remote sensing

MDPI

Article

Extracting Buildings from Remote Sensing Images Using a Multitask Encoder-Decoder Network with Boundary Refinement

Hao Xu [1], Panpan Zhu [1,*], Xiaobo Luo [1], Tianshou Xie [1] and Liqiang Zhang [2]

[1] College of Computer Sciences and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; s200231160@stu.cqupt.edu.cn (H.X.); luoxb@cqupt.edu.cn (X.L.); s200231036@stu.cqupt.edu.cn (T.X.)
[2] State Key Laboratory of Remote Sensing Science, Faculty of Geographical Science, Beijing Normal University, Beijing 100875, China; zhanglq@bnu.edu.cn
* Correspondence: zhupp@cqupt.edu.cn

Abstract: Extracting buildings from high-resolution remote sensing images is essential for many geospatial applications, such as building change detection, urban planning, and disaster emergency assessment. Due to the diversity of geometric shapes and the blurring of boundaries among buildings, it is still a challenging task to accurately generate building footprints from the complex scenes of remote sensing images. The rapid development of convolutional neural networks is presenting both new opportunities and challenges with respect to the extraction of buildings from high-resolution remote sensing images. To capture multilevel contextual information, most deep learning methods extract buildings by integrating multilevel features. However, the differential responses between such multilevel features are often ignored, leading to blurred contours in the extraction results. In this study, we propose an end-to-end multitask building extraction method to address these issues; this approach utilizes the rich contextual features of remote sensing images to assist with building segmentation while ensuring that the shape of the extraction results is preserved. By combining boundary classification and boundary distance regression, clear contour and distance transformation maps are generated to further improve the accuracy of building extraction. Subsequently, multiple refinement modules are used to refine each part of the network to minimize the loss of image feature information. Experimental comparisons conducted on the SpaceNet and Massachusetts building datasets show that the proposed method outperforms other deep learning methods in terms of building extraction results.

Keywords: building extraction; remote sensing; deep learning; multitask method; boundary information

Citation: Xu, H.; Zhu, P.; Luo, X.; Xie, T.; Zhang, L. Extracting Buildings from Remote Sensing Images Using a Multitask Encoder-Decoder Network with Boundary Refinement. *Remote Sens.* **2022**, *14*, 564. https://doi.org/10.3390/rs14030564

Academic Editor: Sander Oude Elberink

Received: 15 December 2021
Accepted: 20 January 2022
Published: 25 January 2022

1. Introduction

Buildings, as important manifestations of urban development and construction, are becoming a newly popular focus of research in regard to segmentation tasks. High-resolution Earth observation systems are growing more mature as remote sensing technology advances, allowing them to swiftly capture high-resolution remote sensing images across large areas. Building footprint maps based on such high-resolution remote sensing images are essential for building change detection [1], urban planning [2], and disaster emergency assessment. However, building type diversity and background complexity make it challenging to automatically extract buildings from high-resolution remote sensing images.

Since high-resolution remote sensing images contain a large amount of intrinsic characteristic information, such as spectral [3,4], textural [4,5], geometric [6,7], and contextual information [8], most studies on building extraction algorithms are based on features of these kinds. Considering that the geometric features of buildings are regular, Izadi et al. [8] used the perceptual grouping method to construct polygons on the tops of buildings and

combined the results with sensor metadata information to automatically detect buildings. Mohan et al. [9] used geometric features to extract roofs of different materials from multiple aerial remote sensing images and generated results based on wall and shadow context information. Jin et al. [10] used mathematical morphology to extract buildings by integrating basic information such as spectral content and context. Compared with original remote sensing images, high-resolution remote sensing images have more refined ground object geometries and textural features, with clear background characteristics. Thus, Xiang et al. [11] performed superpixel segmentation on the original image and extracted buildings by filtering out nonbuilding areas using feature information such as color, spectral, geometric, and background information. However, such methods rely on manually selected features and have limited generalization ability.

In recent years, the rapid development of deep learning has led to its widespread employment in automatic building extraction methods. Deep learning methods automatically learn features from the original image data themselves, thereby avoiding the subjectivity of manually selecting features. The recent pixel-to-pixel fully convolutional network (FCN) proposed by Long et al. [12] has significantly improved the accuracy of building extraction. Some FCN-based encoder–decoder networks [13,14] are enabling remarkable achievements in the field of image segmentation.

It is undeniable that these deep learning networks have achieved excellent results in natural scenarios. Therefore, an increasing number of studies are focusing on applying the above methods in the field of high-resolution remote sensing. Yuan [15] designed a deep convolutional network integrated with multilayer activation for the pixelwise prediction of buildings. Xu et al. [16] proposed an improved segmentation model based on an FCN to optimize building extraction results by using hand-crafted features and a guided filter. Huang et al. [17] proposed a progressive residual refinement network for building extraction by combining high-resolution aerial images with light detection and ranging (LiDAR) point clouds. Although all of the above methods have yielded decent results, some problems still need to be further explored. (1) Holes may exist in an extracted building object due to gaps between different feature levels. (2) The boundaries tend to be blurred because most building extraction methods focus only on comprehensive semantic information while neglecting detailed edge features, as shown in Figure 1.

To address the above issues, we propose a multitask network for automatic building footprint extraction from remote sensing images. Based on the encoder–decoder network structure, we implement two parallel decoders to generate clear contour maps and distance transform maps, which can help the network learn more boundary information and clarify blurry building boundaries. Considering that many details are lost in the downsampling process, we design an intra-refinement module to repair the feature map after each pair of convolutions. In addition, coarse prediction maps are generated for each upsampling output, and the combined loss of each coarse prediction map is calculated to drive the low-level information and high-level information to better complement each other. To this end, a residual refinement module is designed to repair the coarse prediction map during the last stage of downsampling before the final output is produced by the network. Finally, our multitask building extraction network is trained to improve the segmentation accuracy of the developed model on the SpaceNet [18] and Massachusetts [19] building datasets. The main contributions of this paper can be summarized as follows:

1. We propose a multitask deep learning model for building extraction that consists of an encoder–decoder network for generating coarse building maps, with two auxiliary decoders for correcting blurred boundaries, and a multitask loss is applied to monitor the accuracy of the building extraction process.
2. Two parallel auxiliary branches, namely, a contour decoder and a distance decoder, are introduced into the encoder–decoder network architecture to enrich the building boundary information and improve the building extraction accuracy of the model.
3. An intra-refinement module is introduced to enhance the constructed feature map after each round of downsampling, while coarse prediction maps are separately

generated for the multistage outputs of the upsampling process for the subsequent loss calculation. A residual refinement module is also designed to refine the coarse prediction map generated in the last stage in order to obtain the final output.

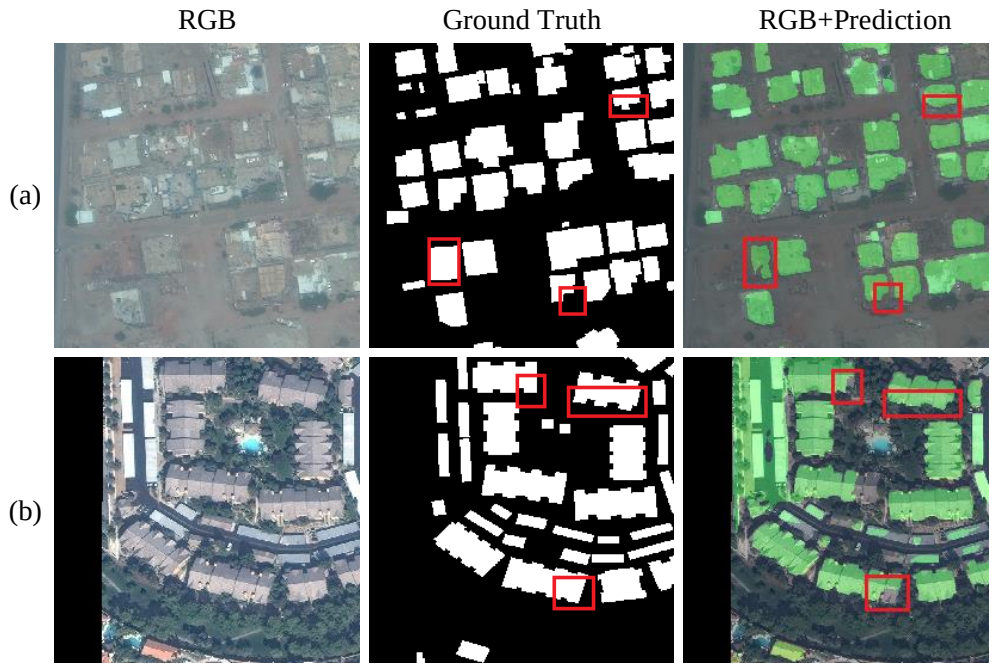


Figure 1. Illustration of semantic segmentation results obtained using the baseline model on the SpaceNet building dataset: (**a**) Khartoum and (**b**) Las Vegas. One problem with convolutional neural network (CNN)-based building extraction methods is that the boundaries of the buildings tend to be blurred, with most boundary pixels belonging to the category of interest being misclassified as background pixels.

The remaining sections of this article are structured as follows. Section 2 introduces the related work. The proposed method, including the network structure and additional modules, is described in detail in Section 3. Section 4 introduces an ablation study and a comparative experiment, including the utilized dataset, the implementation details, the obtained results, and analyses thereof. An Additional dataset is utilized to prove the generalization ability and robustness of the proposed model in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Due to the rapid development of convolutional neural networks (CNNs) [20], deep-learning-based methods are gradually replacing traditional manual extraction methods and becoming mainstream, achieving remarkable results in the field of building extraction. Because of their ability to extracting effective feature information from images without any prior knowledge, CNNs have been widely applied in image classification [21], object detection [22], and image segmentation [23,24] in recent years. In contrast to a CNN, the FCN proposed by Long et al. [12] attempts to recover the corresponding category of each pixel from abstract features; this approach has attracted much attention in the field of semantic segmentation. However, an FCN classifies each pixel without fully considering the relationships between pixels, and the upsampling results obtained via deconvolution are insensitive to image details. Different from the original FCN, the U-Net model proposed by Ronneberger et al. [13] introduces a symmetric structure that can combine low-level features with high-level features via skip connections to enhance semantic information. Based on the U-Net architecture, Ji et al. [25] designed a Siamese U-Net (SiU-Net), in which two input branches share the same U-Net structure and weights

for corresponding inputs. In addition, some variants of FCNs have been presented, such as DeepLabV3+ proposed by Chen et al. [26], which extracts multiscale features and enlarges receptive fields through dilated convolution at different dilation rates. Several examples of works [27–29] using the Deeplab architecture as a backbone model demonstrate its effectiveness in segmentation tasks.

Multitask learning [30], in which representations are shared between related tasks to enhance the generalization ability and robustness of a model, has been introduced into the task of building extraction. For example, Hui et al. [31] applied a multitask learning approach to incorporate the structure information of buildings. Bischke et al. [32] proposed a multitask learning method for building extraction that incorporates building boundary information to improve the segmentation results. However, it is insufficient to generate regular building boundaries based only on distance estimation. To mitigate the influence of the large intraclass variations among buildings, Guo et al. [33] designed a multitask parallel attention network that can overcome the semantic gaps among different kinds of buildings. Li et al. [34] used adaptive weights in a multitask learning method to learn boundary information by means of a special boundary loss. For the refinement of segmentation boundaries, several researchers have attempted to achieve improved segmentation accuracy by adding auxiliary tasks on the basis of an encoder–decoder framework. Chen et al. [35] added a parallel decoder to the original encoder–decoder network to generate clear contours for separating clustered objects. Tan et al. [36] treated boundary recognition as a regression problem and added a branch for distance transformation to an encoder–decoder network in order to ensure smoothness of the segmentation contours. The above studies show that auxiliary tasks are helpful for improving the predictions generated for the main task.

Furthermore, some researchers have refined image details by using residual blocks. Peng et al. [37] designed a residual-based module for boundary refinement. To obtain multiscale features, Zhang et al. [38] proposed a residual block containing dilated convolutional layers with different dilation rates. Qin et al. [39] designed a U-Net-like residual block to further refine the coarse prediction maps of image boundaries.

In light of the above methods, we propose an end-to-end multitask building extraction method to improve segmentation accuracy while using two additional branches for boundary refinement. Different from the aforementioned methods, we design a novel residual block containing a proposed intra-refinement module to further enhance the semantic information of the input image by refining the feature maps at each stage. Furthermore, we utilize multiple intermediate outputs in the loss calculation to drive the low-level and high-level features to better complement each other. In this way, the proposed method combines multiple tasks to overcome semantic gaps and achieve more accurate building extraction results.

3. Methodology

In this section, we discuss the details of the proposed framework. As shown in Figure 2, our framework consists of an encoder–decoder network that is used to obtain a segmentation prediction map and two auxiliary branches trained for contour extraction and distance transformation. In the encoder, we design an intra-refinement module to refine the details of the constructed feature map after every two convolutions. For the intermediate decoder, we propose utilizing multistage outputs to obtain multiple coarse segmentation maps during the upsampling process, which can be used to combine the multilevel information from different stages. Before the generation of the mask segmentation map, a residual refinement module is introduced to refine the coarse segmentation map. Finally, a multitask loss is designed to train the end-to-end network.

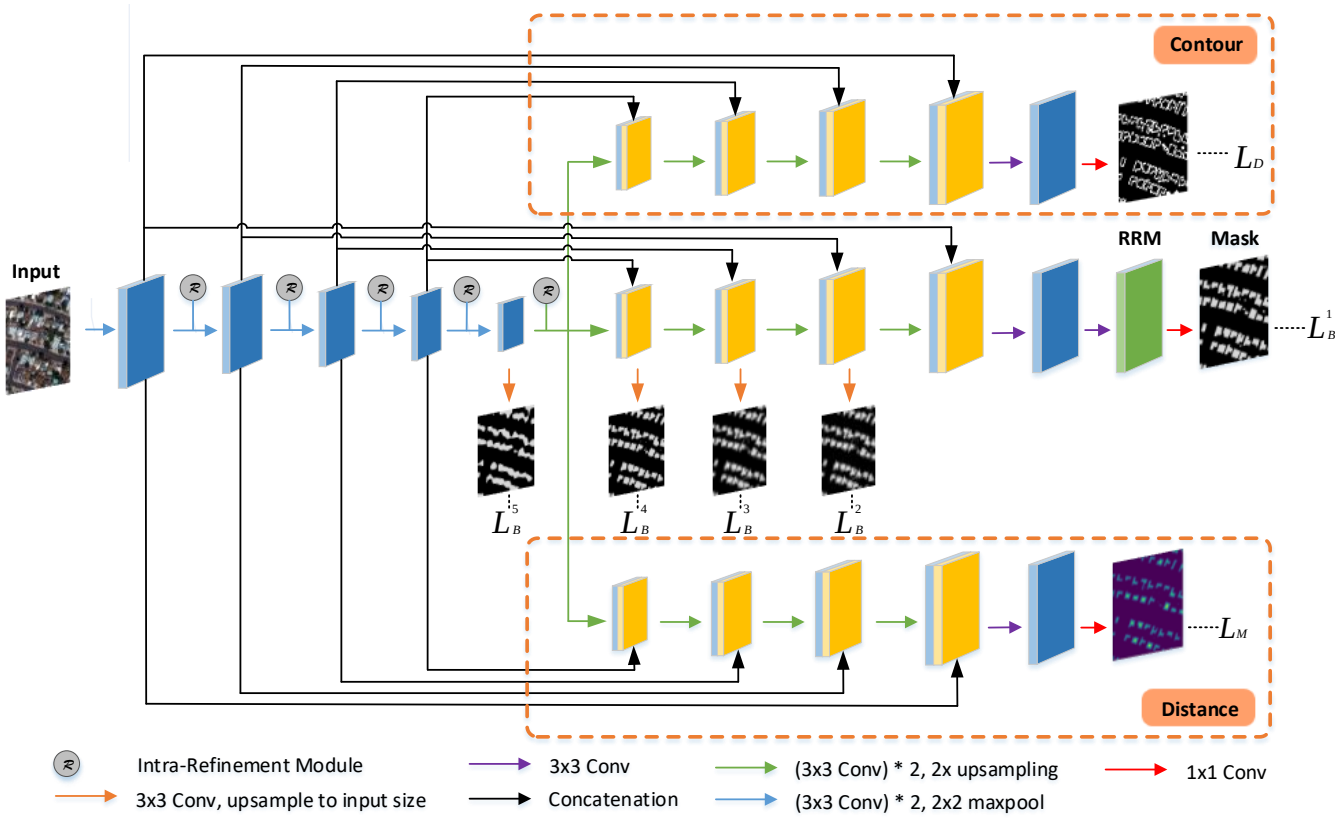


Figure 2. Overview of the proposed model. The encoder applies repeated downsampling operations, each of which is followed by an intra-refinement module, and three decoders perform different tasks. In the intermediate decoder, a residual refinement module (RRM) is inserted after the final 1×1 convolution layer to refine the coarse mask prediction. The coarse prediction maps obtained from four intermediate stages are utilized in the loss calculation.

3.1. Encoder-Decoder Network Architecture

The proposed network has a standard encoder–decoder architecture with a single encoder on the left and three parallel decoders on the right. The encoder applies repeated downsampling operations that halve the size of the feature map. Each downsampling operation includes two 3×3 convolutions, rectified linear unit (ReLU) activation, and a 2×2 max pooling operation with a stride of 2. During the downsampling process, we add an intra-refinement module after each pair of convolutions for detail refinement, as described in Section 3.2. The three parallel decoders are consistent in terms of the structure until the last 1×1 convolutional layer. The number of output channels of the distance decoder is 1; this decoder returns a regression value after computing a sigmoid function. The other two decoders each have 2 output channels, representing background and buildings.

Each decoder represents a separate task, in which repeated upsampling operations are applied to reach the same resolution as the original input. The intermediate decoder is designed for mask segmentation and is followed by a residual refinement module to refine the coarse output before the final mask prediction map is generated. The residual refinement module is described in Section 3.3. To assist the network in capturing more boundary information, we introduce two auxiliary decoders, one for contour extraction and one for distance transformation. The former branch provides semantic building boundary information, while the latter estimates the position of a building by calculating the distance from any point inside the building instance to the nearest background point.

3.2. Intra-Refinement Module

Due to the gaps between different feature levels, there will typically be holes in the segmentation map. To repair these holes, we introduce an intra-refinement module to refine the feature map after each double convolution (doubleconv), as shown in Figure 3. After two successive 3×3 convolutions in the downsampling step, we obtain a feature map denoted by f_{in}. We first apply a 3×3 convolution without changing the number of output channels to obtain the feature vector f_{mid}. Then, we utilize another 3×3 convolution to expand the number of output channels by a factor of 2 and obtain the intermediate feature map $\tilde{f}$. We then divide the feature map evenly into two parts according to the number of channels. The former part (f_1) is multiplied by the feature vector f_{mid} to obtain an intermediate result, while the latter part (f_2) is added to this intermediate result to obtain the output feature map f_{out}. Finally, we send f_{out}, which is constructed as follows, to the subsequent downsampling operation:

$$f_{mid} = R(N(Conv_1(f_{in}))) \tag{1}$$

$$\tilde{f} = Conv_2(f_{in}) \tag{2}$$

$$f_{out} = R((f_1 \odot f_{mid}) + f_2) \tag{3}$$

where $Conv_1(\cdot)$ is a 3×3 convolution that does not change the number of channels, $Conv_2(\cdot)$ is a 3×3 convolution that doubles the number of channels, $R(\cdot)$ is the ReLU function, $N(\cdot)$ denotes batch normalization, and $\odot$ and $+$ denote the elementwise multiplication and addition, respectively, of two vectors.

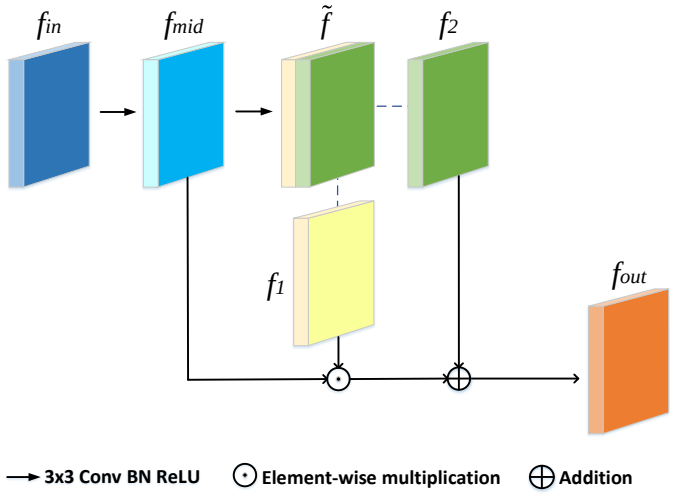


Figure 3. Architecture of the intra-refinement module.

3.3. Residual Refinement Module

To further refine the coarse prediction map obtained after upsampling, we design a residual refinement module. Different from previously developed residual blocks [37–39], the proposed module uses doubleconv to enhance the semantic image information during the downsampling process, followed by the previously introduced intra-refinement module to refine lost details. In Figure 4, the left part of the chart represents a downsampling operation that uses max pooling with a stride of 2, while the right part represents an upsampling operation that uses bilinear interpolation with a factor of 2. Each doubleconv operation in Figure 4 consists of two successive 3×3 convolutions followed by batch normalization and ReLU activation. The features in each decoder layer are concatenated with those in the corresponding encoder layer to fuse multiscale features. In particular, we introduce the intra-refinement module into the residual block to mitigate the effects of the pooling operations and refine the details of the feature maps. Finally, we combine the

coarse prediction results with the residual refinement results via an addition operation to obtain the final refined map. The refined feature map can be described as follows:

$$M_f = M_c + M_s \tag{4}$$

where M_c, M_s, and M_f denote the coarse feature map, the residual feature map, and the refined feature map, respectively.

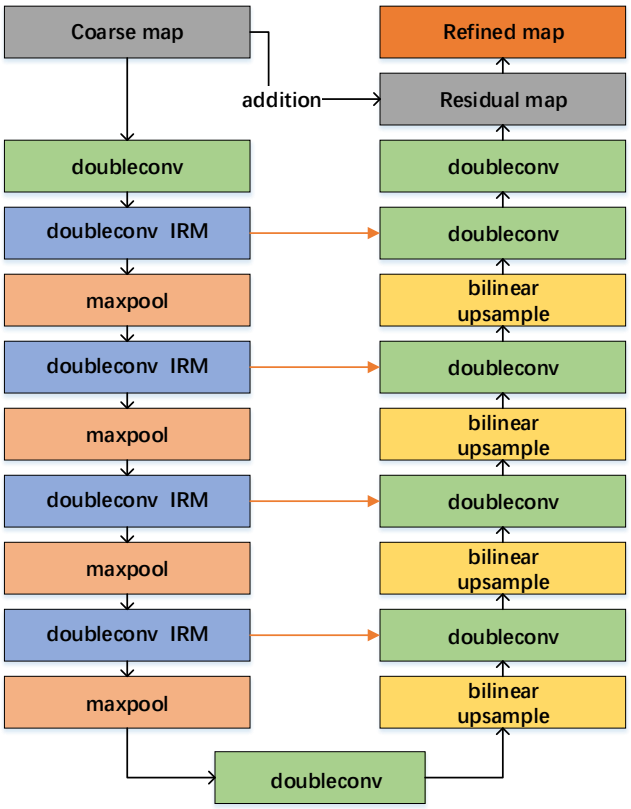


Figure 4. Structure of the proposed residual refinement module. Doubleconv refers to two successive convolutions applied to extract features, followed by an intra-refinement module (IRM) to refine details.

3.4. Loss Function

3.4.1. Multistage Mask Loss

For building mask prediction, we use a cross-entropy loss to calculate the prediction map losses and update the network parameters. The multistage mask loss L_M is formulated as follows:

$$L_M = -\sum_{i,j}\left[y_{i,j}\log\left(\hat{y}_{i,j}\right) + \left(1 - y_{i,j}\right)\log\left(1 - \hat{y}_{i,j}\right)\right] \tag{5}$$

where $y_{i,j} \in \{0, 1\}$ stands for the ground-truth label and $\hat{y}_{i,j}$ is the predicted label probability for pixel (i, j).

3.4.2. Contour Loss

In building boundary prediction, negative examples account for the majority of all samples. Under the effect of the cross-entropy loss, the model tends to ignore difficult positive examples in the learning process; consequently, poor prediction results on positive examples will directly lead to a low F1-score. In our experiment, we choose the Dice loss as the loss function for boundary prediction, which encourages the model to pay more attention to difficult samples during training, reduces the degree of learning based on

simple negative examples, and improves the value of the F1-score. The contour loss L_C is defined as follows:

$$L_C = \frac{2\sum_{i,j} y_{i,j} * \hat{y}_{i,j}}{\sum_{i,j} y_{i,j}^2 + \sum_{i,j} \hat{y}_{i,j}^2} \tag{6}$$

where $y_{i,j} \in \{0, 1\}$ denotes the ground truth and $\hat{y}_{i,j}$ is the predicted segmentation probability for pixel (i, j).

3.4.3. Distance Loss

To estimate the building distance map, we utilize a simple mean square error (MSE) loss as the loss function for the distance transform to gradually reduce the gap between the predicted result and the ground truth; this loss function is formulated as follows:

$$L_D = \sum_{i,j} (\hat{D}(i, j) - D(i, j))^2 \tag{7}$$

where $\hat{D}(i, j)$ stands for the estimated distance map and $D(i, j)$ denotes the ground truth of pixel (i, j).

Our final loss function is defined as follows:

$$L = \alpha_1 \sum_{n=1}^{N} L_M^{(n)} + \alpha_2 L_C + \alpha_3 L_D \tag{8}$$

where α_1, α_2, and α_3 are the weights of each loss component, $L_M^{(n)}$ denotes the loss on the n-th predicted output, and N is the total number of outputs. Let $L_M^{(n)} = \{L_M^1, L_M^2, L_M^3, L_M^4, L_M^5\}$ in our model denote the coarse prediction maps obtained in the four intermediate stages during downsampling and the final mask prediction map.

4. Experiments and Analysis

4.1. SpaceNet Dataset

For a building segmentation dataset, attention should be given to the resolution of the images, the number of building instances, the correspondence between the pixel positions and the ground truth, and the band information. We introduce the SpaceNet dataset (SpaceNet dataset: https://spacenet.ai/spacenet-buildings-dataset-v2/, accessed on 1 December 2021), which consists of images of four cities, namely, Las Vegas, Khartoum, Shanghai, and Paris, as shown in Figure 5. For each city, the dataset contains GeoJson labels of buildings, panchromatic images, 8-band multispectral images, pansharpened versions of the red-green-blue (RGB) bands from the multispectral product (RGB-PanSharpen), and pixel-based labels for buildings. Here, the RGB-PanSharpen images are used as our training dataset.

Four cities are used to evaluate the performance of the proposed method. This dataset contains 10,593 image scenes with labels, as shown in Table 1. We utilize the annotations to generate single-band binary images containing values of 0 and 1 to denote whether a pixel belongs to a nonbuilding or building area, respectively, as the labeled dataset used to evaluate our proposed method.

Figure 5. Examples from four cities in the SpaceNet building dataset (from left to right): Las Vegas, Khartoum, Shanghai, and Paris.

Table 1. Numbers of images and building footprints across the areas of interest in SpaceNet.

City	Number of Images	Resolution (m)	Raster Area (km^2)
Las Vegas	3851	0.3	216
Khartoum	1012	0.3	765
Shanghai	4582	0.3	1000
Paris	1148	0.3	1030
Total	10,593		3011

4.2. Dataset Preprocessing

To facilitate the model training process, we resize the images from 650 × 650 to 256 × 256. The bit depth is changed from 16 to 8, and the RGB images are converted from the Tagged Image File (TIF) format to the Portable Network Graphics (PNG) format. We eliminate images that do not contain buildings, and finally, the numbers of RGB images used for network training and validation from the above four cities are 3615, 831, 3352, and 633, respectively. To obtain the corresponding contour and distance datasets, we rely on the findContours and distanceTransform functions of OpenCV. The dataset for each city is randomly divided into three separate sets for training (80%), validation (10%), and testing (10%), while ensuring that each type of geography is contained in each of these splits.

4.3. Evaluation Metrics

In this paper, we measure different aspects of the performance of the proposed method by using six common metrics, namely, the precision, recall, F1-score, overall accuracy (OA), intersection over union (IoU), and boundary IoU [40].

As shown in Formulas (9)–(11), the precision, recall, and F1-score are used to quantify the model accuracy. The F1-score is the harmonic mean of precision and recall, and thus combines the results of both.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

where "true positive" (TP) denotes the number of building footprint pixels that are correctly detected, "false positive" (FP) denotes the number of pixels where other objects are

incorrectly detected as building footprint polygons, and "false negative" (FN) denotes the number of building footprint pixels that are not detected.

We utilize the OA to quantify the performance of the model on the validation dataset; the OA represents the percentage of correctly classified pixels out of the total number of pixels. The formulation is defined as follows:

$$\mathrm{OA} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{12}$$

where "true negative" (TN) denotes the number of pixels belonging to nonbuilding objects that are correctly detected.

We use the pixelwise IoU score as our evaluation metric to measure the proximity between the ground-truth building area and the predicted building area; the IoU is defined as follows:

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{13}$$

To prove the effectiveness of the proposed method for boundary restoration, we use the boundary IoU to measure the quality of the segmentation boundaries; the boundary IoU is defined as follows:

$$\text{Boundary IoU} = \frac{|(\mathrm{G_d} \cap \mathrm{G}) \cap (\mathrm{P_d} \cap \mathrm{P})|}{|(\mathrm{G_d} \cap \mathrm{G}) \cup (\mathrm{P_d} \cap \mathrm{P})|} \tag{14}$$

where G denotes the ground-truth binary mask, P denotes the predicted binary mask, d denotes the pixel width of the boundary region, and $\mathrm{G_d}$ and $\mathrm{P_d}$ refer to the sets of ground-truth mask pixels and predicted mask pixels, respectively, that are within a predefined distance from each contour.

4.4. Implementation Details

In this study, the proposed method is implemented using the PyTorch deep learning framework [41] on a single Nvidia RTX 3090 GPU. Our model is trained with the root-mean-square propagation (RMSprop) optimizer [42], and the learning rate is initialized at 0.0001 with a weight decay of 1×10^{-8}. The total number of epochs is set to 150, with a batch size of 16. During the training process, in each epoch, the validation set is used to calculate the induced error once the model training process is complete. We use an early stopping strategy [43] to prevent model overfitting; specifically, the training procedure is stopped automatically when the IoU value of the model on the validation set has not improved within 10 epochs. Similarly, we utilize the ReduceLROnPlateau strategy [41] to reduce the learning rate when the network performance on the verification dataset does not improve within a certain number of epochs. In the testing stage, the test dataset is used to evaluate the proposed model, which generates a binary mask with values of 0 and 1.

4.5. Ablation Study

In this section, we verify the contribution of each key component to model performance through an ablation study. For simplicity, we select two of the four city datasets, Las Vegas and Paris, as the ablation datasets. Our baseline model is constructed on the basis of the proposed network, to which contour extraction (C), distance transformation (D), the residual refinement module (R), the intra-refinement module (I), and multistage outputs (M) are added. The OA, precision, recall, F1-score, IoU, and boundary IoU are used to evaluate the performance of these components.

The results of the ablation study are shown in Tables 2 and 3. To facilitate observation, the maximum value of each evaluation metric is presented in bold.

Table 2 shows the results of the ablation study for the Las Vegas dataset. With the addition of contour extraction, the IoU value is increased by 1.3%, indicating that auxiliary contour information can make the boundaries more regular and obviously improve the model accuracy. Once contour extraction has been added, the further addition of the

distance transform module can also improve the model accuracy, increasing the IoU value by 2.7%. With the further addition of the residual refinement module as well as the previous two components, the building extraction result is significantly improved, with an IoU increase of 3.8%. Even without the residual refinement module, after the addition of the intra-refinement module and multistage outputs, the IoU is improved by 3.6% and 3.5%, respectively. The above results indicate that our model achieves its best performance through the combined influence of different modules.

Table 2. Results of an ablation study of different modules obtained on the Las Vegas dataset. C, contour extraction; D, distance estimation; R, residual refinement module; I, intra-refinement module; M, multistage outputs.

Method	OA (%)	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)	Increase in IoU (%)
Baseline	95.4	85.9	87.5	85.9	77.0	-
Baseline + C	95.7	87.3	87.9	86.9	78.3	1.3
Baseline + CD	96.0	88.3	88.4	88.0	79.7	2.7
Baseline + CDR	96.4	89.7	88.1	88.6	80.8	3.8
Baseline + CDI	96.3	89.4	88.4	88.5	80.6	3.6
Baseline + CDM	96.2	89.4	88.4	88.6	80.5	3.5
Baseline + ALL	**96.6**	**91.2**	**88.8**	**89.6**	**82.4**	**5.4**

'Increase in IoU' denotes the increase in the value achieved over that of the baseline. The best results are shown in bold. For all metrics, the higher the value is, the better the evaluation.

Table 3 shows the ablation experiment results obtained for the proposed method on the Paris dataset. The first row of Table 3 corresponds to the baseline model. After the addition of contour extraction, the IoU value is increased by 0.5%. When the distance transform is added in addition to the contour information, the IoU is improved by 0.8%. Rows 4, 5, and 6 of Table 3 show that when the residual refinement module, the intra-refinement module, and multistage outputs are introduced in addition to the contour and distance information, these additions obviously improve the segmentation performance of the model, increasing the IoU by 1.7%, 1.3%, and 1.4%, respectively, thus further proving the effectiveness of the various components proposed in this paper.

Table 3. Results of an ablation study of different modules obtained on the Paris dataset. C, contour extraction; D, distance estimation; R, residual refinement module; I, intra-refinement module; M, multistage outputs.

Method	OA (%)	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)	Increase in IoU (%)
Baseline	95.3	77.6	73.9	73.8	61.9	-
Baseline + C	95.5	77.5	72.7	74.0	62.4	0.5
Baseline + CD	95.6	76.5	73.4	74.3	62.7	0.8
Baseline + CDR	95.8	79.0	**75.9**	75.5	63.6	1.7
Baseline + CDI	95.7	**79.1**	73.2	74.1	63.2	1.3
Baseline + CDM	95.7	78.9	75.0	75.6	63.3	1.4
Baseline + ALL	**95.9**	78.9	74.9	**75.8**	**64.6**	**2.7**

'Increase in IoU' denotes the increase in the value achieved over that of the baseline. The best results are shown in bold. For all metrics, the higher the value is, the better the evaluation.

To validate the proposed modules in terms of building boundary restoration, we introduce the boundary IoU metric to evaluate the boundary segmentation results obtained for the two cities. The results are shown in Figure 6. Compared to the baseline, each of our proposed modules achieves boundary repair improvements. Notably, the performance of our model is improved more significantly after the incorporation of each refinement module. The boundary IoU values are improved by 6.6% and 3.1% on the Las Vegas dataset and the Paris dataset, respectively.

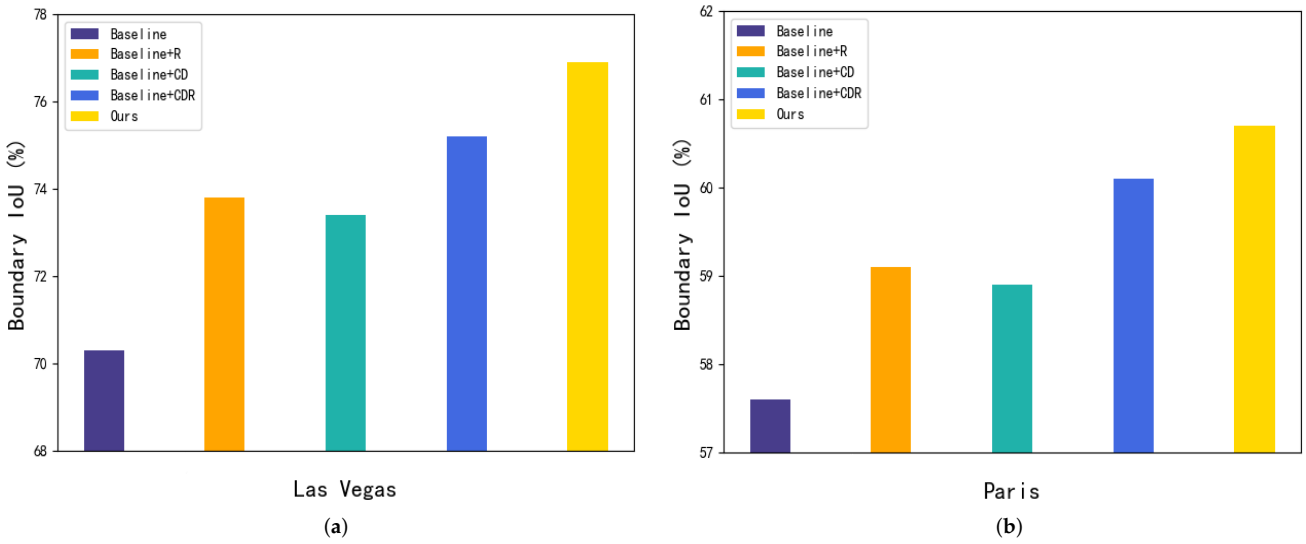


Figure 6. Boundary IoU values for two cities after the addition of different modules: (**a**) the Las Vegas dataset and (**b**) the Paris dataset.

4.6. Comparisons with State-of-the-Art Methods

4.6.1. Introduction of the Models for Comparison

Since our model is an improved version of an encoder–decoder network, we select U-Net [13] as the baseline model. In addition, DeepLabV3+ [26], the pyramid scene parsing network (PSPNet) [44], the unified perception network (UPerNet) [45], and TransUNet [46] are selected for comparison. The details of these models are described as follows.

1. U-Net was proposed by Ronneberger et al. [13] and was originally used to solve medical image segmentation problems. It has a U-shaped network structure that includes an encoder and decoder. The skip connections in the decoder stage effectively combine low-level features with high-level features to recover the lost feature information.
2. DeepLabV3+, proposed by Chen et al. [26], also uses an encoder–decoder structure to fuse multiscale information. In the encoder stage, atrous convolution is introduced with the aim of extracting more effective features, and the receptive field is enlarged without information loss so that each convolution output contains a large range of information. In the experiment, we first select ResNet-50 [47] as the backbone for feature extraction. To extract and fuse features, an atrous convolution with four distinct rates is used, followed by a 1×1 convolution to compress the features.
3. PSPNet was proposed by Zhao et al. [44] and is a semantic segmentation method that utilizes a pyramid pooling module to aggregate contextual information obtained from different regions. In the PSP structure, the acquired feature layers are divided into grids of different sizes, and internal pooling is performed evenly in each grid cell. This structure improves the ability of the network to obtain global information via context aggregation. In this experiment, we select ResNet-50 to extract features, and a four-level pyramid is applied to obtain different pooling kernels.
4. Xiao et al. [45] proposed UPerNet as an improved network based on PSPNet that can simultaneously handle three tasks: semantic segmentation, scene classification, and texture parsing. This algorithm uses a feature pyramid network [48] as the backbone and adds a pyramid pooling module to obtain a fused feature map.
5. TransUNet was proposed by Chen et al. [46] as a combination of U-Net and Transformer [49]. It adopts the U-shaped structure of U-Net to effectively utilize low-level convolution characteristics. In addition, by transforming the input image into a sequence, the Transformer architecture can be used to capture a large amount of global contextual information to achieve more precise positioning.

4.6.2. Results Obtained on the SpaceNet Dataset

Figures 7–10 compare the results obtained by the different methods for the four cities in the SpaceNet dataset. To facilitate the comparisons, we add two distinguishable colors to the prediction maps, with red representing areas where the background is misclassified as a building and blue representing areas where a building is misclassified as background.

(1) Las Vegas Dataset

Figure 7 compares the building extraction results obtained for different regions of the Las Vegas dataset using the six methods. We select two representative images from the Las Vegas test dataset for pixel-level comparisons. We can see that the results of our proposed method are visually superior to those of the other five methods.

The distributions of the buildings in the two test images in Figure 7 are fairly regular, but buildings of different sizes are difficult to extract accurately. The first row of Figure 7 shows that compared with the other five methods, the proposed method obtains finer boundaries, indicating that the addition of auxiliary boundary information can play a certain role in the restoration of building boundaries. In the second image in Figure 7, the distribution of the buildings is generally related to the road layout, providing additional convenience for building extraction in this kind of scenario. As the second row of Figure 7 shows, the boundaries in the prediction map generated by our proposed model are more regular, and the pixel-level results indicate that the proposed method can achieve better segmentation than the other models.

Table 4 presents a quantitative comparison of the proposed method with the other deep learning methods on the same dataset and under the same experimental environment. The OA, precision, recall, F1-score, and IoU metrics are used for quantitative evaluation. As seen from Table 4, the results obtained by our network are superior to those of the other networks. In terms of the IoU values, our proposed method shows the best performance, with 3.9–5.4% improvements in the results; these findings prove the effectiveness of our model for building extraction.

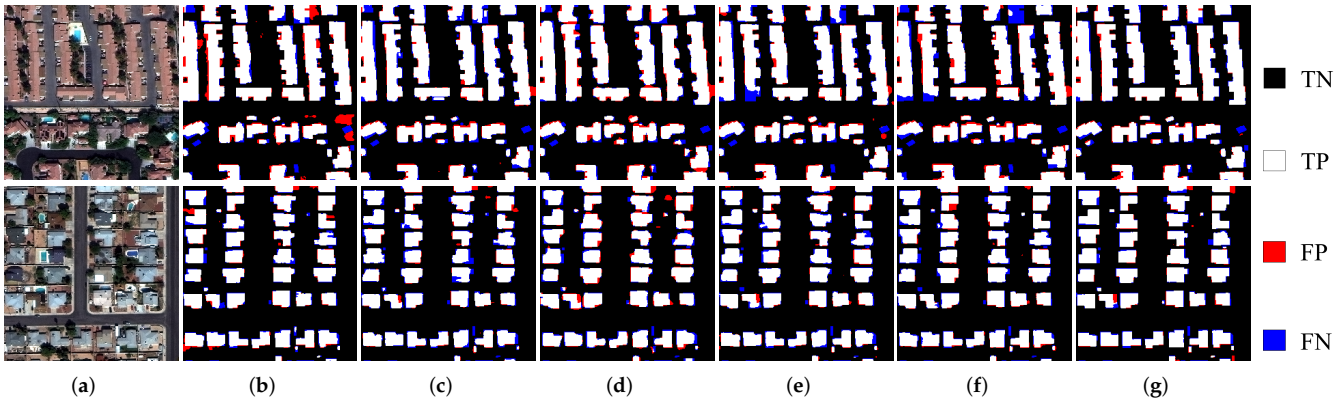


Figure 7. Results of building segmentation in different regions of the Las Vegas dataset using six methods: (**a**) input image, (**b**) U-Net, (**c**) DeepLabV3+, (**d**) PSPNet, (**e**) UPerNet, (**f**) TransUNet, and (**g**) ours. Pixel-based true negatives, true positives, false positives, and false negatives are marked in black, white, red, and blue, respectively.

Table 4. Comparison of the building extraction results obtained by different methods on the Las Vegas test dataset.

Method	OA (%)	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
U-Net	95.4	85.6	87.5	85.9	77.0
DeepLabV3+	95.8	89.3	85.9	87.2	78.5
PSPNet	95.5	87.5	87.3	88.7	78.1
UPerNet	95.7	89.2	86.1	87.2	78.6
TransUNet	95.6	87.8	86.1	86.3	77.4
Ours	**96.6**	**91.2**	**88.8**	**89.6**	**82.4**

The best results are shown in bold. For all metrics, the higher the value is, the better the evaluation.

(2) Khartoum Dataset

We select two representative images from the Khartoum dataset for building extraction. As shown in Figure 8, the proposed method suffers from fewer extraction errors than the other five methods.

The first row of Figure 8 contains many buildings of different shapes and sizes, and the boundaries of some areas are indistinguishable. The second row of Figure 8 shows buildings with an uneven distribution and a complex landscape. The proposed method performs the best visually, while the other methods produce more errors, with results that contain more unsegmented areas and are missing small targets.

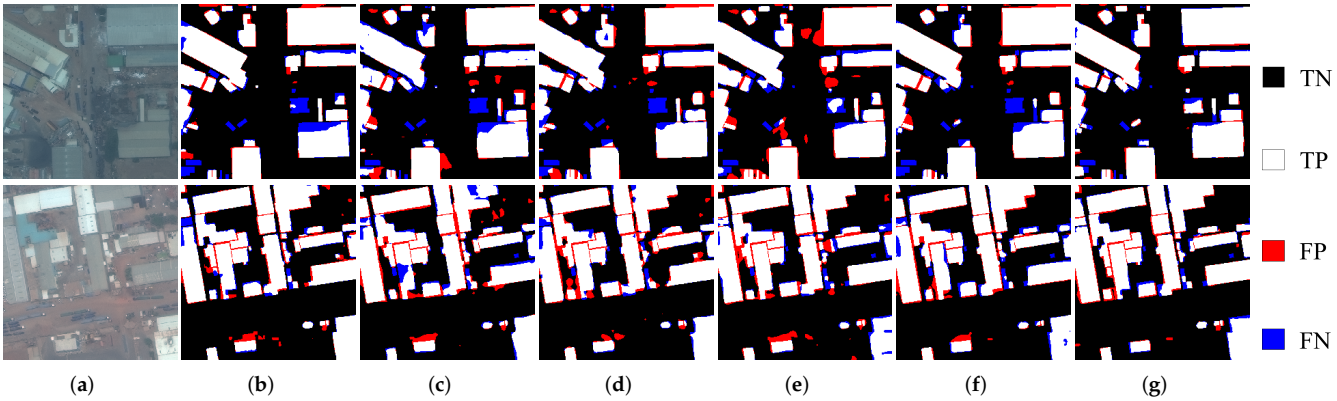


Figure 8. Results of building segmentation in different regions of the Khartoum dataset using six methods: (**a**) input image, (**b**) U-Net, (**c**) DeepLabV3+, (**d**) PSPNet, (**e**) UPerNet, (**f**) TransUNet, and (**g**) ours. Pixel-based true negatives, true positives, false positives, and false negatives are marked in black, white, red, and blue, respectively.

Table 5 presents an evaluation analysis on the Khartoum dataset utilizing the same computational performance measures introduced above. Compared with that of U-Net, the IoU value of our model is improved by 2.4%; similarly, the IoU of our proposed model is improved by 7.5% and 4.3% over those of DeepLabV3+ and TransUNet, respectively. Both DeepLabV3+ and TransUNet require large datasets to reap the benefits of their networks. These evaluation results show that our proposed model is also superior to the other models in terms of the remaining indices, with the exception of precision.

Table 5. Comparison of the building extraction results obtained by different methods on the Khartoum test dataset.

Method	OA (%)	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
U-Net	92.7	**77.3**	71.5	71.5	58.3
DeepLabV3+	91.7	71.7	66.9	67.2	53.2
PSPNet	92.5	72.3	69.8	69.5	56.3
UPerNet	91.7	71.4	68.7	67.1	53.1
TransUNet	92.1	71.8	72.6	70.1	56.4
Ours	**92.8**	76.7	**73.8**	**73.6**	**60.7**

The best results are shown in bold. For all metrics, the higher the value is, the better the evaluation.

(3) Shanghai Dataset

As shown in Figure 9, we again use six different methods to obtain building extraction results for different areas of Shanghai, and we select two images from the test results for detailed analysis.

Figure 9 shows that the buildings in the Shanghai dataset are densely distributed, making it difficult to extract contour information. It can be seen from the segmentation results that the boundary extraction effects for both test images are not ideal. Among the compared methods, DeepLabV3+, PSPNet, and UPerNet do not perform well in generating smooth boundaries, indicating that these methods do not focus on the boundary information in high-resolution remote sensing images. Compared with the other five methods, our proposed method produces fewer segmentation errors while generating more obvious geometric structures.

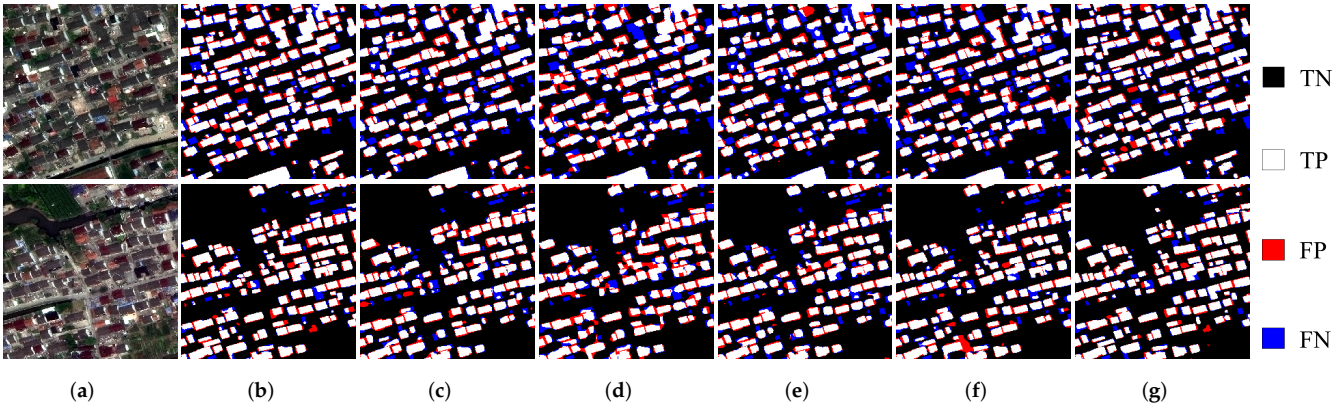


Figure 9. Results of building segmentation in different regions of the Shanghai dataset using six methods: (**a**) input image, (**b**) U-Net, (**c**) DeepLabV3+, (**d**) PSPNet, (**e**) UPerNet, (**f**) TransUNet, and (**g**) ours. Pixel-based true negatives, true positives, false positives, and false negatives are marked in black, white, red, and blue, respectively.

As shown in Table 6, we use the same computational performance measures for model evaluation on the Shanghai dataset. Compared with that of U-Net, the IoU value of the proposed method is increased by 1.3%, which proves that the proposed method can achieve improved building extraction performance. Moreover, the IoU value of the proposed method is 6.0% higher than that of TransUNet, indicating that TransUNet has some shortcomings regarding the processing of complex remote sensing image scenes. Compared with DeepLabV3+, PSPNet, and UPerNet, the proposed model achieves improvements of 4.7%, 1.9%, and 5.6%, respectively. In general, the extraction results of the proposed method show improvement over those of other methods.

Table 6. Comparison of the building extraction results obtained by different methods on the Shanghai test dataset.

Method	OA (%)	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
U-Net	94.5	75.8	**71.7**	72.0	60.2
DeepLabV3+	93.7	73.9	67.5	69.1	56.8
PSPNet	94.4	75.6	70.4	71.4	59.6
UPerNet	93.9	75.2	65.7	68.0	55.9
TransUNet	93.7	71.8	67.4	67.6	55.5
Ours	**94.6**	**77.6**	71.2	**73.1**	**61.5**

The best results are shown in bold. For all metrics, the higher the value is, the better the evaluation.

(4) Paris Dataset

Figure 10 presents the two images selected from the evaluation results for the Paris dataset to demonstrate the validity of our proposed method.

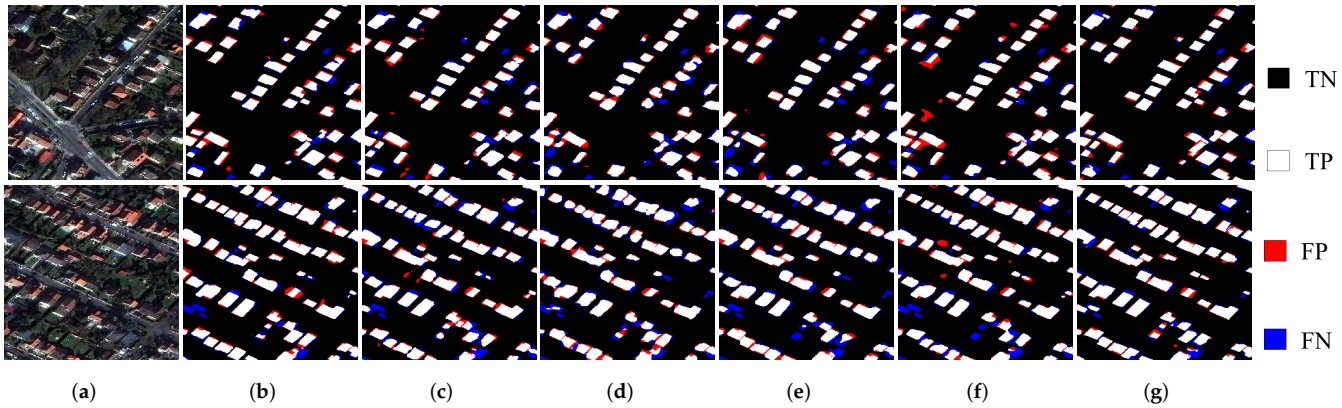


Figure 10. Results of building segmentation in different regions of the Paris dataset using six methods: (**a**) input image, (**b**) U-Net, (**c**) DeepLabV3+, (**d**) PSPNet, (**e**) UPerNet, (**f**) TransUNet, and (**g**) ours. Pixel-based true negatives, true positives, false positives, and false negatives are marked in black, white, red, and blue, respectively.

The first test image in Figure 10 shows building roofs constructed with a variety of materials and shapes. The second test image in Figure 10 contains various small building targets with uniform roofing materials. Although both test images show fairly regular building distributions, the buildings are of different shapes and sizes, and there are many shaded areas around the buildings. As shown in Figure 10, the other five methods produce many unsegmented building areas, whereas the proposed method can obtain better visual results with more obvious geometric features and regular boundaries.

Table 7 shows a quantitative comparison of the building extraction results obtained by the different methods on the Paris test dataset. Compared with those of the other methods, the IoU value of the proposed method is increased by 2.7–7.9%. The OA, precision, recall, F1-score, and IoU of the proposed method are all significantly higher than those of the other methods.

Table 7. Comparison of the building extraction results obtained by different methods on the Paris test dataset.

Method	OA (%)	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
U-Net	95.3	77.6	73.9	73.8	61.9
DeepLabV3+	94.7	74.1	70.3	70.1	57.2
PSPNet	95.3	77.3	72.1	73.6	61.1
UPerNet	94.6	75.8	67.2	69.9	56.7
TransUNet	94.5	69.0	74.7	70.9	57.8
Ours	**95.9**	**78.9**	**74.9**	**75.8**	**64.6**

The best results are shown in bold. For all metrics, the higher the value is, the better the evaluation.

4.7. Parameter Sensitivity Analysis

The three parameters of the total loss function, α_1, α_2, and α_3, need to be tuned for each dataset. α_1 is used to adjust the weight of the multistage loss, whereas α_2 and α_3 are weight parameters for the contour loss and distance loss, respectively. To explore the influence of different parameter values on the building extraction results, we first adjust α_2 and α_3 to find their optimal values, and the results are shown in Figure 11.

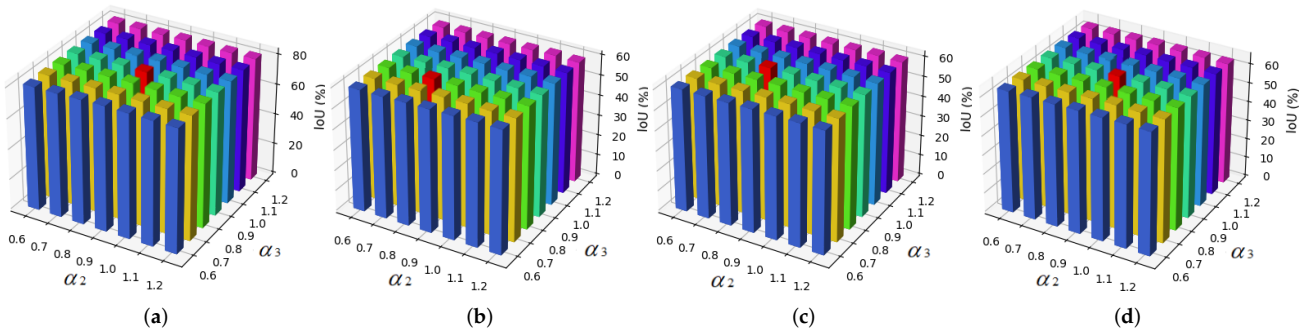


Figure 11. Influences of different parameter values on the IoU values obtained for all four cities (each red pillar represents the best value). (**a**) IoU values on the Las Vegas dataset with different values of the parameters α_2 and α_3. (**b**) IoU values on the Khartoum dataset with different values of the parameters α_2 and α_3. (**c**) IoU values on the Shanghai dataset with different values of the parameters α_2 and α_3. (**d**) IoU values on the Paris dataset with different values of the parameters α_2 and α_3.

Once these two parameters are fixed, we obtain the optimal IoU value by adjusting α_1, as shown in Figure 12.

From Figure 11, it is observed that each city has its own optimal parameter values due to its own distinctive image features: $\alpha_2 = 0.9$ and $\alpha_3 = 0.9$ for Las Vegas, $\alpha_2 = 0.8$ and $\alpha_3 = 0.8$ for Khartoum, $\alpha_2 = 0.8$ and $\alpha_3 = 0.9$ for Shanghai, and $\alpha_2 = 0.9$ and $\alpha_3 = 0.9$ for Paris. As shown in Figure 12, our method obtains the best IoU values with $\alpha_1 = 0.8$ for all four cities, and all cities are insensitive to the parameter α_1.

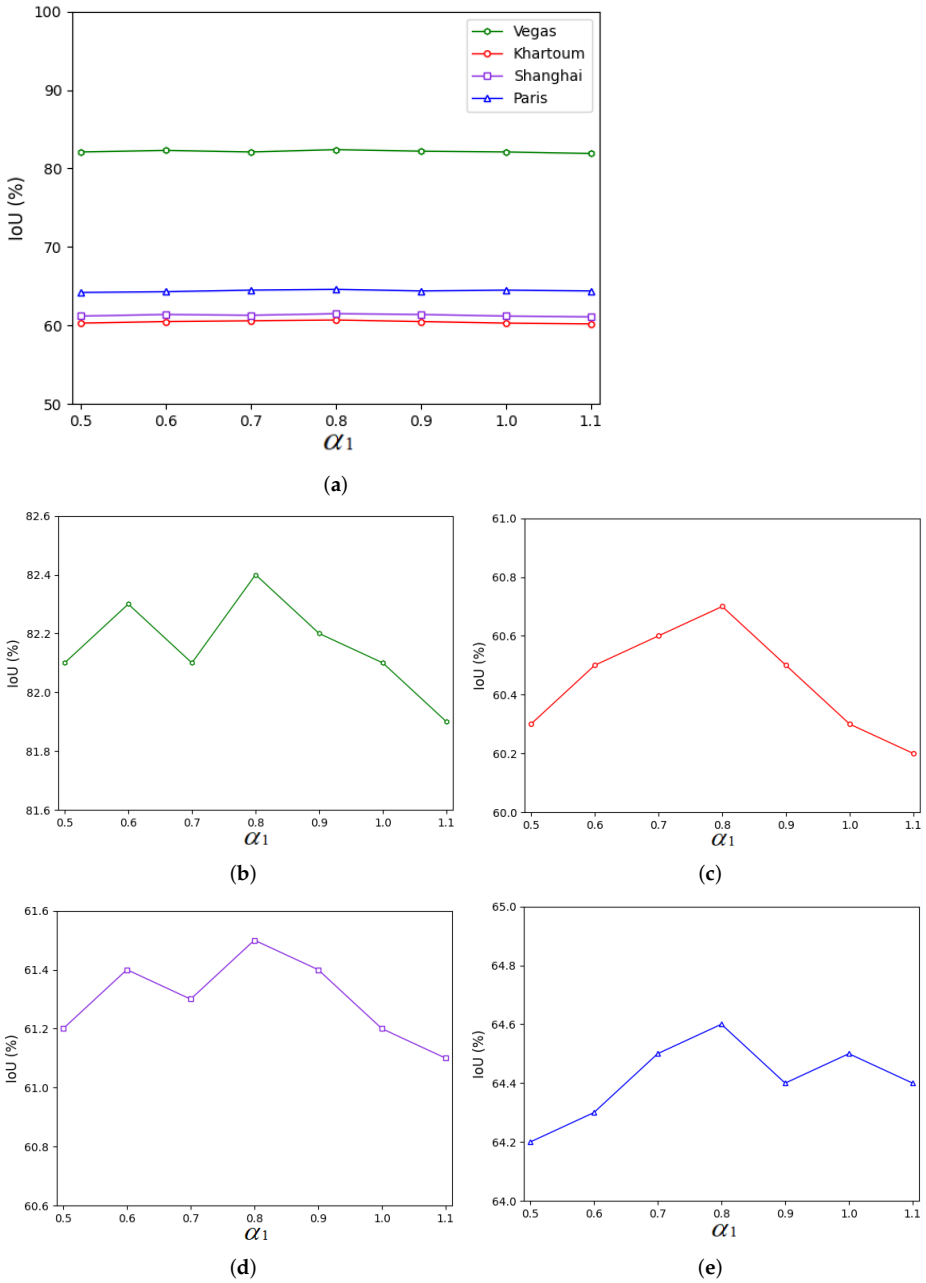


Figure 12. Influences of different parameter values on the IoU values obtained for all four cities. (**a**) IoU values for all four cities with different values of the parameter α_1. (**b**) IoU values on the Las Vegas dataset with different values of the parameter α_1. (**c**) IoU values on the Khartoum dataset with different values of the parameter α_1. (**d**) IoU values on the Shanghai dataset with different values of the parameter α_1. (**e**) IOU values on the Paris dataset with the parameter α_1.

5. Discussion

5.1. Additional Dataset

Another dataset, the Massachusetts building dataset, is used to evaluate the performance of different networks for building extraction. Figure 13 shows some examples from the Massachusetts building dataset. This dataset covers a wider area with relatively smaller building objects, making it more difficult to extract them accurately. In this experiment, this dataset is cut into small patches of 256 × 256 pixels in size with an overlap rate of 0.5. To facilitate training, we remove tiles without buildings. Finally, we obtain 14,745 patches for training, 455 patches for validation, and 1118 patches for testing. Note that all compared methods are trained in the same computational environment as the proposed method.

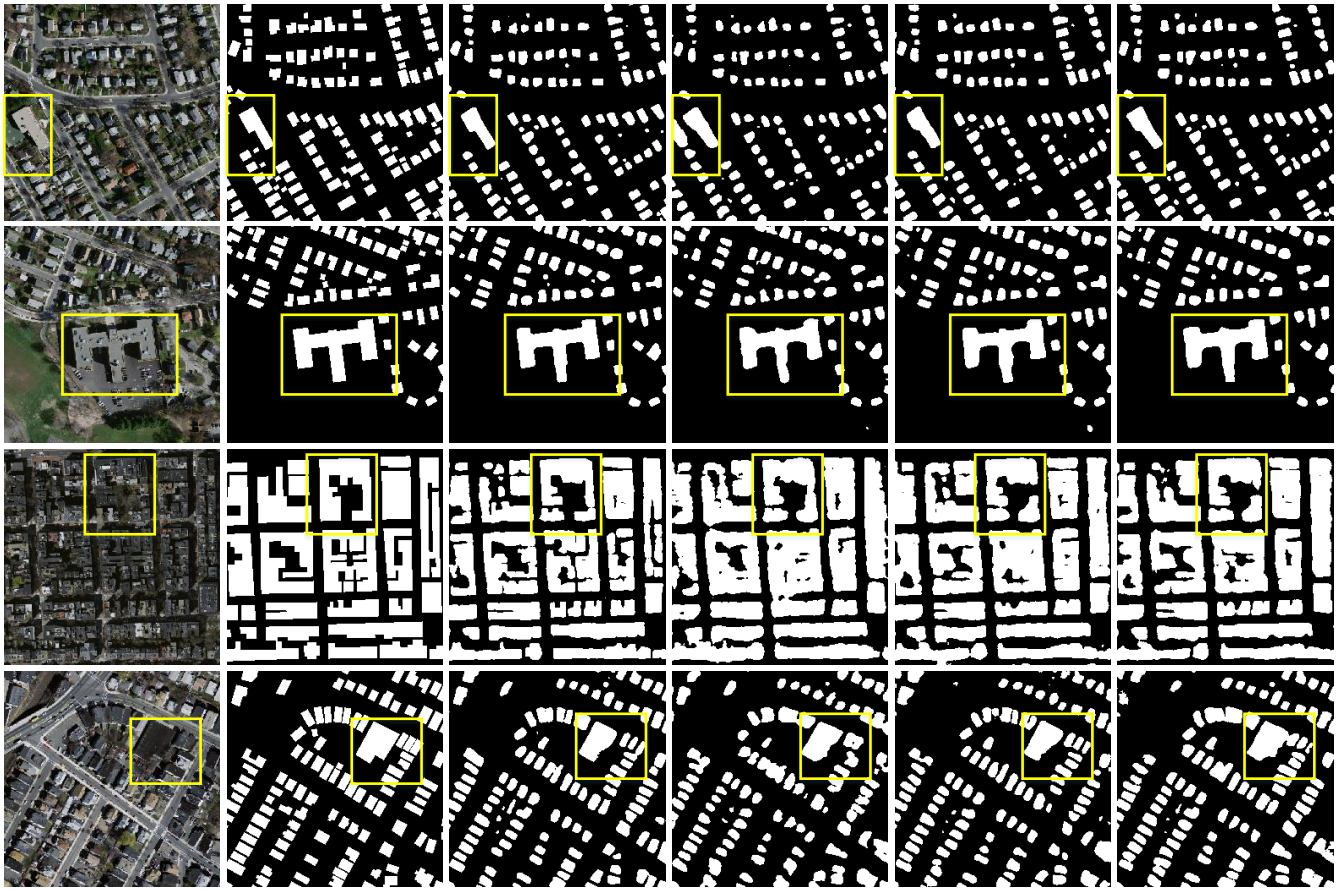
Figure 13. Samples from the Massachusetts building dataset. RGB imagery and corresponding reference maps are displayed in the first and second rows, respectively.

5.2. Comparison with Building Extraction Results from Recent Research

In this section, we further compare the proposed method with several studies on building extraction in recent years to measure its performance. The networks considered for comparison include DDCM-Net [50], MAResU-Net [51], and MANet [52]. DDCM-Net was proposed by Liu et al. [50] and utilizes a combination of dense dilated convolutions with different dilation rates to obtain fused contextual information. MAResU-Net and MANet were proposed by Li et al. [51,52]; in the former, the dot-product attention mechanism is replaced with a linear attention mechanism to reduce the memory and computational costs, and the latter extracts contextual dependencies through multiple kernel attention modules.

Some examples from the test results on the Massachusetts building dataset are shown in Figure 14. The first and second rows of Figure 14 show that our method generates more regular building results than the methods considered for comparison, and the yellow boxes clearly show that the proposed method achieves significant boundary improvements. The third row shows that the proposed method can effectively extract the edge areas of buildings. In particular, the building boundaries extracted by our method are more continuous, while the results of other methods are relatively lacking in continuity. From the yellow boxes in the last row, it can be observed that the proposed method can distinguish the boundaries of different building instances more clearly than other methods, which further proves the effectiveness of the proposed boundary refinement modules.

Table 8 shows the quantitative results on the Massachusetts building dataset. The proposed method achieves the best results in terms of all metrics. In particular, the proposed method obtains an IoU of 73.89% and an F1-Score of 84.98%. Compared with DDCM-Net, MAResU-Net, and MANet, the proposed method achieves improvements of approximately 3.15%, 2.03%, and 1.70%, respectively, in the IoU value. This improvement of the results proves the effectiveness of the proposed method for building extraction.



(a) Original image (b) Ground Truth (c) Ours (d) DDCM-Net (e) MAResU-Net (f) MANet

Figure 14. Qualitative comparisons between our method and several recent studies on the Massachusetts building dataset. (**a**) Original image, (**b**) ground truth, (**c**) prediction map of ours, (**d**) prediction map of DDCM-Net, (**e**) prediction map of MAResU-Net, and (**f**) prediction map of MANet.

Table 8. Comparison of the building extraction results obtained by the proposed method and recent methods on the Massachusetts building dataset.

Method	OA (%)	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
Ours	**94.11**	**88.20**	**81.99**	**84.98**	**73.89**
DDCM-Net [50]	93.30	86.19	79.78	82.86	70.74
MAResU-Net [51]	93.49	85.46	81.87	83.63	71.86
MANet [52]	93.69	87.32	80.64	83.85	72.19

The best results are shown in bold. For all metrics, the higher the value is, the better the evaluation.

6. Conclusions

In this paper, we strive to improve the accuracy of building extraction while refining blurry boundaries. A novel multitask deep learning method was designed to extract buildings from high-resolution remote sensing images. In particular, the proposed method was designed with two auxiliary branches to help refine fuzzy building boundaries, and a multitask loss function guided the network to learn more boundary information. In addition, we proposed several refinement modules to generate accurate prediction results by refining each part of the network. To take full advantage of the information generated in each stage, we calculated a loss using the output of each downsampling step to guide the learning process to consider the advantages of each stage. An ablation study demonstrated the effectiveness of each module in our proposed model, and comparative experiments

showed that the proposed method offers significantly improved building extraction accuracy on a public dataset. Finally, an additional dataset was utilized to further prove the generalization ability and robustness of the proposed model. In future work, we plan to extend our research to the field of instance segmentation and achieve accurate building extraction from high-resolution remote sensing images.

Author Contributions: Conceptualization, H.X.; methodology, H.X. and P.Z.; software, X.L.; validation, T.X. and P.Z.; formal analysis, H.X.; investigation, P.Z.; data curation, H.X.; writing—original draft preparation, H.X.; writing—review and editing, P.Z. and T.X.; visualization, X.L.; supervision, P.Z.; project administration, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grants, grant number 41871226.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The link to download the Massachusetts building dataset can be found in the online version, at https://www.cs.toronto.edu/~vmnih/data/, accessed on 1 December 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tian, J.; Cui, S.; Reinartz, P. Building change detection based on satellite stereo imagery and digital surface models. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 406–417. [CrossRef]
2. Rathore, M.M.; Ahmad, A.; Paul, A.; Rho, S. Urban planning and building smart cities based on the internet of things using big data analytics. *Comput. Netw.* **2016**, *101*, 63–80. [CrossRef]
3. Shackelford, A.K.; Davis, C.H. A combined fuzzy pixel-based and object-based approach for classification of high-resolution multispectral data over urban areas. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 2354–2363. [CrossRef]
4. Zhang, Y. Optimisation of building detection in satellite images by combining multispectral classification and texture filtering. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 50–60. [CrossRef]
5. Su, W.; Li, J.; Chen, Y.; Liu, Z.; Zhang, J.; Low, T.M.; Suppiah, I.; Hashim, S.A.M. Textural and local spatial statistics for the object-oriented classification of urban areas using high resolution imagery. *Int. J. Remote Sens.* **2008**, *29*, 3105–3117. [CrossRef]
6. Ferraioli, G. Multichannel InSAR building edge detection. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1224–1231. [CrossRef]
7. Sun, Z.; Fang, H.; Deng, M.; Chen, A.; Yue, P.; Di, L. Regular shape similarity index: A novel index for accurate extraction of regular objects from remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3737–3748. [CrossRef]
8. Izadi, M.; Saeedi, P. Three-dimensional polygonal building model estimation from single satellite images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *50*, 2254–2272. [CrossRef]
9. Mohan, R.; Nevatia, R. Using perceptual organization to extract 3D structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 1121–1139. [CrossRef]
10. Jin, X.; Davis, C.H. Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information. *EURASIP J. Appl. Signal Process.* **2005**, *2005*, 1–11. [CrossRef]
11. Xiang, Y.; Sun, Y.; Li, C. A rooftop extraction method using color feature, height map information and road information. *Proc. SPIE* **2012**, *8537*, 85370T.
12. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
13. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
14. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
15. Yuan, J. Learning building extraction in aerial scenes with convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2793–2798. [CrossRef]
16. Xu, Y.; Wu, L.; Xie, Z.; Chen, Z. Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. *Remote Sens.* **2018**, *10*, 144. [CrossRef]
17. Huang, J.; Zhang, X.; Xin, Q.; Sun, Y.; Zhang, P. Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network. *ISPRS J. Photogramm. Remote Sens.* **2019**, *151*, 91–105. [CrossRef]
18. Van Etten, A.; Lindenbaum, D.; Bacastow, T.M. Spacenet: A remote sensing dataset and challenge series. *arXiv* **2018**, arXiv:1807.01232.

19. Mnih, V. Machine Learning for Aerial Image Labeling. Ph.D. Thesis, University of Toronto (Canada), Toronto, ON, Canada, 2013.
20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Proc. NIPS* **2012**, *25*, 1097–1105. [CrossRef]
21. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1155–1167. [CrossRef]
22. Zhang, G.; Lu, S.; Zhang, W. CAD-Net: A context-aware detection network for objects in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 10015–10024. [CrossRef]
23. Tsai, Y.H.; Hung, W.C.; Schulter, S.; Sohn, K.; Yang, M.H.; Chandraker, M. Learning to adapt structured output space for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7472–7481.
24. Na, Y.; Kim, J.H.; Lee, K.; Park, J.; Hwang, J.Y.; Choi, J.P. Domain adaptive transfer attack-based segmentation networks for building extraction from aerial images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5171–5182. [CrossRef]
25. Ji, S.; Wei, S.; Lu, M. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 574–586. [CrossRef]
26. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
27. Tabernik, D.; Šela, S.; Skvarč, J.; Skočaj, D. Segmentation-based deep-learning approach for surface-defect detection. *J. Intell. Manuf.* **2020**, *31*, 759–776. [CrossRef]
28. Russo, P.; Tommasi, T.; Caputo, B. Towards multi-source adaptive semantic segmentation. In *International Conference on Image Analysis and Processing*; Springer: Heidelberg/Berlin, Germany, 2019; pp. 292–301.
29. Liu, C.; Chen, L.C.; Schroff, F.; Adam, H.; Hua, W.; Yuille, A.L.; Fei-Fei, L. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 82–92.
30. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
31. Hui, J.; Du, M.; Ye, X.; Qin, Q.; Sui, J. Effective building extraction from high-resolution remote sensing images with multitask driven deep neural network. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 786–790. [CrossRef]
32. Bischke, B.; Helber, P.; Folz, J.; Borth, D.; Dengel, A. Multi-task learning for segmentation of building footprints with deep neural networks. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1480–1484.
33. Guo, H.; Shi, Q.; Du, B.; Zhang, L.; Wang, D.; Ding, H. Scene-driven multitask parallel attention network for building extraction in high-resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 4287–4306. [CrossRef]
34. Li, A.; Jiao, L.; Zhu, H.; Li, L.; Liu, F. Multitask Semantic Boundary Awareness Network for Remote Sensing Image Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*. [CrossRef]
35. Chen, H.; Qi, X.; Yu, L.; Heng, P.A. DCAN: Deep contour-aware networks for accurate gland segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2487–2496.
36. Tan, C.; Zhao, L.; Yan, Z.; Li, K.; Metaxas, D.; Zhan, Y. Deep multi-task and task-specific feature learning network for robust shape preserved organ segmentation. In Proceedings of the IEEE 15th International Symposium on Biomedical Imaging (ISBI), Washington, DC, USA, 4–7 April 2018; pp. 1221–1224.
37. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large kernel matters–improve semantic segmentation by global convolutional network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4353–4361.
38. Zhang, L.; Dai, J.; Lu, H.; He, Y.; Wang, G. A bi-directional message passing model for salient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 1741–1750.
39. Qin, X.; Zhang, Z.; Huang, C.; Gao, C.; Dehghan, M.; Jagersand, M. Basnet: Boundary-aware salient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7479–7489.
40. Cheng, B.; Girshick, R.; Dollár, P.; Berg, A.C.; Kirillov, A. Boundary IoU: Improving Object-Centric Image Segmentation Evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 15334–15342.
41. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
42. Hinton, G.; Srivastava, N.; Swersky, K. Neural networks for machine learning. *Coursera Video Lect.* **2012**, *264*, 2146–2153.
43. Dodge, J.; Ilharco, G.; Schwartz, R.; Farhadi, A.; Hajishirzi, H.; Smith, N. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv* **2020**, arXiv:2002.06305.
44. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
45. Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; Sun, J. Unified perceptual parsing for scene understanding. In Proceedings of the European Conference Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 418–434.

46. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**, arXiv:2102.04306.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
48. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
49. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
50. Liu, Q.; Kampffmeyer, M.; Jenssen, R.; Salberg, A.B. Dense dilated convolutions merging network for land cover classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 6309–6320. [CrossRef]
51. Li, R.; Zheng, S.; Duan, C.; Su, J.; Zhang, C. Multistage Attention ResU-Net for Semantic Segmentation of Fine-Resolution Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*. [CrossRef]
52. Li, R.; Zheng, S.; Zhang, C.; Duan, C.; Su, J.; Wang, L.; Atkinson, P.M. Multiattention network for semantic segmentation of fine-resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*. [CrossRef]