ISPRS Journal of Photogrammetry and Remote Sensing 151 (2019) 91–105

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network

Jianfeng Huang[a], Xinchang Zhang[b,c,*], Qinchuan Xin[a,d,*], Ying Sun[a,d], Pengcheng Zhang[e]

[a] School of Geography and Planning, Sun Yat-Sen University, Guangzhou 510275, China
[b] School of Geographical Sciences, Guangzhou University, Guangzhou 510006, China
[c] The College of Environment and Planning of Henan University, Henan University, Kaifeng 475000, China
[d] Guangdong Key Laboratory of Urbanization and Geo-simulation, Guangzhou 510275, China
[e] Guangzhou Urban Planning and Design Survey Research Institute, Guangzhou 510060, China

ARTICLE INFO

Keywords:
Building extraction
Deep learning
Convolutional neural networks
Image classification
Semantic segmentation

ABSTRACT

Automated extraction of buildings from remotely sensed data is important for a wide range of applications but challenging due to difficulties in extracting semantic features from complex scenes like urban areas. The recently developed fully convolutional neural networks (FCNs) have shown to perform well on urban object extraction because of the outstanding feature learning and end-to-end pixel labeling abilities. The commonly used feature fusion or skip-connection refine modules of FCNs often overlook the problem of feature selection and could reduce the learning efficiency of the networks. In this paper, we develop an end-to-end trainable gated residual refinement network (GRRNet) that fuses high-resolution aerial images and LiDAR point clouds for building extraction. The modified residual learning network is applied as the encoder part of GRRNet to learn multi-level features from the fusion data and a gated feature labeling (GFL) unit is introduced to reduce unnecessary feature transmission and refine classification results. The proposed model - GRRNet is tested in a publicly available dataset with urban and suburban scenes. Comparison results illustrated that GRRNet has competitive building extraction performance in comparison with other approaches. The source code of the developed GRRNet is made publicly available for studies.

1. Introduction

Extracting building information from remotely sensed data is important for a wide range of geographic and environmental applications. Geographic information of buildings is valuable to traditional applications such as cartography and image interpretation (Noronha and Nevatia, 2001) that greatly relies on manual interpretation and building vectorization and advanced applications like three-dimensional city modeling, urban expansion analysis, and environment surveying (Huang and Zhang, 2011). Developing automatic and robust algorithms of building extraction is therefore a research frontier in the field of remote sensing.

Despite that tremendous efforts have been made in the last few decades to develop various building extraction methods, accurate and automatic building extraction is still challenging for both remote sensing and computer vision communities. There are mainly two difficult issues, including that (1) buildings in most of the scenes, especially the urban areas, have varied sizes and band reflectance and are often obscured by trees and their shadow (Liu et al., 2017a), and (2) the high intra-class and low inter-class variation of building objects in high-resolution remote sensing images makes it complex to extract the spectral and geometrical features of buildings (Alshehhi et al., 2017).

Based on the used data, the building extraction methods (Lee et al., 2008) mainly include image-based (Ghanea et al., 2016; Huang and Zhang, 2012), LiDAR-based (Du et al., 2017; Mongus et al., 2014; Niemeyer et al., 2014; Sampath and Shan, 2010; Wang et al., 2016) and data fusion-based methods (Meng et al., 2012; Rottensteiner et al., 2005; Zarea and Mohammadzadeh, 2016). High-resolution aerial and/or satellite images provide valuable spectral, texture, and geometric information to distinguish buildings from non-building objects (e.g., roads, water, shadows, and vegetation) (Ghanea et al., 2016). The fast-developing technology of airborne LiDAR offers three-dimensional information of the land surface and allows for retrieval of high-precision digital terrain model (DTM) that is useful to extract aboveground objects (Yan et al., 2015; Zhang, 2010). As using multi-source data could provide complementary information on building objects (Awrangjeb

* Corresponding authors at: School of Geography and Planning, Sun Yat-Sen University, Guangzhou 510275, China.
E-mail addresses: eeszxc@mail.sysu.edu.cn (X. Zhang), xinqinchuan@gmail.com (Q. Xin).

https://doi.org/10.1016/j.isprsjprs.2019.02.019
Received 17 September 2018; Received in revised form 15 January 2019; Accepted 25 February 2019
Available online 20 March 2019

et al., 2010), the methods that use both high-resolution images and LiDAR data (i.e., the data fusion-based methods) have shown to improve the building extraction results effectively rather than using data from single sources alone (Zhang, 2010).

Among the data fusion-based methods, pixel-based and object-oriented image classification are two commonly used approaches for building extraction. Haala and Brenner (1999) conducted building extraction by using a pixel-based classification framework based on the combination of the normalized digital surface model (nDSM) and multispectral images. Gilani et al. (2016) proposed a data-driven building extraction and regularization method using the detected candidate building regions and line segments in an image. Khoshelham et al. (2010) conducted a comprehensive evaluation of five different data fusion-based methods that extract features at the pixel or object levels. Based on the ISPRS WG II/4 challenging dataset (http://www2.isprs.org/commissions/comm3/wg4/detection-and-reconstruction.html) that consists of both airborne high-resolution images and LiDAR point clouds, Rottensteiner et al. (2014) made comparisons on the state-of-the-art methods of building extractions. They found that most building extraction algorithms can produce satisfactory results for buildings larger than 50 m^2, but there is still room for improvement in the accurate extraction of small buildings and building boundaries. There remain limitations in the current data fusion-based methods. First, many methods (Awrangjeb et al., 2010; Meng et al., 2012) use low- or mid-level features to distinguish building objects from non-building objects and it often needs to apply certain threshold settings or empirical decisive rules when using low- or mid-level features. Second, many algorithms (Hermosilla et al., 2011) take image segmentation as a prerequisite step, of which the results are highly dependent on the segmentation parameter settings and are often easily affected by factors such as solar radiation, shadows and even random noise in remote sensing images (Fu et al., 2017).

Recent studies have demonstrated that deep Convolutional Neural Networks (CNNs) could achieve impressive performance on processing remote sensing images, such as scene classification and object detection. CNNs could automatically learn not only low- and middle-level features but also high-level semantic features from the raw images (Zhao et al., 2017). Prior methods have used CNNs for semantic segmentation of remotely sensed data (Paisitkriangkrai et al., 2016; Saito and Aoki, 2015), in which each pixel is labeled with the category of its enclosing region. However, these methods in the frame for category classification often generate lower resolution feature maps than the input images and shows coarse results in pixel-wise labeling (Badrinarayanan et al., 2017). Long et al. (2015) proposed a fully convolutional neural network (FCN) that performs both image segmentation and pixel-wise labeling synchronously via an end-to-end encoder-decoder framework. Benefitting from the ability to get full-resolution classification maps, FCN has now become a common framework for some state-of-the-art semantic segmentation methods (Marmanis et al., 2018). CNNs and FCNs have been widely used in urban objects extraction such as buildings, roads, and trees (Cheng et al., 2017; Kaiser et al., 2017; Liu et al., 2017a; Sun et al., 2018). Mnih and Hinton (2010) established a deep neural network for large-scale road and building detection using aerial images in Massachusetts. Paisitkriangkrai et al. (2016) proposed a new method that uses both deep features and hand-crafted features to perform semantic labeling of aerial images. Yuan (2017) designed a deep convolutional network that integrates hierarchical layer activations for pixel-wise prediction of buildings. Xu et al. (2018) designed a new FCN model for building extraction and employed both hand-crafted features and the guided filtering technique to improve the classification results. Wu et al. (2018) proposed an end-to-end segmentation network that synthesizes the multi-stage supervision technique. The above-mentioned studies not only develop and validate new CNNs and FCNs models on building extractions but also generously offer public available remote sensing datasets for comparative scientific studies.

There are still issues to be properly addressed in the current studies based on FCNs. First, FCNs perform pixel-wise classification by using the high-level but coarse-resolution semantic features from CNNs. Because the rich low-level image features such as building edges and building corners are largely neglected, FCNs often produce “blobby” extraction results (Bischke et al., 2017). An effective solution is to transmit the low-level features into the decoder part of FCN by skipping connections (Lin et al., 2017; Liu et al., 2017b) or reusing the maximum indices of the pooling layers (Badrinarayanan et al., 2017; Noh et al., 2015). Many existing methods do not consider feature selection during the process of transmission (Wang et al., 2017), such that redundant features are generated and the learning efficiency of the network is reduced. Second, the transmitted features often contain categorical ambiguity (Islam et al., 2017) or non-boundary related information that has no effect on the refinement of the classification results. Marmanis et al. (2018) fused the boundary detection result from HED network (Xie and Tu, 2017) with the deep encoder-decoder network to compensate for the edge loss during the down-sampling process. Wang et al. (2017) proposed a gated segmentation neural network (referred to as GSN) that can adaptively select the effective information for feature fusion and obtained competitive segmentation results on the ISPRS labeling benchmark. Islam et al. (2017) introduced a gate unit to the decoder part of FCN models (referred to as G-FRNet) that could filter out the features of categorical ambiguity. Inspired by the gate mechanisms proposed in GSN and G-FRNet, we intend to design a new gated network to optimize FCN in building extraction as the useful features can be selected via gate units. Third, most FCN models focus on extracting buildings from aerial images and there is still a need to understand their performance on the dataset fused from both high-resolution optical images and LiDAR point clouds. Adding LiDAR data to FCN models could potentially improve the accuracy of building extraction but also increase the learning difficulties of the networks because the initialization parameters of many FCN models are learned from natural images. Comprehensive evaluation of different FCN models using the fused data for building extraction would also be valuable.

The main contributions of this study are: (1) proposing a new gated semantic segmentation neural network that could be used to extract buildings from high-resolution aerial images and LiDAR data, (2) analyzing the effect of gated feature labeling unit for the refinement of the coarse classification results, and (3) comparing the performance of the state-of-the-art deep models using a large dataset from different city scenes.

2. Methodology

Here we develop an end-to-end trainable gated residual refinement network (GRRNet) for building extraction using both high-resolution aerial images and LiDAR data. The developed network is based on a modified residual learning network (He et al., 2016) that extracts robust low/mid/high-level features from remotely sensed data. A new gated feature labeling (GFL) unit is introduced to reduce the unnecessary feature transmission and refine the coarse classification maps in each decoder stage of the network. We first introduce background that is relevant to our proposed model briefly and then describes GRRNet in details.

2.1. Background

2.1.1. Deep residual network

A typical CNN consists of three types of layers, including the convolutional layer, the rectified linear unit (ReLU) layer, and the pooling layer. The convolutional layer convolves the input image with a set of filters and each filter generates a feature map in the output image. The ReLU layer generates an output of 0 if the value in the feature map is less than 0 and otherwise generates an output that is equal to the input

feature. The pooling layer obtains abstract features by compressing the input feature maps and simplifies the computational complexity of the network. By combining these layers in an orderly manner, CNN learns the low/mid/high-level features that are more robust than traditional hand-crafted features in the input images (Zhang et al., 2016). Previous studies (He et al., 2016) have found that increasing the depth of neural networks do not necessarily improve the performance of CNN. Instead, the accuracy may degrade after a saturation. This phenomenon is often referred to as the degradation problem. He et al. (2016) recently proposed a Residual Network (ResNet) that is considerably deeper than previous networks like VGG-16 (Simonyan and Zisserman, 2014) and solved the degradation problem effectively. Our study chooses ResNet-50, the well pre-trained ResNet model (https://github.com/KaimingHe/deep-residual-networks) for the ImageNet challenge, as our basic block for building extraction.

2.1.2. Encoder-decoder network architecture

FCN models, such as FCN8s (Long et al., 2015), SegNet (Badrinarayanan et al., 2017), DeconvNet (Noh et al., 2015), and CNN-FPL (Volpi and Tuia, 2017), often have an encoder-decoder architecture. The encoder-decoder architecture could fully utilize CNNs to extract image features and effectively solve the end-to-end learning problem of semantic segmentation (Long et al., 2015). The encoder part could be a deep CNN (e.g., VGG-16, ResNet-50) that consists of a series of convolutional operations, non-linear operations, and pooling operations and the encoder part obtains high-level semantic features with spatial dimensions smaller than the input images. Different from the encoder part, the decoder part enlarges the features obtained by the encoder using upsampling layers or max unpooling layers and produces the final prediction result with spatial dimensions the same as the input images. The max unpooling layer enlarges the features by reusing the locations of maxima within each max-pooling layer (Badrinarayanan et al., 2017) and the upsampling layer is often a learnable deconvolutional layer or bilinear interpolation layer (Noh et al., 2015). The proposed model has the encoder-decoder architecture and employs the upsampling layer of bilinear interpolation.

2.1.3. FCNs for building extraction

The FCN model allows for predicting the probability that each pixel belongs to the classes of building or non-building in an image. The final classification map for a given image can be obtained by calculating the category corresponding to the maximum probability of each pixel. The function to extract buildings can be described as follows:

$$\hat{k} = \underset{k\in\{0,1\}}{\operatorname{argmax}} p_k(x^n|\theta), \quad \forall\ n \in \{1, 2, \cdots, N\} \tag{1}$$

where x^n denotes the n-th pixel and N is the total number of pixels in the given image; k is a binary value, where 0 and 1 represent the non-building category and building category, respectively; $p_k(x^n|\theta)$ is the posterior probability of x^n belonging to the category k, and it is estimated by an FCN model with parameters θ.

2.2. Network architecture

Our proposed GRRNet consists of an encoder network and a decoder network (Fig. 1). The encoder network can receive multiple input images from both remote sensing images and LiDAR data. This study uses the red (R), green (G) and near-infrared (NIR) bands from multispectral images and the LiDAR-derived nDSM as the input data for tests. The output of the encoder network is a binary classification map, where 0 and 1 represent non-building and building, respectively. The spatial dimensions of each input image and output classification map are set as 480 × 480 pixels.

The encoder network (Fig. 1) is largely based on ResNet-50 and the layers are grouped into blocks according to the size of output features (the blocks in different sizes are displayed in different colors in Fig. 1). A more detailed description of ResNet-50 can be found in (He et al., 2016) and here ResNet-50 is modified as follows to improve the model performance. First, a convolutional layer is added at the beginning of ResNet-50, followed by a batch-normalized (BN) layer and a ReLU layer. The newly added convolutional layer allows for receiving multiple input image bands and produces 64 features in the same size as the input data. The idea is to break the limitations of three-band input of ResNet-50 and provide the same size feature maps for subsequent upsampling operations. As a result, the band number for the next convolutional layer in ResNet-50 is changed from 3 to 64. Second, the last three layers of ResNet-50 are replaced with a dropout layer to avoid overfitting. In each block of ResNet-50, identity shortcuts are repeatedly used two or more times to optimize the training of the network while maintaining the feature size in a block unchanging. The projection shortcuts are used between every two different size blocks to increase the number of output bands and reduce the size of features. Within the encoder network, we could obtain 6 feature blocks with different sizes that range from 15 × 15 to 480 × 480 pixels.

The output features of the encoder (with 2048 bands) are passed to the decoder and are then convolved into coarse labeling maps with only two bands (2 × 15 × 15 pixels). A standard 2 × upsampling operation and a convolutional operation are repeatedly used in each coarse labeling map for five times to obtain the final prediction map (480 × 480 pixels). Note that two issues need to be solved explicitly here. First, the decoder network does not make good use of rich low/mid-level features obtained from the encoder, making it easily produce blobby extraction results (Bischke et al., 2017), especially for small building objects. Second, transmitting the encoder features into the decoder network without any feature selection has no effect on the refinement of coarse labeling maps, because these features usually contain a large amount of non-boundary related information and are of categorical ambiguity. A new component, named as the gated feature labeling (GFL) unit, is therefore introduced to solve the issues related to feature selection and feature transmission. Details on the GFL are illustrated in the following section.

2.3. Gated feature labeling unit

Fig. 2 illustrates an example for different refined modules of CNNs that transmit the rich low/mid-level features into the upsampling stages. SegNet (Badrinarayanan et al., 2017) upsamples the decoder features using the max pooling indices followed by a trainable decoder filter bank (Fig. 2a). Res-U-Net (Xu et al., 2018) upsamples the decoder features and the corresponding encoder features separately and concatenates them for the next upsampling stage (Fig. 2b). TD-CEDN (Liu et al., 2017b) first applies a deconvolutional layer to the decoder features and then concatenated the upsampled results with the lower encoder features followed by convolutional, batch-normalized, ReLU and dropout layers (Fig. 2c). In the upsampling stages of the above-mentioned modules, all encoder features (or max pooling indices) are transmitted into the decoder without selection such that the number of decoder features is often numerous. Different from these modules, the proposed GFL unit integrates the upper encoder features into the upsampling stages such that the decoder features in GRRNet are then restricted.

The GFL unit is illustrated in Fig. 2e and described as follows. For a GFL unit G_c, its inputs include the lower encoder features f_e^l and upper encoder features f_e^u coming from a specific encoder stage, respectively. f_e^l have large spatial dimensions and small receptive fields and f_e^u have small spatial dimensions and large receptive fields. The way that the GFL unit integrates the input features can be described as follows:

$$M_e^c = BN(BN([f_e^l \otimes w_{3\times3}^c]) \odot UP([f_e^u \otimes w_{3\times3}^c])) \tag{2}$$

where c denotes the number of bands of the output features; and $BN(\cdot)$, $UP(\cdot)$, $\otimes$, and $\odot$ denote the batch normalization operator, the

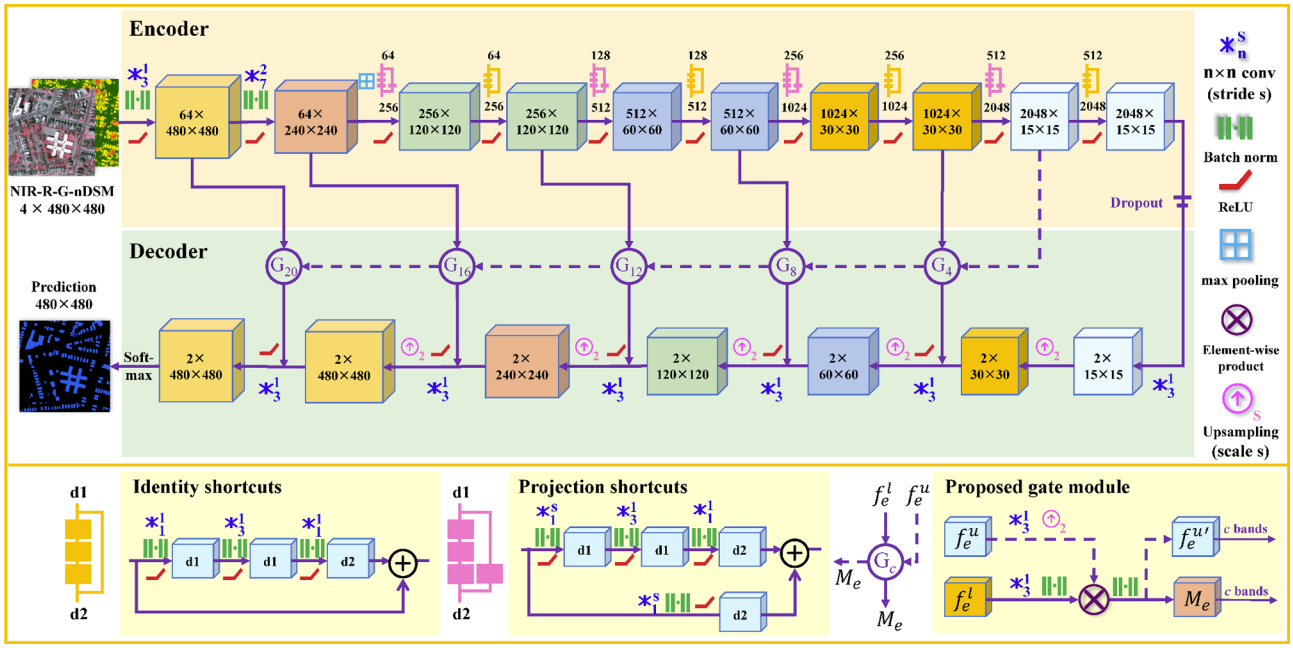

Fig. 1. An overview of the gated residual refinement network (GRRNet).

upsampling operator, the convolutional operator, and the element-wise product operator, respectively; $w^c_{3\times3}$ denotes 3 × 3 convolutional kernels of ⊗ and the outputs of ⊗ contain c bands; M^c_e denotes the selected encoder features as derived from f^l_e and f^u_e.

The selected encoder features M^c_e are then transmitted into the decoder network and fused with the coarse labeling maps f^l_d as follows:

$$M^{c+2}_d = ReLU\,(CONCAT\,(M^c_e,\ UP(f^l_d))) \tag{3}$$

where $ReLU(\cdot)$ and $CONCAT\,(a, b)$ denotes the non-linear operator and the concatenation operator, respectively; M^{c+2}_d denotes the fused decoder features with $(c + 2)$ bands.

The fused decoder features M^{c+2}_d are convolved into the upper coarse labeling maps f^u_d as follows:

$$f^u_d = M^{c+2}_d \otimes w^2_{3\times3} \tag{4}$$

One difference between the Gate unit as described in (Islam et al., 2017) (Fig. 2d) and the GFL unit is that the selected encoder features M^c_e are reused in the next gated stage in the GFL unit. In other words, the upper encoder features $f^{u'}_e$ of the next encoder stage are replaced by M^c_e. The idea is to connect the uppermost encoder features to the lowest encoder features, not only limited to interaction with the adjacent encoder features. Another difference is that the number of features

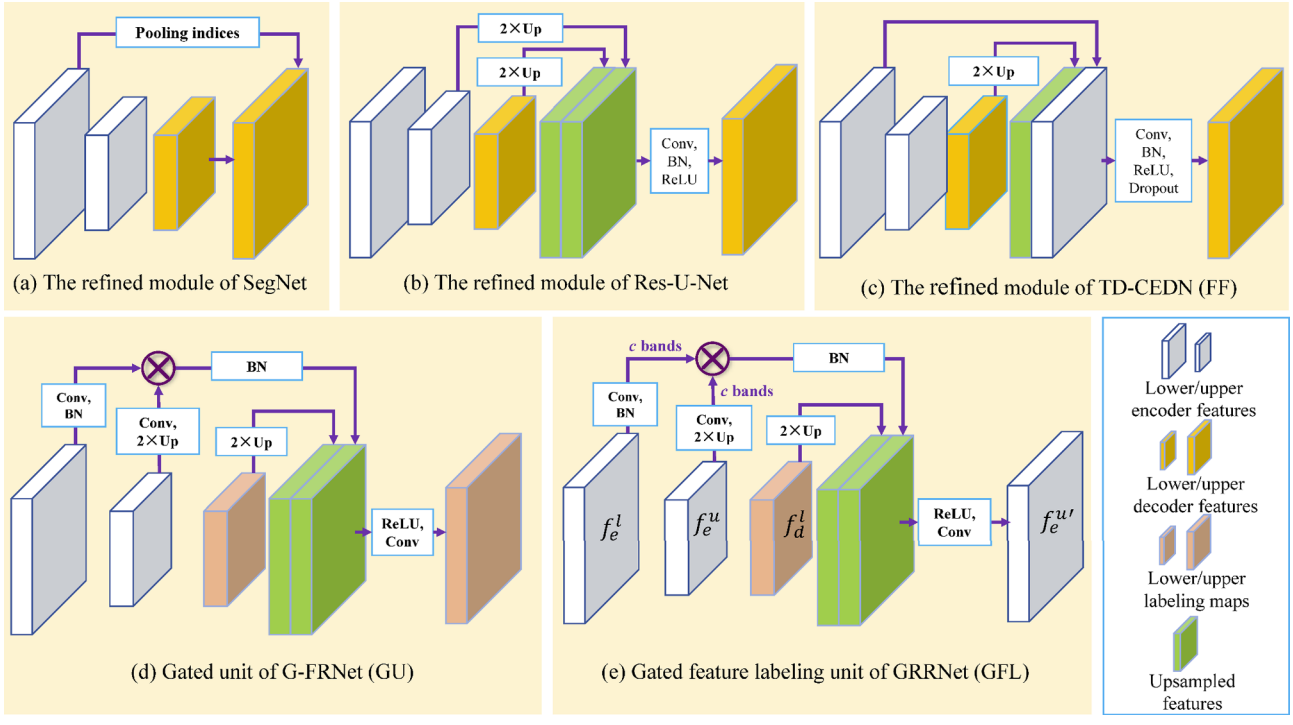

Fig. 2. Schemes are illustrated for different refined modules in SegNet, Res-U-Net, TD-CEDN, G-FRNet and GRRNet.

transmitted by the GFL unit varies with the encoder level at which the GFL unit is located. The lower the encoder stage, the more the features are transmitted. Because lower-level features are considered to contain more detailed geometry, color, and texture characteristics of building objects than the upper-level ones, the GFL unit allows for preserving useful low-level features for building object identification. As shown in Fig. 1, five GFL units in total are applied in GRRNet, and the numbers of features that are transmitted by the GFL unit are 20, 16, 12, 8 and 4, respectively.

2.4. Class balancing

The objective function for training GRRNet is the cross-entropy loss function as calculated by summing up all pixels in a mini-batch. Let $x^{(n,b)}$ be the n-th pixel in the b-th patch image and $y^{(n,b)}$ be the category label. The loss calculation is defined as follows:

$$p_k^{(n,b)} = \frac{\exp(m_k^{(n,b)})}{\sum_{k\in\{0,1\}}\exp(m_k^{(n,b)})}, \quad \forall\, n \in \{1, 2, \cdots, N\}, \quad \forall\, b \in \{1, 2, \cdots, B\} \tag{5}$$

$$Loss = -\frac{1}{N\times B}\left[\sum_{n=1}^{N}\sum_{b=1}^{B}\left(I\{y^{(n,b)}=0\}\log p_0^{(n,b)} + I\{y^{(n,b)}=1\}\log p_1^{(n,b)}\right)\right] \tag{6}$$

where k is binary where 0 represents non-building and 1 represents building; N is the total number of pixels in a patch image; B is the mini-batch size; $m_k^{(n,b)}$ and $p_k^{(n,b)}$ denote the response parameter obtained from uppermost decoder stage and the category probability of $x^{(n,b)}$, respectively; and $I\{y^{(n,b)} = k\}$ is an indicator function that has the value of 0 when $y^{(n,b)} \neq k$ and the values of 1 for other cases.

The loss weights of both building and non-building categories are the same in Eq. (6) but in most scenes, non-building pixels are more than building pixels. Class imbalance could result in an unbalanced distribution of features in the training datasets, making the classifiers tend to classify a pixel into a majority class. To solve the problem of class imbalance, we apply the *median frequency balancing* method (Eigen and Fergus, 2015) to calculate the loss weights of different categories as described as follows:

$$\begin{cases} w_k = f_{\text{median}}/f_k \\ f_k = pix\_num_k/(img\_num_k \times W \times H) \end{cases} \tag{7}$$

where w_k denotes the loss weight of category k; W and H denote the width and height of a single image; pix_num_k denotes the pixel numbers in category k; img_num_k denotes the number of images where the pixel in category k is present; f_k denotes the pixel frequency in category k; and f_{median} denotes the median of all f_k.

Finally, the class-weighted loss function is modified as follows:

$$Loss = -\frac{1}{N\times B}\left[\sum_{n=1}^{N}\sum_{b=1}^{B}\left(I\{y^{(n,b)}=0\}\log p_0^{(n,b)} w_0 + I\{y^{(n,b)}=1\}\log p_1^{(n,b)} w_1\right)\right] \tag{8}$$

where w_0 and w_1 denote loss weights for non-building and building categories, respectively.

3. Experiment design

3.1. Study materials

A publicly available dataset (https://dx.doi.org/10.6084/m9.figshare.3504413) compiled by the 2016 Data Plus Energy Analytics Group (hereinafter referred to as the DataPlus dataset) is used. The used dataset contains (1) high-resolution aerial orthoimages with the spatial resolutions ranging from 0.15 m to 0.3 m as obtained from the United States Geological Survey (USGS), (2) LiDAR point clouds obtained from the LiDAR surveys conducted by state and federal agencies, and (3) building footprints corresponding to the images as downloaded from the Open Street Maps (OSM).

14 orthoimages that have the size of 5000 × 5000 pixels that cover five different cities were chosen as the experimental dataset. All high-resolution images have four spectral bands, including near-infrared (NIR), red (R), green (G) and blue (B). The LiDAR point clouds were used to derive the nDSM data as inputs for the networks. To mitigate the sparse noise effects, the noise points were first filtered from the raw point clouds using LasTools. The bare-earth and first-return LiDAR data were then interpolated using the natural neighbor interpolation method into the digital elevation model (DEM) and DSM. nDSM was simply generated as the difference between DSM and DEM. All nDSM data derived from LiDAR were processed to have the same resolution as the corresponding orthoimages. The OSM polygons were processed to label the ground reference images into the building and non-building categories. Because the used remote sensing images and OSM vector data were acquired at different time, the OSM polygons, if incorrect or missing, were manually corrected. The OSM polygons were rasterized to obtain building footprints for network training and evaluation. In total, 9 out of the 14 annotated images were used for network training and the remaining 5 images (each for one individual city) were used for network testing. As shown in Table 1, the DataPlus dataset consists of images with varied spatial resolutions, surface elevations, and building densities in urban and suburban scenes, representing varied environmental conditions that are ideal for understanding the robustness of the developed and tested methods.

Fig. 3 shows the examples of the training set images and the corresponding close-ups in the marked areas. The urban areas (e.g., Fig. 3e in San Francisco) have higher building densities than the suburban areas (e.g., Fig. 3a in Arlington). Buildings in the suburban areas often

Table 1
Information on all training set images and test set images for five cities. The size of each image is 5000 × 5000 pixels.

Type	Image name	Scene	Location	Year	Resolution	Mean elevation	Number of buildings
Training images	Arlington_02	Suburban	Massachusetts	2013	0.3 m	30.2 m	2139
	Arlington_03	Suburban	Massachusetts	2013	0.3 m	70.4 m	1570
	NewHaven_02	Suburban	Connecticut	2012	0.3 m	11.1 m	1640
	NewYork_02	Urban	New York	2014	0.5 ft	11.3 m	1253
	NewYork_03	Urban	New York	2014	0.5 ft	25.6 m	1287
	Norfolk_02	Suburban	Virginia	2013	1 ft	2.5 m	2079
	Norfolk_03	Suburban	Virginia	2013	1 ft	3.8 m	2158
	SanFrancisco_02	Urban	California	2015	0.3 m	165.3 m	4186
	SanFrancisco_03	Urban	California	2015	0.3 m	30.2 m	5305
Test images	Arlington_01	Suburban	Massachusetts	2013	0.3 m	9.7 m	2232
	NewHaven_01	Suburban	Connecticut	2012	0.3 m	17.4 m	1174
	NewYork_01	Urban	New York	2014	0.5 ft	9.6 m	871
	Norfolk_01	Suburban	Virginia	2013	1 ft	2.9 m	2053
	SanFrancisco_01	Urban	California	2015	0.3 m	92.4 m	4123

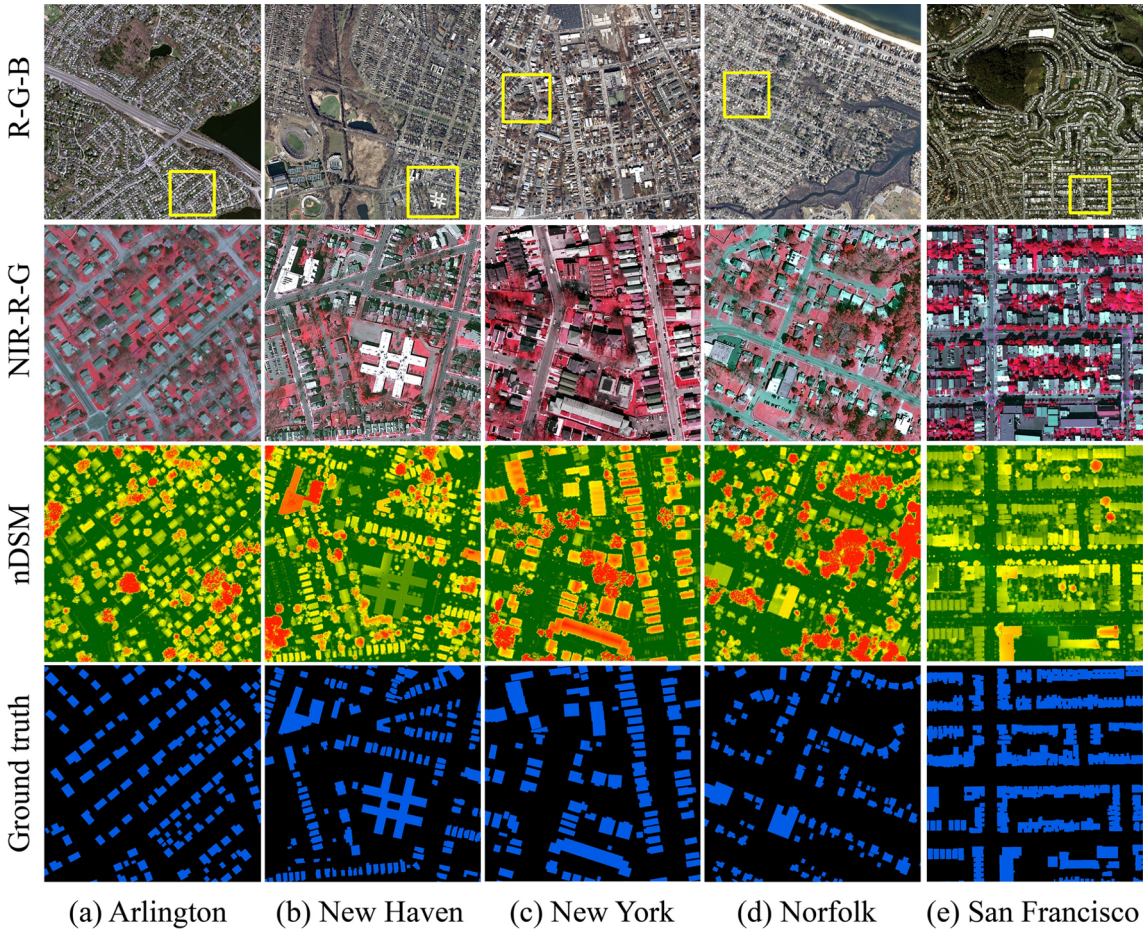

Fig. 3. Examples for the DataPlus training set images (top row) and the corresponding subset images (marked in yellow rectangles) for NIR-R-G false-color composite images (second row), nDSMs (third row), and ground references (bottom row), respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

have relatively low heights and are easily overshadowed by surrounding trees (e.g., Fig. 3d in Norfolk), whereas buildings in the urban areas are obviously higher than adjacent objects (e.g., Fig. 3c in New York). These scenes with contrasting building densities, building sizes, and surrounding circumstances make it challenging for the task of automatic building extraction.

3.2. Comparative studies using different networks

Five state-of-the-art FCN models, including SegNet (Badrinarayanan et al., 2017), DeconvNet (Noh et al., 2015), CNN-FPL (Volpi and Tuia, 2017), V-FuseNet (Audebert et al., 2017), and Res-U-Net (Xu et al., 2018), were used for comparisons. These methods were selected because all of them have already proven effective in semantic labeling and/or building extraction for remote sensing images and all of them are open source with easy implementation and accept multiple input images including the height information data. Details for each network can be found in the corresponding publication and here we only provide a brief summary to highlight the network characteristics.

3.2.1. SegNet

SegNet is a classic deep learning method and is often used as a baseline for evaluating the performance of semantic segmentation methods because it has elegant encoder-decoder architecture and has high efficiency. The pooling indices in the encoder part in SegNet are reused in the decoder part (Badrinarayanan et al., 2017).

3.2.2. DeconvNet

Noh et al. (2015) proposed a deconvolution network where the encoder part is based on VGG-16 and the decoder part consists of a series of deconvolutional and unpooling layers. This model achieved the state-of-the-art performance on the PASCAL VOC 2012 dataset.

3.2.3. CNN-FPLaaa

Volpi and Tuia (2017) designed a full patch labeling CNN model where deconvolutions are used to upsample the spatially coarse feature maps back to the initial resolution. CNN-FPL achieved the results aligned with state-of-the-art models on ISPRS Vaihingen and Potsdam challenging datasets without any postprocessing.

3.2.4. V-FuseNet

Audebert et al. (2017) proposed a novel Fuse-Net based architecture for early fusion of the LiDAR data and multispectral images. A “virtual” encoder that fuses activations from the other encoders with convolution and summation operations is embedded in the encoder network. V-FuseNet obtained results that are competitive with the state-of-the-art methods.

3.2.5. Res-U-Net

Xu et al. (2018) designed a fully convolutional network for building extraction, where the deep residual network acts as the encoder part and a guided filter is used for postprocessing. The input images include four spectral bands (NIR-R-G-B) and additional hand-crafted features like NDVI and nDSM.

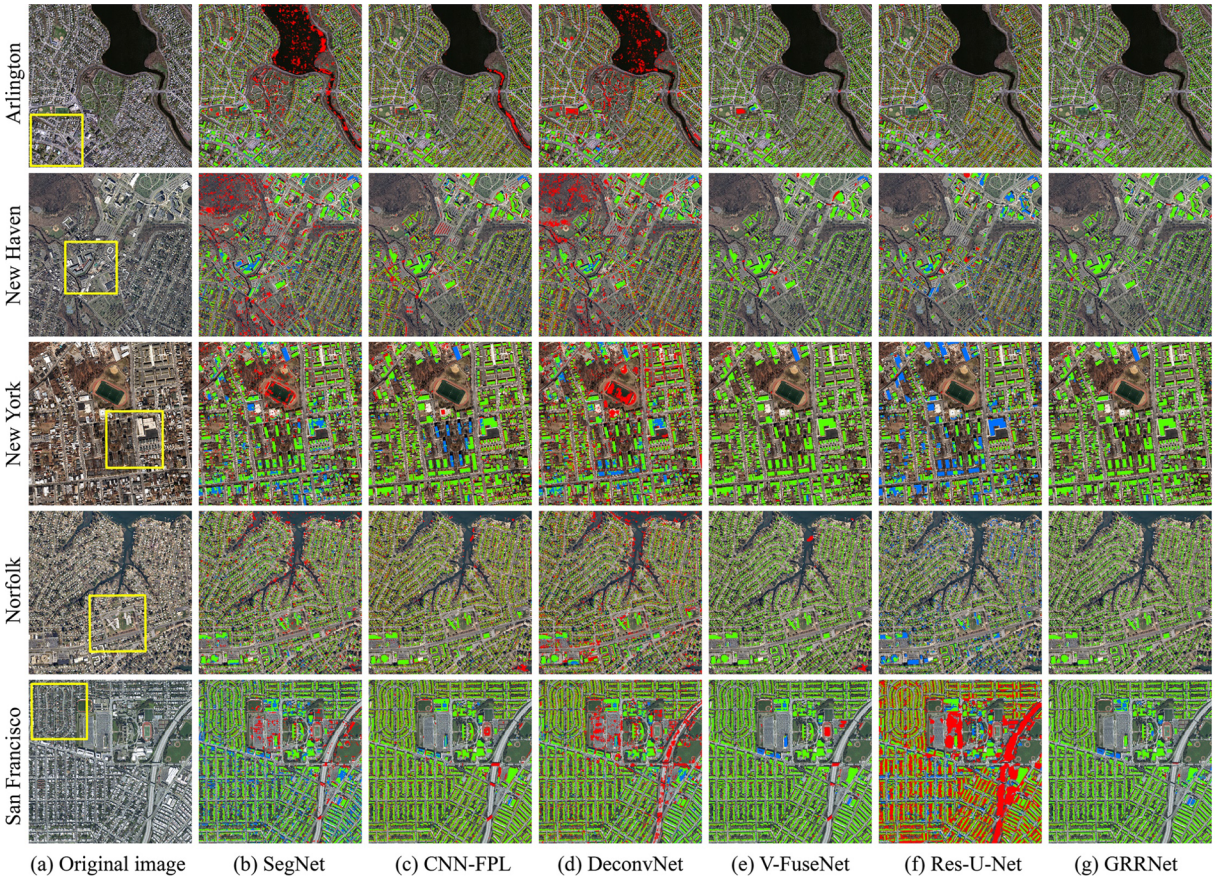

Fig. 4. Images are shown for the original true-color composite images and the classification results using the state-of-the-art deep learning methods across five cities. The true positive (TP), false positive (FP) and false negative (FN) are marked in green, red, and blue, respectively. The yellow rectangles in the original images are enlarged for close-up inspection in Fig. 5. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.3. Comparative studies using different refined modules

To understand whether the proposed gated feature labeling unit gain an advantage over others, we take GRRNet with only 2× upsampling operations in the decoder part as the baseline and added four different refined modules to the baseline for comparisons, including (1) baseline with the feature fusion unit (Fig. 2c) (referred to as Baseline + FF), (2) baseline with the gate unit (Fig. 2d) (referred to as Baseline + GU), (3) baseline with the GFL unit (Fig. 2e) but only data of two bands are transmitted (referred to as Baseline + GFL-2), and (4) the proposed GRRNet (referred to as Baseline + GFL).

3.4. Method implementation

Data augmentation techniques were applied to the training set images to avoid network overfitting and improve model efficiency. Each input image was cropped to create a sequence of 480 × 480 pixel patches with an overlap of 200 pixels. The patches were rotated every 90 degrees in the clockwise direction and were also mirrored in both the horizontal and vertical directions. As a result, the training dataset contains 15,606 patches in total. At the inference stage, we create patches with an overlap of 100 pixels from each test image without performing data augmentation. The final probability map was obtained by merging all the patch probability maps and the prediction values of the overlapping pixels were derived as the mean values of all the predictions.

The proposed network of GRRNet was implemented using Caffe (Jia et al., 2014) on an NVIDIA GTX Titan X GPU. The network was trained with stochastic gradient descent (SGD) using the initial learning rate of 0.01, the weight decay of 0.0005, the momentum of 0.9 and the bath size of 4. The total iteration number was set as 40,000 and the learning rate was reduced to one-tenth of the original every 8000 iterations. The encoder part parameters in GRRNet were initialized with the pre-trained ResNet-50 model. All the other parameters were initialized using the techniques introduced by (He et al., 2015). The NIR-R-G composite images and nDSMs were fed into GRRNet and other comparative networks. The ground reference data were used for supervised training.

Note that: (1) V-FuseNet was trained using a smaller patch size (128 × 128 pixels) than GRRNet as the GPU memory was limited to 12 GB, (2) compared to the literature, the input images of Res-U-Net in this study only include four bands (i.e., NIR-R-G and nDSM, the same as GRRNet used) other than seven bands, and no post-processing was performed for the classification results. All configurations for the other networks are the same as GRRNet, and all the model weights were initialized using the corresponding pre-trained models (e.g., VGG-16 and ResNet-101) for semantic segmentation.

3.5. Accuracy assessment

Two commonly used metrics, namely the overall accuracy (OA) and the mean intersection over union (mean IoU), were used to assess the performance of different methods. OA is the ratio of the correctly classified pixel numbers to the total pixel numbers in an individual image or the entire image dataset. IoU, also known as the Jaccard similarity coefficient, provides statistical accuracies that penalize false positives (FPs). For each category, the IoU score is the ratio of the correctly classified pixel numbers to the total number of ground reference pixels and the detected pixels in the corresponding category as follows:

$$IoU_k = \frac{TP_k}{TP_K + FP_k + FN_k} \quad (9)$$

where TP_k (true positive) denotes the number of pixels in the category k that present in both reference and prediction maps; FP_k (false positive) denotes the number of pixels in the category k that only present in the prediction map but not in the reference map; FN_k (false negative) denotes the number of pixels in the category k that only present in reference map but not detected in the prediction map. In our experiments, the mean IoU score of all categories is calculated for both the individual image and the entire image dataset, respectively.

The receiver operating characteristic (ROC) curve is plotted to compare the binary classification accuracy of different FCN models. The ROC curve illustrates the trade-off between the true positive rate (TPR) and the false positive rate (FPR) with varying probability thresholds. The larger area under the ROC curve (AUC) indicates better model performance.

4. Experiment results

4.1. Deep learning model comparisons

Fig. 4 displays the test images and the classification results using different deep models across five city scenes. Visually, V-FuseNet (Fig. 4e) and GRRNet (Fig. 4g) obtained better classification results than other models in the urban and suburban scenes. SegNet (Fig. 4b) and DeconvNet (Fig. 4d) performed well in Norfolk than in the suburban areas of Arlington and New Haven, where water pixels and pixels with higher elevations were easily misclassified as buildings. In the urban area of San Francisco, there are many FNs and FPs in the classification results of both SegNet and DeconvNet, implying that the use of unpooling layers in SegNet and DeconvNet for feature transmission could not refine the upsampled results in this study. Although the unpooling layers are used in the decoder part of V-FuseNet, the corresponding encoder features are first convolved and transmitted by a third "virtual" encoder that performs feature selection and could refine the classification results. CNN-FPL (Fig. 4c) generally performed better than SegNet and DeconvNet but still frequently misclassified water pixels as building pixels in Arlington and did not detect the building pixels in the central area of New York City. Res-U-Net (Fig. 4f) performed better in the suburban areas than in the urban areas and correctly classified the water pixels in both Arlington and Norfolk. The FNs and FPs produced in the New York City and San Francisco indicate that Res-U-Net does not perform well enough in the urban scenes.

Fig. 5 shows the close-ups (as marked in yellow rectangles in Fig. 4a) of the tested images and the classification results for detailed inspection. Most buildings were identified correctly and completely using both V-FuseNet and GRRNet, except that some FPs were generated in the classification result of V-FuseNet in New Haven (Fig. 5e). The results for building boundaries demonstrated that V-FuseNet and GRRNet performed well on boundary refinement. The other four models however failed to detect the buildings completely in New Haven and New York City scenes as many FPs were found in the areas covered by roadside trees or shadows (Fig. 5b, c, d, and f), indicating that these models did not fully utilize the features of the input data.

Table 2 summarizes the quantitative results obtained using different methods. GRRNet generated the best result with the overall accuracy of 96.20% and the mean IoU score of 88.05% among all methods that have the encoder-decoder network architectures. GRRNet outperforms all other models in the test images of Arlington, New Haven, Norfolk, and San Francisco. V-FuseNet achieved the best performance for the data of the New York City. CNN-FPL achieved an overall accuracy of 92.44% but the resulted mean IoU score was nearly 9% lower than that of GRRNet. Res-U-Net generated comparable results with CNN-FPL in suburban areas of Arlington, New Haven, and Norfolk but did not perform well enough in urban scenes like San Francisco, where Res-U-Net only obtained the mean IoU score of 50.44% and the overall accuracy of 68.09%. The results imply that the strategy that Res-U-Net directly concatenates the encoder features with the decoder features without feature selection is not stable for building objects refinement.

Fig. 6 shows the ROC curves of all deep learning models for different scenes. Consistent with the statistical results in Table 2, GRRNet and V-FuseNet are shown to perform reasonably well. CNN-FPL has a stable classification performance for different scenes and is superior to Res-U-Net, SegNet, and DeconvNet. Res-U-Net has a competitive performance with respect to CNN-FPL in the Arlington scene, but it has low accuracies and is unstable in other scenes.

4.2. Model comparisons using different refined modules

Fig. 7 exhibits the building extraction results of GRRNet (Baseline + GFL) and its variants. As shown in Fig. 7c, Baseline + FF did not perform well as compared with other models. Due to the categorical ambiguity of the low-level encoder features, simply transmitting low-level features into a decoder network does not improve the baseline results. Both the gate unit and the GFL unit could improve the classification ability as compared to the baseline method.

Fig. 8 exhibits close-up views of the results. Baseline + FF frequently misclassified water pixels and building pixels (Fig. 8c). The baseline model produced better classification results than Baseline + FF and the obtained results in New York city were competitive with the gate units applied models (Fig. 8d–f). The gate units applied models obtain similar classification results in Arlington, New York City, and Norfolk. However, in the New Haven suburban area, an obvious FP patch appeared in Baseline + GU result (Fig. 8d). Moreover, in the San Francisco urban scene, a football field is found misclassified as a building object. Both Baseline + GU and Baseline + GFL-2 cannot completely detect the white-roofed building, whereas Baseline + GFL extract this building successfully and can identify the areas covered by overpasses correctly (Fig. 8f).

Table 3 lists the quantitative results of GRRNet and its variants with different refined modules. The baseline model only achieves overall accuracy of 94.20% and mean IoU of 83.37%, but it still outperforms SegNet, DeconvNet, CNN-FPL, and Res-U-Net (Table 2), indicating that the basic network architecture of GRRNet is suitable for building extraction task. Baseline + FF achieves a better result in San Francisco but poor results in other city scenes. However, Baseline + GU significantly outperforms Baseline + FF, indicating that the feature selection operations are needed and join the upper encoder features to the transmission helps to improve the classification ability of the network. With the GFL unit, both Baseline + GFL-2 and Baseline + GFL further improve the baseline overall accuracy and mean IoU score by nearly 2% and 5%, respectively. The more mIoU scores increase, the better refinement in the edge of the classification results. Our proposed GFL units exhibit better performance than the original gate unit. Fig. 9 illustrates the gate units applied models (e.g. Baseline + GU, GRRNet) perform better in the task of building extraction.

4.3. The influence of data input strategies

Fig. 10 shows the statistical results for different data input strategies on GRRNet. The NIR-R-G and R-G-B composite images produced similar classification accuracies and the combinations of LiDAR-derived images and spectral images significantly improved the classification results as compared with using spectral images alone. When the nDSM image is included, the overall accuracies and the mean IoU scores of GRRNet could increase by approximately 2.5% and 6.0%, respectively. Using nDSM as the unique input can obtain a slightly lower accuracies than the "NIR-R-G-nDSM", which indicates that the relative elevation information of nDSM has positive effects on building extraction. On the flip side, the introduction of spectral information can further improve the performance of nDSM. Using DSM instead of nDSM could not

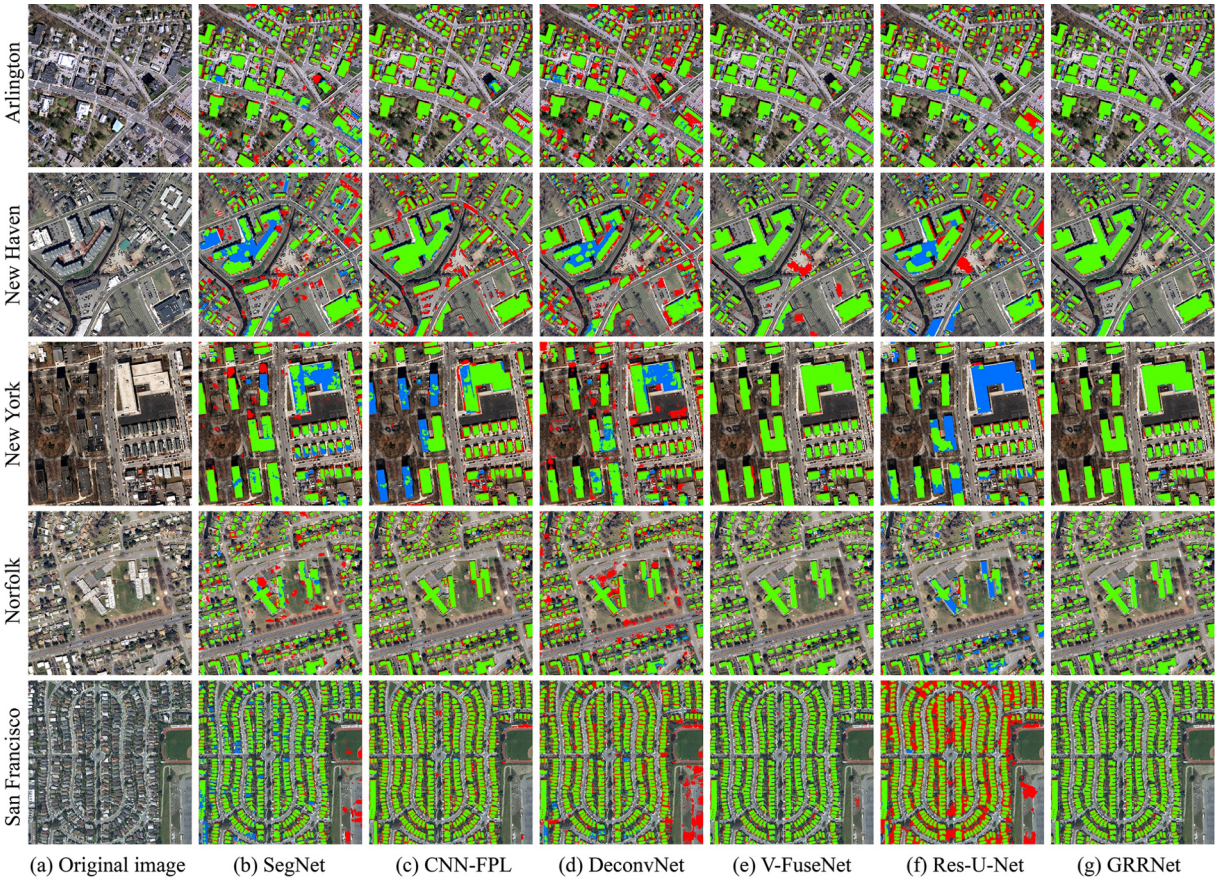

Fig. 5. Close-up views of images are shown for the original true-color composite images and the classification results using the state-of-the-art deep learning methods across five cities. The images are the subset from the yellow rectangles marked in Fig. 4a. The true positive (TP), false positive (FP) and false negative (FN) are marked in green, red, and blue, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

improve the results greatly because the nDSM data remove most "bare-earth noise" in the data preprocessing step. In our experiments, the NDVI data do not have obvious impacts on the results.

4.4. Model efficiency analysis

Table 4 lists the computing statistics of the deep learning models and the variants of GRRNet. CNN-FPL requires fewer computing resources and less inference time than others because CNN-FPL has the custom and shallow encoder-decoder architecture. DeconvNet needs much larger computing resources and longer training time than other models except V-FuseNet. Because the FuseNet-based architecture has to deal with two feature branches from the optical image and the height image separately, V-FuseNet has the longest inference time. GRRNet (i.e., Baseline + GFL) requires less inference time of 81.44 ms and smaller model size of 91.57 MB than other models because the ResNet-based encoder and the gated features transmitted at each upsampling stage. Overall, GRRNet shows to be more efficient than most models.

5. Discussions

The reasons for the excellent performance of our model are as follows. First, the encoder part of GRRNet is based on the modified version of ResNet-50, which can effectively solve the degradation problem and

Table 2
The statistical results obtained using the deep learning models. OA denotes the overall accuracy and mIoU denotes mean intersection over union. The bold values denote the best result and the underlined values denote the second best result achieved by models.

Model	Arlington		New Haven		New York City		Norfolk		San Francisco		Overall	
	mIoU	OA	mIoU	OA	mIoU	OA	mIoU	OA	mIoU	OA	mIoU	OA
SegNet	68.13	88.74	65.26	87.93	71.70	88.40	75.53	91.80	73.88	87.87	71.33	88.95
CNN-FPL	76.07	92.52	76.10	93.00	81.41	92.75	78.72	92.83	81.21	91.11	79.46	92.44
DeconvNet	66.50	87.53	64.23	86.89	67.57	85.32	70.98	89.26	71.22	85.09	68.80	86.82
V-FuseNet	80.92	94.60	87.02	96.88	**90.87**	**96.74**	89.43	97.03	86.04	93.80	87.24	95.81
Res-U-Net	75.51	92.33	73.92	93.05	69.58	88.68	73.77	92.70	50.44	68.09	67.53	86.97
GRRNet	**82.34**	**95.19**	**87.15**	**97.02**	90.58	96.71	**90.46**	**97.43**	**87.62**	**94.64**	**88.05**	**96.20**

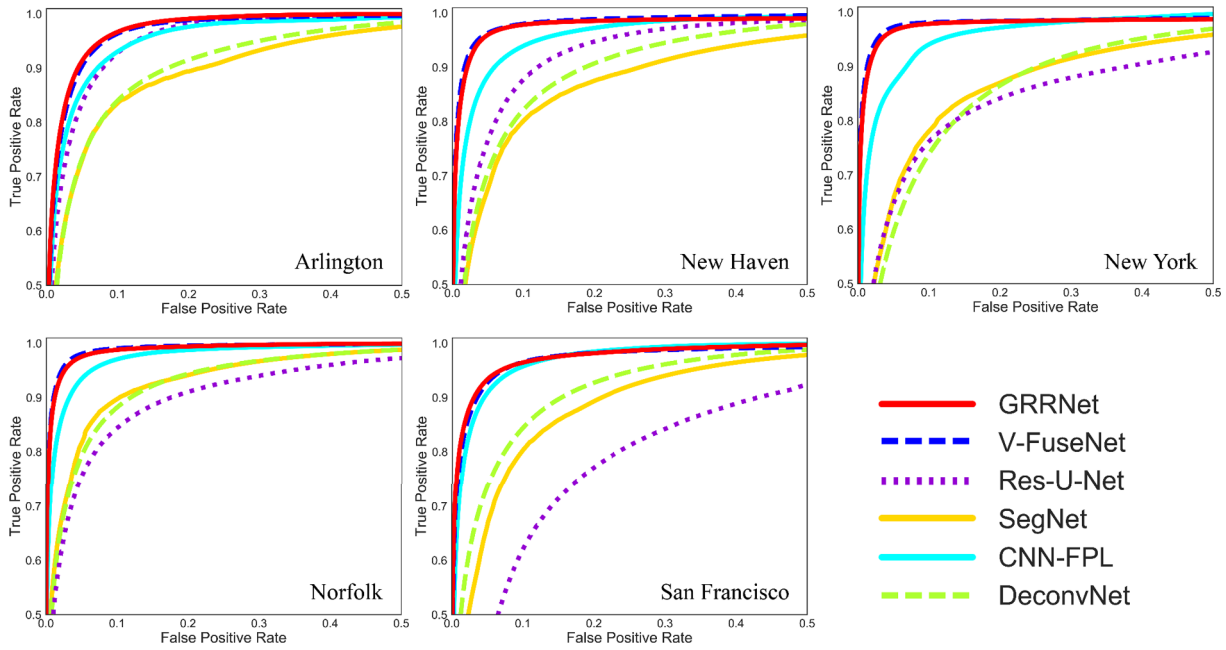

Fig. 6. The receiver operating characteristic (ROC) curves of all deep learning methods on the DataPlus test set images.

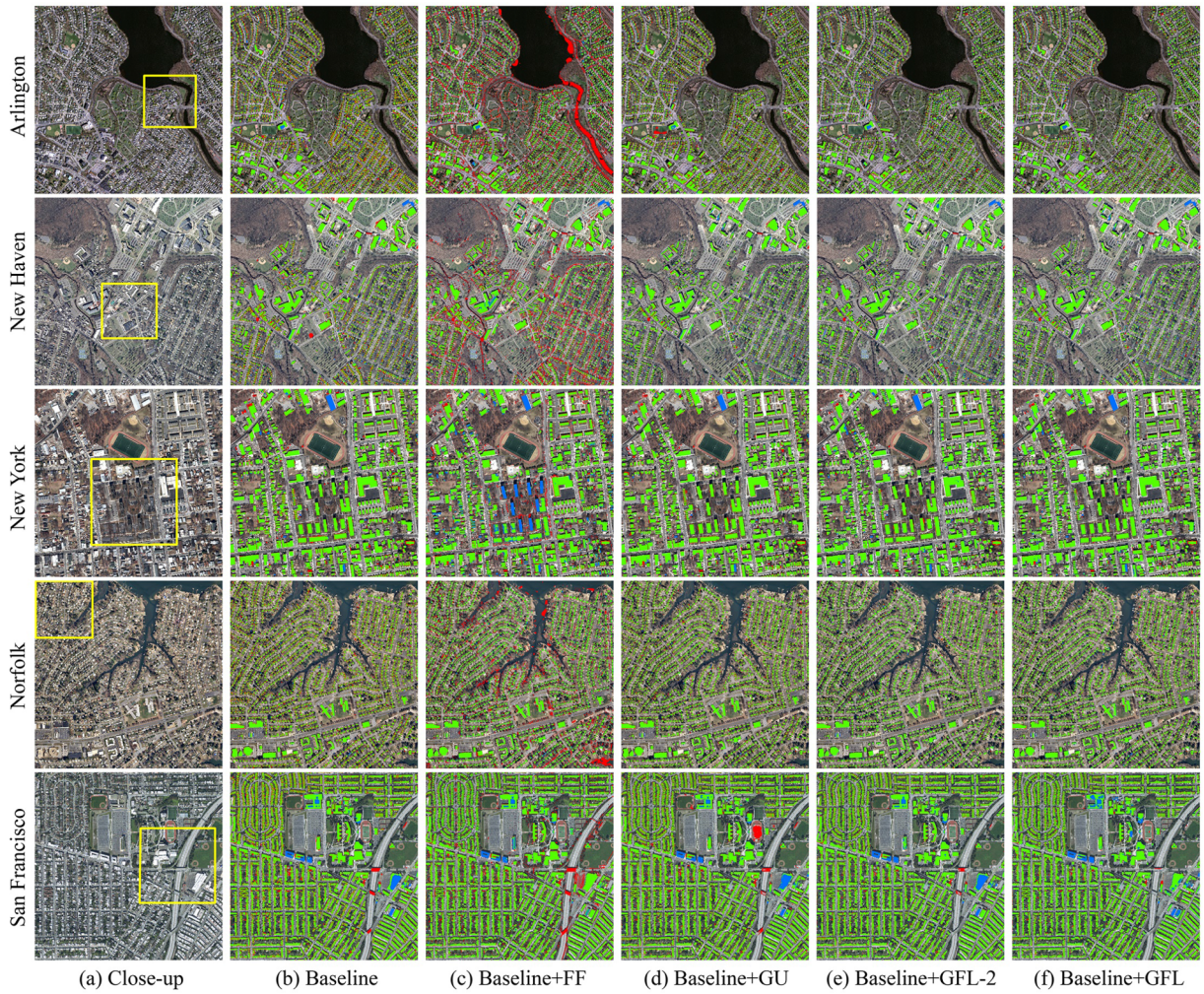

Fig. 7. Visual comparison with GRRNet (Baseline + GFL) and its variants on the full-resolution DataPlus test set images.

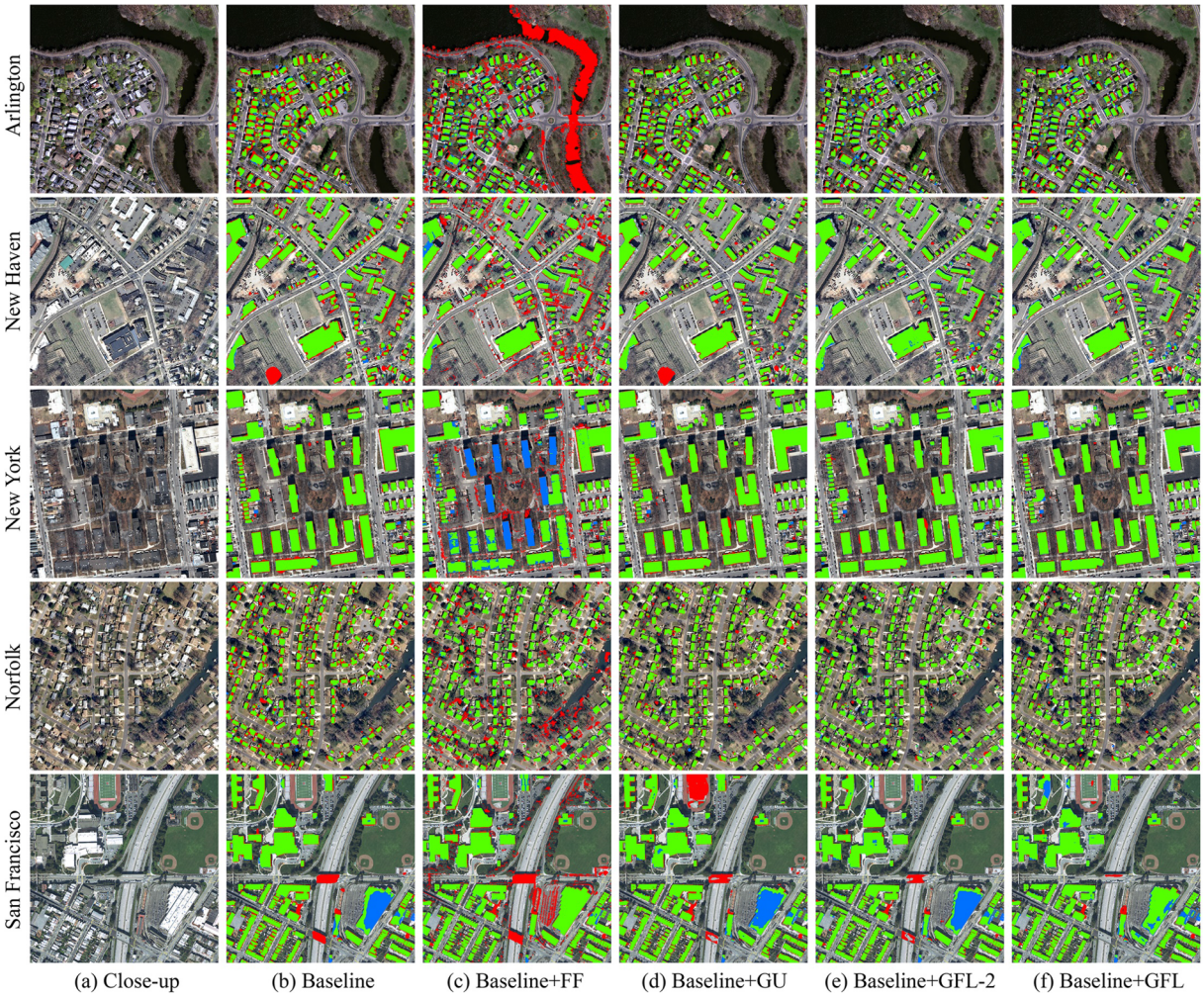

Fig. 8. Visual comparison with GRRNet (Baseline + GFL) and its variants on the close-ups of DataPlus test set images.

easily be adapted to different tasks (e.g., from image classification to semantic segmentation task). The analysis of different data fusion strategies on GRRNet (Fig. 10) also demonstrates that the encoder network has robust feature extraction capability even though only the spectral images are used, and it can still obtain consistent results when different types of input images are fed. Second, the gated feature labeling unit (GFL) solves the problem of redundant feature transmission in FCN models. That means GRRNet just passes the essential features with the aid from the upper encoder features that with larger receptive fields and less category-ambiguous, instead of transmitting all the encoder features to the decoder network. This way enhances the cross-level information exchange during the training process and helps to improve the learning ability of the network. Both qualitative and quantitative comparison results show that the GFL unit outperforms many commonly used modules, e.g., max unpooling and simple feature fusion unit, etc.

Scalability is another advantage of our proposed model. GRRNet can receive multiple input images and output more than two categories in the classification stage. Therefore, it can be adapted for multi-class semantic segmentation of remote sensing images, as long as the training samples are sufficient. Meanwhile, the model is end-to-end trainable and any post-classification processing is not needed.

To further confirm the availability of GRRNet, our model was also trained and validated on the ISPRS Vaihingen and Potsdam 2D semantic labeling datasets (http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html). Both the training and test datasets are

Table 3
The statistical results obtained using GRRNet (Baseline + GFL) and its variants. OA denotes the overall accuracy and mIoU denotes the mean intersection over union. The bold values denote the best result and the underlined values denote the second best result achieved by models.

Model	Arlington		New Haven		New York City		Norfolk		San Francisco		Overall	
	mIoU	OA	mIoU	OA	mIoU	OA	mIoU	OA	mIoU	OA	mIoU	OA
Baseline	77.68	93.20	82.63	95.47	88.73	95.87	82.43	94.52	82.63	91.98	83.37	94.20
Baseline + FF	67.47	88.23	72.05	91.14	81.03	92.83	79.35	93.34	83.46	92.41	77.42	91.59
Baseline + GU	80.74	94.42	85.50	96.43	89.97	96.37	88.41	96.68	84.54	92.99	86.21	95.38
Baseline + GFL-2	**82.43**	**95.24**	86.70	96.90	**90.59**	**96.71**	90.13	97.35	86.88	94.24	87.76	96.09
Baseline + GFL	82.34	95.19	**87.15**	**97.02**	90.58	**96.71**	**90.46**	**97.43**	**87.62**	**94.64**	**88.05**	**96.20**

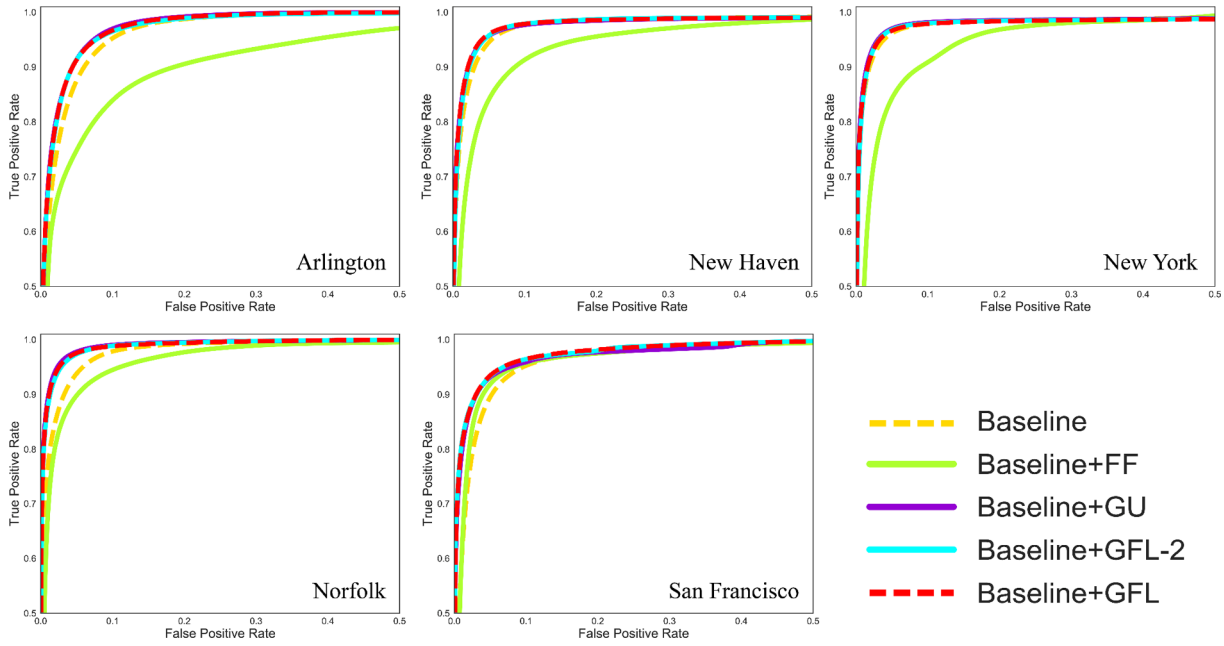

Fig. 9. The receiver operating characteristic (ROC) curves of GRRNet (Baseline + GFL) and its variants on the DataPlus test set images.

randomly selected from the released annotated images and each test dataset contains 6 images. The training process and hyper-parameters used are the same as those of DataPlus dataset. Figs. 11 and 12 show the classification results of GRRNet on Vaihingen dataset and Potsdam dataset, respectively. The overall accuracy of 98.09% and the mIoU score of 95.38% were achieved by GRRNet on Potsdam dataset, and the overall accuracy of 96.52% and the mIoU score of 91.74% were obtained on Vaihingen dataset. In particular, the classification results on Potsdam are superior to that reported by Xu et al. (2018). Overall, the performances of GRRNet on Potsdam dataset and Vaihingen dataset are better than that of DataPlus dataset, which is probably due to the higher image resolution and more accurate image registration processing of the ISPRS datasets.

The developed GRRNet could still be improved potentially. First, the registration between multi-source data is still a key factor affecting the accuracy of building extraction, and it also brings challenges for many existing studies (Yang and Chen, 2015). In the DataPlus dataset, we found that the ground truth annotations are more consistent with nDSM than with aerial images. This is mainly due to the geometric distortion of the images. We do not have accurate registration of images and point cloud data because it is still a challenging task and may cause more uncertainty in the experiments. Second, like many other methods,

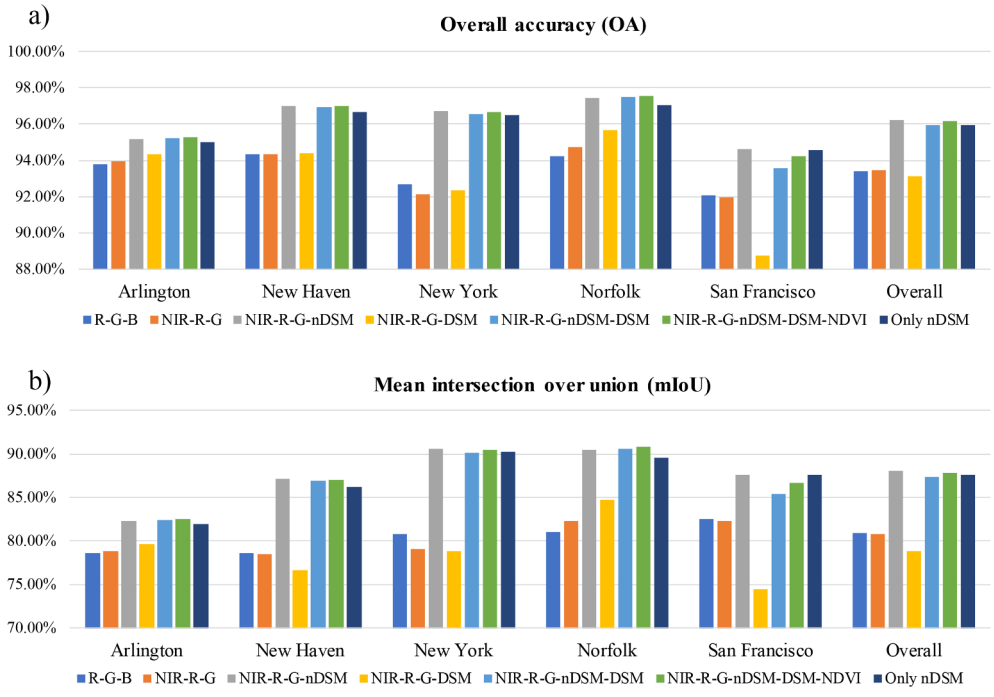

Fig. 10. Accuracy assessment of different data input strategies as implemented with GRRNet.

Table 4
Comparisons of network efficiency among the tested deep learning models and the variants of GRRNet (Baseline + GFL).

Model	Forward pass (ms)	Backward pass (ms)	Model size (MB)
SegNet	121.69	279.08	112.33
CNN-FPL	70.72	154.74	13.87
DeconvNet	187.68	307.23	960.57
V-FuseNet	227.22	528.25	225.09
Res-U-Net	117.38	237.53	388.77
Baseline	53.66	132.94	90.78
Baseline + FF	87.73	197.57	209.07
Baseline + GU	82.69	192.80	91.07
Baseline + GFL-2	86.94	193.21	91.06
Baseline + GFL	81.44	207.17	91.57

The Caffe time command was used to compute time requirement as averaged over 50 iterations with an image size of 480 × 480 pixels. The model size was obtained from the model file size.

LiDAR point clouds are first rasterized into 2D height images and then efficiently processed by GRRNet. However, this way may lead to loss of accurate 3D scene information. Therefore, it is promising to develop 3D CNN models for feature extraction and further fuse with image features learned from FCN models. Finally, in the experiments, we only focus on analyzing the performance of different FCN models. In other words, traditional methods, such as the thresholding-based (Hermosilla et al., 2011) and object-based (Khoshelham et al., 2010) methods, have not been compared, since the learning and inference procedures of FCN models are fully automated and quite different from traditional methods. In the future, it is needed to provide a more comprehensive comparison of these methods.

6. Conclusions

Deep CNNs have now become increasingly important in semantic segmentation of remote sensing images and have found to be efficient in the extraction of urban objects. This study proposed a novel end-to-end trainable gated residual refinement network (GRRNet) for building extraction using both high-resolution aerial images and airborne LiDAR data. The encoder part of GRRNet is based on a modified residual learning network and the decoder part uses the gated feature labeling unit to refine the upsampled classification results. The proposed GRRNet could learn multi-level image features and perform semantic pixel labeling of an entire image, thereby simplifying the extensive processes of both designing hand-crafted features and pre-segmentation of input images. The upsampling stage of FCN models is also designed to deal with the effect of gated feature transmission. The model is evaluated using a publicly available dataset that consists of images with different spatial resolutions, surface elevations, and building densities in urban and suburban scenes. Comparison results illustrated that GRRNet has competitive building extraction performance in comparison with other approaches. The model code is now publicly available at https://github.com/CHUANQIFENG/GRRNet for further studies.

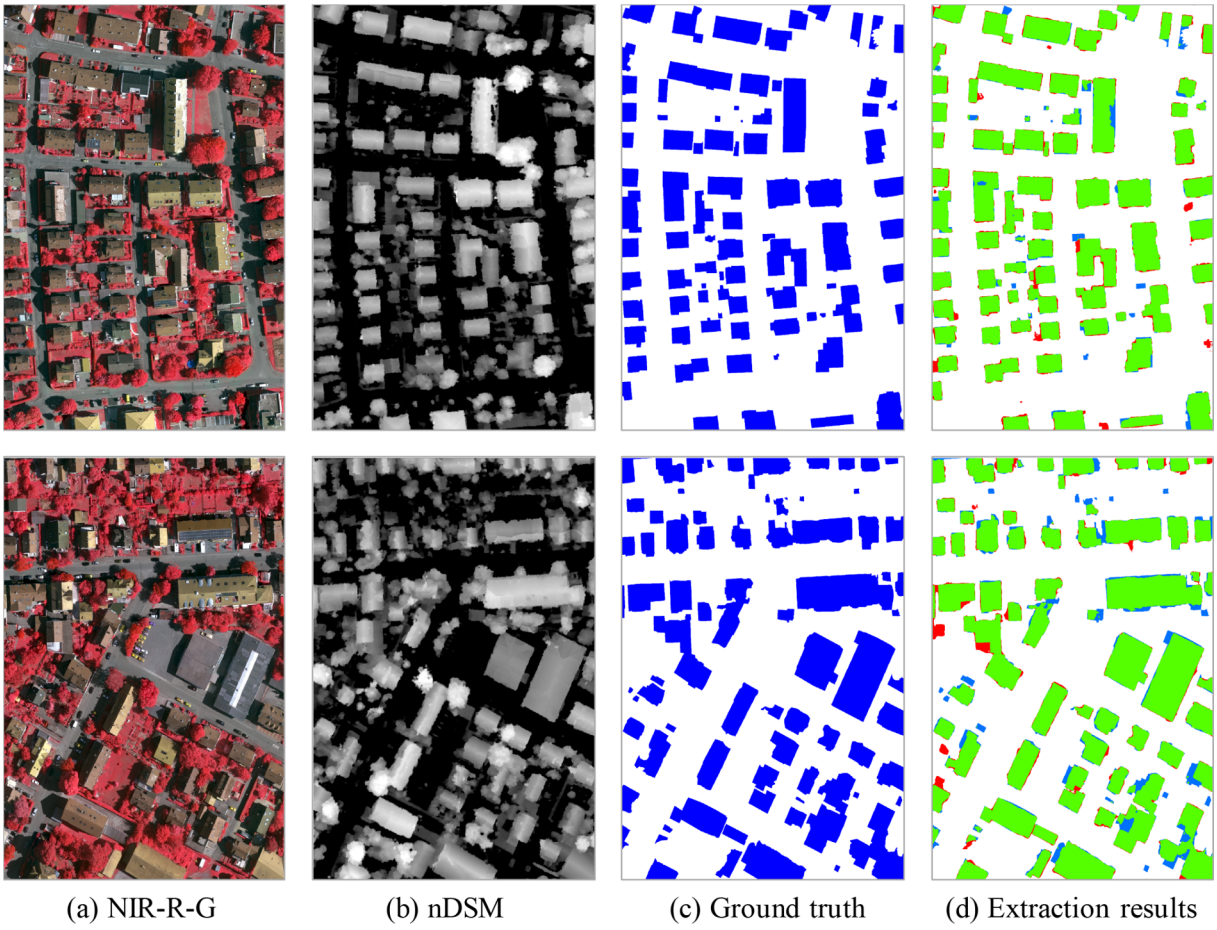

Fig. 11. Building extraction results of GRRNet on Vaihingen dataset. (a) NIR-R-G false-color composite images; (b) nDSMs; (c) ground truth images; (d) building extraction results (green: true positive, blue: false negative, red: false positive). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

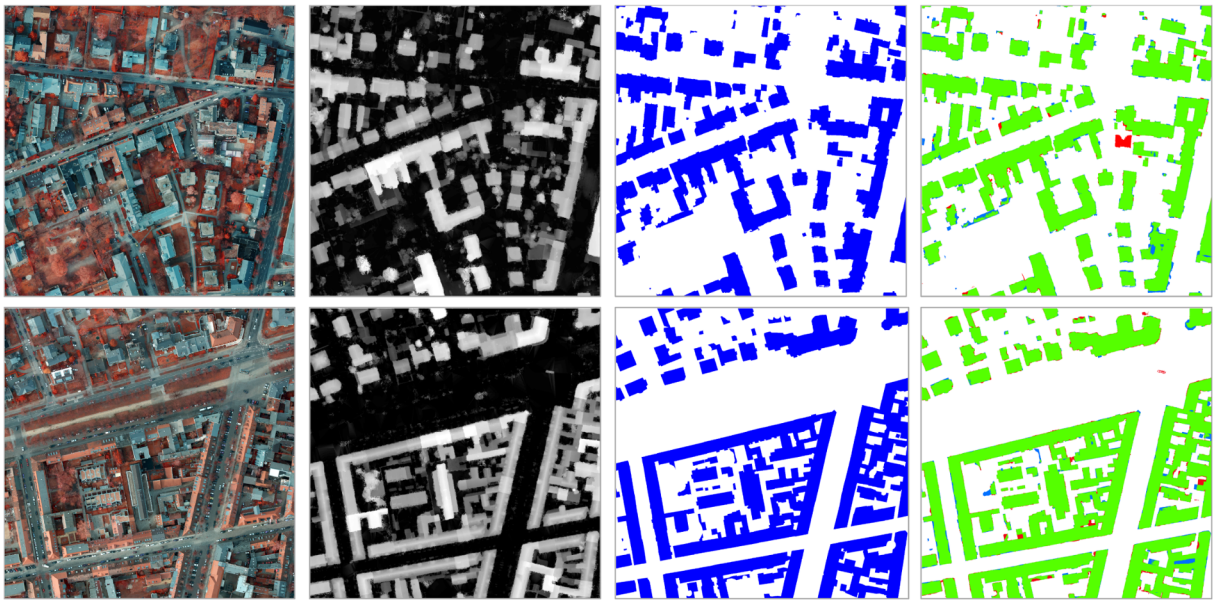

Fig. 12. Building extraction results of GRRNet on Potsdam dataset. (a) NIR-R-G false-color composite images; (b) nDSMs; (c) ground truth images; (d) building extraction results (green: true positive, blue: false negative, red: false positive). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Acknowledgments

We acknowledge the Data+ and Bass Connections programs in Duke University for providing the experiment data set. We thank the ISPRS for making the Vaihingen and Potsdam dataset available for evaluating the proposed model. This research is supported by the National Natural Science Foundation of China (grant nos. 41431178, 41801351, 41671453 and 41875122), the Natural Science Foundation of Guangdong Province, China (grant no. 2016A030311016), the National Administration of Surveying, Mapping and Geoinformation of China (grant no. GZIT2016-A5-147), the Research Institute of Henan Spatio-Temporal Big Data Industrial Technology (grant no. 2017DJA001), the Key Projects for Young Teachers at Sun Yat-sen University (grant no. 17lgzd02). We thank anonymous reviewers for their constructive comments.

References

Alshehhi, R., Marpu, P.R., Woon, W.L., Dalla Mura, M., 2017. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 130, 139–149.

Audebert, N., Le Saux, B., Lefèvre, S., 2017. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS J. Photogramm. Remote Sens.

Awrangjeb, M., Ravanbakhsh, M., Fraser, C.S., 2010. Automatic detection of residential buildings using LIDAR data and multispectral imagery. ISPRS J. Photogramm. Remote Sens. 65, 457–467.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2481–2495.

Bischke, B., Helber, P., Folz, J., Borth, D., Dengel, A., 2017. Multi-task learning for segmentation of building footprints with deep neural networks. arXiv preprint arXiv:1709.05932.

Cheng, G.L., Wang, Y., Xu, S.B., Wang, H.Z., Xiang, S.M., Pan, C.H., 2017. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. IEEE Trans. Geosci. Remote Sens. 55, 3322–3337.

Du, S.J., Zhang, Y.S., Zou, Z.R., Xu, S.H., He, X., Chen, S.Y., 2017. Automatic building extraction from LiDAR data fusion of point and grid-based features. ISPRS J. Photogramm. Remote Sens. 130, 294–307.

Eigen, D., Fergus, R., 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2650–2658.

Fu, G., Liu, C.J., Zhou, R., Sun, T., Zhang, Q.J., 2017. Classification for high resolution remote sensing imagery using a fully convolutional network. Remote Sens. 9, 498.

Ghanea, M., Moallem, P., Momeni, M., 2016. Building extraction from high-resolution satellite images in urban areas: recent methods and strategies against significant challenges. Int. J. Remote Sens. 37, 5234–5248.

Gilani, S.A.N., Awrangjeb, M., Lu, G.J., 2016. An automatic building extraction and regularisation technique using LiDAR point cloud data and orthoimage dagger. Remote Sens. 8, 258.

Haala, N., Brenner, C., 1999. Extraction of buildings and trees in urban environments. ISPRS J. Photogramm. Remote Sens. 54, 130–137.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034.

He, K.M., Zhang, X.Y., Ren, S.Q., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), pp. 770–778.

Hermosilla, T., Ruiz, L.A., Recio, J.A., Estornell, J., 2011. Evaluation of automatic building detection approaches combining high resolution images and LiDAR data. Remote Sens. 3, 1188–1210.

Huang, X., Zhang, L., 2012. Morphological building/shadow index for building extraction from high-resolution imagery over urban areas. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 5, 161–172.

Huang, X., Zhang, L.P., 2011. A multidirectional and multiscale morphological index for automatic building extraction from multispectral GeoEye-1 imagery. Photogramm. Eng. Remote Sens. 77, 721–732.

Islam, M.A., Rochan, M., Bruce, N.D.B., Wang, Y., 2017. Gated feedback refinement network for dense image labeling. In: 30th IEEE Conference on Computer Vision and Pattern Recognition (Cvpr 2017), pp. 4877–4885.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, pp. 675–678.

Kaiser, P., Wegner, J.D., Lucchi, A., Jaggi, M., Hofmann, T., Schindler, K., 2017. Learning aerial image segmentation from online maps. IEEE Trans. Geosci. Remote Sens. 55, 6054–6068.

Khoshelham, K., Nardinocchi, C., Frontoni, E., Mancini, A., Zingaretti, P., 2010. Performance evaluation of automated approaches to building detection in multi-source aerial data. ISPRS J. Photogramm. Remote Sens. 65, 123–133.

Lee, D.H., Lee, K.M., Lee, S.U., 2008. Fusion of lidar and imagery for reliable building extraction. Photogramm. Eng. Remote Sens. 74, 215–225.

Lin, G.S., Milan, A., Shen, C.H., Reid, I., 2017. RefineNet: multi-path refinement networks for high-resolution semantic segmentation. In: 30th IEEE Conference on Computer Vision and Pattern Recognition (Cvpr 2017), pp. 5168–5177.

Liu, Y., Fan, B., Wang, L., Bai, J., Xiang, S., Pan, C., 2017a. Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. ISPRS J. Photogramm. Remote Sens.

Liu, Y., Yao, J., Li, L., Lu, X., Han, J., 2017b. Learning to Refine Object Contours with a Top-Down Fully Convolutional Encoder-Decoder Network. arXiv preprint arXiv:1705.04456.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

Marmanis, D., Schindler, K., Wegner, J.D., Galliani, S., Datcu, M., Stilla, U., 2018. Classification with an edge: Improving semantic image segmentation with boundary

detection. ISPRS J. Photogramm. Remote Sens. 135, 158–172.
Meng, X.L., Currit, N., Wang, L., Yang, X.J., 2012. Detect residential buildings from lidar and aerial photographs through object-oriented land-use classification. Photogramm. Eng. Remote Sens. 78, 35–44.
Mnih, V., Hinton, G.E., 2010. Learning to detect roads in high-resolution aerial images. In: European Conference on Computer Vision. Springer, pp. 210–223.
Mongus, D., Lukac, N., Zalik, B., 2014. Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. ISPRS J. Photogramm. Remote Sens. 93, 145–156.
Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. ISPRS J. Photogramm. Remote Sens. 87, 152–165.
Noh, H., Hong, S., Han, B., 2015. Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528.
Noronha, S., Nevatia, R., 2001. Detection and modeling of buildings from multiple aerial images. IEEE Trans. Pattern Anal. Mach. Intell. 23, 501–518.
Paisitkriangkrai, S., Sherrah, J., Janney, P., van den Hengel, A., 2016. Semantic labeling of aerial and satellite imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 9, 2868–2881.
Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J.D., Breitkopf, U., Jung, J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. ISPRS J. Photogramm. Remote Sens. 93, 256–271.
Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2005. Using the Dempster-Shafer method for the fusion of LIDAR data and multi-spectral images for building detection. Inform. Fusion 6, 283–300.
Saito, S., Aoki, Y., 2015. Building and road detection from large aerial imagery. In: Image Processing: Machine Vision Applications VIII. International Society for Optics and Photonics, p. 94050K.
Sampath, A., Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. IEEE Trans. Geosci. Remote Sens. 48, 1554–1567.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Sun, Y., Zhang, X., Xin, Q., Huang, J., 2018. Developing a multi-filter convolutional neural network for semantic segmentation using high-resolution aerial imagery and LiDAR data. ISPRS J. Photogramm. Remote Sens.
Volpi, M., Tuia, D., 2017. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. IEEE Trans. Geosci. Remote Sens. 55, 881–893.
Wang, H.Z., Wang, Y., Zhang, Q., Xiang, S.M., Pan, C.H., 2017. Gated convolutional neural network for semantic segmentation in high-resolution images. Remote Sens. 9, 446.
Wang, R.S., Hu, Y., Wu, H.Y., Wang, J., 2016. Automatic extraction of building boundaries using aerial LiDAR data. J. Appl. Remote Sens. 10.
Wu, G., Shao, X., Guo, Z., Chen, Q., Yuan, W., Shi, X., Xu, Y., Shibasaki, R., 2018. Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks. Remote Sens. 10, 407.
Xie, S., Tu, Z., 2017. Holistically-nested edge detection. Int. J. Comput. Vision 125, 3–18.
Xu, Y.Y., Wu, L., Xie, Z., Chen, Z.L., 2018. Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. Remote Sens. 10, 144.
Yan, W.Y., Shaker, A., El-Ashmawy, N., 2015. Urban land cover classification using airborne LiDAR data: a review. Remote Sens. Environ. 158, 295–310.
Yang, B., Chen, C., 2015. Automatic registration of UAV-borne sequent images and LiDAR data. ISPRS J. Photogramm. Remote Sens. 101, 262–274.
Yuan, J., 2017. Learning building extraction in aerial scenes with convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell.
Zarea, A., Mohammadzadeh, A., 2016. A novel building and tree detection method from LiDAR data and aerial images. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 9, 1864–1875.
Zhang, J., 2010. Multi-source remote sensing data fusion: status and trends. Int. J. Image Data Fusion 1, 5–24.
Zhang, L., Zhang, L., Du, B., 2016. Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geosci. Remote Sens. Mag. 4, 22–40.
Zhao, W.Z., Du, S.H., Wang, Q., Emery, W.J., 2017. Contextually guided very-high-resolution imagery classification with semantic segments. ISPRS J. Photogramm. Remote Sens. 132, 48–60.