy of this method is relatively high only in urban areas because the height of buildings is higher than the surrounding vegetation. ENVINet 5 model, which was built in ENVI version 5.6 and the ENVI Deep Learning version 1.1, was established in (Liu & Wang, 2021) to train and segment buildings. This method is difficult to evaluate the geometric shape of buildings with pixels, and the results of ENVINet5 are required for post processing to solve this problem. (M. H. Liu et al., 2020) proposed an airborne LiDAR point-cloud building extraction method based on the combination of Point Cloud Library region growth segmentation and histogram. The method easily identifies non-buildings whose top is similar to the surface of buildings as buildings.

All the afore mentioned models have achieved great success in building extraction, but the boundary, the accuracy, and the multi-shape and scale prediction are
Encoder

The goal of the encoder unit is to complete feature extraction from the remote sensing images. Because the accuracy of remote sensing images has reached the decimeter level, a large building often covers a large number of slice units. Moreover, background information is very important in the process of building image extraction (Tan et al., 2016). To extract more information, we use the larger size of the input image. Larger size increases the training time (Kang et al., 2019); therefore, we propose an efficient encoder unit. While many researchers used three (J. B. Lin et al., 2019; Marcu et al., 2018) or four (Chen et al., 2018; Ruan et al., 2019) downsampling layers to obtain as much detail as possible, we used five downsampling units in our method. The pooling operation is replaced by a convolutional block to downsample the feature map in our encoder unit.

A deeper neural network can extract richer feature information but will result in gradient vanishing and exploding (Buduma & Locascio, 2017); meanwhile, degradation problems may also occur (He et al., 2016). Gradient vanishing and exploding can be partially reduced by adding a batch normalization (BN) layer (Ioffe & Szegedy, 2015). The degradation problem can be addressed using the residual neural network (He et al. 2016; Z. Zhang et al., 2018), which we integrated into our method. Residual neural networks are composed of multiple residual units and are used to complete identity mapping, as expressed in the following:

\[ y_l = h(x_l) + F(x_l, W_l), \]
\[ x_{l+1} = f(y_l) \]  

(1)

Where \( x_l \) and \( x_{l+1} \) represent the input and output of the \( l \)-th residual unit, respectively; \( h(x_l) \) represents the identity mapping function; \( W_l \) is the convolution

Methods

In this paper, a building extraction model based on U-Net architecture is designed, enabling very precise semantic segmentation results to be obtained with very few training images. The specific architecture of the proposed model is shown in Figure 1. The model is divided into three parts: the encoder, bridge, and decoder units. The encoder extracts low-level features, and the decoder extracts the corresponding high-level features. Several feature channels are added to create a path between the low-level features and the corresponding high-level features to facilitate backward propagation during training (Ronneberger et al., 2015). The residual network unit is adopted by each block of the encoder and decoder units, and the ASPP is adopted by the bridge unit.

![Figure 1. Architecture of the proposed residual U-Net.](image-url)
operation used to adjust the channel dimension of $x_l$; $F(\cdot)$ is the residual function; and $f(y_l)$ represents the activation function. This is a transformation of the full pre-activation proposed in He et al. (2016).

Incorporating the strengths of residual learning, our encoder has the benefits of facilitating training and decreasing degradation. The encoder consists of five units that use residual learning to facilitate the training of the network. As shown in Figure 2 (a), each unit consists of two convolution blocks and a residual mapping. Each convolution block includes a BN layer, a ReLU activation layer, and convolutional $3 \times 3$ kernel layer. As opposed to the most common feature identity mapping, the residual mapping includes a convolutional $3 \times 3$ kernel layer and a BN layer. The output consists of the results of the two convolution blocks and the residual mapping. The output $x_{l+1}$ of each unit serves as the input to the next unit and establishes a feature channel to the decoder unit.

**Bridge**

In the traditional U-Net model, the results of the downsampling in the encoder are directly taken as the input of the upsampling in the decoder. To increase the reception field and obtain multiscale features, we built a bridge linking the encoder to the decoder. In the bridge unit, the ordinary convolution is replaced by the atrous convolution, the architecture of which is shown in Figure 3 (Yu & Koltun, 2016). In Figure 3 only the red points are convolved with the $3 \times 3$ kernel and the rest of the points are omitted so the reception fields of Figure 3(a-c) are $3 \times 3$, $7 \times 7$, and $15 \times 15$, respectively. It helps to expand the reception field of the CNN without increasing the parameters, thus making the convolutional layer contain broad spatial characteristics. Chen et al. (2018) adopted an ASPP strategy in which different dilated rates were used in the same convolutional layer to connect the convolution feature maps with different dilated rates. This facilitates the detailed expression of multiscale information; in our method, it is used to process multiscale building extraction from high-resolution images.

The design of the dilation rate in this paper is inspired by Wang’s idea of Hybrid Dilated Convolution (Wang et al., 2018), which designed the dilation rates in a sawtooth structure (the [1,2,4,6] structure is adopted in this paper). The sawtooth structure reduces the effectiveness of cavity convolutional grids and helps meet the segmentation requirements of objects at different scales. The dilated rate of the sawtooth structure covers...
Table 1. Network structure of residual U-Net.

| Block     | Conv layer | Filter       | Stride | Outputsize |
|-----------|------------|--------------|--------|------------|
| input     | Block1     | Conv1-3      | [3 × 3,16] x 3 | 1          | 512 × 512 x 3 |
|           | Block2     | Conv4-6      | [3 × 3,32] x 3 | 1          | 256 × 256 x 32 |
| encoder   | Block3     | Conv7-9      | [3 × 3,64] x 3 | 1          | 128 × 128 x 64 |
|           | Block4     | Conv10-12    | [3 × 3,128] x 3 | 1         | 64 × 64 x 128 |
|           | Block5     | Conv13-15    | [3 × 3,256] x 3 | 1         | 32 × 32 x 256 |
| bridge    | Block6     | Conv16       | [3 × 3,256] x 3 | 1         | 32 × 32 x 1024 |
|           | Block7     | Conv17       | [3 × 3,256] x 3 | 1         | 32 × 256 x 64  |
|           | Block8     | Conv18       | [3 × 3,256] x 3 | 1         | 32 × 256 x 128 |
|           | Block9     | Conv19       | [3 × 3,256] x 3 | 1         | 32 × 256 x 256 |
| decoder   | Block10    | Conv20-22    | [3 × 3,128] x 3 | 1         | 32 × 256 x 128 |
|           | Block11    | Conv23-25    | [3 × 3,64] x 3  | 1         | 128 × 128 x 64 |
|           | Block12    | Conv26-28    | [3 × 3,32] x 3   | 1         | 256 × 256 x 32  |
|           | Block13    | Conv29-31    | [3 × 3,16] x 3   | 1         | 512 × 512 x 16  |
|           | Block14    | Conv32-34    | [1 × 1] x 1      | 1         | 512 × 512 x 1   |

A larger area and avoids the loss of information continuity. A modified ASPP model is adapted as the bridge module; this model can obtain features containing different types of receptive field information from the encoder.

Decoder

The architecture of the decoder is the same as that of the encoder. The decoder consists of five units that use residual learning to facilitate the training of the network. As shown in Figure 2(b), each unit consists of an upsampling, a concatenate, two convolution blocks, and a residual mapping. The upsampling can be achieved via deconvolution. The concatenate accepts the channel information from the encoder unit. Each convolution block includes a BN layer, a ReLU activation layer, and a convolutional 3x3 kernel layer. The residual mapping includes a convolutional 3x3 kernel layer and a BN layer. The output consists of the results of two convolution blocks and a residual mapping. Prior to the final output of the model, a x11 convolution and a layer using sigmoid as the activation function are used to construct an expected segmental multichannel characteristic graph.

In this paper, the WHU aerial image dataset and the Inria aerial image labeling dataset are used as the experimental data. In order to obtain as much image information as possible, the size of the input image is 512×512, and the output and parameters of each block are given in Table 1.

Focal loss

To address the imbalance between the foreground and background categories and that between the positive and negative samples, the FL method proposed in (T. Lin et al., 2017) was incorporated as the loss function of our method. In this paper, it is used to deal with the species imbalance between the building and others. FL is from the CE of binary classification, and the CE of binary classification is expressed as follows:

\[
CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases}
\]

(2)

Where \( y \in \{0, 1\} \) is the ground-truth label and \( p \in [0, 1] \) is the probability value of the positive example predicted by the model. To ease the presentation, we make the transformation:

\[
p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases}
\]

(3)

The final form of the binary CE is

\[
CE(p, y) = CE(p_t) = -\log(p_t)
\]

(4)

Through experiments in (H. Wu et al., 2019), it has been found that even easy examples have high loss. The influence of the positive samples is drowned out by this loss. Equilibrium factors are used to reduce the influence of easy examples; this generally consists of the specific gravity of the opposite class. For convenience, we define \( \alpha_t \) analogously to \( p_t \); the balanced CE is defined as

\[
CE(p_t) = -\alpha_t \log(p_t).
\]

(5)

Where \( \alpha_t \in [0, 1] \) is the weighting coefficient of the categories.

Even though \( \alpha_t \) can balance the importance of the positive and negative samples, it cannot solve the problem of easy and hard samples. Therefore, a modulating coefficient \((1 - p_t)^\gamma\) is added to the CE with an adjustable focusing parameter \( \gamma \geq 0 \). FL can be defined as follows:

\[
FL(p_t) = -(1 - p_t)^\gamma \log(p_t).
\]

(6)

FL is expressed as several values of \( \gamma \in [0, 5] \). In Eq.(6), \( p_t \to 1 \) and the factor \((1 - p_t)^\gamma \to 0 \). This means that \( p_t \) is relatively large when dealing with simple samples and that the weight naturally decreases, \( p_t \) is relatively...
small for hard samples and has a large weight, which makes the network tends to use such samples for updating parameter. Moreover, the weight changes dynamically, if the complex samples gradually become more differentiated, their influence will gradually decline. The balanced CE is then added to the final form of FL:

$$\text{FL}(p_i) = -\alpha_i (1 - p_i)^\gamma \log(p_i).$$ (7)

This solves not only the problem of the negative and positive sample imbalance but also the problem of easy and hard sample imbalance. Additionally, it slightly improves the accuracy compared with the non-α-balanced form and results in greater numerical stability. (T. Lin et al., 2017) pointed out that this method works best when $\alpha = 0.25$ and $\gamma = 2$. Therefore, we adopt those settings in this paper.

**Results**

**Implementation details**

We implemented our RU-Net using the Keras API in the Tensorflow framework. In the pre-processing stage, the original images were cropped into 512×512 without an overlap. In the training process, to increase the diversity of the training samples, the same random cutting method was used to process the images to 256×256 during each iteration. The hyperparameters are set as follows: epoch 200, batch size 32, and learning rate 0.001. The model was trained using the NVIDIA GeForce GTX 2080Ti and Intel(R) Xeon(R) CPU E5-2690 v4.

**Dataset description**

The WHU aerial image dataset was established in (Ji & Wei, 2019), includes an aerial dataset and a satellite dataset, mainly in and around densely populated areas. In this paper, the aerial image dataset was adopted; this dataset downsampled the original aerial image data to a ground resolution of 0.3 m. It was cropped into 8189 tiles with 512×512 pixels and corresponding 8189 labels. The dataset was divided into three parts: a training set of 130,500 buildings, a validation set of 14,500 buildings, and a test set of 42,000 buildings, with 4736, 1036, and 2416 tiles, respectively. The WHU dataset has high accuracy labels and several positive samples, so FL is not required. The sample result is shown in Figure 4.

Figure 5 shows the advantages and disadvantages of using RU-Net to extract buildings from the WHU dataset. In Figure 5(a), the manual label identifies the three small buildings separated by plants as a whole building, and the error is corrected in the prediction result of RU-Net. In Figure 5(b), the manual label identified an irregular building and the connected concrete floor as a whole, which was avoided in the prediction result of RU-Net. In Figure 5(c), the shadows on the right side of the building were wrongly identified as a part of the building in the prediction results of RU-Net. In Figure 5(d), the building was almost completely covered by plants and was missed in the prediction results of RU-Net.

The Inria aerial image labeling dataset was proposed in (Maggioli et al., 2017), which is an aerial image dataset that includes 180 images with and 180 without labels. To facilitate the quantitative analysis, we used the labeled dataset. This dataset contains remote sensing images of five different cities, from densely populated areas to alpine towns, each of which includes 36 images. Because its annotation accuracy is lower than that of the WHU dataset, we used this dataset evaluate the generalization ability of the model. The Inria dataset has a ground resolution of 0.3 m and an image size of 5000×5000 pixels. To enable a better comparison, we cropped the dataset to 512×512 pixels without an overlap. The entire dataset was cropped into 14,580 tiles. A total of 10,000 tiles were used as a training set, 2430 tiles were used as a validation set, and 2150 tiles were used as a test set. FL was used to avoid lower annotation accuracy and more negative samples. The sample result is shown in Figure 6.

**Evaluation metrics**

The most commonly used evaluation metrics to evaluate a binary classification are precision, recall, F1, and intersection over union (IoU). They are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}.$$ (8)
Recall \(= \frac{TP}{TP + FN}\) (9)

\[ F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \] (10)

\[ \text{IoU} = \frac{TP}{TP + FP + FN} \] (11)

where TP, FP, and FN represent the number of true positive, false positive, and false negative, respectively. Precision represents the ratio of images correctly predicted to be positive samples in all the images predicted as positive samples. Recall is the ratio of the number of correctly identified positive samples to all positive samples in the test set. F1 is used to measure precision and recall. The harmonic average values of the two values are given. The IoU represents the overlap ratio between the predicted and original image. In addition, we present the confusion matrix in Figure 7. Figure 7(a) is the confusion matrix of the WHU dataset, and Figure 7(b) is the confusion matrix of the Inria dataset.

**Discussion**

**Comparing methods**

The accuracy and efficiency of RU-Net were compared with those of U-Net (Ronneberger et al., 2015), FastFCN (H. Wu et al., 2019), DeepLabv3+ (Chen et al., 2018), Web-Net (Y. Zhang et al., 2019), and SegNet (Badrinarayanan et al., 2017). We reproduce the five networks using the Keras API and train them on both datasets.

**U-Net:** The U-Net model is a classical FCN which proposed in (Ronneberger et al., 2015). The encoder extracts the feature using a series of down-sampling operations, which include convolution and maxpooling.
The decoder completes upsampling by concatenating the map derived from deconvolution with the corresponding feature map from the encoder.

**FastFCN**: The FastFCN model was proposed in (H. Wu et al., 2019) and achieved the most outstanding results on the ADE20K dataset and the PASCAL Context dataset. The method proposes a Joint Pyramid Upsampling (JPU) module to replace dilated convolutions in the trunk network. Due to the high computational efficiency of the JPU module, both time and memory consumption can be saved.

**DeepLabV3+**: The DeepLab networks were proposed by Chen et al., and have been improved for many times, including V1 (D. Y. Chen et al., 2014), V2 (Chen et al., 2017), V3 (Chen et al., 2017), and V3+ (Chen et al., 2018). The backbone of encoder was CNN with dilated convolution and the ASPP module with dilated convolution. The DeepLabV3+ also introduced the decoder module, which integrated the low-level features with the high-level features to improve the accuracy of the segmentation boundary.

**Web-Net**: The Web-Net (Y. Zhang et al., 2019) is a nested encoder-decoder deep network for building extraction. To balance the local cues and the structural consistency, Ultra-Hierarchical Sampling (UHS) blocks were used to extract and fuse inter-level features. It had the advantages of less resource occupation and strong tailoring ability. For the best results, one must use the pretrained parameters from ImageNet.

**SegNet**: The SegNet model (Badrinarayanan et al., 2017) is a deep network for image semantic segmentation, which has been applied to autonomous driving and intelligent robots. The encoder part of SegNet uses the first 13 convolutional networks of VGG16. Each encoder layer corresponds to a decoder layer. The final output of the decoder is put into the soft-max classifier to generate a class probability for each pixel.

**Comparing with deep models**

**WHU dataset**

The parameters and four evaluation metrics of building extraction for RU-Net and the comparison methods are shown in Table 2. The results show that RU-Net outperforms the other methods on all evaluation metrics, and our RU-Net gives the best balance between recall and precision.

Compared with U-Net, FastFCN, DeepLabv3+, Web-Net, and SegNet, the IoU of RU-Net increased by 0.67%, 35.21%, 15.2%, 4.49%, and 0.96%, respectively. The parameters of SegNet are more than 5.9 times that of RU-Net, and the parameters of FastFCN and DeepLabv3+ are more than 12.9 times that of RU-Net. As a result, our RU-Net consumes less time than other methods except for the U-Net. Our method requires 88 seconds per epoch, while FastFCN and DeepLabv3+ require 290 seconds and 222 seconds per epoch, respectively. In other words, the efficiency of our model is more than 2.5 times of the latter. The Web-Net and the SegNet are also much slower than our model.

Figure 8 shows a comparison of the six methods for building extractions from the WHU dataset for different scenes. The first scene was for the identification of large buildings; all methods detected the building successfully. However, RU-Net achieves a more precise boundary compared with DeepLabV3+ and SegNet. FastFCN and Web-Net have not completed building shape integrity detection. The second scene was for the identification of medium-sized buildings; all methods partly misclassified the cement floors as buildings. The results of RU-Net, Web-Net, and SegNet showed a more precise boundary, and some buildings in the lower left corner were ignored in U-Net, FastFCN, and DeepLabv3+. The third scene was for the identification of small-sized buildings; all methods included more or less misclassifications. RU-Net

| Method | Parameters | Recall (%) | Precision (%) | F1(%) | IoU (%) | Time(s) |
|--------|------------|------------|---------------|-------|---------|---------|
| RU-Net | 5,644,401  | 96.98      | 97.48         | 97.23 | 94.61   | 88      |
| U-Net  | 1,768,945  | 96.81      | 96.93         | 96.87 | 93.94   | 43      |
| FastFCN| 79231809   | 70.52      | 79.03         | 74.53 | 59.40   | 290     |
| DeepLabv3| 73,324,698| 86.57      | 90.57         | 88.52 | 79.41   | 222     |
| WebNet | 1,434,241  | 94.13      | 95.48         | 94.79 | 90.12   | 90      |
| SegNet | 33,392,517 | 96.58      | 96.86         | 96.72 | 93.65   | 96      |
misclassified the cement floor in the upper left corner as a building, but it achieved more precise shapes and boundaries for the correctly detected buildings. Meanwhile, the result of RU-Net revealed the best segmentation of adjacent buildings. The fourth scene was for the identification of a negative sample; U-Net gave an incorrect prediction, while the other methods gave correct predictions. In summary, our RU-Net had better results than the other methods in terms of accuracy, shapes and boundaries in the building extraction. In summary, our method performed better in the integrity of building shape, the accuracy of boundary, and the distinction of adjacent buildings.

**Inria dataset**

The parameters and four evaluation metrics of building extraction for RU-Net and a comparison of the methods are shown in Table 3. Due to more complex scenarios, lower resolution, and label quality, the results of the Inria dataset are worse than that of the WHU dataset. The results show that our RU-Net outperforms the other methods in F1 and IoU. Compared with U-Net, FastFCN, DeepLabv3+, Web-Net, and SegNet, the F1 of RU-Net increased by 0.14%, 13.36%, 0.54%, 3.21%, and 2.39%, respectively. And the IoU of RU-Net increased by 0.22%, 21.48%, 0.98%, 5.64%, and 4.24%, respectively. The U-Net achieves the best recall and the DeepLabv3+ gets the best precision. Our RU-Net consumes less time than other methods except for the U-Net. It requires 120 seconds per epoch, while FastFCN and DeepLabv3+ require 471 seconds and 365 seconds per epoch, respectively. In other words, the efficiency of our model is more than 3 times of the latter. The Web-Net and the SegNet are much slower than our model.

**Table 3.** Evaluation results for the test set of the Inria dataset for the experimental methods.

| Method     | Parameters | Recall(%) | Precision(%) | F1(%) | IoU(%) | Time(s) |
|------------|------------|-----------|--------------|-------|--------|---------|
| RU-Net     | 5,644,401  | 93.73     | 96.14        | 94.92 | 90.33  | 120     |
| U-Net      | 1,768,945  | 94.72     | 94.86        | 94.78 | 90.11  | 64      |
| FastFCN    | 79231809   | 84.37     | 78.92        | 81.56 | 68.85  | 471     |
| DeepLabv3+ | 73,324,698 | 92.52     | 96.31        | 94.38 | 89.35  | 365     |
| Web-Net    | 1,434,241  | 91.26     | 92.16        | 91.71 | 84.69  | 125     |
| SegNet     | 33,392,517 | 92.03     | 93.03        | 92.53 | 86.09  | 143     |

Figure 8. Building extraction from the WHU dataset using the four methods on different scenes.

Figure 9 compares the building extraction from different scenes of the Inria dataset for the six methods. Due to the lower annotation accuracy and incorrect labels of the Inria dataset, the semantic segmentation results of all the methods were not very high. The first scene was for the identification of large buildings. RU-Net identified the buildings well, with less information loss than that in the other methods, and the boundaries were basically consistent with the labels. U-Net and FastFCN correctly identified the buildings but also incorrectly identified the surrounding concrete as buildings. DeepLabv3+ and Web-Net performed poorly, missing nearly all the information. SegNet failed to detect any buildings. The second scene was for the identification of medium-sized buildings. RU-Net identified the buildings well but also misidentified part of the concrete as buildings. The U-Net, FastFCN, and DeepLabv3+ had difficulty in correctly identifying the buildings. FastFCN misidentified concrete floors and even plants as buildings. The results of Web-Net and SegNet were terrible, both
failed to detect any buildings. The third scene was for the recognition of small-sized buildings. All the methods had poor recognition results; the accuracy of RU-Net and Web-Net were relatively high. In the fourth scene, for the identification of a negative sample, U-Net and FastFCN incorrectly identified the bridge as a building. The fifth scene was used to examine the buildings and the concrete that connects them, and there is a lot of shadow in this scene. The results show that only RU-Net performs well with respect to accuracy and generates sharp boundaries. Other methods lose some boundary information or incorrectly divide the cement floor into buildings. Compared with other methods, our proposed method achieves better results in terms of accuracy of building identification and building boundary. At the same time, it avoided the misclassification caused by the shadow.

**Learning rate and epoch**

The learning rate and the number of epochs had a significant impact on the performance of the model. We determine the convergence speed of the model. If the learning rate is too small, the model converges very slowly, and if it is too high, the loss function oscillates and iterates greatly, and even the gradient descent method may pass to the lowest point or even diverge. Therefore, we set several different learning rate values from large to small and run the model separately to investigate the iteration effect. This is effective only if the loss function becomes smaller; otherwise, the step size needs to be increased. Further, we conducted several comparative experiments to obtain the optimal learning rate. Taking the WHU aerial image dataset as an example, the learning rate was 0.001 at the start of the experiment and was gradually increased to 0.01, 0.02, 0.05, and 0.08. We used the same number of epochs (200), and the results are shown in Figure 10. When the learning rate was greater than 0.001, the loss function decreased slowly. When the learning rate is 0.05 or 0.08, the values of the loss function are much greater than those for other learning rates. As a result, the RU-Net model was trained on the building dataset with a learning rate of 0.001.

After selecting the appropriate learning rate for the model, we used as many epochs as possible to train the model. With an increasing number of epochs, the updating time of the neural networks increased, and the loss and accuracy curves became appropriate for underfitting. As shown in Figure 11, the training and validation accuracy curves are smoothed when the epoch is increased to 200; the training loss curve is flattened at the same epoch. However, the validation loss curve moved upward and fluctuated greatly when the number of epochs was more than 50. The validation accuracy was lower than the training accuracy and did not show an obvious upward trend, indicating that overfitting did not occur in our model. Therefore, 200 epochs are sufficient to train the RU-Net model.

**Suitable sample size and batch size**

The size of the training image affects the training model; it is easier for a model to obtain more detailed information from a sample with higher resolution (Tan & Le, 2019). Conversely, the batch size is closely related to the computer hardware conditions. Due to limited hardware conditions, the most common
practice is to sacrifice the batch size to guarantee the sample size. The sensitivity of batch size to gradient disappearance and gradient explosion is solved by BN in our method. To obtain the optimal performance of our model, we need to choose the most suitable sample size and batch size.

Three sets of comparative models with sample sizes and batch sizes of (512×512,16), (256×256,32), (128×128,64) were set up. The input shape of the test dataset was 512×512, and it was resized as appropriate in the experiment. We tested it on the WHU aerial image dataset, where all of the models were operated for 200 epochs, and the results are shown in Table 4. We can see that the combination (512×512,16) outperformed the other combinations for all evaluation metrics but consumed almost twice the time of consumed by (256×256,32) and seven times the time of (128×128,64). Compared with the combinations (256×256,32), the results of recall, precision, F1, and IoU for the combinations (256×256,32) were increased by 0.25%, 0.18%, 0.11%, and 0.41% separately; however, as previously noted, the former took more than double the time of the latter. The results of the combination (128×128,64) were the worst, and this combination was the most unbalanced between the recall and precision, it’s IoU was also 0.78% lower than that of the combination (256×256,32). Even though it consumed the minimum time because its final image size was smaller than the dilated rate of our ASPP, this leads to information loss. Considering the accuracy, efficiency, and time consumption, the best sample size, and batch size selection tested here was (256×256,32).

Table 4. Evaluation results for different sample sizes and batch sizes.

| (512×512,16) | 97.23 | 97.66 | 97.44 | 95.02 | 178
| (256×256,32) | 96.98 | 97.48 | 97.23 | 94.61 | 88
| (128×128,64) | 96.21 | 97.41 | 96.80 | 93.81 | 25

Atrous spatial pyramid pooling

In this section, we discuss the influence of ASPP. A comparison of our method with and without ASPP is shown in Table 5. We show that the precision, F1, and IoU of the method with ASPP are higher than those of the method without ASPP on the WHU dataset. Since the entire building can be separated using ASPP, the recall of the method with ASPP is 0.19% lower than that of the method without ASPP. For the Inria dataset, the recall, F1, and IoU metrics of the method with ASPP are much higher than those of the method without ASPP, and the precision is nearly the same. The only advantage of the method without ASPP is the time saved.

The main function of ASPP is to extract the features of multiscale buildings. Figure 12 shows the extraction results for buildings with and without ASPP. The results indicate that the modified RU-Net with ASPP can improve the extraction accuracy for large and medium-sized, irregular, and adjacent buildings. The sawtooth structure of the atrous convolution is adopted in this paper to obtain a variety of complex building features that can improve the precision of extraction. The first three scenarios in Figure 12 compare methods with and without ASPP on the WHU dataset. Scenario 1 is the identification of medium and large buildings. The method with ASPP achieves almost the integrity of the buildings, and the correctness of the boundary, which is closer to the label. Scenario 2 is an evaluation of adjacent building extraction. Although the method with ASPP does not completely extract all the buildings, the basic contours and shapes are recognized. The method without ASPP

Table 5. Evaluation results for architectures with and without ASPP.

| Method                  | Dataset | Recall (%) | Precision (%) | F1 (%) | IoU (%) | Time(s) |
|-------------------------|---------|------------|---------------|--------|---------|---------|
| Method with ASPP        | WHU     | 96.98      | 97.48         | 97.23  | 94.61   | 88      |
|                         | Inria   | 93.73      | 96.14         | 94.92  | 90.33   | 136     |
| Method without ASPP     | WHU     | 97.17      | 97.17         | 97.17  | 94.50   | 61      |
|                         | Inria   | 91.26      | 96.15         | 93.64  | 88.04   | 112     |
misses a large part of the adjacent buildings. Scenario 3 is the extraction of circular buildings. The extraction result with ASPP is almost the same as the label, and the contour of the method without ASPP is incomplete. The last three scenarios in Figure 12 compare the method with and without ASPP on the Inria dataset. They are also used to extract large and medium-sized, adjacent, and arc-shaped buildings. Similar to the WHU dataset, the method with ASPP obtains better extraction results for large- and medium-sized, irregular, and adjacent buildings.

Loss function

The foreground and background classes are extremely unbalanced, and a large number of easy negative examples exist. Considering the category imbalances between the building images and the other images in the semantic segmentation, especially the fact that there are a large number of negative samples in urban areas, FL was adopted in this paper to solve the problem of hindering error labels. The performance of FL is verified in

Figure 12. Building extraction results produced with and without ASPP.
Table 6; the sample and batch size of the experiment are the same. The WHU dataset has high accuracy labels and several positive samples. When FL was used as the loss function, only the precision increased by 0.27%, but the recall, F1, and IoU decreased by 0.3%, 0.03%, and 0.06%, respectively, due to the increase in false negatives caused by FL. The Inria dataset has a lower annotation accuracy, and the number of negative samples in this dataset is high. When FL was used as the loss function, the precision decreased by 0.34% because the false positive increased, but the recall, F1, and IoU increased by 2.33%, 0.97%, and 1.73%, respectively.

### Conclusion

In this paper, we propose an RU-Net model for remote sensing image-building extraction. The proposed architecture takes advantage of residual networks, U-Net, ASPP, and FL. The model adopts the basic framework of U-Net, and it is divided into encoder, bridge, and decoder units. Modified residual networks are adopted by the encoder and decoder units, ASPP is adopted as the bridge unit, and FL is used as the loss function. Experiments were conducted using the WHU aerial image dataset and the Inria aerial image labeling dataset, the experimental results of which show that our proposed method not only saved training time but also achieved improved results compared with other methods. In addition, we discuss suitable values of the learning rate, epoch, sample size and batch size and the performance of ASPP and FL.

Although our model has obtained satisfactory classification in quantitative performance and smooth boundary conditions, the accuracy of building boundary is still insufficient. There are still some problems, such as misclassification of concrete floor, poor identification effect of adjacent buildings, mistaken identification of shadows as buildings, and unrecognition of buildings covered by vegetation. In addition, there is a possibility for improvement with respect to the accuracy of the training dataset and validation dataset. The validation loss curve moves upward and fluctuates greatly, while the training loss curve tends to be flat and there is a huge gap between the two curves; accordingly, the hyperparameters can be adjusted to obtain better results.

### Disclosure statement

No potential conflict of interest was reported by the author(s).

### ORCID

Haifying Wang http://orcid.org/0000-0002-2926-7599

### References

Awrangjeb, M., Zhang, C. S., & Fraser, C. S. (2013). Improved building detection using texture information. International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences, XXXVIII-3/ W22(3), 143–148. Http://doi.org/10.5194/isprsarchives- XXXVIII-3-W22-143-2011

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481–2495. Http://doi.org/10.1109/TPAMI.2016.2644615

Bengio, Y. (2009). Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2(1), 1–127. Http://doi.org/10.1561/2200000006

Buduma, N., & Locascio, N. (2017). Fundamentals of deep learning. O’Reilly Media, Inc.

Chen, D. Y., Shang, S. B., & Wu, C. (2014). Shadow-based building detection and segmentation in high-resolution remote sensing image. Journal of Multimedia, 9(1), 181–188. Http://doi.org/10.4304/ jmm.9.1.181-188

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv 2014, arXiv:1412.7062.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834–848. Http://doi.org/10.1109/ TPAMI.2017.2699184

Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv 2017, arXiv:1706.05587.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F, and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. Computer Vision-ECCV 2018 (Lecture Notes in Computer Science), 11211, 801–818. http://doi.org/10.1007/978-3- 030-01234-2_49.
Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. Computer Vision-ECCV 2018 (Lecture Notes in Computer Science), 11211, 801–818. Http://doi.org/10.1007/978-3-030-01234-2_49

Chen, R. X., Li, X. H., & Jonathan, L. (2018). Object-based features for house detection from RGB high resolution images. Remote Sensing, 10(3), 451. Http://doi.org/10.3390/rs10030451

Dunaeva, A. V., & Kornilov, F. A. (2017). Specific shape building detection from aerial imagery in infrared range. Bulletin of the south ural state university. Series “Mathematical Computations and Software Engineering”, 6(3), 84–100. Http://doi.org/10.14529/cmse170306

Ferraioli, G. (2010). Multichannel InSAR building edge detection. IEEE Transactions on Geoence & Remote Sensing, 48(3), 1224–1231. Http://doi.org/10.1109/TGRS.2009.2029338

Gamal, A., Wibisono, A., Wicaksono, S. B., & Abyan, M. A. (2020). Automatic LIDAR building segmentation based on DGCNN and Euclidean clustering. Journal of Big Data, 7(1), 102. Http://doi.org/10.1186/s40537-020-00374-x

Guo, J. Y., Pan, Z. X., Lei, B., & Ding, C. B. (2017). Automatic color correction for multisource remote sensing images with Wasserstein CNN. Remote Sensing, 9(5), 483. Http://doi.org/10.3390/rs9050483

He, C. Y., Cao, X., Shi, P. J., & Dong, J. (2004). Incorporation of texture and structure information for urban building detection by using Landsat 7 ETM+ panchromatic image. Editorial Board of Geomatics & Information Ence of Wuhan University, 29(9), 800–804.

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Deep residual learning for image recognition. IEEE conference on computer vision & pattern recognition, 770–778. Http://doi.org/10.1109/CVPR.2016.90

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. Computer Vision-ECCV 2016 (Lecture Notes in Computer Science), Amstterdam, The Netherlands, 9908, 630–645.

He, K., Zhang, X., Ren, S., & Sun. J. (2016a). Computer Vision-ECCV 2016 (Lecture Notes in Computer Science), 9908, 630–645. Http://doi.org/10.1007/978-3-319-46493-0_38

Hinton, G. E., Osindero, S., & The, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18(7), 1527–1554. Http://doi.org/10.1162/neco.2006.18.7.1527

Huang, X., Yuan, W. L., Li, J. Y., & Zhang, L. P. (2017). A new building extraction postprocessing framework for high-spatial-resolution remote-sensing imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10(2), 654–668. Http://doi.org/10.1109/JSTARS.2016.2587324

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

Ji, S. P., & Wei, S. Q. (2019). Building extraction via convolutional neural networks from an open remote sensing building dataset. Acta Geodaetica et Cartographica Sinica, 48(4), 448–459.

Ji, S., Wei, S., & Lu, M. (2019). Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. IEEE Transactions on Geoose and Remote Sensing, 57(1), 574–586. Http://doi.org/10.1109/TGRS.2018.2858817

Kang, W. C., Xiang, Y. M., Wang, F., & You, H. J. (2019). EU-Net: An efficient fully convolutional network for building extraction from optical remote sensing images. Remote Sensing, 11(23), 2813. Http://doi.org/10.3390/rs11232813

Lam, E. Y., Niel, K. S., Saito, & Aoki, Y. (2015). Building and road detection from large aerial imagery. Image Processing: Machine Vision Applications VIII (SPIE Proceedings), 9405, 94050K. Http://doi.org/10.1117/12.2083273 k.

Lam, E. Y., Niel, K. S., Saito, & Aoki, Y. (2015). Building and road detection from large aerial imagery. Image Processing: Machine Vision Applications VIII (SPIE Proceedings), 9405, 94050K. Http://doi.org/10.1117/12.2083273

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In Advances in neural information (pp. 396–404).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. Http://doi.org/10.1109/5.726791

LeCun, Y., Kayukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. Proceedings of 2010 IEEE international symposium on circuits and systems, . Paris, France, 253–256. Http://doi.org/10.1109/ISCAS.2010.5537907 .

Li, Y., & Wu, H. Y. (2008). Adaptive building edge detection by combining LiDAR data and aerial images. Remote Sensing and Spatial Information Science, 37(1B), 197–202.

Lin, J. B., Jing, W. P., Song, H. B., & Chen, G. S. (2019). ESNet: Efficient network for building extraction from high-resolution aerial images. IEEE Access, 7, 54285–54294. Http://doi.org/10.1109/ACCESS.2019.2912822

Lin, T., Goyal, P., Girshick, R., He, K., & Dollr, P. (2017). Focal Loss for Dense Object Detection. 2017 IEEE international conference on computer vision (ICCV), Venice, Italy, 99, 2999–3007. Http://doi.org/10.1109/ICCV.2017.324

Liu, L. Y., & Wang, C. K. (2021). Building segmentation in agricultural land using high resolution satellite imagery based on deep learning approach. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43(B3), 587–594.

Liu, M. H., Shao, Y., Li, R. R., Wang, Y., Sun, X. B., Wang, J. K., & You, Y. C. (2020). Method for extraction of airborne LiDAR point cloud buildings based on segmentation. PLOS ONE, 15(5), e0232778. Http://doi.org/10.1371/journal.pone.0232778

Liu, P. H., Liu, X. P., Liu, M. X., & Shi, Q. (2019). Building footprint extraction from high-resolution images via spatial residual inceptive convolutional neural network. Remote Sensing, 11(7), 830. Http://doi.org/10.3390/rs11070830

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 640–651.

Maggiore, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, United States, 3226–3229. Http://doi.org/10.1109/IGARSS.2017.8127684 .

Marcu, A., Costea, D., Slusanschi, E., & Leordeanu, M. (2018). A multi-stage multi-task neural network for aerial scene interpretation and geolocalization. arXiv:1804.01322.
Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. 2015 IEEE international conference on computer vision, Santiago, Chile. 1520–1528. Http://doi.org/10.1109/ICCV.2015.178.

Ok, A. O., Senaras, C., & Yuksel, B. (2013). Automated detection of arbitrarily shaped buildings in complex environments from monocular VHR optical satellite imagery. IEEE Transactions on Geoscience & Remote Sensing, 51(3), 1701–1717. Http://doi.org/10.1109/TGRS.2012.2207123

Pan, X., Yang, F., Gao, L., Chen, Z., Zhang, B., Fan, H., & Ren, J. (2019). Building extraction from high-resolution aerial imagery using a generative adversarial network with spatial and channel attention mechanisms. Remote Sensing, 11(8), 917. Http://doi.org/10.3390/rs11080917

Psaltis, D., Sideris, A., & Yamamura, A. (1988). A multilayered neural network controller. IEEE Control Systems Magazine, 8 (2), 17–21. Http://doi.org/10.1109/37.1868

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. International conference on medical image computing and computer-assisted intervention, Springer International Publishing, 234–241.

Ruan, T., Liu, T., Huang, Z., Wei, Y., Wei, S., & Zhao, Y. (2019). Devil in the details: Towards accurate single and multiple human parsing. Proceedings of the AAAI conference on artificial intelligence, 33, 4814–4821. Http://doi.org/10.1609/aaai.v33i01.33014814

Senaras, C., Ozay, M., Yarman, V., & Fatos, T. (2013). Building detection with decision fusion. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6(3), 1295–1304.

Shrestha, S., & Vanneschi, L. (2018). Improved fully convolutional network with conditional random fields for building extraction. Remote Sensing, 10(7), 1135. Http://doi.org/10.3390/rs10071135

Sirmacek, B., & Unsalan, C. (2008). Building detection from aerial images using invariant color features and shadow information. International symposium on computer & information sciences, Istanbul, turkey, 1–5. Http://doi.org/10.1109/ISICS.2008.4717854

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv:1905.11946.

Tan, Y., Yu, Y., & Xiong, S. (2016). Semi-automatic building extraction from very high resolution remote sensing imagery via energy minimization model. 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 657–660. Http://doi.org/10.1109/IGARSS.2016.7729165

Vakalopoulou, M., Karantzalos, K., Komodakis, N., & Paragios, N. (2015). Building detection in very high resolution multispectral data with deep learning features. 2015 IEEE international geoscience and remote sensing symposium, Milan, Italy, 1873–1876. Http://doi.org/10.1109/IGARSS.2015.7326158.

Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z. H., Hou, X., & Garrison, C. (2018). Understanding convolution for semantic segmentation. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV/CA, USA, 1451–1460. Http://doi.org/10.1109/WACV.2018.00163

Wu, G. M., Shao, X. W., Guo, Z. L., Chen, Q., Yuan, W., Shi, X. D., Xu, Y. W., & Shibasaki, R. (2018). Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks. Remote Sensing, 10(3), 407. Http://doi.org/10.3390/rs10030407

Wu, H., Zhang, J., Huang, K., & Liang, K. (2019). FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation. arXiv:1903.11816

Xu, Y., Wu, L., Xie, Z., & Chen, Z. (2018). Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. Remote Sensing, 10(1), 144. Http://doi.org/10.3390/rs10010144

Yu, F., & Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122v3

Yuan, Y., Jiang, Z. G., Zhang, H. P., & Cai, B. W. (2017). Chimney and condensing tower detection based on faster R-CNN in high resolution remote sensing images. 2017 IEEE international geoscience and remote sensing symposium, Fort Worth, USA, 3329–3332. Http://doi.org/10.1109/IGARSS.2017.8127710

Zhang, Y., Gong, W., Sun, J., & Li, W. (2019). Web-Net: A novel nest networks with ultra-hierarchical sampling for building extraction from aerial images. Remote Sensing, 11(16), 1897. Http://doi.org/10.3390/rs11161897

Zhang, Y. (1999). Optimisation of building detection in satellite images by combining multispectral classification and texture filtering. Isprs Journal of Photogrammetry & Remote Sensing, 54(1), 50–60. Http://doi.org/10.1016/S0924-2716(98)00027-6

Zhang, Z., Liu, Q., & Wang, Y. (2018). Road extraction by deep residual U-Net. IEEE Geoscience and Remote Sensing Letters, 15(5), 749–753. Http://doi.org/10.1109/LGRS.2018.2802944

Zhong, S. H., Huang, J. J., & Xie, W. X. (2008). A new method of building detection from a single aerial photograph. International conference on signal processing, Beijing, China, 1219–1222. Http://doi.org/10.1109/ICOSP.2008.4697350

Zhou, K., Chen, Y., Smal, I., & Lindenbergh, R. (2019). Building segmentation from airborne VHR images using MASK R-CNN. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W13, 155–161. Http://doi.org/10.5194/isprs-archives-XLII-2-W13-155-2019

Zhu, Y. T., Liang, Z. L., Yan, J. W., Chen, G., & Wang, X. Q. (2021). E-D-Net: Automatic building extraction from high-resolution aerial images with boundary information. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 4595–4606.