remote sensing

MDPI

Article

Multi-Modal Feature Fusion Network with Adaptive Center Point Detector for Building Instance Extraction

Qinglie Yuan [1,2,*] and Helmi Zulhaidi Mohd Shafri [1]

[1] Department of Civil Engineering and Geospatial Information Science Research Centre (GISRC), Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia
[2] School of Civil and Architecture Engineering, Panzhihua University, Panzhihua 617000, China
* Correspondence: yuanqinglie@pzhu.edu.cn

Abstract: Building information extraction utilizing remote sensing technology has vital applications in many domains, such as urban planning, cadastral mapping, geographic information censuses, and land-cover change analysis. In recent years, deep learning algorithms with strong feature construction ability have been widely used in automatic building extraction. However, most methods using semantic segmentation networks cannot obtain object-level building information. Some instance segmentation networks rely on predefined detectors and have weak detection ability for buildings with complex shapes and multiple scales. In addition, the advantages of multi-modal remote sensing data have not been effectively exploited to improve model performance with limited training samples. To address the above problems, we proposed a CNN framework with an adaptive center point detector for the object-level extraction of buildings. The proposed framework combines object detection and semantic segmentation with multi-modal data, including high-resolution aerial images and LiDAR data, as inputs. Meanwhile, we developed novel modules to optimize and fuse multi-modal features. Specifically, the local spatial–spectral perceptron can mutually compensate for semantic information and spatial features. The cross-level global context module can enhance long-range feature dependence. The adaptive center point detector explicitly models deformable convolution to improve detection accuracy, especially for buildings with complex shapes. Furthermore, we constructed a building instance segmentation dataset using multi-modal data for model training and evaluation. Quantitative analysis and visualized results verified that the proposed network can improve the accuracy and efficiency of building instance segmentation.

Keywords: building extraction; instance segmentation; multi-modal feature fusion; remote sensing images; LiDAR; object detection

Citation: Yuan, Q.; Mohd Shafri, H.Z. Multi-Modal Feature Fusion Network with Adaptive Center Point Detector for Building Instance Extraction. *Remote Sens.* **2022**, *14*, 4920. https://doi.org/10.3390/rs14194920

Academic Editor: Mohammad Awrangjeb

Received: 16 August 2022
Accepted: 28 September 2022
Published: 1 October 2022

1. Introduction

Building information has wide applications in many domains, such as urban planning, cadastral mapping, geographic information censuses, and land-cover change analysis [1–3]. Building feature extraction algorithms have a profound effect on promoting intelligent city construction [4–6]. With the emergence of advanced sensors and platforms, multi-modal remote sensing data, such as LiDAR and aerial images, can be obtained. These remote sensing data provide accurate spatial information and abundant spectral features. For instance, high-spatial-resolution remote sensing images contain a fine-grained, 2D, geometric structure and texture, while high-precision, 3D spatial information can be acquired via LiDAR technology. Multi-modal feature fusion is conducive to improving the accuracy and efficiency of building detection.

Feature construction using remote sensing data is vital for improving method performance. However, building feature extraction is challenging due to many factors, such as the complexity of scenes, shadow, multiple scales, occlusion, illumination, and diverse shapes. Some traditional methods apply spectral features and establish the morphological

index to distinguish buildings from the background [7]. However, these indicators can significantly change according to season and environment. Additionally, the diversity of geographical objects brings discriminative difficulties when applying shallow features due to intra-class spectral variation and inter-class similarity. In addition, some methods establish geometric structure information, such as corners, height variation, and normal vectors, to extract buildings using 3D point clouds [8–10]. Recently, multi-modal feature-fusion-based methods have been developed and effectively improved building extraction by combining 2D and 3D information [11–13]. Nevertheless, traditional approaches based on prior knowledge are suitable for specific data and are vulnerable to parameter setting. Moreover, these methods cannot obtain object-level information, such as position, size, and count for each building, by semantic segmentation only. Therefore, automatic building instance segmentation has high algorithm complexity and is still a challenge in remote sensing data processing.

Deep learning algorithms have shown great potential in target detection, semantic segmentation, and classification using image data in recent years. Instead of relying on prior knowledge, deep neural networks can learn multi-level features. In particular, these methods provide an automatic and robust solution for the automatic extraction of buildings. Convolutional neural networks (CNNs), the popular neural network frameworks in deep learning, have been widely used in remote sensing image processing. Compared with traditional approaches, CNNs can extract hierarchical features from shallow level to deep level with semantic representation by stacking convolutional blocks. Instead of over-reliance on manual designs, a deep convolution network can automatically complete multiple tasks and flexibly construct modules to achieve different requirements.

Some algorithms based on CNNs have achieved excellent building detection and extraction performance. For instance, Wen et al. [12] constructed a detection framework for rotated building objects using the improved Mask RNN [13]. Meanwhile, atrous convolution and inception blocks were introduced into the backbone network to optimize feature extraction. This network obtained object-level segmentation results from complex backgrounds. Similarly, Ji et al. redesigned a U-Net structure and created a WHU building dataset [14] for building extraction and instance change detection using Mask R-CNN. Moreover, some methods use a one-stage, anchor-free instance segmentation framework to improve speed and segmentation accuracy. For example, Zhang et al. proposed a one-stage change detection network combined with a spatial-channel attention mechanism for newly built buildings [15]. Multi-scale, built-up change regions were effectively detected in the public LEVIR and AICD datasets. Wu et al. improved the detectors based on CenterMask [16], obeying a one-stage detection paradigm, and established an attention-guided mask branch to segment building instances [17]. Experimental results showed that the method achieved better speed and accuracy than state-of-the-art methods.

Although deep learning methods provide various feature optimization strategies and can achieve excellent performance in processing remote sensing data, some issues still should be addressed: (1) Most remote sensing datasets are mainly used for the semantic segmentation of buildings or natural scenes. An instance segmentation dataset of buildings needs to be constructed for the training and evaluation of the model. (2) Many methods apply remote sensing data with a single modality, such as optical remote sensing images. However, some spectral information is similar to the buildings, especially information pertaining to roads, cars, and artificial ground. Misclassification often exists in semantic segmentation networks. (3) The detectors based on the region proposal framework require a number of predefined anchors and a filtering mechanism, which significantly reduces the training and inference efficiency of the model. (4) Buildings display significant shape differences and scale variability. Some anchor-free detectors cannot accurately regress the location due to fixed grid computation in convolution. The main features of some buildings with small sizes are omitted after the down-sampling operation. Large-scale buildings often occupy most of the area in the sample patches, while the global context

is insufficient due to the limited receptive field [18]. These factors could cause inaccurate detection and segmentation.

To address the above problems, we designed a new convolutional neural network (CNN) framework for object-level building extraction. Some novel modules were developed to integrate multi-modal remote sensing data advantages, including 2D, high-spatial-resolution images and 3D LiDAR point cloud data. The main contributions of this study are summarized as follows:

- We constructed an end-to-end instance segmentation CNN, combining anchor-free detection and semantic segmentation methods. Meanwhile, a local spatial–spectral perceptron was developed to optimize and fuse multi-modal features. This module can interactively compensate for spectral and spatial information in the convolutional operators and effectively recalibrates the semantic features. Furthermore, a cross-level global feature fusion module was constructed to enhance long-range context dependence;
- An adaptive center point detector, based on the CenterNet, was proposed for multi-scale buildings and the complex shapes of buildings, introducing the explicit deformable convolution under supervised learning to enhance the size regression ability and the central point semantic intensity;
- We created a building instance segmentation dataset using high-resolution aerial imagery and LiDAR data. This dataset provides highly precise instance labels for model training and evaluation.

2. Related Work

2.1. Instance Segmentation

Based on the different detectors, instance segmentation algorithms can be divided into two categories: two-stage instance segmentation and one-stage instance segmentation. The former generates predefined anchors by a region proposal network (RPN) and then conducts binary classifications within each ROI. Mask R-CNN is a representative two-stage instance segmentation method based on Faster R-CNN and has achieved state-of-the-art performance using the ROI Align strategy [13]. Similarly, PANet [19] constructs a bottom-up feature aggregation path and applies the adaptive pooling strategy to improve the mask prediction accuracy for Mask R-CNN. However, some drawbacks hinder the application of these models in practical engineering tasks. Due to the number of predefined anchors, the inference time is significantly increased. In addition, the segmentation result generates local masks and presents a rough delineation due to the fixed-scale ROI features and limited spatial resolution.

On the other hand, like the anchor-free methods being proposed, such as FCOS [20] and CenterNet [21], single-stage instance segmentation methods outperform two-stage instance segmentation algorithms in terms of accuracy and efficiency for special tasks. For instance, the SOLO [22] algorithm contains the concept of "instance categories" by global mask prediction. This paradigm establishes two-branch networks and transforms the problem of instance segmentation into classification tasks, including location and scale prediction. Wang et al. proposed CenterMask [16], complying with the CenterNet framework. This network combines local shape representation and a global saliency map to segment instance objects, effectively separating overlapping objects and improving mask prediction accuracy. YOLACT [23] generates a set of prototypes using FCN and predicts corresponding mask coefficients for each instance. High-resolution masks can be generated by the linear combination of the prototype with template coefficients. InstanceFCN [24] uses the FCN to generate multiple instance-sensitive score maps and then applies the assembly module to output the target. Unlike the above methods, the multi-task network cascade strategy [25,26] is applied to instance segmentation in which semantic segmentation and target detection are conducted synchronously. The results confirm that features can be optimized due to the commonness of multiple subtasks [27,28]. Considering the advantages of the above methods, in this paper, we combine the task of object detection and

semantic segmentation to achieve building instance segmentation. We adopt an anchor-free framework in the detection task and improve the CenterNet detector. Multi-modal features are effectively fused in the semantic segmentation task to increase prediction accuracy.

2.2. Multi-Modal Feature Fusion

In the natural environment, different sensors can capture various modal data, such as sound, text, raster data, and vector information. LiDAR and image are two typical modalities that mutually compensate for 2D and 3D information for semantic segmentation or object detection in deep learning tasks. On the one hand, some methods manage to integrate optical image features into 3D representation. For example, Yoo et al. [29] established auto-calibrated projection and a gated feature fusion network to transform and fuse features for 3D target detection. Qi et al. [30] obtained 2D semantic segmentation results using images and projected them to a 3D frustum space combined with PointNet for localization.

On the other hand, spatial information from LiDAR is converted into digital geographic models and fused with optical image features. For example, Hosseinpour et al. developed a two-stream residual network with feature-gated units to extract buildings [31]. Based on nDSM and optical image data, Cao et al. constructed a cross-modal feature calibration module by effectively modeling the attention mechanism to aggregate context [32]. Moreover, LiDAR point cloud data can be transformed into depth images. From this perspective, depth-aware modules were introduced to feature construction. For instance, Wang et al. proposed a depth-aware CNN (D-CNN) [33], which explicitly models depth information and integrates it into convolutional operators. Similarly, Chen et al. proposed DPANet [34] for salient object detection with depth potential perception and combined the complementary advantages of RGB-D images. However, the above methods ignore semantic and spatial correlations when fusing multi-modal features. The gap in different feature domains brings obstacles to parameter optimization and even introduces noise information. To solve the above problems, we construct a multi-modal feature fusion module and introduce a D-CNN structure and involution to enhance the complementarity of spectral and spatial features.

3. Method

3.1. Building Instance Segmentation Architecture

As illustrated in Figure 1, the network architecture is composed of an encoder-decoder. LiDAR data and images as multi-modal data are transmitted into encoders to extract features, while the decoder predicts the instance mask by semantic segmentation and detection approaches for each building. Meanwhile, two novel feature optimization modules were developed to dynamically fuse multi-modal features and enhance local and global contexts.

Concretely, in the encoders, one branch, as the backbone network, adopts the improved ResNet50 [35] for feature extraction with images as inputs, while another applies ResNet18 for LiDAR data products. In ResNet50, convolutional layers can be divided into different stages, including stage 1~stage 5, as presented in Figure 1. Two 3×3 convolutions replace a 7×7 convolution in stage 1 to reduce parameters. In stage 5, the residual blocks are modified to enlarge the receptive field with high spatial resolution. Specifically, the dilated 3×3 convolution with a dilation ratio of 2 is applied to residual blocks and keeps the same spatial resolution with the feature maps of stage 4. The drop layer follows the last residual blocks with a ratio of 50% to prevent overfitting.

For the decoders, the feature from the encoders is transmitted to different subnetworks to complete semantic segmentation and object detection. The feature pyramid [36] structure is constructed to recover fine-grained information in the segmentation task. In the object detection tasks, we developed an adaptive center point detector based on the CenterNet's detector [21] to predict the spatial position of each building. Finally, the above outputs are combined to generate the masks for each building. The whole learning process obeys an end-to-end paradigm without any post-processing.

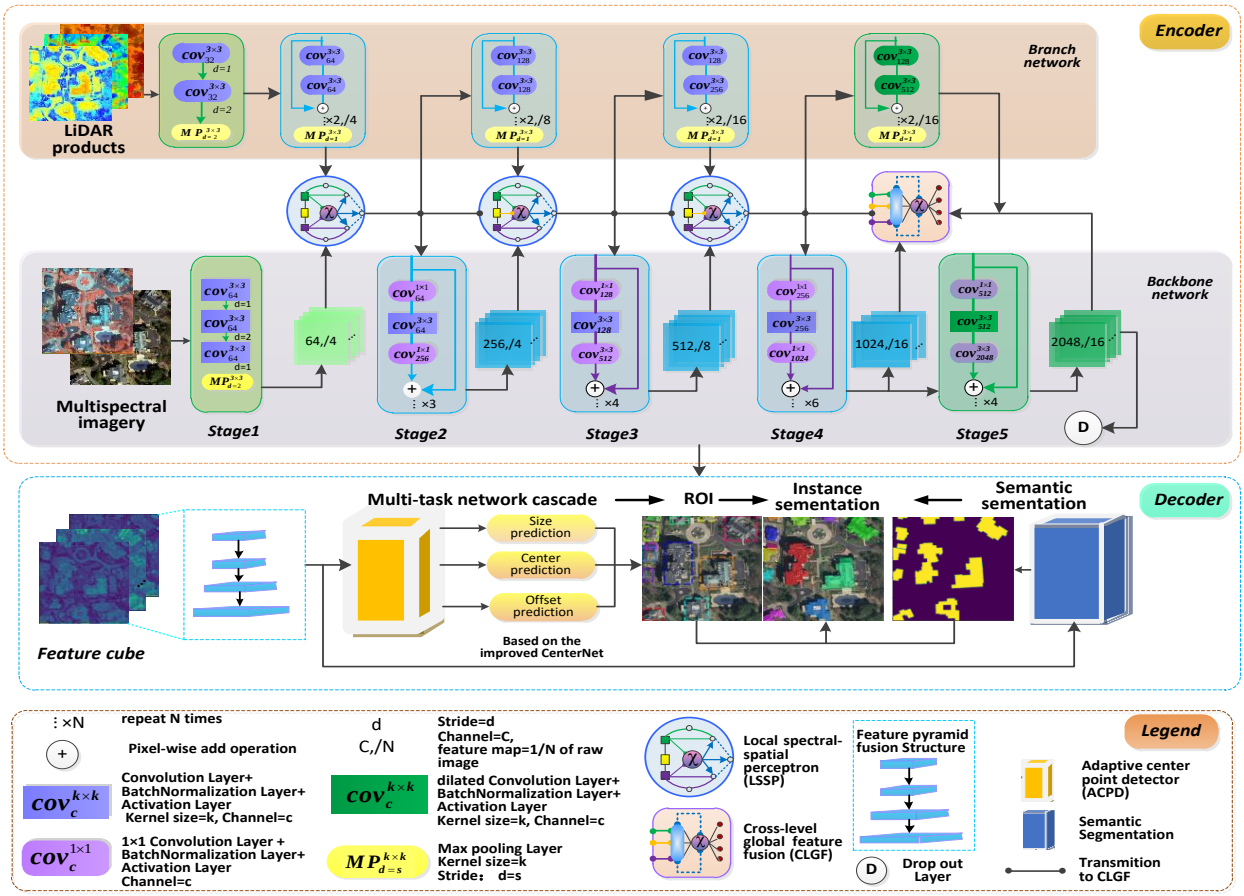


Figure 1. The network architecture for building instance segmentation.

3.2. Local Spectral–Spatial Perceptron

To effectively fuse multi-modal features, we established a local spectral–spatial perceptron (LSSP) in the encoders and dynamically integrated the dual-modal information into the convolutional operation. Figure 2 exhibits the details of the module structures. The LSSP can be flexibly embedded into residual network blocks with learning parameters. This module unit consists of two network structures with complementary functions: the local spatial perceptron and the local spectral perceptron.

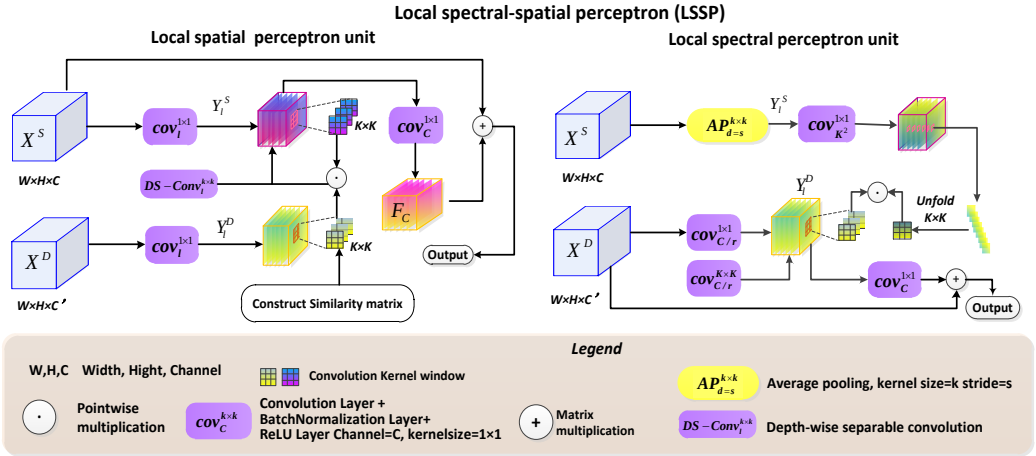


Figure 2. Local spectral–spatial perceptron. This module consists of two units: the local spatial perceptron and spectral perceptron.

3.2.1. Local Spatial Perceptron

Considering the image depth and spectral feature correlations in CNN, the depth-aware CNN (D-CNN) [33] operator explicitly establishes the depth similarity coefficient for any central point and its neighborhoods. Convolution kernels matching these weights extract local features, seamlessly incorporating depth information into convolution operation without increasing additional parameters. Nevertheless, D-CNN cannot adaptively encode the spatial context in different spectral channels. Probably, irrelevant background features are enhanced synchronously. To address the above issues, the spatial perceptron unit extends the D-CNN structure with multiple spatial representations and recalibrates local optical image features using LiDAR products instead of depth images.

Specifically, the input feature cube is defined as $X \in \mathbb{R}^{H \times W \times C_{in}}$, where H, W, and C_{in} represent the height, width, and channel numbers of the feature maps, respectively. Conventionally, if there are output feature maps with channels of C_{out}, the 2D convolution operation can be represented as follows:

$$Y_{i,j,m} = \left(\sum_{n=1}^{C_{in}} \sum_{(u,v)\in\Theta} W_{m,n,u,v} X(i+u, j+v, n)\right) + b_{i,j,m}, \quad (1)$$

$$\Theta = \{(-1,-1), (0,-1), (1,-1) \ldots, (0,1), (1,1)\}, \quad (2)$$

where $Y_{i,j,m}$ denotes the convolutional feature on the 2D position of the (i, j) index and the m-th channel ($m \in C_{out}$); and Θ denotes convolution kernel neighborhood with size $k \times k$. For example, if k is set to 3, Θ can be defined by Equation (2); (u, v) presents the spatial index position on Θ; $X(\cdot)$ is the input feature from the Θ; and $W_{m,n,u,v}$ and $b_{i,j,m}$ correspond to the learning weights and basic parameters, respectively.

As illustrated in Figure 2, X^D and X^S represent the input feature maps from LiDAR and the optical image, respectively. The function $\varphi(\cdot)$ maps X^D and X^S to ${Y_l}^D$ and ${Y_l}^S$, respectively, where l is low rank for the channel dimension representing the 3D spatial feature response in some aspects. This process can be modeled as Equation (3), and Conv2D1 $\times$ 1 is used with the activation function *Relu*. To fuse spatial features, we established the local spatial similarity weights using W^D and combined the depth-wise separable convolution (DS-Conv) [37] to extract features using ${Y_l}^S$, as defined in Equation (4), where α is a regulatory factor (referring to [37], α is set to 8.3).

$${Y_l}^D = \varphi(X^D), {Y_l}^S = \varphi(X^S), \quad (3)$$

$${W^D}_{u,v,m} = exp(-\alpha \left| Y_l^D(i+u, j+v, m) - Y_l^D(i,j,m) \right|), (u, v \in \Theta), \quad (4)$$

Equation (4) implies that, if a central point has similar spatial features to its neighborhoods, these points are assigned greater weights. In Equation (5), $F_{i,j,m}$ enumerates each separable feature on the (i, j, m) spatial index position (m = 1, 2,$\cdots$, l), and ${W^S}_{u,v,m}$ is the learning weight parameters from DS-Conv. Finally, Conv2D1 $\times$ 1 is a residual bottleneck structure map $F_{i,j,m}$ of initial C_{in} dimensions, as presented in Equation (6), where $Z_{i,j,n}$ is the fused feature on (i, j, n) position, and ${W^S}_{m,n}$ is the mapping parameter.

$$F_{i,j,m} = \sum_{(u,v)\in\Theta} {W^D}_{u,v,m} {W^S}_{u,v,m} {Y_l}^S(i+u, j+v, m), \quad (5)$$

$$Z_{i,j,n} = X^S(i,j,n) + \sum_{n}^{C_{in}} {W^S}_{m,n} F_{i,j,m}, \quad (6)$$

3.2.2. Local Spectral Perceptron

Local spatial perceptron can explicitly fuse 3D spatial features into spectral features by constructing local spatial similarity. However, some errors can arise from unreliable 3D information if point cloud data exist or noise or buildings present significant elevation

variation, such as non-planar roofs and large-scale buildings with different heights. Additionally, LiDAR-based products (e.g., DSM) present inherent errors derived from particular environments when some interpolation-based or triangulation-mesh-generation-based algorithms are used. If the convolution kernel is directly assigned to unreliable weights, these errors can cause intra-class inconsistency for the feature extraction.

To address the above problems, we managed to integrate spectral confidence into LiDAR features and guided convolution to recalibrate kernels for X^D. Involution [38] is a network structure with channel-agnostic and spatial-specific properties, which "squeeze" local spatial features and learn channel attention in an arbitrary position. Inspired by this operator, we constructed the local spectral perceptron to obtain a spectral response. Firstly, as illustrated in Figure 2, average pooling operation $avpool(\cdot)$ is applied to the aggregate, 2D, spatial context in each channel domain of X^S, as defined in Equation (7). Then, calibration parameters $W^S{}_{i+u,\,j+v}$ are learned at a spatial location of (i, j, n), as defined in Equation (8), where $u,v \in \Theta$ (the size of Θ is k^2) and $\sigma(\cdot)$ are the activation functions using *Relu*; and $w^S{}_{i+u,j+v,n}$ denotes weights corresponding to the n-th channel. $W^S{}_{i+u,j+v}$ represents the spectral response in channel domains and shares all positions. Finally, convolution kernels $W_{m,n,u,v}$ match $W^S{}_{i+u,\,j+v}$ to recalibrate X^D, as defined in Equation (9), where $Y^D{}_{i,j,m}$ is the fused feature.

$$Y^S_{u,v} = avpool(X^S), (u, v \in \Theta), \tag{7}$$

$$W^S_{i+u,j+v} = \sigma(\sum_{n=1}^{C_{in}} w^S{}_{i+u,j+v,n} Y^S_{u,v}(i, j, n)), \tag{8}$$

$$Y^D{}_{i,j,m} = (\sum_{n=1}^{C_{in}} \sum_{(u,v)\in\Theta} W^S_{i+u,j+v} W_{m,n,u,v} X^D(i+u, j+v, n)) \tag{9}$$

3.3. Cross-Level Global Feature Fusion

The LSSP formulates feature fusion fashion based on the local region and shallow encoders. However, as the depth of the network increases, spatial information significantly decreases due to the low spatial resolution of the feature maps. Feature interpretability becomes difficult, while semantic information is enhanced. Hence, the LSSP is not suitable for high-level feature transformation and fusion. Additionally, local operators based on convolution cannot obtain long-range interdependence. The semantic gap in different levels causes difficulties in decoding context features. To address the above problems, we constructed a cross-level global feature fusion (CLGF) module to decode contextual information from different level features, as shown in Figure 3.

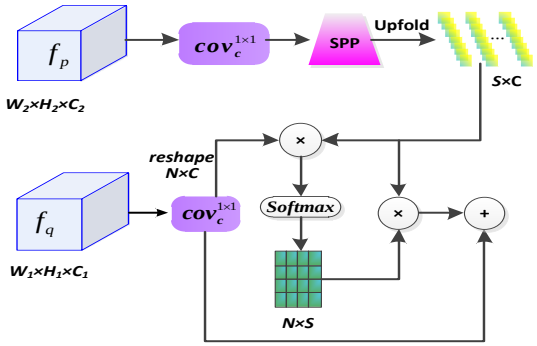


Figure 3. Cross-level global feature fusion module.

Specifically, if features $X_i{}^D$ and $X_i{}^S$ are derived from stages i, fused feature maps f_i can be obtained. Concretely, function $\phi_i(\cdot)$ maps $X_i{}^D$ and $X_i{}^S$ into embedding space, as defined in Equation (10), and the channel is reduced by C_{in}/r, where $\phi_i(\cdot)$ is Conv1 $\times$ 1 operator following activation function *Relu*, and r is reduction ratio. A non-local [39] operator is applied to the CLGF module and generates the global context. High-level features

have abundant semantic information with a large receptive field, which is suitable for identifying categories, while shallow features have fine-grained features that are conducive to recovering details of buildings. Hence, unlike conventional, non-local networks, CLGF constructs cross-level semantic similarity maps from different layers. In the network structure, as shown in Figure 1, f_5 as a high-level feature in stage 5 and f_3/f_4 (as low-level and middle-level features from stage 3 and stage 4) are used to achieve a similarity matrix. In Equation (11), $F(f_p, f_q)$ presents similarity between f_5 and f_3/f_4, where (u, v, m) and (u', v', m') denote arbitrary positions on the feature maps. Softmax function is used for the normalization of similarity maps.

$$f_i = \phi(X_i^D + X_i^S) = Relu(w_i(X_i^D + X_i^S)), \tag{10}$$

$$F(f_p, f_q) = \frac{exp(SPP(f_p(u\prime, v\prime, m)) \otimes f_q(u, v, m))}{\sum exp(SPP(f_p(u\prime, v\prime, m)) \otimes f_q(u, v, m))}, \{p, q \in i | i = 3, 4, 5\}, \tag{11}$$

Additionally, since high-level feature maps have lower spatial resolution with a larger receptive field than shallow layers, pixel-wise interdependence is converted into pixel–regional correlations. To reduce the computational burden, a spatial pooling pyramid (SPP) [40] is applied to resample feature maps using average pooling for f_3~f_5 and keeping the same spatial resolution. In the network, we adopted the average pooling operator using different pooling rates in SPP to aggregate multi-scale regional features. The above process can be modeled as Equation (12), where ${F_q}^U$ is a new feature fusing global and local contexts.

$$F_q^U = f_q(u, v, m) + \sum_{m=1}^{C_{in}/r} SPP(f_p(u, v, m))F(f_p, f_q), q = 3, 4 \tag{12}$$

3.4. Adaptive Center Point Detector

CenterNet [21] provides a simple and efficient one-stage network framework based on an anchor-free strategy. This network represents the object as a point in the bounding box center by the CNN. Hence, the object center prediction determines the detection accuracy of the whole network. However, buildings exhibit multi-scale variation with different shapes and complex geometric structures. Convolution operation obeys the fixed grid computation structure with a limited receptive field, which causes contextual information loss for the center position regression. As illustrated in Figure 4, some buildings present irregular shapes, such as the "T" shape, "E" shape, ring shape, and narrow, long shape, where the building center does not exist in itself. As a result, the center point detector lacks corresponding context and semantic features, resulting in deviation of position regression. Moreover, the same building instance can be predicted with multiple center points due to its large size or spectral heterogeneity. To deal with this issue, we proposed an adaptive center point detector (ACPD) based on the CenterNet formulation, as shown in Figure 4.

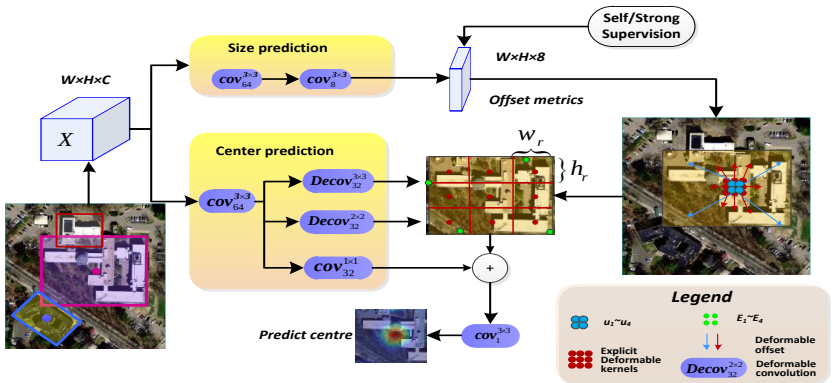


Figure 4. Adaptive center point detector.

The ACPD constructs two branches to extract contextual information: central position and scale. Deformable convolution [41] adopts an adaptive computation paradigm to capture features by predicting offset parameters for the convolution kernels and extracting features instead of fixed geometric structures. In Equation (13), $Y^s_{i,j,m}$ is the corresponding deformable feature; $(\Delta u, \Delta v)$ denotes the coordinate offset of convolution kernels; Ω is the range of deformation; and Θ is the kernel neighborhood in conventional convolution with a size of $k \times k$, which is similar to Equation (1). Nevertheless, deformable convolution implicitly models inference without supervision for offset prediction. An unrelated feature or other instance information is probably introduced into regression analysis due to uncertain offset variation. Additionally, weight parameters and complexity significantly increase since each feature position requires two offsets. Therefore, we redesigned the deformable range Ω and managed to explicitly enhance contextual representation for the center point regression under supervised learning.

$$Y^s_{i,j,m} = (\sum_{n=1}^{C_{in}} \sum_{\substack{(u,v) \in \Theta, \\ (\Delta u, \Delta v) \in \Omega}} W_{m,n,u,v} X(i+u+\Delta u, j+v+\Delta v, n)), \tag{13}$$

Concretely, the ACPD predicts coordinate offsets to limit deformable range using four groups of offset parameters. In Equation (14), E_1~$E_4 \in \Omega$ are point coordinates defining each object range depending on the minimum bounding box size. The abscissa offsets and ordinate offsets from E_1~E_4 are limited to a certain extent, which can improve detection sensitivity at the object boundary. Then, deformable convolution kernels with k = 2 are assigned to positions E_1~E_4, in which features can be obtained by bilinear interpolation.

$$\begin{aligned} E_1 &= (u_1 + \Delta u_1, v_1 + \Delta v_{\min}), E_2 = (u_2 + \Delta u_2, v_2 + \Delta v_{\max}), \\ E_3 &= (u_3 + \Delta u_{min}, v_3 + \Delta v_3), E_4 = (u_4 + \Delta u_{max}, v_4 + \Delta v_4), \end{aligned} \tag{14}$$

where $(\Delta u_1, \Delta u_2) \in \Omega^{\Delta u_{\min} \times \Delta u_{\max}}$ and $(\Delta v_3, \Delta v_4) \in \Omega^{\Delta v_{\min} \times \Delta v_{\max}}$; u_1~u_4 and v_1~$v_4 \in \Theta$; $\Delta u_{\min}$ and Δu_{max} are the minimum and maximum on the abscissa offset, respectively; and $\Delta v_{\min}$ and $\Delta v_{\max}$ are the minimum and maximum on the ordinate offset, respectively.

Considering the scale variation of the objects, another branch of the ACPD captures holistic instance features. Specifically, predicted instance regions are regularly divided into rectangular grids based on offset range and aligned with the corresponding feature map. Each feature in the grid cell is aggregated using the average pooling operator. Equation (15) defines the grid cell size as $h_r \times w_r$. Rectangular grid size can be defined by Equation (16), where H_r and W_r are the height and width, respectively. In the network, we sequentially assigned convolution kernels with k = 3 to match these grids and extract features, as defined in Equation (17), where $(\Delta u, \Delta v) \in \Omega^{H_r \times W_r}$. Finally, fused feature maps $Y^H_{i,j,m}$ from the dual branches of the ACPD can be obtained, as defined in Equation (18). The ACPD is embedded in the residual block and generates the heatmap with 3 × 3 convolution.

$$h_r = \lceil H_r / k \rceil, w_r = \lceil W_r / k \rceil, \tag{15}$$

$$H_r = \max(k, v_2 - v_1 + \Delta v_{\max} - \Delta v_{\min}), W_r = \max(k, u_4 - u_3 + \Delta u_{max} - \Delta u_{min}), \tag{16}$$

$$Y^P_{i,j,m} = \sum_{n=1}^{C_{in}} \sum_{\substack{(u,v) \in \Theta, \\ (\Delta u, \Delta v) \in \Omega}} W_{m,n,u,v} avepool(X(i+u+\Delta u, j+v+\Delta v, n)), \tag{17}$$

$$Y^H_{i,j,m} = Y^s_{i,j,m} + Y^p_{i,j,m}, \tag{18}$$

4. Model Loss Function

The multi-task learning model adopts different losses, including semantic segmentation loss and object regression loss, to optimize network training. Softmax function is used to normalize predicted results in the segmentation task. We used the binary cross-entropy loss L_{seg} for pixel-wise segmentation. Following the CenterNet loss function [21], L_k and L_{off} denote focal loss and offset loss for the center point regression, respectively.

To predict deformable offset under supervised learning, we established self-supervision and strong-supervision loss functions for the scale and position regression. In self-supervision, E_1~E_4 are the extreme points located on the boundary of the bounding box. Hence, these coordinates are satisfied with geometric relationships. As presented in Figure 4, predicted central coordinates denote the midpoints for E_1~E_4 in abscissa and ordinate. The network applies the smooth L_1 function to constrain the above relationship, as defined in Equation (19). Similarly, in strong supervision, the ACPD applies offset to regress the scale of the bounding box for each object, as defined in Equation (20), where S_u and S_v denote the width and height of the bounding box in ground truth, respectively. Finally, the total loss can be expressed by Equation (21), where constant coefficient λ is used to adjust the loss proportion in the multiple-task training (referring to CenterNet [21] $\lambda_{seg} = 2$, $\lambda_k = \lambda_{off} = 1$, $\lambda_{ps} = 0.1$ in the experiments).

$$L_{pos} = \frac{1}{N}\sum_i \left| \frac{u_3 + \Delta u_{min} + u_4 + \Delta u_{max}}{2} - x^c \right| + \left| \frac{v_1 + \Delta v_{\min} + v_2 + \Delta v_{\max}}{2} - y^c \right|, \quad (19)$$

$$L_{scale} = \frac{1}{N}\sum_i |u_4 + \Delta u_{max} - u_3 - \Delta u_{min} - S_u| + |v_2 + \Delta v_{\max} - v_1 - \Delta v_{\min} - S_v|, \quad (20)$$

$$L_{total} = \lambda_{seg} L_{seg} + \lambda_k L_k + \lambda_{off} L_{off} + \lambda_{ps}(L_{pos} + L_{scale}), \quad (21)$$

5. Experiments

5.1. Datasets Description

Many public, open building extraction datasets have been created to train advanced deep neural network models and verify their performance or accuracy [14]. However, these building datasets mainly serve the semantic segmentation of buildings with a single data source. Therefore, to train the proposed model and verify its effectiveness, we created an open building instance segmentation dataset using multi-modal remote sensing data (BISM), including high-spatial-resolution multispectral images and LiDAR data. Table 1 displays the metadata information for the BISM dataset.

Table 1. Metadata information in the BISM dataset.

Dataset Name	Data Resource		Sensor Platform	GCD (m)	Sample Numbers	Area (km^2)	Data Size (pixel)	Ground Truth
	2D	3D						
BISM (ours)	image/R-G-B-NIR	point cloud data (.las)	aerial/LiDAR	0.3	2496	60	512 × 512	Polygon vectors/raster

Generally, the BISM dataset covers 60 km^2 in Boston, Massachusetts, the United States, and comprises approximately 39,527 building objects, accounting for 23.39% of the total experimental area. The experimental area consists of various features, as shown in Figure 5. Some details are shown in the yellow rectangle for close-up inspection. Category imbalance brings challenges to the reasonable design of the model structure. Additionally, these buildings exhibit diverse textures and colors with complex geometric shapes. The above factors can enhance the potential of different models for building automatic interpretation and evaluating their generalization ability. Multispectral aerial orthoimages were obtained in 2013 from the United States Geological Survey (USGS) [42] with 0.3 m spatial resolution and red–green–blue–near-infrared (RGB-NIR) channels. Thirty orthoimages

with a size of 5000 × 5000 pixels were integrated and cropped into a mosaic image of 26,624 × 24,576 pixels. LiDAR point data (.las format) were derived from the National Oceanic and Atmospheric Administration (NOAA) [43] in 2013 with an estimated point spacing of 0.35 m, vertical accuracy of 5.2 cm, and horizontal accuracy of 36 cm.

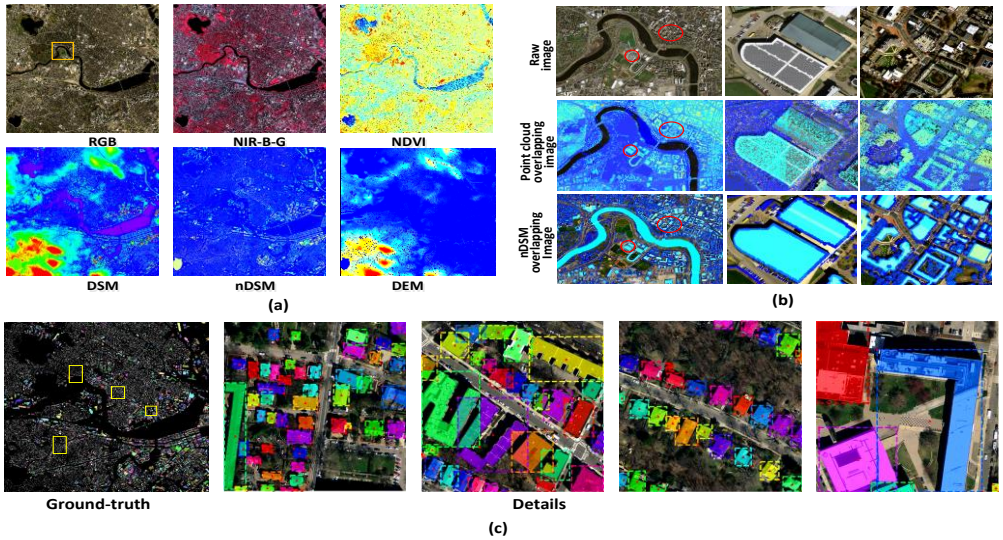


Figure 5. BISM dataset visualization. (**a**) presents the original image and lidar products; (**b**) shows detail of the yellow rectangle in (**a**); (**c**) presents the ground truth and corresponding details distinguishing each building instance with different colors. Triangle denotes the central position, and the dashed rectangle denotes the bounding box of the object.

5.2. Data Preprocessing

In the BISM datasets, noise points and outliers were removed for the 3D LiDAR point cloud data using the open-source software Cloud Compare [44]. LiDAR products were generated for different input strategies, such as normalized difference vegetation index (NDVI), digital elevation model (DEM), digital surface model (DSM), and normalized digital surface model (nDSM), as shown in Figure 5. The cloth simulation filter (CSF) [45] algorithm was used to generate nDSM and DEM. Finally, the products derived from the LiDAR point cloud data were rasterized and resampled to a spatial resolution of 0.3 m.

LIDAR point cloud data and images were geographically registered in the same projection coordinate system to reduce spatial shift. Due to the oblique errors of photogrammetry, we used DEM derived from LiDAR to complete the orthorectification for images. However, the edges of some buildings are not accurate in DSM due to the sparsity of point cloud data and the inherent errors of some interpolation algorithms. Some buildings still have oblique facades and shadows. Therefore, in the ground truth, we comprehensively considered the spatial relationship between the image and DSM to draw the correct vector boundary for the buildings. Concretely, we manually edited polygon vectors (.shp format) using ArcGIS software by visual interpretation. The results referenced the open street map (OSM).

For model training, the entire dataset was cropped into 2496 tiles with a size of 512 × 512 pixels. These tiles were divided into several subsets, including the training subset (1747 tiles), validation subset (500 tiles), and test subset (248 tiles). The data augmentation strategy was applied to the model training to increase the number of samples and enhance the model generalization ability. These sample patches were processed by fundamental image transformation, such as rotation 180° counterclockwise, adding random noise, and mirror transformation along the vertical or horizontal directions. As a result, the training data subset was increased to 5241 tiles. We used the minimum bounding box to mark the location and range of each building object. In addition, a subset (3000 tiles) was created in distinct areas from the WHU dataset [14] for the subsequent experimental comparison. The

WHU subset was augmented and divided into the training subset (4200 tiles), validation subset (600 tiles), and test subset (300 tiles).

5.3. Experimental Configuration and Metrics

In the model training phase, all experiments were completed on the Keras/Tensorflow platform using the configuration of 3 × 32 GB RAM, NVIDIA Tesla V100 GPU. Each network model was trained with 400 epochs with an initial learning rate of 0.001 and a batch size of 16. The Adam algorithm was applied to optimize training parameters with a momentum rate of 0.9. The learning rate decreased if the validation accuracy did not improve every five iterations. The weight parameters in ResNet50 were initialized by the pre-trained model in the public dataset ImageNet. Other network layers were initialized by the Xavier [45] method. Several experiments were completed to verify the performance of the proposed method.

The proposed network model uses a multi-task learning paradigm to obtain the results of instance segmentation. The accuracy of the results is affected by segmentation and detection. As a result, we evaluated its performance in an ablation experiment. Intersection over union (IOU), F_1-score, and average accuracy (AP) are widely used in the evaluation of semantic segmentation and object detection. These metrics can be calculated by other evaluation parameters, including TP, FP, FN, precision, and recall. In the regression detection task, an object is marked as a positive sample (TP) when the IOU (between the predicted bounding box and its ground truth) is greater than the threshold; otherwise, it is a false positive (FP). If the object is not identified, it is marked as a false negative (FN). In the experiment, the threshold of IOU was set to 0.5. Precision and recall were defined by Equations (22) and (23). Therefore, AP was calculated by Equation (24). Similarly, we only counted the number of pixels for each object within the bounding box to achieve the metrics in the semantic segmentation task. The predicted probability of each pixel was obtained by the Softmax function. The F_1-score was applied to the segmentation task, as defined by Equation (25).

$$Precision = \frac{TP}{TP + FN}, \tag{22}$$

$$Recall = \frac{TP}{TP + FP}, \tag{23}$$

$$AP = \int_0^1 P(R)d(R) \tag{24}$$

$$F_{1-score} = 2 \cdot \frac{\mathrm{Pr}ecision \cdot recall}{\mathrm{Pr}ecision + recall} \tag{25}$$

5.4. Ablation Study on Multi-Modal Data

To verify the influence of multi-modal data on the model prediction, we arranged seven groups using different input strategies in the BISM dataset, as shown in Figure 6. When RGB-NIR-NDVI was input, we only retained the backbone network, and others were removed. If the input contained LiDAR products, backbone and branch networks were reserved. Multispectral images were fed into the backbone network, while DEM/DSM/nDSM was fed into the branch network.

We first completed the comparative analysis in the experiment with RGB, RGB–NDVI, and RGB–NIR. Compared with inputs only using RGB, it can be observed in Figure 6 that the accuracy was improved by 0.3% AP and 1.6% F_1-score when using RGB-NIR. Similarly, RGB-NDVI as input achieved a slight increase compared to RGB-NIR. However, when DEM was introduced into the model, the prediction accuracy decreased by 2% AP and 1.5% F_1-score. Probably, the DEM could not represent the height variation of the buildings, and noise was introduced into model training. In contrast, when RGB-NDVI-DSM was fed into the network, the prediction accuracy was significantly increased by 1.5% AP and 6.7% F_1-score compared to RGB, which indicates that LiDAR features can increase the accuracy of

segmentation. The accuracy did not obviously change using RGB-NDVI-nDSM. However, the last group with DEM and nDSM achieved better results than the other groups, with 89.8% AP and 86.3% F_1-score. Therefore, we used RGB-NDVI-nDSM-DEM as multi-modal data in subsequent experiments.

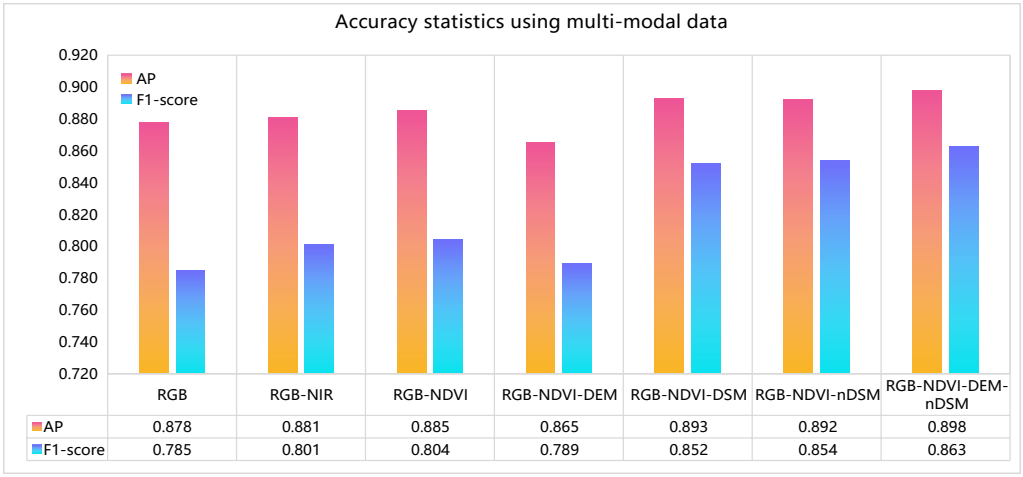


Figure 6. Accuracy assessment using multi-modal data strategies as implemented with proposed methods.

5.5. Contributions of Modules in LSSP and CLGF

Based on the above analysis, the LSSP fuses local features from different modal information, while CLGF integrates global context. Hence, we compared and analyzed the complementary ability of these two modules using the BISM dataset. Firstly, in the experiments with the LSSP, we verified the model performance using the spatial perceptron (SPA) and the spectral perceptron (SPE) in the LSSP, respectively. Meanwhile, some hyperparameters were determined by quantitative analysis and comparison. Each sub-module was applied to the model separately, and others were removed. Input data included multispectral images and LiDAR products. The feature maps from the four stages were fused through FPN. The fused feature maps with a size of 128 × 128 were used for regression and upsampled to 512 × 512 for the segmentation. ResNet50 and the CenterNet detectors were combined as the backbone model (BM) for the comparative analysis.

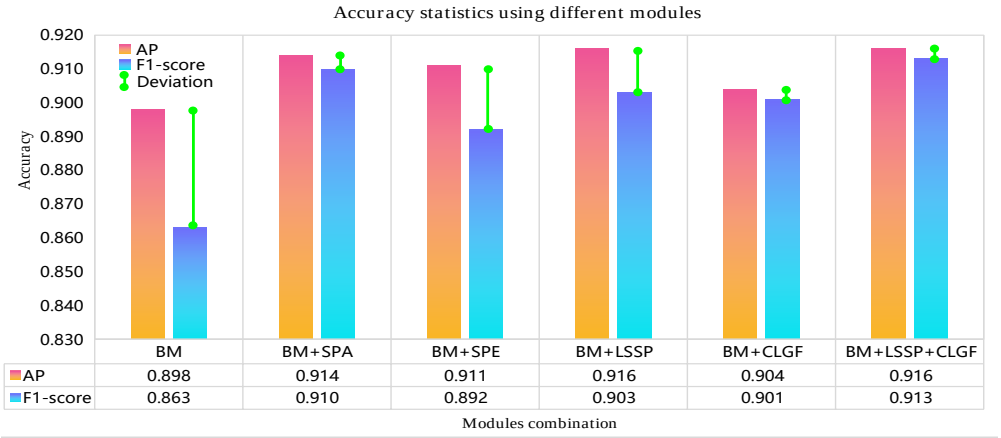
The SPA module is subject to two parameters, l and k. l determines some influence on optical features when the LiDAR feature is decomposed into multiple spatial representations. For testing module sensitivity, l is set to 2/3, 1/4, 1/8, or 1/16 of the input channel numbers, and k is set to 3, 5, or 7. Table 2 presents the accuracy of the results using different modules. It can be observed that the AP and F_1-score improved when l was increased from 1/16 to 1/4, but the performance remained stable and even decreased when l was set at 2/3. Probably, more parameters were introduced to the network structure, which increased the difficulty of training optimization. Similarly, as the k increased, AP did not show regular changes, but the F_1-score increased with a large k. Hence, based on the best computation efficiency and performance, l = 1/4 and k = 5 were set in the experiments.

Table 2. Comparison with different hyperparameters in SPA.

Metric		**AP**				**F_1-Score**			
l		**1/16**	**1/8**	**1/4**	**2/3**	**1/16**	**1/8**	**1/4**	**2/3**
k	3	0.873	0.884	0.901	0.900	0.834	0.876	0.873	0.872
	5	0.875	0.871	0.914	0.911	0.867	0.865	0.910	0.870
	7	0.872	0.883	0.917	0.896	0.866	0.867	0.879	0.874

Figure 7 shows the impact of the ablation studies on different modules with the accuracy of the results. Although AP presented a slight increase of 1.3%~1.6% when using

BM + SPA or BM + SPE compared to BM, segmentation results significantly increased with an F_1-score about 5% higher than BM. In general, the LSSP module improved prediction accuracy from 89.8% AP to 91.6% AP and 86.3% F_1-score to 90.3% F_1-score. Compared with BM, detection accuracy using CLGF slightly increased by 0.4% AP, but segmentation accuracy improved by 3.8% F_1-score. In addition, detection and segmentation tasks had a large accuracy deviation between AP and F_1-score when using BM or BM + SPE, while other combinations had relatively small variations. This indicates that the SPA made more contributions to the segmentation task than the SPE. In general, the detection accuracy was higher than the segmentation accuracy since the ROI region had an impact and limitation on the segmentation results. As a result, the above analysis demonstrates that the proposed modules can improve the prediction accuracy, and their combination gains more.



	BM	BM+SPA	BM+SPE	BM+LSSP	BM+CLGF	BM+LSSP+CLGF
AP	0.898	0.914	0.911	0.916	0.904	0.916
F1-score	0.863	0.910	0.892	0.903	0.901	0.913

Figure 7. Accuracy assessment using different modules as implemented with proposed methods.

Figure 8 displays the results of feature variation and instance segmentation. Test image A and test image B contain buildings with different scales and very similar textures to roads. To analyze the influence of the LSSP on the shallow encoders, we used feature maps via stage 2 to generate a heatmap by calculating the mean value along the channel dimension. Visually, it can be observed in the heatmap that the LSSP enhanced the boundary information and regional feature. Although the prediction range of some buildings is inaccurate, multi-scale buildings are correctly segmented. In contrast, there exist some false negatives using BM in image A. Additionally, as shown in image B, BM has a weak detection ability for large buildings. The prediction results were easily subject to the road feature, as shown in the details, which implies that the LSSP can effectively fuse spatial–spectral features to distinguish heterogeneous features.

For the CLGF, Figure 9 displays the heatmap overlapped on the raw images, revealing some variation. The background features exhibited a weak response after using CLGF, especially for roads and ground. Large-scale building areas have inconsistent feature responses in local regions, as shown in the heatmaps marked by black ellipses. In contrast, CLGF alleviated this heterogeneity and recalibrated feature distribution. The result indicates that this module can assist the network in filtering out redundant information and enhancing semantic correlations.

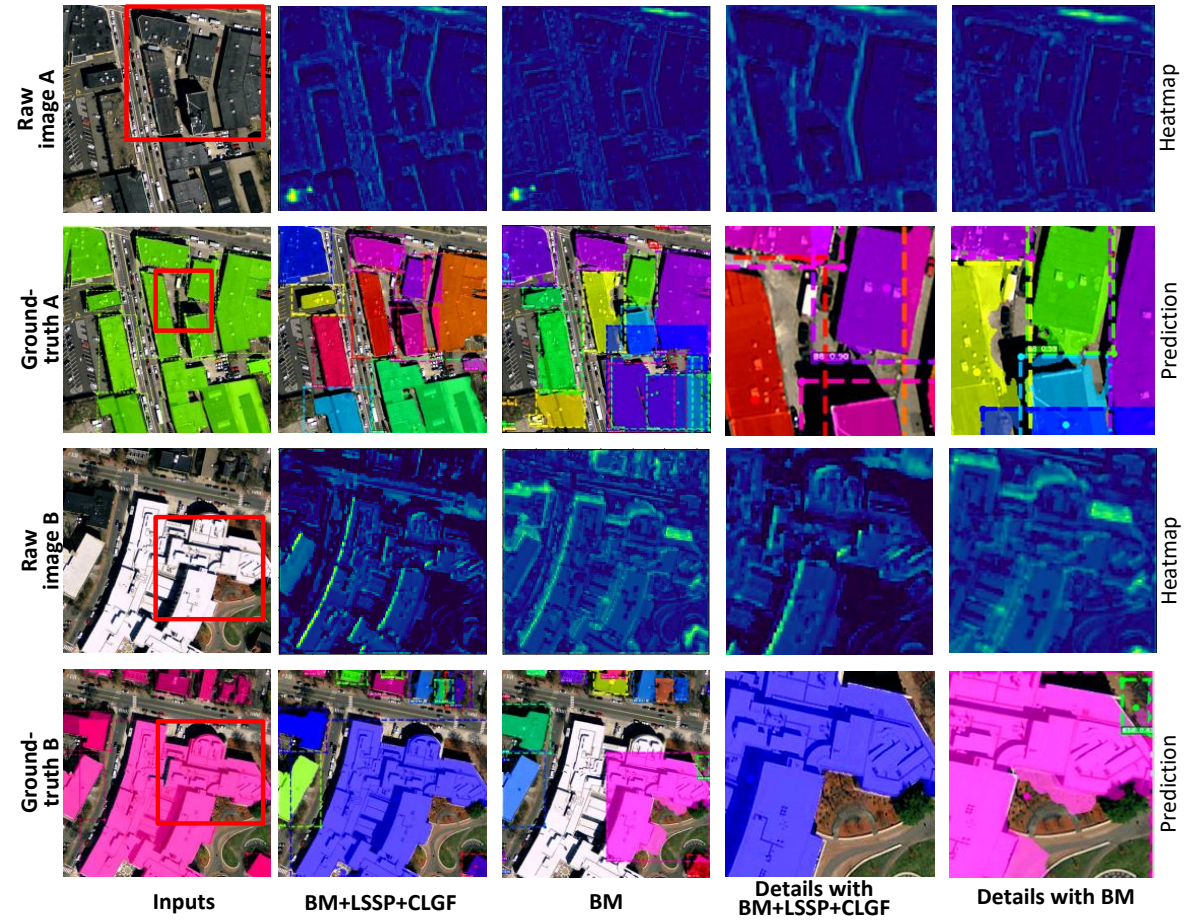


Figure 8. Each predicted building is labeled with random colors. The dotted boxes denote the predicted location. The circular points represent the predicted center points. The details in the red rectangular are from close-up inspection.

Figure 9. Feature heatmap overlapped on the raw image before and after applying CLGF. Some feature changes are marked by a black ellipse.

5.6. Center Point Detector Accuracy Analysis

In this experiment, the multispectral and DSM images were fed into the model, and other modules were removed. In the experiment, we set a threshold of 0.5 for the final center points. As shown in Table 3, compared with BM, AP increased by about 2.2% in the BISM dataset for BM + AP, which implies that the ACPD can improve the detector accuracy. Meanwhile, segmentation accuracy improved by 1.9% F_1-score.

Table 3. Influence of ACPD on prediction accuracy using the BISM dataset.

Metric	**AP**		**F_1-Score**	
Modules	BM	BM + ACPD	BM	BM + ACPD
Accuracy	0.898	0.920	0.863	0.882

The prediction results from three images were visualized, as displayed in Figure 10, to visualize the result performance. Dense and small-scale buildings exist in test image A with shadows and cement surfaces. BM + ACPD achieved better building segmentation results. However, some buildings were not detected using BM, as shown by the green ellipse.

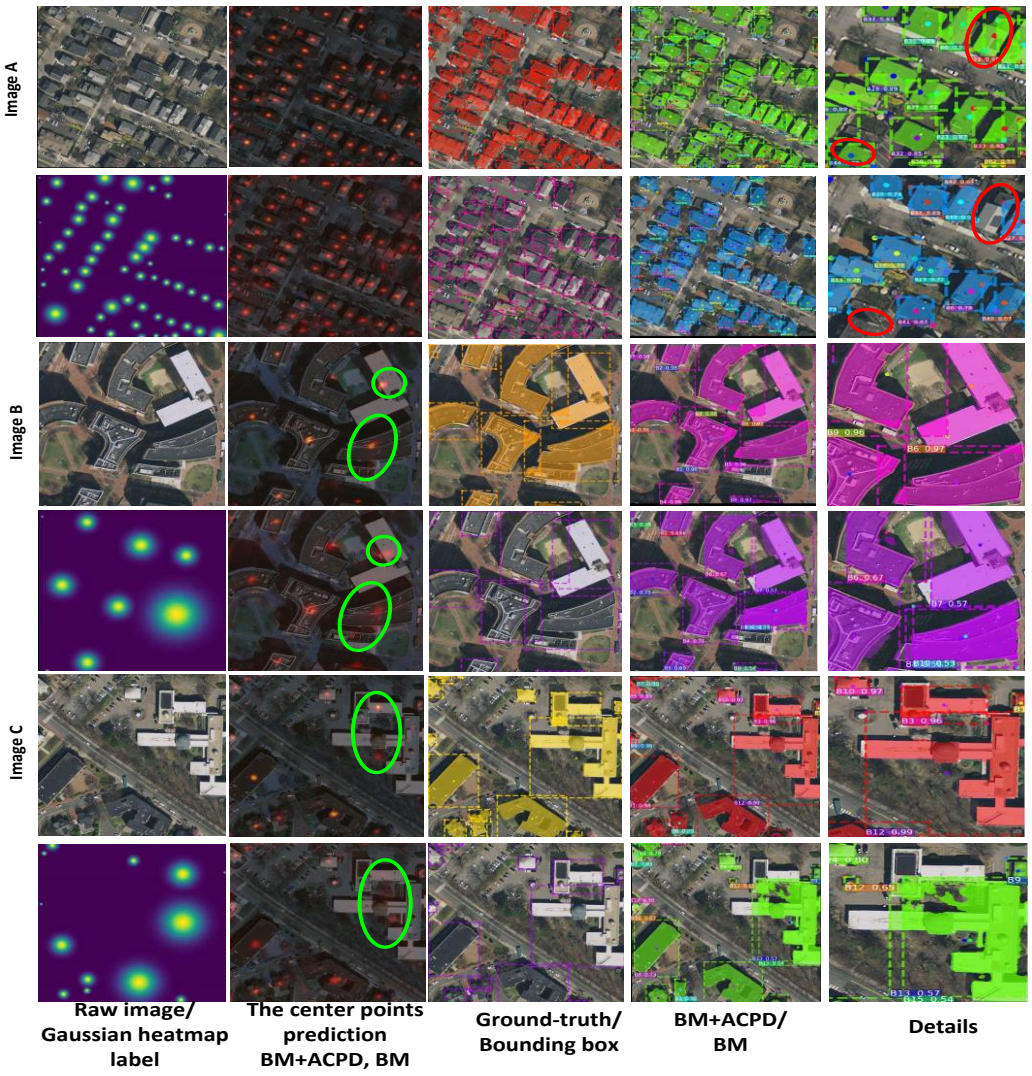


Figure 10. Three experimental results are presented. The second column is the central point prediction map overlapped on the raw image, where the red area brightness denotes the center semantic intensity. The first and third columns are the raw data and corresponding labels. The fourth and fifth columns show instance segmentation results using different methods.

Test image B has large-size and tall buildings with vegetation distribution. These buildings exhibit various shapes where some central positions do not exist in themselves. As shown in the green ellipse of Figure 10, the CenterNet detector presented multiple centers that deteriorated the position precision. In contrast, the ACPD exhibited better performance in large-size buildings. The problem of multiple centers was alleviated, and semantic information was enhanced.

Test image C contains narrow, long buildings. Compared with CenterNet detections, the ACPD identified these central positions with less deviation. Background features using BM were misclassified where central points had a weak response, as shown in the red ellipse. As a result, the ACPD can improve center regression ability, especially for buildings with complex shapes.

Furthermore, we selected other typical samples in the test data and conducted comparative experiments with the ACPD and CenterNet to verify the detection ability of the proposed module for complex-shaped buildings. Figure 11 shows examples of buildings with various irregular boundaries. Although the predicted center of some buildings had a slight deviation, such as in the first and last rows, and presented FN for some small buildings, the ACPD could better correct the central positions for large-scale buildings than the CenterNet detector. In addition, it can be observed that the same building in the sixth column presented multiple prediction centers via CenterNet, which deteriorated the semantic segmentation results, as shown in the third column. The buildings in the second and last rows were not wholly detected since many FT samples, such as some roads, were misclassified. In contrast, the ACPD optimized the center point feature and reduced the interference of background information. The proposed method can significantly suppress FT samples and improve the semantic segmentation performance compared to the results of the third and fourth columns.

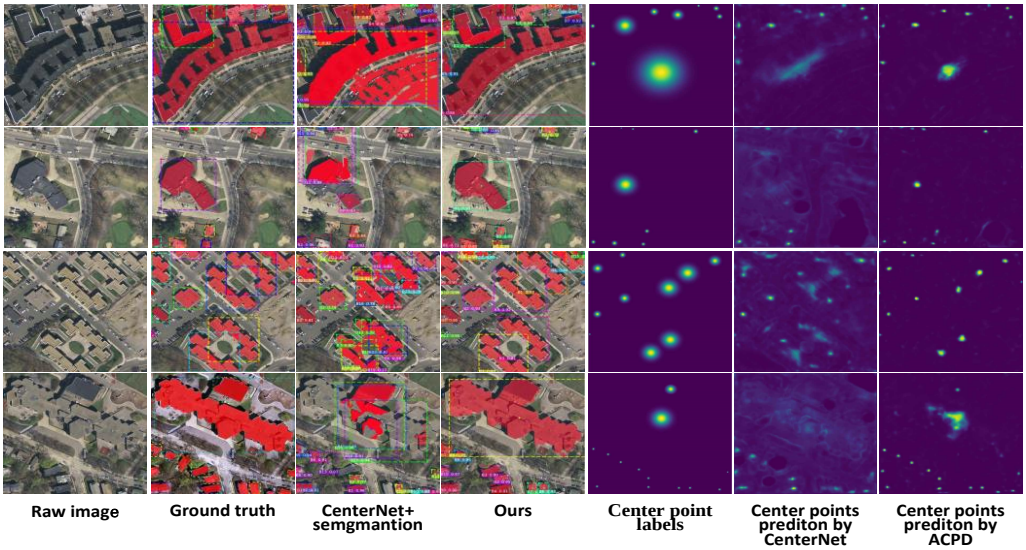


Figure 11. The comparison of CenterNet detectors and ACPD on BISM dataset.

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

This section compares the proposed model with other, state-of-the-art instance segmentation methods. BISM and WHU datasets [14] were used to verify the generality of different methods. Since the WHU dataset only contains RGB images, we used the ACPD and CLFG modules with a backbone network. For comparison and analysis, the framework of all methods adopted ResNet50 as the primary network structure, combining different types of detectors. As mentioned, we selected advanced algorithms, including Mask RCNN [13], PANet [19], SOLOv2 [22], and CenterMask [16], to verify the advantages of the proposed network. All experimental configurations were kept the same.

Tables 4 and 5 show the accuracy in two datasets using different methods. The test datasets were classified into three scales by AP and F_1-score, small (s), middle (m), and large (l), to evaluate multi-scale performance. These methods generally achieved better results in the BISM dataset than in the WHU dataset, indicating that multi-modal data contribute to building detection and segmentation performance.

Table 4. Accuracy comparison using test images on the BISM dataset. The bold format represents the best result, while the underlined values represent the second.

Type	Methods	AP	AP^s	AP^m	AP^l	F_1-Score	F_1-$Score^s$	F_1-$Score^m$	F_1-$Score^l$	Inference Time (ms)
Two-stage	Mask RCNN [13]	0.897	0.894	0.898	0.886	0.864	0.853	0.874	0.872	78.35
	PANet [19]	<u>0.901</u>	0.867	**0.925**	0.876	<u>0.885</u>	<u>0.892</u>	0.893	0.879	79.94
One-stage	CenterMask [16]	0.887	0.873	0.894	<u>0.889</u>	0.878	0.861	<u>0.895</u>	<u>0.883</u>	**44.16**
	SOLOv2 [22]	0.874	<u>0.896</u>	0.901	0.883	0.870	0.856	0.877	0.881	48.47
Multi-task	Ours	**0.929**	**0.928**	<u>0.923</u>	**0.930**	**0.918**	**0.896**	**0.921**	**0.934**	<u>46.29</u>

Table 5. Accuracy comparison using test images on the WHU dataset. The bold format represents the best result, while the underlined values represent the second.

Type	Methods	AP	AP^s	AP^m	AP^l	F_1-Score	F_1-$Score^s$	F_1-$Score^m$	F_1-$Score^l$	Inference Time (ms)
Two-stage	Mask RCNN [13]	0.778	0.672	0.815	<u>0.843</u>	0.757	0.658	0.773	<u>0.839</u>	67.38
	PANet [19]	0.792	0.775	**0.838**	0.786	0.771	0.740	**0.823**	0.757	71.24
One-stage	CenterMask [22]	<u>0.821</u>	<u>0.819</u>	0.812	0.831	0.773	<u>0.764</u>	0.777	0.763	<u>44.07</u>
	SOLOv2 [23]	0.775	0.773	<u>0.827</u>	0.784	<u>0.782</u>	0.754	<u>0.821</u>	0.775	**43.28**
Multi-task	Ours	**0.824**	**0.828**	0.784	**0.863**	**0.795**	**0.783**	0.798	**0.894**	45.21

In the BISM dataset, the proposed model performed better in multi-scale AP and F_1-score than the others, especially for large-scale buildings. Although the inference time was not the best, the speed was higher than in the two-stage detection methods. PANet was superior to other methods but had low AP^l with a long inference time and a low F_1-score for large-scale buildings. The CenterNet detection framework enabled CenterMask to achieve the best inference time and high-precision AP^l. However, this network had poor performance for small-scale buildings. SOLOv2 had the lowest detection accuracy with an AP^l of 88.3% but outperformed other methods in small-size buildings. A large number of buildings introduces more parameters and increases the difficulty of the model training. In addition, the prediction of global masks, depending on the manual setting, is not suitable for a number of dense buildings. Mask RCNN had the lowest segmentation accuracy with an F_1-score of 86.4% due to poor performance on small-scale buildings, which indicates that low spatial resolution is not conducive to high-precision segmentation for small buildings.

In the WHU dataset, although our method did not achieve the best results in AP^m and F_1-$score^m$, the total accuracy outperformed other methods. CenterMask obtained better detection results with 82.1% AP, while SOLOv2 had a higher F_1-score of 78.2% compared to others. Although PANet achieved the best performance in medium-size buildings with an AP^l of 83.8% and 82.3% F_1-$score^m$, it presented weak prediction ability in small-scale and large-scale buildings. Regarding detection efficiency and accuracy, the one-stage method had advantages over the two-stage methods for building instance segmentation.

In addition, we removed the LSSP and LiDAR products with the encoder branch to verify the model's generalization ability, using only RGB images as input in the BISM dataset. In Table A1 of Appendix A, it can be seen that PANet had a F_1-score 1.2% higher than ours in the segmentation task. On different scales, it was observed that the proposed model had poor performance in small-scale buildings, with an F_1-$score^s$ of 79.8%. Nevertheless, our method achieved the best performance in the detection task compared with other methods with 88.5% AP. In addition, comparing the accuracy shown in Table 4, AP decreased by 4.4%, and the F_1-score decreased by about 10%, which indicates that the LSSP fusing multi-modal features can improve the segmentation accuracy significantly.

Four typical areas were visualized in the test dataset to analyze the performance of the results using different methods. As shown in the first column of Figure 12a, Figure 12a,b belong to the BISM dataset, while test images C and D come from the WHU dataset. Test image A contains some buildings with complex shapes. The proposed methods achieved better performance than other models in complex-shape buildings, as illustrated in Figure 12b. Obviously, other networks had a weak ability to regress position for narrow, long buildings. In addition, Mask RCNN and PANet were sensitive to shadow changes misclassified as small holes in the roofs. Although there were accurate results for some buildings with a relatively regular shape using SOLOv2, it had poor performance in "T"-shape buildings.

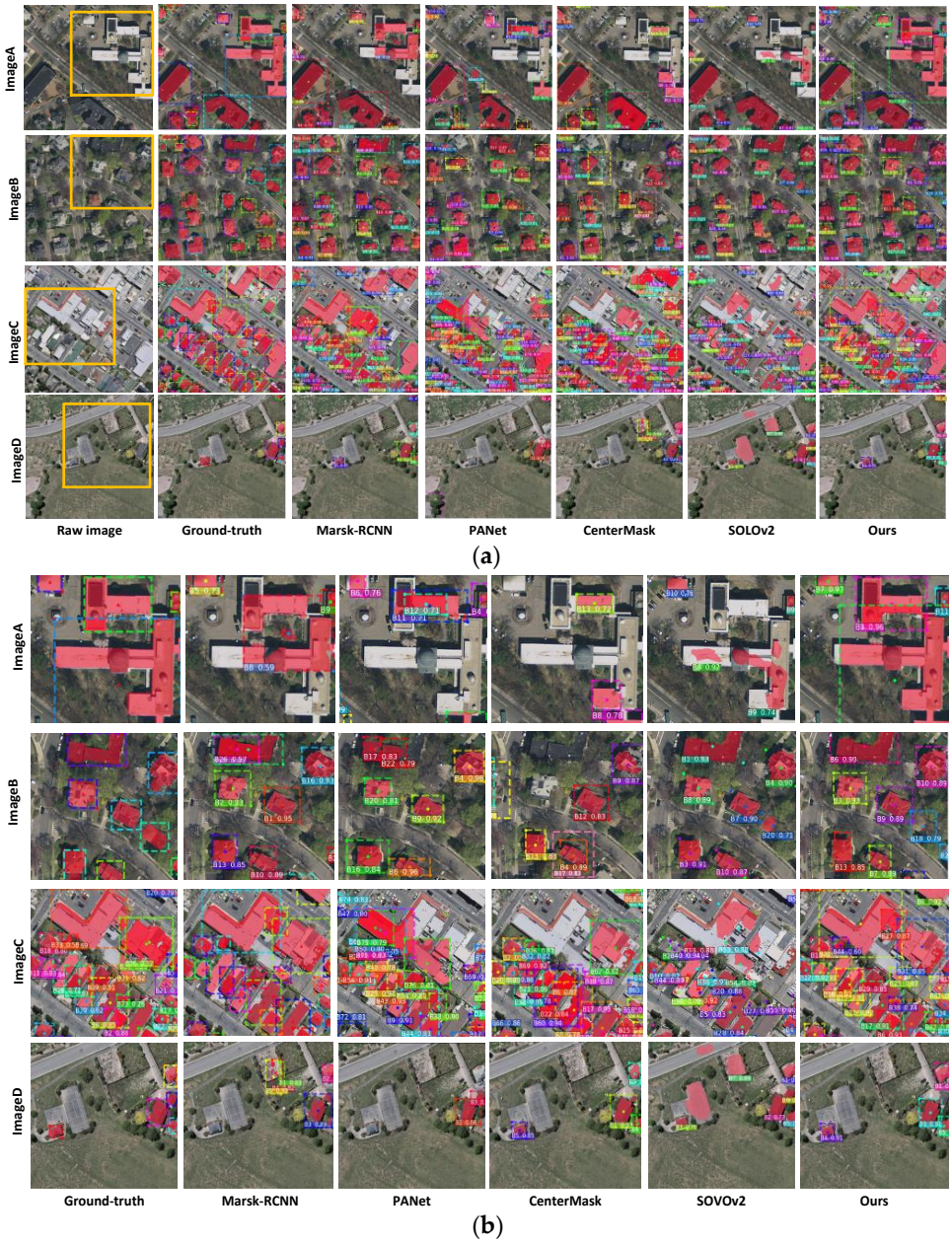


Figure 12. (**a**) presents the building instance segmentation results using different methods. The details in the yellow rectangles are enlarged and closely inspected in (**b**), where the segmentation results are marked with red for comparison, and the detection result is marked with a random color. The dotted rectangle denotes the predicted bounding box, and the dot denotes the geometric center of the bounding box.

Test image B covers relatively small-scale buildings with an amount of vegetation, as shown in the second row of Figure 12a. These methods obtained better segmentation results than test image A, but there were more false negatives for the CenterMask in the detection task. In contrast, test image C contains large-scale industrial plants that cover over 50% of the area in one patch. Moreover, some buildings were closely arranged, as displayed in the third row of Figure 12b. Our method and Mask RCNN outperformed the others, especially for large-size buildings. PANet misdetected many buildings and could not regress size accurately for the large building regions. Although SOLOv2 and CenterMask identified most buildings, the segmentation results were incomplete for some large-scale buildings. In test image D, there are sparsely distributed small buildings surrounded by cement ground and vegetation in the suburbs. Our method exhibited better results than others for small-size buildings. Mask RCNN and SOLOv2 were sensitive to the road feature.

6. Discussion

To further verify the performance of the proposed method, we completed a comparative experiment with traditional methods using the commercial software ENVI 5.3. In the object-based segmentation process of ENVI, we used the method based on edge detection to create objects and the support vector machine (SVM) algorithm to classify them. The results of semantic segmentation are presented in Figure A1 of Appendix A. The proposed method requires building large-scale training sample datasets. In contrast, the object-based method can segment the building area simply and efficiently. However, the segmentation accuracy and performance are inferior to the proposed method. In addition, object-based segmentation methods cannot obtain end-to-end object-level extraction results and require post-processing such as clustering or vectorization. Hence, we used pixel-wise overall accuracy (OA) and F_1-score as evaluation indicators.

As shown in Table A2 of Appendix A, the proposed method had more than 9% OA and 13% F_1-score compared to ENVI. Figure A1 of Appendix A shows that the object-based method had good segmentation ability in small-scale buildings with regular texture. However, many misclassifications existed for roofs and roads with complex textures or similar colors. Therefore, it is difficult to distinguish buildings and other objects with similar spectral and spatial information using only shallow semantic features.

The above experiments confirmed that the proposed method can improve the performance of building instance segmentation with high efficiency. However, some issues can still be potentially explored and optimized. Some buildings cannot be distinguished in overlapping regions of the bounding boxes, which interferes with the automatic detection of building information. It is necessary to develop rotated-object detection and obtain the correct orientation for the buildings. LiDAR products are rasterized into 2D images containing only elevation variation and 2D spatial information. Hence, the effective combination of 3D spatial and spectral information can be explored. Furthermore, instead of 2D instance detection, 3D object-level building detection is a further research direction. In addition, creating large-scale training datasets takes time and costs money. In further research work, knowledge distillation or transfer learning combined with a semi-supervised training mode is worth exploring to reduce the dependence on supervised samples.

The proposed method cannot identify multi-story heights on the same building roof. LiDAR data can provide different elevation information. Thus, in further research work, we will continue to improve the detectors to enhance the sensitivity of CNN to 3D position information. Furthermore, the proposed module consumes a lot of computational memory due to the high-dimensional feature matrix operations. It is necessary to optimize the module structure further and reduce memory consumption. In addition, simultaneous acquisition of LiDAR and image data is not easy, with high operating costs, and the proposed method contains many parameters to train, which increases the algorithm's complexity and brings difficulties to practical application. Further research will improve the encoders to provide users with flexible input modes using a lightweight network

structure. Meanwhile, we will improve datasets and provide abundant building labels (including spatial blocks in multiple heights and different functional zones).

7. Conclusions

Automatic building instance segmentation helps with vital decisions and provides comprehensive analysis for intelligent city construction. This study combined the advantages of multi-modal remote sensing data with multi-task deep learning to interpret objects. A building instance segmentation dataset was created, including high-resolution, multispectral images and LiDAR data. Meanwhile, new modules were developed to optimize and dynamically fuse multi-modal features. The LSSP module constructs spatial and spectral perceptrons for local feature optimization to mutually compensate for semantic features and spatial information. In addition, we established the CLGF module to enhance the global context in the encoders. In the decoders, the ACPD module constructs explicitly deformable convolution under supervised learning, which improves the regression ability for complex shapes and multi-scale buildings. The quantitative analysis and visual results from multiple experiments and datasets demonstrated that the proposed network framework can improve prediction accuracy with high efficiency for building instance segmentation.

Author Contributions: Methodology and writing, Q.Y.; review and supervision, H.Z.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Sichuan Province Key Laboratory of Higher Education Institutions for Comprehensive Development and Utilization of Industrial Solid Waste in Civil Engineering under grant no. SC-QWLY-2021-Y-02; the Doctoral Science Foundation under grant no. 035200242.

Data Availability Statement: The code can be obtained via https://github.com/yuanqinglie/Building-instance-segmentation-combining-anchor-free-detectors-and-multi-modal-feature-fusion.git (accessed on 30 September 2022); the BISM dataset can be downloaded via http://bismdataset.mikecrm.com/Yc5qJZD (accessed on 30 September 2022); WHU building dataset can be downloaded via http://study.rsgis.whu.edu.cn/pages/download/building_dataset.html (accessed on 30 September 2022).

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

Appendix A

Table A1. Accuracy comparison using test images on the BISM dataset only using RGB images. The bold format represents the best result, while the underlined values represent the second.

Type	Methods	AP	AP^s	AP^m	AP^l	F_1-Score	F_1-Score^s	F_1-Score^m	F_1-Score^l	Inference Time (ms)
Two-stage	Mask-RCNN [13]	0.845	0.821	0.834	0.858	0.839	0.773	0.761	0.805	67.25
	PANet [19]	0.833	0.825	0.814	0.867	**0.865**	**0.804**	**0.856**	0.819	72.37
One-stage	CenterMask [22]	0.821	0.813	0.835	0.812	0.774	0.803	0.779	0.758	44.16
	SOLOv2 [23]	0.773	0.634	0.789	0.864	0.801	0.771	0.811	0.793	46.38
Multi-task	Ours	**0.885**	**0.896**	**0.864**	**0.872**	0.853	0.798	0.824	**0.834**	**44.57**

Table A2. Accuracy comparison with ENVI using test images on the BISM and WHU dataset.

Methods	BISM		WHU	
	OA	F_1-Score	OA	F_1-Score
ENVI	0.872	0.747	0.784	0.695
Ours	0.956	0.918	0.872	0.795

Figure A1. The comparison with ours and ENVI software on BISM and WHU datasets. The details are enlarged in the yellow rectangle for view inspection.

References

1. Zheng, H.; Gong, M.; Liu, T.; Jiang, F.; Zhan, T.; Lu, D.; Zhang, M. HFA-Net: High frequency attention siamese network for building change detection in VHR remote sensing images. *Pattern Recognit.* **2022**, *129*, 108717. [CrossRef]
2. Kang, M.S.; Bae, J.H.; Kang, B.S.; Kim, K.T. ISAR cross-range scaling using iterative processing via principal component analysis and bisection algorithm. *IEEE Trans. Signal Process.* **2016**, *64*, 3909–3918. [CrossRef]
3. Xue, J.; Cao, Y.; Wu, Z.; Li, Y.; Zhang, G.; Yang, K.; Gao, R. Simulating the Scattering Echo and Inverse Synthetic Aperture Lidar Imaging of Rough Targets. *Ann. Phys.* **2022**, *534*, 2100491. [CrossRef]
4. Tian, H.; Mao, H.; Liu, Z.; Zeng, Z. Sparse imaging of airborne inverse synthetic aperture lidar micro-moving targets. *Infrared Laser Range* **2020**, 1–10.
5. Giustarini, L.; Hostache, R.; Matgen, P.; Schumann, G.J.P.; Bates, P.D.; Mason, D.C. A change detection approach to flood mapping in urban areas using TerraSAR-X. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 2417–2430. [CrossRef]
6. Yan, J.; Zhang, K.; Zhang, C.; Chen, S.; Narasimhan, G. Automatic Construction of 3-D Building Model From Airborne LiDAR Data Through 2-D Snake Algorithm. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3–14.
7. 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.* **2011**, *5*, 161–172. [CrossRef]
8. Du, S.; Zhang, Y.; Zou, Z.; Xu, S.; He, X.; Chen, S. Automatic building extraction from LiDAR data fusion of point and grid-based features. *ISPRS J. Photogramm. Remote Sens.* **2017**, *130*, 294–307. [CrossRef]
9. Tomljenovic, I.; Tiede, D.; Blaschke, T. A building extraction approach for Airborne Laser Scanner data utilizing the Object Based Image Analysis paradigm. *Int. J. Appl. Earth Obs. Geoinf.* **2016**, *52*, 137–148. [CrossRef]
10. Xia, S.; Wang, R. Extraction of residential building instances in suburban areas from mobile LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2018**, *144*, 453–468. [CrossRef]
11. Chen, S.; Shi, W.; Zhou, M.; Zhang, M.; Chen, P. Automatic building extraction via adaptive iterative segmentation with LiDAR data and high spatial resolution imagery fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2081–2095. [CrossRef]
12. Zarea, A.; Mohammadzadeh, A. A novel building and tree detection method from LiDAR data and aerial images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *9*, 1864–1875. [CrossRef]
13. Yang, S.Y.; Chi, L.; He, J. An inverse synthetic aperture lidar imaging algorithm. *Laser Infrared* **2010**, *40*, 904–909. [CrossRef]
14. Ji, S.; Shen, Y.; Lu, M.; Zhang, Y. Building instance change detection from large-scale aerial images using convolutional neural networks and simulated samples. *Remote Sens.* **2019**, *11*, 1343. [CrossRef]
15. Zhang, L.; Hu, X.; Zhang, M.; Shu, Z.; Zhou, H. Object-level change detection with a dual correlation attention-guided detector. *ISPRS J. Photogramm. Remote Sens.* **2021**, *177*, 147–160. [CrossRef]
16. Lee, Y.; Park, J. CenterMask: Real-Time Anchor-Free Instance Segmentation. In Proceedings of the CVPR 2020: Computer Vision and Pattern Recognition, Virtual, Seattle, WA, USA, 14–19 June 2020; pp. 13906–13915.
17. Wu, T.; Hu, Y.; Peng, L.; Chen, R. Improved anchor-free instance segmentation for building extraction from high-resolution remote sensing images. *Remote Sens.* **2020**, *12*, 2910. [CrossRef]
18. Yuan, Q.; Shafri, H.Z.M.; Alias, A.H.; Hashim, S.J.B. Multi-scale semantic feature optimization and fusion network for building extraction using high-resolution aerial images and LiDAR data. *Remote Sens.* **2021**, *13*, 2473. [CrossRef]

19. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
20. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9626–9635.
21. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2020**, arXiv:1904.07850.
22. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. Solov2: Dynamic and fast instance segmentation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17721–17732.
23. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9157–9166.
24. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2359–2367.
25. Dai, J.; He, K.; Sun, J. Instance-aware semantic segmentation via multi-task network cascades. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, 27–30 June; pp. 3150–3158.
26. Tseng, K.K.; Lin, J.; Chen, C.M.; Hassan, M.M. A fast instance segmentation with one-stage multi-task deep neural network for autonomous driving. *Comput. Electr. Eng.* **2021**, *93*, 107194. [CrossRef]
27. 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.
28. Wen, Y.; Hu, F.; Ren, J.; Shang, X.; Li, L.; Xi, X. Joint multi-task cascade for instance segmentation. *J. Real-Time Image Process.* **2020**, *17*, 1983–1989. [CrossRef]
29. Yoo, J.H.; Kim, Y.; Kim, J.; Choi, J.W. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 720–736.
30. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum pointnets for 3d object detection from rgb-d data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 918–927.
31. Hosseinpour, H.; Samadzadegan, F.; Javan, F.D. CMGFNet: A deep cross-modal gated fusion network for building extraction from very high-resolution remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 96–115. [CrossRef]
32. Cao, Z.; Diao, W.; Sun, X.; Lyu, X.; Yan, M.; Fu, K. C3net: Cross-modal feature recalibrated, cross-scale semantic aggregated and compact network for semantic segmentation of multi-modal high-resolution aerial images. *Remote Sens.* **2021**, *13*, 528. [CrossRef]
33. Wang, W.; Neumann, U. Depth-aware cnn for rgb-d segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 135–150.
34. Chen, Z.; Cong, R.; Xu, Q.; Huang, Q. DPANet: Depth potentiality-aware gated attention network for RGB-D salient object detection. *IEEE Trans. Image Process.* **2020**, *30*, 7012–7024. [CrossRef]
35. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
37. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
38. Li, D.; Hu, J.; Wang, C.; Li, X.; She, Q.; Zhu, L.; Chen, Q. Involution: Inverting the inherence of convolution for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12321–12330.
39. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
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. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
42. Available online: https://earthexplorer.usgs.gov/ (accessed on 30 July 2022).
43. Available online: https://coast.noaa.gov/ (accessed on 30 July 2022).
44. Available online: https://www.cloudcompare.org (accessed on 30 July 2022).
45. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.