Article
A Multiscale and Multitask Deep Learning Framework for Automatic Building Extraction

Jichong Yin 1, Fang Wu 1,*, Yue Qiu 1, Anping Li 2, Chengyi Liu 1 and Xianyong Gong 1

1 Institute of Geospatial Information, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China
2 78098 Troops, Chengdu 610000, China
* Correspondence: wufang_630@126.com

Abstract: Detecting buildings, segmenting building footprints, and extracting building edges from high-resolution remote sensing images are vital in applications such as urban planning, change detection, smart cities, and map-making and updating. The tasks of building detection, footprint segmentation, and edge extraction affect each other to a certain extent. However, most previous works have focused on one of these three tasks and have lacked a multitask learning framework that can simultaneously solve the tasks of building detection, footprint segmentation and edge extraction, making it difficult to obtain smooth and complete buildings. This study proposes a novel multiscale and multitask deep learning framework to consider the dependencies among building detection, footprint segmentation, and edge extraction while completing all three tasks. In addition, a multitask feature fusion module is introduced into the deep learning framework to increase the robustness of feature extraction. A multitask loss function is also introduced to balance the training losses among the various tasks to obtain the best training results. Finally, the proposed method is applied to open-source building datasets and large-scale high-resolution remote sensing images and compared with other advanced building extraction methods. To verify the effectiveness of multitask learning, the performance of multitask learning and single-task training is compared in ablation experiments. The experimental results show that the proposed method has certain advantages over other methods and that multitask learning can effectively improve single-task performance.

Keywords: building extraction; multilevel and multitask; deep learning framework; high-resolution remote sensing image

1. Introduction

Buildings, as an integral part of human life, are among the most important elements on a map. Accordingly, building extraction is extremely important in urban planning, land use analysis, and map-making. With the rapid development of earth observation technology, the available spatial resolution of remote sensing imagery has increased year by year. By capturing rich and detailed information on ground objects, high-resolution remote sensing images enable fine extraction of ground objects (e.g., buildings and roads), providing important data support for the automatic extraction of large-scale buildings. However, because of the complexity of high-resolution remote sensing images and the lag in the development of extraction techniques [1], the automatic and precise extraction of buildings in high-resolution remote sensing images has remained a central challenge in remote sensing applications and cartography.

Most early building extraction methods including the mathematical morphology-based methods [2–5] and methods based on shape, color, and texture features [6–9], relied on manually extracted features for judgment. However, due to their limited capabilities for image expression, manually designed features are usually applicable only to specific regions and provide minimal support for model generalization. In recent years, deep
learning technology has been widely applied in various fields related to computer vision and image processing, such as image classification [10], object detection [11], and image segmentation [12,13]. Considering the similarity between building extraction from remote sensing images and computer vision tasks, some building extraction methods based on deep learning technology [14–16] have been applied in remote sensing, enhancing the intelligence of building extraction methods to a new level. Compared with traditional building extraction methods that rely on manually designed features, deep learning methods have the advantage of powerful feature representation capabilities, enabling them to address more complex tasks.

In the computer vision field, building extraction tasks are commonly divided into three categories: building detection tasks, building footprint segmentation tasks, and building edge extraction tasks.

Building detection involves recognizing each building instance in a remote sensing image, applying object recognition techniques to obtain the location of each building instance as a rectangular bounding box, and determining the quantity of buildings. In recent years, as deep learning technology has rapidly advanced, a series of outstanding object detection algorithms have become available. These algorithms can be approximately divided into two categories. One category includes the two-stage object detection algorithms represented by the region-based convolutional neural network (RCNN) [17], Fast R-CNN [18] and Faster R-CNN [19] methods, whose main concept is to generate regional proposal boxes first and then input them into a convolutional neural network (CNN) for further classification. The other category includes single-stage object detection algorithms represented by the single-shot multibox detector (SSD) [20] and You Only Look Once (YOLO) [21] series of models, which constitute an end-to-end object detection framework that can directly output the category of each detected object. The aforementioned object detection methods have been applied for building detection in remote sensing images. Based on Faster R-CNN, Ding et al. [22] used deformable convolution to improve the adaptability to arbitrarily shaped collapsed buildings and proposed a new method of estimating the intersected proportion of objects (IPO) to describe the degrees to which bounding boxes intersect, thus offering better detection precision and recall. Bai et al. [23] proposed a Faster R-CNN method based on DRNet and ROI Align and utilized texture information to solve region mismatch problems. Building detection method can approximately recognize building locations but cannot achieve pixel-level segmentation, which is more accurate.

Building footprint segmentation refers to the pixel-level segmentation of remote sensing images, in which each pixel in an image is assigned either a building or nonbuilding label. In most building footprint segmentation methods, the fully convolutional network (FCN) architecture [12] or one of its variants is used as the basic architecture, and various measures are implemented to improve the multiscale learning capability of the model. Xie et al. [24] proposed MFCNN, a symmetric CNN with ResNet [25] as the feature extractor, which contains many complex designs, such as dilated convolution units and pyramid feature fusion. MAP-Net, proposed by Zhu et al. [26], has an HRNet-like [27] architecture with multiple feature encoding branches and a channel attention mechanism. Ma et al. [28] proposed the global and multiscale encoder–decoder network (GMEDN), which consists of a U-Net-like [29] network and a nonlocal modeling unit. These methods have greatly enhanced the accuracy of footprint segmentation for differently sized buildings in remote sensing images. However, the aforementioned methods focus only on distinguishing between building and nonbuilding pixel values and rarely closely observe building edge information, often resulting in blurred contours and failure to obtain regular boundaries in the segmentation results.

Building edge extraction refers to marking and extracting the outer boundaries of building instances in remote sensing images. Building edge extraction methods prioritize building boundary information and attempt to reach beyond pixel-level footprint segmentation to directly obtain regular and accurate building contour lines. Recently, some researchers have introduced building boundary information into deep learning networks.
to improve their building extraction accuracy. Lu et al. [30] adopted a deep learning network to extract building edge probability maps from remote sensing images and applied postprocessing to the edge probability maps based on geometrical morphological analysis to achieve refined building edge extraction. Wu et al. [31] proposed a novel deep FCN architecture, named the boundary regulated network (BRNet) architecture, which utilizes local and global information to simultaneously predict segments and contours to achieve better building segmentation and more accurate contour extraction. Jiwani et al. [32] improved the DeepLabV3+ [33] model by introducing a feature pyramid network (FPN) [34] module to achieve cross-scale feature extraction and designing a special weighted boundary diagram to penalize incorrect predictions of building boundaries. Li et al. [35] combined a graph-based conditional random field model with a segmentation network to preserve clear boundaries and fine-grained segmentation. However, due to the structural diversity of buildings and their complex environments, accurately locating and recognizing building edges remain significant challenges.

As a basis for interpreting remote sensing images, building detection provides a foundation for coping with higher-level interpretation tasks, such as building footprint segmentation and building edge extraction. Building detection determines the general locations for building footprint segmentation and edge extraction, while building footprint segmentation and edge extraction allow building shape features to be enhanced for building detection. Simultaneously, building footprint segmentation provides closed shape information for edge extraction, while building edge extraction yields exact boundary information for footprint segmentation. To an extent, building detection, building footprint segmentation, and building edge extraction have a symbiotic relationship of mutual dependence and information complementarity. Nevertheless, although many deep learning-based methods have been used to extract buildings and achieved good performance, the existing methods have been developed for certain tasks, e.g., footprint segmentation of buildings. Hence multitask learning frameworks are required to simultaneously perform multiple tasks, e.g., the detection, footprint segmentation, and edge extraction of the buildings.

Multitask learning can improve the performance on each task by learning better feature representations from the shared information for multiple related tasks. The classic instance segmentation framework Mask R-CNN [13] is based on the object detection framework Faster R-CNN with the addition of a branch for object mask prediction. It first locates the objects in an image and then segments the target objects in the positioning boxes, effectively combining the semantic segmentation and object detection tasks to facilitate their mutual performance enhancement. Considering the symbiotic relationship between road detection and centerline extraction, Lu et al. [36] proposed the MSMT-RE framework for performing these two tasks simultaneously, and this framework has delivered excellent road detection results. MultiNet [37] consists of a shared encoder and three independent decoders for simultaneously completing the three scene perception tasks: scene classification, object detection, and drivable area segmentation. Wu et al. [38] proposed the panoptic driving perception network YOLOP to simultaneously perform the tasks of traffic object detection, drivable area segmentation, and lane detection, significantly improving performance on each single task. Bischke et al. [39] adopted a multitask learning framework to combine the learning of boundaries and semantic information to improve the semantic segmentation of building boundaries. As seen above, multitask learning methods have been widely applied in segmentation tasks, but there is a lack of multitask learning frameworks that can solve the tasks of building detection, footprint segmentation and edge extraction simultaneously.

To address the abovementioned problems, this study proposes a multiscale and multitask deep learning framework called MultiBuildNet for simultaneously performing the building detection, footprint segmentation, and edge extraction tasks. This framework is also integrated with a multiscale feature fusion network to combine features from different scales, aiming to improve the robustness of feature extraction against complex backgrounds. In addition, to minimize the loss function across the three tasks, this study introduces a
multitask loss function that fully considers any deviations between the predicted values and true values in all three tasks to obtain the best training effect.

The main contributions of this study are as follows:

(1) An effective multiscale and multitask deep learning framework is designed that can simultaneously handle the three key tasks in building extraction: building detection, footprint segmentation, and edge extraction.

(2) A multitask loss function is introduced that can address the imbalances between positive and negative samples and among sample categories in the three tasks.

The remainder of this study is organized as follows. The proposed MultiBuildNet framework is introduced in detail in Section 2. Then, Section 3 describes experiments conducted with the proposed method on open-source building datasets and large-scale high-resolution remote sensing images and presents comparisons with other advanced building extraction methods. Ablation experiments conducted to compare the performances of multitask learning and single-task training in order to verify the effectiveness of multitask learning are also reported. Section 4 discusses the performance improvements achieved by the MultiBuildNet framework compared with other deep learning methods as well as its limitations and prospects for future work. Finally, conclusions are presented in Section 5.

2. Materials and Methods

The MultiBuildNet framework essentially possesses an encoder–decoder architecture. The encoder part consists of a backbone network and a neck network, while the decoder primarily includes specific networks for the three building extraction tasks. As illustrated in Figure 1, the MultiBuildNet framework consists of three main parts:

![Figure 1. Flowchart of automatic building extraction based on the MultiBuildNet framework for very high-resolution remote sensing images.](image)

(1) A multiscale feature extraction and fusion network. A spatial pyramid pooling (SPP) [40] module and a FPN module are integrated into the encoder portion of the MultiBuildNet framework. The framework can generate and fuse features at different scales and different semantic levels by performing multiscale feature extraction in the spatial context to improve the robustness of feature extraction.

(2) A building extraction module based on multitask learning. The MultiBuildNet framework includes a multitask learning module based on YOLACT [41], which can simultaneously train networks for building detection, footprint segmentation, and edge extraction.
(3) A multitask loss function. A multitask loss function is introduced into the MultiBuildNet framework to avoid situations in which the multiple tasks are not sufficiently collaborative in relation to the network weights and to balance the joint training process for all tasks.

2.1. Multiscale Feature Extraction and Fusion Network

Convolutional layers of different sizes can generally extract feature information at different semantic levels. Therefore, this study introduces a multiscale feature extraction and fusion network, which incorporates a SPP module and a FPN module in the encoder portion while conducting multiscale encoding in the spatial context to yield rich spatial information that can be used to fuse features at different scales and different semantic levels, thus enhancing the robustness of feature extraction in complex environments.

Figure 2 shows the multiscale feature extraction and fusion network. YOLACT serves as the backbone network. The neck network fuses the features generated by the backbone network and is formed by combining a SPP module and a FPN module. In addition, multiple parallel convolutional layers of different sizes are combined in place of the resampling operation to achieve information extraction at different semantic levels and feature extraction at different scales. The convolutional layer for each level integrates convolution kernels of three different sizes, $1 \times 1$, $3 \times 3$, and $5 \times 5$, each of which generates a corresponding number of feature maps. The cascaded multiscale features are fed to the next convolution operation and subjected to a cubic convolution operation on the contracting path.

![Figure 2. Multiscale feature extraction and fusion network.](image)

2.2. Building Extraction Module Based on Multitask Learning

As shown in Figure 3, the building extraction module based on multitask learning consists of three networks, namely, a building detection network, a footprint segmentation network, and an edge extraction network, all of which run in parallel during training.
Figure 3. Building extraction module based on multitask learning.

The building extraction module takes red–green–blue (RGB) images as input. In the encoder part, there are four repeated groups of convolutional layers, where each group consists of two convolutional (Conv) layers, each followed by a batch normalization (BN) layer and a rectified linear unit (ReLU) activation function. Every group is followed by a max-pooling (Pooling) layer for downsampling. The decoder part primarily consists of the specific networks for each of the three building extraction tasks.

The building detection network adopts a multiscale detection scheme based on anchor boxes. The encoder part transfers semantic features from top to bottom through the FPN, while the decoder part uses a path aggregation network (PAN) [42] to transfer localization features from bottom to top. The combination of the FPN and PAN in the building detection network achieves an improved feature fusion effect and allows building detection to be performed using multiscale fusion feature maps in the PAN. Each grid point of the multiscale feature maps is used to generate three prior anchor boxes with different aspect ratios. The building detection network head predicts the position offset and the longitudinal and transverse scaling for each prior anchor box and the corresponding probabilities of building instances and predicted confidence values.

The building footprint segmentation network and building edge extraction network have the same network structure. The final prediction map is obtained by upsampling the underlying feature map from the FPN three times to restore the output feature map to the input image size. Every upsampling layer is followed by a deconvolutional (DConv) layer, a BN layer and a ReLU activation. Every group is followed by a convolutional layer. In addition, instead of the deconvolution operation, the bilinear interpolation method is adopted in the upsampling layers to reduce computational costs. Thus, the building contour segmentation network not only delivers high-performance segmentation results but also ensures an efficient inference speed.

2.3. Multitask Loss Function

Multitask learning is actually a problem of multiobjective optimization, and one of the complex challenges in solving such a problem is the optimization process itself, i.e., designing an appropriate loss function. In the extreme case in which the loss value for one task is considerable, but those for the other tasks are very small, multitask learning approximately degenerates into single-task learning and no longer has the advantage of multitask information sharing. Therefore, to avoid a situation in which one or more tasks dominate the network weights during the training process, an appropriate loss function is necessary to balance the loss values of each task. To this end, this study introduces a
multitask loss function $L_{\text{ext}}$, which consists of three components: a building detection loss function $L_{\text{det}}$, a building footprint segmentation loss function $L_{\text{seg}}$, and a building edge extraction loss function $L_{\text{edg}}$. The symbols used for the loss functions in this paper and their meanings are shown in Table 1. The multitask loss function is a weighted sum of all three parts, as shown in Equation (1):

$$L_{\text{ext}} = a_1 L_{\text{det}} + a_2 L_{\text{seg}} + a_3 L_{\text{edg}}$$

### Table 1. The symbols used for the loss functions in this paper and their meanings.

| Symbol | Meaning |
|--------|---------|
| $L_{\text{ext}}$ | Multitask loss function |
| $L_{\text{det}}$ | Building detection loss function |
| $L_{\text{seg}}$ | Building footprint segmentation loss function |
| $L_{\text{edg}}$ | Building edge extraction loss function |
| $L_{\text{cla}}$ | Classification loss function |
| $L_{\text{obj}}$ | Objective loss function |
| $L_{\text{box}}$ | Bounding box loss function |
| $L_{\text{fl}}$ | Focal loss function |
| $L_{\text{CIoU}}$ | Complete intersection over union loss function |
| $L_{ce}$ | Cross-entropy loss function |

The building detection loss function is a weighted sum of the classification loss $L_{\text{cla}}$, the objective loss $L_{\text{obj}}$, and the bounding box loss $L_{\text{box}}$, as shown in Equation (2):

$$L_{\text{det}} = \beta_1 L_{\text{cla}} + \beta_2 L_{\text{obj}} + \beta_3 L_{\text{box}}$$

The focal loss $L_{\text{fl}}$ is adopted for the classification loss and objective loss, which is used to control the weights of positive and negative samples and of difficult and easy samples. Two penalty factors are introduced to reduce the weights of easy samples so that the model can focus more on difficult samples during the training process, thus solving the issue of unbalanced sample categories. The classification loss is used to penalize problems in the binary classification of building/nonbuilding samples, while the objective loss is used to penalize confidence in building instance predictions. The complete intersection over union loss function $L_{\text{CIoU}}$ is adopted for the bounding box loss. It is used to penalize the predicted frame confidence and takes into account three important geometric factors, namely, the overlap area between the predicted and real frames, the center point distance, and the aspect ratio, as shown in Figure 4. In this figure, the solid yellow rectangle represents the real frame for building detection, the solid green rectangle represents the predicted frame for building detection, the dotted black rectangle indicates the minimum closed area that can cover both the predicted and real frames, $c$ is the diagonal length of the minimum closed area covering both the predicted and real frames, and $d$ is the distance between the center points of the predicted and real frames. The focal loss and the complete intersection over union loss are calculated as shown in Equations (3) and (4), respectively:

$$L_{\text{fl}} = - (a(1 - \hat{p})^\gamma p \log(\hat{p}) + (1 - a)(1 - \hat{p})^\gamma \log(1 - \hat{p}))$$

$$\begin{aligned}
    d &= \rho^2 \left( b_1 b_2 \right) \\
    \beta &= \frac{v}{1 - \text{IoU}} \\
    v &= \frac{4}{\pi} \left( \arctan \frac{w_{gt}}{h_{gt}} - \arctan \frac{w_{pr}}{h_{pr}} \right)^2 \\
    L_{\text{CIoU}} &= 1 - \text{IoU} + \frac{d}{\hat{c}} + \beta v
\end{aligned}$$

---

**Note:** The symbols $\rho$, $b_1$, $b_2$, $\hat{c}$, $\text{IoU}$, $w_{gt}$, $h_{gt}$, $w_{pr}$, $h_{pr}$ are used in the equations but not defined in the text. The expressions for $d$, $\beta$, and $v$ are given in terms of $\rho$, $b_1$, $b_2$, $c$, and $d$. The complete intersection over union loss function $L_{\text{CIoU}}$ is used to penalize the predicted frame confidence and takes into account three important geometric factors, namely, the overlap area between the predicted and real frames, the center point distance, and the aspect ratio.
Figure 4. Calculation of geometric consistencies between predicted and real frames of building.

For the building footprint segmentation loss function, the cross-entropy loss $L_{ce}$ is adopted to minimize the classification errors between the predicted pixels and building instances.

$$L_{ce} = - (p \log(\hat{p}) + (1 - p) \log(1 - \hat{p}))$$

(5)

On the basis of the cross-entropy loss, the building edge extraction loss function additionally includes the complete intersections over union loss to improve the prediction performance for building edges in sparse areas, as shown in Equation (6):

$$L_{edg} = L_{ce} + L_{CIoU}$$

(6)

where $\hat{p}$ is the probability for a model-predicted sample and $p$ is the sample label. For positive samples, $p$ is 1; otherwise, $p$ is 0. $w$, $h$, $w^\text{gt}$, and $h^\text{gt}$, respectively, represent the widths and heights of the predicted and real frames; IoU is the intersection over union of the building pixels, i.e., the ratio between the intersection and union of the predicted and real building pixels; $v$ is the similarity between the aspect ratios of the predicted and real building frames; $\alpha$ is a penalty factor for negative samples, with a range of [0, 1], which is used to control the ratio of positive to negative samples; $\gamma$ is the modulation coefficient, with a range of [0, +\infty], which is used to distinguish the complexity of samples and enables the model to focus more on difficult samples by reducing the weights of easy samples during training; $\beta$ is a balance factor for the overlapping areas of the predicted and real building frame; $b$ and $b^\text{gt}$ represent the center points of the predicted frame and real frame, respectively; $d$ is the distance between the center points of the predicted and the real frames, respectively; and $c$ is the diagonal length of the smallest closed area that can cover both the predicted and the real frames.

3. Experiments and Results

3.1. Experimental Setup

In this study, three groups of experiments were conducted to verify the effectiveness of the proposed MultiBuildNet framework: experiments based on open-source building datasets, experiments based on large-scale high-resolution remote sensing images, and ablation experiments. The hardware used for the experiments has the following specifications: an Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz, 128 GB of RAM, and an NVIDIA Corporation (https://www.nvidia.cn/) TU102GL [Quadro RTX 6000/8000] graphics card, with Ubuntu 20.04 as the operating system and Python 3.7 and PyTorch (https://pytorch.org/) 1.7.1 as the programming environments.
3.2. Accuracy Evaluation Indicators

To thoroughly and quantitatively evaluate the performance of the MultiBuildNet framework, Recall and AP50 are adopted in this study as evaluation indicators for building detection accuracy. In addition, six indicators, specifically Recall, Precision, F1-score, IoU (intersection over union), mIoU (mean IoU), and Kappa, are used to evaluate the precision of building footprint segmentation and edge extraction. Recall refers to the proportion of correctly predicted building pixels among all real building pixels, Precision refers to the proportion of correctly predicted building pixels among all predicted building pixels, the F1-score is a combined metric considering both Recall and Precision, IoU refers to the ratio between the intersection and union of the predicted and real building pixels, mIoU is the mean of the IoU mean values for buildings and background, Kappa reflects the accuracy for both buildings and background, and AP50 represents the building detection accuracy when detected buildings with an IoU greater than 50% are regarded as correct. The above indicators are calculated using the following equations [43–45]:

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (7)
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (8)
\]

\[
F1 - \text{score} = \frac{2TP}{2TP + FN + FP} \quad (9)
\]

\[
\text{IoU} = \frac{TP}{TP + FP + FN} \quad (10)
\]

\[
\text{mIoU} = \frac{1}{2} \left( \frac{TP}{TP + FP + FN} + \frac{TN}{TN + FN + FP} \right) \quad (11)
\]

\[
\begin{align*}
P_0 &= \frac{TP + FN}{(TP + FN)(TP + FP) + (FN + TN)(FP + TN)} \\
P_e &= \frac{(TP + FN + FP)^2}{(TP + FN + FP) + (FN + TN)(FP + TN)} \\
\text{Kappa} &= \frac{P_0 - P_e}{1 - P_e} \quad (12)
\end{align*}
\]

where \(TP\) represents the number of correctly predicted building pixels, \(TN\) represents the number of correctly predicted background pixels, \(FP\) represents the number of background pixels incorrectly predicted to be building pixels, and \(FN\) represents the number of building pixels incorrectly predicted to be background pixels.

3.3. Experiments Based on Open-Source Building Datasets

3.3.1. Dataset

In this experiment, we used the WHU building dataset (WHU) [14], the Massachusetts buildings dataset (Massachusetts) [46] and the remote sensing imagery for building extraction (RSIBE) dataset [47] to evaluate the performance of the proposed MultiBuildNet framework. The original images in these datasets are orthoimages derived from very high-resolution aerial imagery. The details of each dataset are listed in Table 2.

| Dataset     | Resolution (m) | Pixels       | Area (km²) | Tiles        |
|-------------|----------------|--------------|------------|--------------|
| WHU         | 0.3            | 512 × 512    | 450        | 8189         |
| Massachusetts | 1.0          | 1500 × 1500  | 240        | 151          |
| RSIBE       | 0.075          | 1024 × 1024  | 525        | 103,439      |

Among the above datasets, the RSIBE dataset natively supports multitask learning. We reconstructed the detection labels and edge labels from the segmentation labels provided by the WHU and Massachusetts datasets for multitask building learning. Figure 5 shows the dataset images and labels used in our experiment.
3.3.2. Results of Building Detection

In this experiment, the MultiBuildNet framework was used to perform building detection on the three open-source building datasets and was compared with four classic object detection methods: YOLOv4, Faster R-CNN, Mask R-CNN, and YOLACT. The visualized results are shown in Figures 6–8. Among the tested methods, YOLOv4 and Faster RCNN can only detect buildings, while Mask RCNN, YOLACT and MultiBuildNet can simultaneously detect buildings and segment building footprints. Figures 6–8 indicate that the MultiBuildNet framework has an advantage over the other models when detecting small buildings, adjacent buildings, buildings with unique shapes, and building groups with large dimensional differences. Other models often fail to detect small buildings shaded by trees, while the MultiBuildNet framework can successfully adapt to building detection tasks in different scenarios. Tables 3–5 compare the quantitative results for the building detection accuracy of the tested methods. According to Tables 3–5, all models achieve satisfactory building detection results on the three open-source datasets, but the method proposed in this study is superior in terms of both Recall and AP50.
Figure 6. Results of building detection on the WHU dataset. (a) Images. (b) Labels. (c) YOLOv4. (d) Faster R-CNN. (e) YOLACT. (f) Mask R-CNN. (g) MultiBuildNet.

Figure 7. Results of building detection on the Massachusetts dataset. (a) Images. (b) Labels. (c) YOLOv4. (d) Faster R-CNN. (e) YOLACT. (f) Mask R-CNN. (g) MultiBuildNet.
Figure 8. Results of building detection on the RSIBE dataset. (a) Images. (b) Labels. (c) YOLOv4. (d) Faster R-CNN. (e) YOLACT. (f) Mask R-CNN. (g) MultiBuildNet.

Table 3. Accuracy of building detection on the WHU dataset.

| Network       | YOLOv4 | Faster R-CNN | YOLACT | Mask R-CNN | MultiBuildNet |
|---------------|--------|--------------|--------|------------|---------------|
| Recall (%)    | 77.4   | 75.3         | 77.2   | 75.9       | 78.5          |
| AP50 (%)      | 79.5   | 77.6         | 79.6   | 78.4       | 80.2          |

Table 4. Accuracy of building detection on the Massachusetts dataset.

| Network       | YOLOv4 | Faster R-CNN | YOLACT | Mask R-CNN | MultiBuildNet |
|---------------|--------|--------------|--------|------------|---------------|
| Recall (%)    | 66.8   | 65.4         | 63.6   | 57.2       | 65.1          |
| AP50 (%)      | 62.5   | 61.9         | 62.3   | 55.5       | 63.4          |

Table 5. Accuracy of building detection on the RSIBE dataset.

| Network       | YOLOv4 | Faster R-CNN | YOLACT | Mask R-CNN | MultiBuildNet |
|---------------|--------|--------------|--------|------------|---------------|
| Recall (%)    | 82.4   | 81.3         | 79.8   | 73.5       | 83.7          |
| AP50 (%)      | 84.4   | 80.6         | 84.1   | 82.8       | 85.1          |

3.3.3. Results of Building Footprint Segmentation

In this experiment, the MultiBuildNet framework was utilized to conduct building footprint segmentation on the three open-source building datasets and was compared with four classic deep learning methods: U-Net, PSPNet [48], HRNet and DeepLabV3+. The visualized results are shown in Figures 9–11. Figures 9–11 indicate that these deep learning methods achieve satisfactory results on the three open-source datasets, and they all can fundamentally and accurately recognize buildings in remote sensing images. However, due to the range limitations of the receptive fields range obtained by their convolution kernels, U-Net, PSPNet, HRNet, and DeepLabV3+ are prone to producing holes in the extracted buildings, and fragmentation can easily occur in areas with dense buildings. In contrast, the MultiBuildNet framework adopts a multiscale feature fusion strategy. It can extract complete large buildings and obtain segmentation results with regular contours, and it does not often overlook small buildings shaded by trees. Even for circular buildings that are difficult to extract, MultiBuildNet can deliver better results than the other models.
Figure 9. Results of building footprint segmentation on the WHU dataset. (a) Images. (b) Labels. (c) U-Net. (d) PSPNet. (e) HRNet. (f) DeepLabV3+. (g) MultiBuildNet.

Figure 10. Results of building footprint segmentation on the Massachusetts dataset. (a) Images. (b) Labels. (c) U-Net. (d) PSPNet. (e) HRNet. (f) DeepLabV3+. (g) MultiBuildNet.
Tables 6–8 compare the results for the building footprint segmentation accuracy of the above methods. As indicated in the tables, the MultiBuildNet framework is superior to the other deep learning models on the RSIBE dataset in terms of all indicators, with an IoU of 93.54%, a Precision of 95.23%, and a Recall of 96.34%, presenting significant performance advantages.

Table 6. Accuracy of building footprint segmentation on the WHU dataset.

| Methods      | Recall (%) | Precision (%) | F1-Score (%) | IoU (%)  | mIoU (%) | Kappa (%) |
|--------------|------------|---------------|--------------|----------|----------|-----------|
| U-Net        | 91.47      | 90.54         | 91.76        | 83.25    | 90.32    | 90.94     |
| PSPNet       | 93.15      | 91.47         | 92.3         | 86.29    | 91.55    | 91.58     |
| DeepLabV3+   | 92.85      | 91.91         | 92.38        | 85.84    | 92.55    | 93.36     |
| HRNet        | 93.83      | 92.15         | 92.98        | 86.88    | 92.08    | 92.99     |
| MultiBuildNet| 94.06      | 93.5          | 93.78        | 88.29    | 93.36    | 92.99     |

Table 7. Accuracy of building footprint segmentation on the Massachusetts dataset.

| Methods      | Recall (%) | Precision (%) | F1-Score (%) | IoU (%)  | mIoU (%) | Kappa (%) |
|--------------|------------|---------------|--------------|----------|----------|-----------|
| U-Net        | 82.57      | 73.54         | 77.79        | 63.65    | 79.35    | 75.26     |
| PSPNet       | 78.08      | 75.35         | 76.69        | 62.19    | 76.14    | 71.47     |
| DeepLabV3+   | 81.47      | 80.5          | 80.99        | 68.05    | 79.88    | 76.67     |
| HRNet        | 77.61      | 77.04         | 77.32        | 63.03    | 78.83    | 74.57     |
| MultiBuildNet| 85.82      | 80.08         | 82.85        | 70.72    | 81.74    | 79.09     |
Table 8. Accuracy of building footprint segmentation on the RSIBE dataset.

| Methods       | Recall (%) | Precision (%) | F1-Score (%) | IoU (%)  | mIoU (%)  | Kappa (%) |
|---------------|------------|---------------|--------------|----------|-----------|-----------|
| U-Net         | 94.96      | 93.86         | 94.16        | 91.48    | 93.61     | 93.53     |
| PSPNet        | 95.13      | 94.03         | 94.33        | 91.80    | 93.80     | 93.73     |
| DeepLabV3+    | 94.13      | 93.84         | 94.14        | 91.44    | 93.58     | 93.50     |
| HRNet         | 94.78      | 93.25         | 94.39        | 91.92    | 93.88     | 93.81     |
| MultiBuildNet | 96.34      | 95.23         | 96.15        | 93.54    | 95.06     | 95.88     |

3.3.4. Results of Building Edge Extraction

In contrast to deep learning networks, such as U-Net, PSPNet, HRNet, and DeepLabV3+, the MultiBuildNet framework proposed in this study can refine the footprint segmentation results to additionally achieve building edge information extraction. Figures 12–14 show the visual results of building edge extraction on the three open-source building datasets obtained using the MultiBuildNet framework. As shown in Figures 12–14, the MultiBuildNet framework can improve unclear building edges to some extent and obtain complete boundaries for building instances of various shapes. However, it still has shortcomings. For example, it still has difficulty extracting the edges of small buildings. Moreover, although the right-angle features of buildings can be effectively maintained, the problem of how to construct the edges of irregular buildings still requires attention. Figure 15 shows the accuracy evaluation of building edge extraction on three open-source datasets using the MultiBuildNet framework. The IoU of buildings trained and tested in the RSIBE dataset is 19.86% higher than that of the Massachusetts dataset, because the RSIBE dataset has high-resolution images and high-accuracy labeling. Meanwhile, the Massachusetts dataset directly uses OSM data as labels, which leads to its relatively poor labeling quality.

![Figure 12](image1)

![Figure 13](image2)

---

**Figure 12.** Results of building edge extraction on the WHU dataset. (a) Images. (b) Labels. (c) Results.

**Figure 13.** Results of building edge extraction on the Massachusetts dataset. (a) Images. (b) Labels. (c) Results.
3.4. Experiments Based on Large-Scale High-Resolution Remote Sensing Images

Due to limited GPU memory, it is often necessary to crop remote sensing images into smaller images when deep learning methods are adopted for automatic building extraction. Hence, deep learning models perform well when inferring the contents of smaller cropped images but poorly when conducting direct inference for larger remote sensing images. In addition, when smaller images are spliced to obtain the building extraction results for a larger area, inaccurate segmentation of objects whose edges have been cut by the cropping process will commonly occur. To verify the MultiBuildNet framework’s performance in building footprint segmentation on large-scale high-resolution remote sensing images, three large-scale aerial images of Christchurch in New Zealand were selected for experimentation, each with a resolution of 0.075 m. Residential, commercial, and industrial areas all appear in the three selected images, each of which has an area of 10,000 $\times$ 10,000 pixels. Figure 16 shows the visualized results of building footprint segmentation obtained from these large-scale high-resolution remote sensing images with U-Net, PSPNet, and MultiBuildNet. The building labels and the segmentation results of U-Net, PSPNet, and MultiBuildNet are shown from top to bottom on the original image. As seen from Figure 16, U-Net performs well when extracting relatively regular buildings in residential areas but yields a large number of missed extractions and incomplete segmentations in commercial and industrial areas. In comparison, although PSPNet can achieve relatively complete building extraction, its extraction effect is poor. It cannot maintain the inherent features of buildings, resulting in tortuous edges and unclear right-angle features in the extracted buildings. Meanwhile, MultiBuildNet demonstrates strong feature extraction capabilities in all three areas, with
complete building edge extraction and few missed or incorrect extractions. The results indicate its significant advantages gained through multitask learning and multifeature fusion. Figure 17 shows the quantitative accuracy comparison of three deep learning methods when extracting building footprints from large-scale remote sensing images. Compared with U-Net and PSPNet, MultiBuildNet has obvious advantages in extracting buildings from large-scale remote sensing images.

Figure 16. Results of building footprint segmentation from large-scale remote sensing images. (a) Images. (b) Labels. (c) U-Net. (d) PSPNet. (e) MultiBuildNet.

Figure 17. Accuracy of building footprint segmentation from large-scale remote sensing images.
3.5. Ablation Experiment

Ablation experiments were designed to compare the performances of the multitask and single-task learning strategies to verify the effectiveness of multitask learning and clarify the significance and role of each network in the multitask learning framework. In these experiments, we tested the building detection (Det), footprint segmentation (Seg) and edge extraction (Edg) modules of the MultiBuildNet framework and various combinations thereof on the RSIBE dataset. Table 9 compares the performance of the two strategies for a single specific task. The models using the multitask learning strategy perform better in terms of all indicators than the models adopting the single-task learning strategy. Similar tasks share some relevant features due to their underlying commonality. In the multitask learning strategy, the tasks of building footprint segmentation and edge extraction are highly similar; therefore, the combination of these two tasks is especially helpful in improving the network performance. The building detection network can use the global information of building objects to better identify buildings and to reduce false and missed detections of buildings. The footprint segmentation network can use the local information of surrounding pixels to better segment buildings and capture more details to achieve more complete segmentation. The building edge extraction network can make use of the global information of polygons to generate better contours and obtain smoother building boundaries.

Table 9. Comparison between the multitask learning strategy and single-task learning strategy.

| Strategy   | Recall (%) | AP50 (%) | Precision (%) | IoU (%) | mIoU (%) |
|------------|------------|----------|---------------|---------|----------|
| Det        | 79.8       | 84.1     | -             | -       | -        |
| Seg        | -          | -        | 93.56         | 90.23   | 93.04    |
| Edg        | -          | -        | 92.03         | 89.53   | 92.34    |
| Det+Seg    | 80.16      | 85.34    | 94.28         | 93.31   | 95.38    |
| Det+Edg    | 81.63      | 84.91    | 93.64         | 92.67   | 94.53    |
| Seg+Edg    | 79.35      | -        | 94.83         | 93.76   | 95.68    |
| Det+Seg+Edg| 83.7       | 85.1     | 95.23         | 93.54   | 95.06    |

4. Discussion

4.1. Regarding the Proposed MultiBuildNet Framework

The multitask learning framework proposed in this paper can accomplish building detection, building footprint segmentation and building edge extraction simultaneously. Such an architecture can obtain smooth and complete building extraction results. Compared with the classic building extraction methods, the proposed MultiBuildNet framework can capture more details and obtain more consistent results. The ablation experiments show that the mutual dependence among building detection, footprint segmentation and edge extraction is helpful for feature fusion and extraction. Because of its multitask learning capabilities, the MultiBuildNet framework can utilize both local information from surrounding pixels to segment buildings and global information from polygons to generate building outlines, thus achieving superior performance.

4.2. Limitations and Future Work

The MultiBuildNet framework is currently suitable for the pixel-level semantic segmentation of buildings, but it cannot generate regular building polygons. When the framework was tested on large-scale images for which the data distribution was different from that of the training set, its ability to extract buildings decreased because of the framework’s limited transfer learning ability. In our future work, a more general deep learning framework will be studied for application to data from different sources. In addition to network learning, additional postprocessing could help to vectorize some irregular building edges and improve the network prediction results. For example, Zhao et al. [49] used building boundary regularization in the postprocessing stage to produce better regularized polygons and achieved good performance. Our team has previously carried out extensive research work based on building simplification [50]. Therefore, in our future research, we will consider
integrating building simplification techniques into the network learning process to achieve end-to-end mapping.

5. Conclusions

This study proposes the MultiBuildNet framework to address the common problems associated with the extraction of buildings from high-resolution remote sensing images, such as incorrect extraction, missed extraction, insufficient integrity, and inaccurate boundaries. The proposed method can generate and fuse features at different scales and different semantic levels by means of its multiscale feature extraction and fusion network and learn rich semantic information in the spatial context, thereby improving the robustness of feature extraction against complex backgrounds. In addition, a multitask learning strategy is adopted to simultaneously perform the tasks of building detection, footprint segmentation, and edge extraction. Through the utilization of shared information among multiple tasks, this strategy allows better feature representations to be learned, thereby improving the performance on each task. Experiments on open-source building datasets and large-scale high-resolution remote sensing images indicate that the multiscale and multitask learning frameworks proposed in this study demonstrate considerable performance superiority over other deep learning methods and can better adapt to building extraction tasks in different scenarios. However, the method is currently applicable only for the raster-level semantic segmentation and instance segmentation of buildings and still produces pixel-based building patches. Methods to extend the building extraction process from the raster level to the vector level and to extract building vector polygons directly from remote sensing images merit further study.

Author Contributions: Conceptualization, J.Y.; methodology, J.Y. and F.W.; software, J.Y.; validation, J.Y. and YQ.; formal analysis, A.L.; investigation, C.L.; resources, X.G.; data curation, J.Y.; writing—original draft preparation, J.Y.; writing—review and editing, J.Y.; visualization, J.Y.; supervision, F.W.; project administration, F.W.; funding acquisition, F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation for Distinguished Young Scholars of Henan Province under grant number 212300410014.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive comments which improved considerably the quality of the final manuscript, and also express their gratitude to Springer Nature Author Services (https://authorservices.springernature.cn/, accessed on 15 September 2022) for the expert linguistic services provided.

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

Abbreviations

| Abbreviation | Full Form |
|--------------|-----------|
| BN           | Batch normalization |
| CIoU         | Complete intersection over union |
| CNN          | Convolutional neural network |
| Conv         | Convolutional |
| CPU          | Central processing unit |
| DConv        | Deconvolutional |
| IPO          | Intersected proportion of objects |
| FCN          | Fully convolutional network |
| FN           | False negative |
| FP           | False positive |
| FPN          | Feature pyramid network |
| mIoU         | Mean intersection over union |
Remote Sens. 2022, 14, 4744

References

1. Grekousis, G.; Moutrakis, G.; Kavouras, M. An overview of 21 global and 43 regional land-cover mapping products. Int. J. Remote Sens. 2015, 36, 5309–5335. [CrossRef]
2. Gavankar, N.L.; Ghosh, S.K. Automatic building footprint extraction from high-resolution satellite image using mathematical morphology. Eur. J. Remote Sens. 2018, 51, 182–193. [CrossRef]
3. Huang, X.; Zhang, L. Morphological Building/Shadow Index for Building Extraction From High-Resolution Imagery Over Urban Areas. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2012, 5, 161–172. [CrossRef]
4. Vu, T.T.; Yamazaki, F.; Matsuoka, M. Multi-scale solution for building extraction from LiDAR and image data. Int. J. Appl. Earth Obs. Geoinf. 2009, 11, 281–289. [CrossRef]
5. Bi, Q.; Qin, K.; Zhang, H.; Zhang, Y.; Li, Z.; Xu, K. A Multi-Scale Filtering Building Index for Building Extraction in Very High-Resolution Satellite Imagery. Remote Sens. 2019, 11, 482. [CrossRef]
6. Jabari, S.; Zhang, Y.; Suliman, A. Stereo-based building detection in very high resolution satellite imagery using IHS color system. In Proceedings of the 2014 IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014.
7. Sirmacek, B.; Unsalan, C. Urban-Area and Building Detection Using SIFT Keypoints and Graph Theory. IEEE Trans. Geosci. Remote Sens. 2009, 47, 1156–1167. [CrossRef]
8. Tournaire, O.; Brediif, M.; Boldo, D.; Durupt, M. An efficient stochastic approach for building footprint extraction from digital elevation models. ISPRS J. Photogramm. Remote Sens. 2010, 65, 317–327. [CrossRef]
9. Argyridis, A.; Argialas, D.P. Building change detection through multi-scale GEOBIA approach by integrating deep belief networks with fuzzy ontologies. Int. J. Image Data Fusion 2016, 7, 148–181. [CrossRef]
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. Commun. ACM 2017, 60, 84–90. [CrossRef]
11. Gavrilescu, R.; Fos, C.; Skoczylas, M.; Cotovanu, D. Faster R-CNN: An Approach to Real-Time Object Detection. In Proceedings of the 2018 International Conference and Exposition on Electrical And Power Engineering (EPE), Iasi, Romania, 18–19 October 2018.
12. Shellhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 60–65. [CrossRef]
13. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. 2020, 42, 386–397. [CrossRef] [PubMed]
14. 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. 2019, 57, 574–586. [CrossRef]
15. Shi, Y.; Li, Q.; Zhu, X.X. Building segmentation through a gated graph convolutional neural network with deep structured feature embedding. ISPRS J. Photogramm. Remote Sens. 2020, 159, 184–197. [CrossRef] [PubMed]
16. Yuan, W.; Xu, W. MSST-Net: A Multi-Scale Adaptive Network for Building Extraction from Remote Sensing Images Based on Swin Transformer. Remote Sens. 2021, 13, 4743. [CrossRef]
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
18. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 1137–1149. [CrossRef]
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision–European Conference on Computer Vision 2016, Amsterdam, The Netherlands, 8–16 October 2016.
21. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934.
22. Ding, J.; Zhang, J.; Zhan, Z.; Tang, X.; Wang, X. A Precision Efficient Method for Collapsed Building Detection in Post-Earthquake UAV Images Based on the Improved NMS Algorithm and Faster R-CNN. Remote Sens. 2022, 14, 663.
23. Bai, T.; Pang, Y.; Wang, J.; Han, K.; Luo, J.; Wang, H.; Lin, J.; Wu, J.; Zhang, H. An Optimized Faster R-CNN Method Based on DRNet and RoI Align for Building Detection in Remote Sensing Images. Remote Sens. 2020, 12, 762. [CrossRef]
24. Xie, Y.; Zhu, J.; Cao, Y.; Feng, D.; Hu, M.; Li, W.; Zhang, Y.; Fu, L. Refined Extraction Of Building Outlines From High-Resolution Remote Sensing Imagery Based on a Multi-feature Convolutional Neural Network and Morphological Filtering. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2020, 13, 1842–1855.

25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.

26. Zhu, Q.; Liao, C.; Hu, H.; Mei, X.; Li, H. MAP-Net: Multiple Attending Path Neural Network for Building Footprint Extraction From Remote Sensed Imagery. *IEEE Trans. Geosci. Remote Sens.* 2021, 59, 6169–6181. [CrossRef]

27. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.

28. Ma, J.; Wu, L.; Tang, X.; Liu, F.; Zhang, X.; Jiao, L. Building Extraction of Aerial Images by a Global and Multi-Scale Encoder-Decoder Network. *Remote Sens.* 2020, 12, 2330. [CrossRef]

29. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, Munich, Germany, 5–9 October 2015.

30. Lu, T.; Ming, D.; Lin, X.; Hong, Z.; Bai, X.; Fang, J. Detecting Building Edges from High Spatial Resolution Remote Sensing Imagery Using Richer Convolution Features Network. *Remote Sens.* 2018, 10, 1496. [CrossRef]

31. Wu, G.; Guo, Z.; Shi, X.; Chen, Q.; Xu, Y.; Shibasaki, R.; Shao, X. A Boundary Regulated Network for Accurate Roof Segmentation and Outline Extraction. *Remote Sens.* 2018, 10, 1195. [CrossRef]

32. Jiwani, A.; Ganguly, S.; Ding, C.; Zhou, N.; Chan, D.M. A Semantic Segmentation Network for Urban-Scale Building Footprint Extraction Using RGB Satellite Imagery. *arXiv 2021*, arXiv:2104.01263.

33. Chen, L.-C.; Zhiu, Y.; Papandreou, G.; Schröff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Computer Vision–European Conference on Computer Vision 2018, Munich, Germany, 8–14 September 2018.

34. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

35. Li, Q.; Shi, Y.; Huang, X.; Zhu, X.X. Building Footprint Generation by Integrating Convolution Neural Network With Feature Pairwise Conditional Random Field (FPCRF). *IEEE Trans. Geosci. Remote Sens.* 2020, 58, 7502–7519. [CrossRef]

36. Lu, X.; Zhong, Y.; Zheng, Z.; Liu, Y.; Zhao, J.; Ma, A.; Yang, J. Multi-Scale and Multi-Task Deep Learning Framework for Automatic Road Extraction. *IEEE Trans. Geosci. Remote Sens.* 2019, 57, 9362–9377. [CrossRef]

37. Teichmann, M.; Weber, M.; Zillstner, M.; Cipolla, R.; Urtasun, R. MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018.

38. Wu, D.; Liao, M.; Zhang, W.; Wang, X.; Bai, X.; Cheng, W.; Liu, W. YOLOP: You Only Look Once for Panoptic Driving Perception. *arXiv 2021*, arXiv:2108.11250.

39. 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 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019.

40. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 2015, 37, 1904–1916. [CrossRef]

41. Bolta, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOALCT: Real-Time Instance Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.

42. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–21 June 2018.

43. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv 2017*, arXiv:1704.06857.

44. Chen, J.; Zhang, D.; Wu, Y.; Chen, Y.; Yan, X. A Context Feature Enhancement Network for Building Extraction from High-Resolution Remote Sensing Imagery. *Remote Sens.* 2022, 14, 2276. [CrossRef]

45. Qiu, Y.; Wu, F.; Yin, J.; Liu, C.; Gong, X.; Wang, A. MSL-Net: An Efficient Network for Building Extraction from Aerial Imagery. *Remote Sens.* 2022, 14, 3914. [CrossRef]

46. Mnih, V. Machine Learning for Aerial Image Labeling. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2013.

47. YICO-HUE. GitHub Repository. Available online: https://github.com/Yico-Yue/RSIBE (accessed on 11 July 2022).

48. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

49. Zhao, K.; Kang, J.; Jung, J.; Sohn, G. Building Extraction from Satellite Images Using Mask R-CNN with Building Boundary Regularization. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018.

50. Zhai, R.; Li, A.; Yin, J.; Du, J.; Qiu, Y. A Progressive Simplification Method for Buildings Based on Structural Subdivision. *ISPRS Int. J. Geo-Inf.* 2022, 11, 393. [CrossRef]