Efficient Building Detection for Large-scale Urban Remote Sensing

Feiyu Qin, Tongcun Zuo, Xuejin Chen

***Abstract*—Extracting buildings from remote sensing images plays an important role in urban applications (e.g., urban planning and digital city). However, this task is quite difficult due to the great diversity of buildings and similarities between buildings and other categories. Recent approaches have attempted to harness the capabilities of deep learning techniques for building extraction. In this paper, we propose a robust method which extracts buildings from large-scale remote sensing images efficiently via deep learning. And further we extend our method to the 3D building reconstruction to accelerate the overall process. Our study demonstrate that learning low-level appearance information and high-level semantic information are equally important in building extraction task since buildings possess various scales and aspect ratios in the scene. Hence, to make full use of the information extracted from each layer, we propose a simple but effective hierarchical fusion operation which fuses the feature maps between channels stage by stage. In this paper, a novel network named hierarchical fused fully convolution network(HF-FCN) is also described which fuses the information through combining the fusion operation the general networks. The experiments on several available remote sensing image datasets show that our method achieves state-of-the-art performance. (xuejin:Problem: Over-emphasized the buidling detection part, without clearly describe the scope of this paper and the relationship between detection and reconstruction.)**

***Index Terms*—building extraction, hierarchical fusion operation, Hierarchically Fused Fully Convolutional Network (HF-FCN), 3D city modelling**

I. Introduction

(xuejin:Is your goal reconstruction or building extraction? The introduction should explain the overall goal. What are the challenges for building modeling from remote sensing images? What kind of work has been done in the literature? Why do we focus on building detection? What are our contributions?)

BUILDING extraction, which aims to extract rooftop[1] in a large-scale remote sensing image, remains one of the main challenges have been studied for decades in the field of remote sensing. Moreover, automatic extraction of building rooftops from aerial and satellite imagery is an important step in many applications, such as: urban planing, automated map making, 3D city modeling, updating geographical dataset and military reconnaissance. It is particularly difficult to extract rooftop from remote sensing images at the pixel level because of the following three reasons:

- Different density of buildings in the scene. A rural scene has low density but an urban scene has high density, with a suburban scene in between (medium density).
- Diverse shapes of the buildings. Buildings come in many shapes from simple rectangular blocks with flat roof to complex shapes with intricate roof shape.
- The quality of remote sensing images. Images vary in terms of contrast, resolution, and image principle [1].

[1]Because the data sets used in our article are high altitude remote sensing images which could be considered as the top views of the ground. Therefore, we do not distinguish the concepts of buildings and rooftops in the subsequent description.

(a) (b) (c) (d)

Fig. 1. Examples of remote sensing patches with different kinds of challenges. (a) Shadow occlusion in green frame. (b) Low inter-class differences. (c) High intra class variance. (d) A lot of tiny buildings close to each other.

Several remote sensing patches are shown in Fig. 1 (xuejin:do not use the figure no directly. using ref..) , which illustrate the challenges of building extraction task.

In the past decades, many researchers have made some experimental investigations to extract buildings automatically. At first, many simple knowledge-based methods were put forward by [1], [2], [3], [4], [5]. Their basic ideas are derived from prior knowledge of buildings that buildings are closed polygons made up of some straight lines. Some others are energy based methods which involves the variational level set evolution, improved snake model and graph cut [6], [7], [8]. Early methods are excessively dependant on prior and initialization. It's hard to apply to today's application.

In recent years, with the development of machine learning, many techniques via machine learning are gradually introduced into the remote sensing domain. At first, some shallow networks were proposed for multiple object extraction [9], [10], [11], [12]. By using patches to classify, the methods are inefficient and inaccurate for the pixel-wise segmentation task. Further, with increasing computer power, deep learning developed rapidly and brought into the field of remote sensing. Some researchers tried Convolutional Neural Networks (CNNs) for aerial images classification and semantic pixel labelling [13], [14], [15], [16], [17]. Unfortunately, they didn't consider the problem that multi-scale information extracted by the network are critical to the final prediction. Owing to ignoring the hierarchical information, they could not deal with the case that scenes of close-packed buildings well.

To fix the above problems, a relatively simple, but very effective manner is proposed in this work. And it could be combined into a general CNN architecture easily for building extraction. We take full advantages of the low-level appearance

information as well as high-level semantic information by the novel fusion operation in a way of stage by stage. Inspired by FCN [18], a noval hierarchically fused FCN is proposed, named HF-FCN whose output is in the same resolution of input for pixel-wise classification. Differ from the traditional FCN, a set of hierarchical fusion operations are used to fuse the intra layer information and inter layer information respectively which improve the performance of FCN greatly. And numerous experiments conducted on three remote sensing image datasets all obtain fairly good results. We further extend our work to the field of 3D modeling as the part of building detection. It is easily intergrated into the pipeline of building reconstruction. Our technical contributions are:

1) A effective hierarchical fusion operation which is specially designed for multi-scale building extraction is proposed. Combining with a general FCN, a novel network is presented, named HF-FCN that can deal with the problems of different sizes, diverse appearance and mutual occlusion of buildings and etc.
2) HF-FCN is an end-to-end network that does not need any post processing. And the approach is significantly computationally efficient than existing techniques. Besides, the overall accuracy based on HF-FCN exceeds the state-of-art algorithms.

The remainder of this paper is organized as follows. Sec. II sums up the related works in the past. In Sec. III, we introduce the fusion operation and architecture of HF-FCN. The training loss are also presented. And in Sec. IV, a brief description of the dataset used for our task is provided. HF-FCN training strategies, details and its evaluation metrics are also described. In Sec. V, we display and analysis the experimental results. Extension in 3D building modeling are presented in Sec. VI. Finally, the conclusion is discussed in Sec. VII.

II. Related Work

Building extraction is one of the most fundamental problems in remote sensing domain, which has been studied for nearly 30 years. As time goes by, many research achievements have sprung up. We roughly divide these methods into three groups: one is based on the shape prior, another is based on the energy function and machine learning third. Meanwhile, the methods of machine learning could be divide into two parts: one is shallow networks and the other is deep learning methods. Here we briefly review some representative methods that have evolved in the past decades in the different groups respectively. Moreover, some related work which are popular in compute vision similar to our task are also introduced. (xuejin:More related work on city modeling/urban modeling. Maybe facade modeling.)

Shape Prior based Methods During early days, methods are mainly based on the hypothesis of prior knowledge. Huertas and Nevatia [1] assumed that buildings are rectangular or composed of rectangular components. Based on it, the approach detected lines and corners, traced object boundaries and used shadows to verify. Later, a system [2] for building detection and modelling was proposed with the assumption that the roofs were flat or symmetrical and walls were vertical. Using known ground height and detected rooftop, the reconstructed models could be soon obtained. Further, Noronha and Nosrati [3] transformed the line and intersection points of the image into a graph presentation, and turned the problem of polygon finding into the one that finding loops in the graph. However, it was still estimated on assumption that the buildings are polygonal. Izadi and Saeedi [4] presented a complete system for building detection and modelling. In the stage of building detection, a tree consisting of intersection points of lines was created and refined based on the found hypotheses. The sun azimuth and elevation angles were used to estimate the height with existing shadows afterwards. In recent years, very high resolution (VHR) optical satellite imagery could be obtained easily. Wang et al [5] proposed an efficient method for automatic rectangular building extraction from VHR remote sensing images by detecting line segments and grouping lines based on path integrity and closed contour search. It depends on the clear remote sensing images and accurate line extraction . (xuejin:If there are two authors, say A and B proposed.... if more than two authors, say A et al. ..)

The aforementioned shape-based methods have a good performance in rural scenes with low density of buildings. Neverthless, there are several limitations of these methods. First, the shape-based methods inherently limited to handle buildings of arbitrary shapes. Second, they may failed to deal with complicated cases, for instance, buildings are close to each other, which thereby is hard to adapt to today's applications. Third, the algorithms using shadows to verify corners and estimate height are greatly limited to obvious shadows and sparse building environment.

Energy Function based Methods Later, several energy-based methods in image segmentation domain have been applied in automatic rooftop extraction. Cote and Saeedi [6] employed corner detection as an initial estimate of the roof, and then refined by the level set evolution. Peng et al [7] proposed an approach that segments remote sensing images into high objects, ground and shadow regions, with further refined by an improved snake model. Later, the urban-region-detection problems were casted as one of multiple subgraph matching by Sirmacek and Unsalan [8]. They considered each SIFT keypoint as a vertex, neighborhood between vertexs as edges of the graph and formulated the problem of building detection in terms of graph cut.

Experiment results in the above works reveal that energy function based methods limit to a good initialization. And it is generally known that energy-based methods are greatly influenced by the nature of the images. That is to say, the above mentioned methods do not apply to the building extraction task for high altitude remote sensing images of dense buildings with severe shadow occlusion.

Shallow Networks Over the past decade, CNNs have achieved great success in the field of computer vision. There are significant amount of efforts on semantic pixel-level classification for extraction buildings in remote sensing. A shallow patch-based network proposed by Mnih [9] has five layers with a 64 by 64 aerial patch as input. And the output of the network was processed by conditional random fields (CRFs) to constrain the segmentation continuity. Afterwards, Satio

et al. [10] putted forward two major strategies to improve the performance of [9]. One was a channel-wise inhibited softmax (CIS) for getting a multi-label prediction result, the other was model averaging (MA) with spatial displacement for enhancing the prediction result. Later, Alshehhi et al. [11] adjusted the architecture of [9] through changing the kernel size of convolutional layers and replacing the fully connection layer of the last layer with the average pooling layer. Alternative post-processing strategies such as CRFs and multi-scales were used to improve the final prediction results. At the same time, some other methods took advantage of the feature extraction capability of CNNs to generate feature descriptions of patches for further segmentation; for instance, Paisitkriangkrai et al. [13] made use of both the CNN and hand-craft extracted features and combined them together to generate predicted labels of each patch. The CRFs is also used as post-processing to get a sound result. Unlike [13], [17] put forward a multi-label pixelwise classification method using the feature vector extracted by a CNN to train a Support Vector Machine (SVM) for classification. Other appearance information, such as edges [12] are also harnessed to guide the shallow network to extract buildings.

There are several disadvantages of above mentioned methods. (a) Shallow networks cannot adequately express the features of images. (b) The methods using Shallow networks always cast the problem of building segmentation as a patch classification problem. It has a great impact on the accuracy of building extraction. (c) Most of them are processed by at least one kind of post-processing, which is time-consuming.

Deep Learning More recently, Long et al. [18] illustrated that Fully Convolutional Networks (FCN) could better handle the problem of multi-label pixel-wise classification. By up-sampling, final predicted result could be the same resolution of the input. Liu et al. [14] did a further research on the formulation proposed by Paisitkriangkrai [13] but used FCN as the branch of CNN and applied a higher-order CRFs as post-processing. Unlike traditional CRFs, the label consistency for the pixels within the same segment were enforced by higher-order CRFs. In order to reduce the information loss during pooling stage, SegNet [19] delivered pooling indices computed in the max-pooling to the decoder. It eliminated the need of learning during the up-sample stage while preserving segmentation performance. Audebert et al. [15] using SegNet architecture for semantic labeling of remote sensing and got better prediction results compared to the traditional methods. Later, Kampffmeyer at al. [16] proposed a novel idea that using CNN with missing data for urban land cover classification. The idea is came from a modality hallucination architecture proposed by Hoffman et al. [20].

Above-mentioned deep learning models have exceeded the traditional methods significantly, but all of them lost important hierarchical features encoded in the CNNs. And there is no way to recover them. That is to say, the features from the last layer of CNNs upsampled for building segmentation is not enough for pixel-wise classification due to the lost low-level information.

Computer Vision In the filed of computer vision, the FCNs [18] were introduced as a powerful method for semantic segmentation and already achieved great success. But, along with the deepening of network, the feature maps with lower resolution which causes the segmentation accuracy decline. In order to weaken the influence caused by pooling, Chen et al. [21] proposed a atrous convolution which enlarged the receptive field and reduced the number of pooling layers at the same time. Vemulapalli et al. [22] later extended the Deeplab [21] with a pairwise network and proposed a Gaussian Conditional Random Field Network for more continuous segmentation results. Afterwards, with the advent of the powerful networks such as ResNet [23], GoogLeNet [24] and their variants [25] [26] [27], a large amount of literature made use of these networks as their main structure for semantic segmentation. Zhao et al. [12] recently developed a pyramid pooling module following the ResNet [23] to get multi-scale feature maps and connected these feature maps with those which before pyramid pooling to create the final prediction. Zuo et al. [28] described a hierarchically fused fully convolutional network, which combined the feature maps from each group of VGG16 Net to generate the final prediction. In this paper, we extend the work of [28] and proposed a simple but effective fusion operation that could be easily combined to the general network and explore the effect of different layers of features on the final result. The details of our idea will be described in the next section.

The most relevant work of our work are U-Net [29] and FPN [30]. Differ from the U-Net which simple concatenates the feature maps from encoder to decoder, we apply a fusion operation firstly to fuse the feature maps created in the same convolution layers in the path of encoder to get more richer features. In addition, the resolution of encoder-decoder bottleneck is about $1/17$ of input resolution. It is too small for building segmentation task. FPN leveraged the encoder part as a feature pyramid, with predictions made independently at all levels. During the top-down path of FPN, it only exploit the feature maps come from the each stage's last residual block, but we take advantage of the all feature maps from the each stage. Moreover, we upsample the feature maps from each stage to the same resolution of the input and apply a hierarchical fusion operation to fuse the upsampled feature maps to a final prediction instead of directly predicting from each stage. That is to say, we only need one prediction rather than multiple predictions.

(xuejin:Need a paragraph to discuss recent semantic segmentation networks.)

(xuejin:Also cite our accv paper and describe the relationship/difference of this journal paper with it.)

III. Hierarchically Fused Fully Convolutional Network

In this section, we introduce a novel operation for feature fusion, named hierarchical fusion operation and apply it to the common networks, VGG16 Net and ResNet. The overview diagram in Fig. 2 shows where the fusion operations take effect and how they work. Different from other networks for semantic segmentation, we apply the fusion operation twice to integrate information gradually. Our network consists of

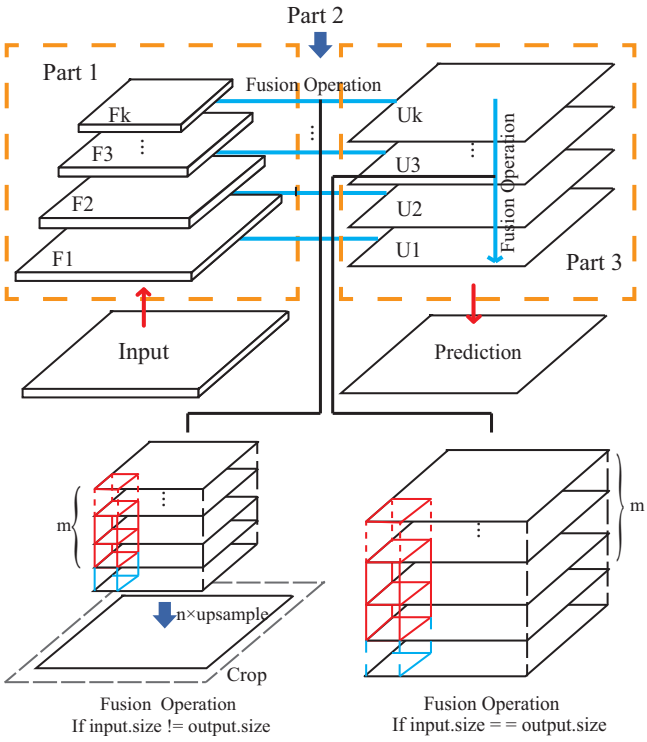


Fig. 2. The first line shows the overview of our network. The second row shows the details of two kinds of fusion operation. One is a case where the input is not equal to the output and the other is the input equal to the output. Fk means the feature maps come from the kth layer. m for number of feature maps. n said the n times of up sampling.

three parts. Part 1 is a bottom-up pathway which plays a role of features extractors at different levels. In theory, arbitrary feature extraction network is applicable to the Part 1. The second part is a process of feature fusion in the first stage, which fuses the feature maps generated from Part 1. Besides, Part 3 is a second stage of feature fusion. In Part 3, we take full advantage of the information extracted from the Part 2 by learning the connection weights between upsampled feature maps.

(xuejin:Put overview here. Explain the main components of our methods.)

A. Network Architecture

Part 1 The Part 1 is a bottom-up pathway, which generates the hierarchical feature maps from the network. With the increase of the field of perception, the extracted semantic information is gradually from the lower level to the high level. At the same time, the extracted information of image is from local to global. Each group of feature maps come from the same convolution(conv) layer contribute to the $\{F_k\}$ in Fig. 2, where $k : \Omega \to \{1, \ldots, K\}$. And K is the number of groups; for instance, K is 13 for VGG16 Net. Specifically, for ResNets, we consider a ResBlock as a feature extractor that K is 15.

Part 2 The Part 2 which fuses the feature maps of each group extracted from Part 1 via a set of hierarchical fusion operations. Due to the feature maps learned from same group including similar types of information, we fuse them into

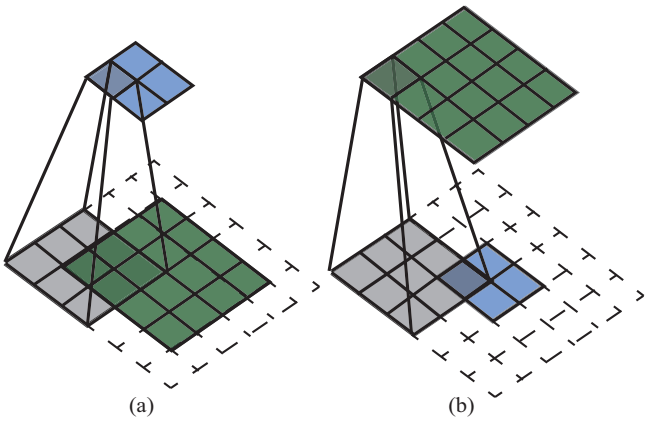


Fig. 3. A diagram of convolution and transposed convolution. The traditional convolution and transposed convolution are shown in the left and right column, respectively. The input of convolution is 4×4, the output is 2×2 and the kernel is 3×3. And the transposed convolution is the opposite.

one feature map, which contains the richer features. The hierarchical fusion operation consists of three steps: A 1×1 conv layer first, deconvolutional (transposed convolution) layer second and crop operation third. The Part 2 in Fig. 2 can be written as:

$$U_k = Crop(UpSample_n(Conv(\{F_k\}))) \tag{1}$$

where $\{F_k\}$ is extracted feature maps from Part 1, $k : \Omega \to \{1, \ldots, K\}$.

The first step is a 1×1 conv operation among the channels of feature maps in the same group. $Y = Conv(\{X\})$ is defined as:

$$Y(i,j) = \sum_{m=1}^{M} w_m X_m(i,j) \tag{2}$$

where M is the number of the feature maps in $\{X\}$. The w_m is the weight of conv kernels, $m : \Omega \to \{1, \ldots, M\}$.

If the input and output were to be unrolled into vectors form left to right, top to bottom, the convolution operation can be expressed as:

$$y = Wx \tag{3}$$

where x, y are flattened vector of input and output. The W is a sparse matrix whose non-zero elements are weights of kernels. A diagram of conv operation is shown in Fig. 3(a). For Fig. 3(a), x is a 16-dimensional vector (the input X is a 4×4 patch), y is a 4-dimensional vector and W is a matrix of 4×16.

The output of $Conv\{F_k\}$ are upsmapled by a transposed convolution which is written as $Y = UpSample_n(X)$, where n means n times up sampling. The n in our network is determined by the size of input X and output Y which is equal to $max\{\lceil Y.weight/X.weight \rceil, \lceil Y.height/X.height \rceil\}$. Contrary to conv operation, the transposed conv is a process of transforming the features of low dimension to high dimension. It can be written in a inverse form of conv operation:

$$y = W_1^T x \tag{4}$$

where x, y are flattened vector of input and output. The W_1^T is a sparse matrix whose non-zeros elements are weights of deconvolutional kernels. Fig. 3(b) shows a diagram of transposed conv. For Fig. 3(b), x is 4-dimensional vector (the input X of transpose conv is a 2×2 patch), y is a 16-dimensional vector and W_1^T is a matrix of 4×16.

The W_1^T of transposed conv layers of different groups are learned separately. It means that the transposed conv layers recovery semantic information from hierarchical feature maps. The $Crop(\{X\})$ operation in Part 2 is just a center alignment crop which cuts the superfluous boundary of upsampled feature maps. The output $\{U_k\}$, $k : \Omega \to \{1, \ldots, K\}$ of Part 2 has the same resolution with input X.

Part 3 Part 3 is the second fusion stage which aims to fuse the cropped feature maps from Part 2. In this part, the fusion operation plays a role of feature weighting. Using a 1×1 conv layer, a set of parameters w_k, $k : \Omega \to \{1, \ldots, K\}$ are learned to combine the hierarchical feature maps. The expression of the Part 3 is as follows:

$$Y = Conv(\{U_k\}) \tag{5}$$

where Y is output feature map. $\{U_k\}$ is input of Part 3 and output of Part 2.

In order to get a probability map of input X, the output Y of Part 3 is normalized by a sigmoid function:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

After normalization, A probability map P of rooftop is generated, each point in P means the probability that pixel belongs to a rooftop. If the pixel $x(i, j)$ belongs to a roftop, the output $P(i, j) \approx 1$.

B. Network Training

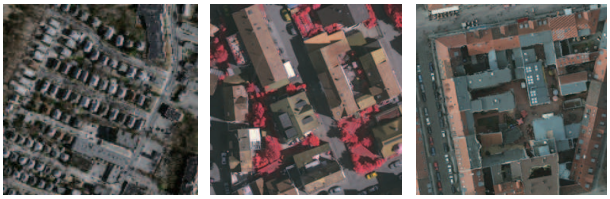
The ground truth G in our dataset is labeled by 0 or 1 to indicate whether a pixel belongs to a roof or not. (xuejin:only roof? or part of the building including facades?) When a remote sensing image X is inputted into the network, the output is a prediction probability map $P(X; W)$ of roof, where W denotes all the parameters that learned by HF-FCN including first part, Part 2 and 3. We use the sigmoid cross-entropy loss function to penalize each position on the prediction map formulated as:

$$L(W) = -\frac{1}{|I|} \sum_{i=1}^{|I|} [\tilde{g}_i \log P(X_i; W) + (1 - \tilde{g}_i) \log(1 - P(X_i; W)] \tag{7}$$

where $\tilde{g}_i$ is label of X_i, $i : \Omega \to \{1, \ldots, |I|\}$, $|I|$ is the number of pixels in the input image X.

IV. Experiments

To verify the effectiveness of the proposed network, extensive experiments have been conducted on three remote sensing datasets. In this section, the experimental setup is described including details of datasets, training settings of HF-FCN and different criterion for evaluation.



(a) (b) (c)

Fig. 4. Sample patches on the three datasets (a) Massachusetts dataset (b) Vaihingen dataset (c) Potsdam dataset

A. Dataset Description

a) Massachusetts dataset: Massachusetts dataset consists of 151 aerial images of the Boston area which covers roughly 340 square kilometers. The resolution of each image is 1500×1500 with the spacial resolution of 1 meter per pixel. And the images are composed of red, green and blue channels. This dataset is built by Mnih while ground-truth is produced by Saito et al. The dataset is split into three parts, a training set of 137 images, a test set of 10 images and validation set of 4 images. To train the network, we create a set of image tiles for training and validation by sliding a 256×256 window with 64 stride from right to left, top to bottom. The detailed description is shown in Table I.

b) Vaihingen dataset: Vaihingen dataset is captured over Vaihingen which is a relatively small village with many detached buildings and small multi story buildings in Germany. This dataset contains 16 labeled images whose spacial resolution is 9cm per pixel. It consists of near infra-red, red, green, blue imagery with corresponding normalized digital surface models (nDSMs) and row DSMs. The dataset is divided into training set, validation set, and test set which have 11 images, 2 images, and 3 images respectively. The same crop operations are done as the Massachusetts dataset.

c) Potsdam dataset: In the Potsdam dataset, there are 24 labeled images whose ground sampling distance is 5cm. This dataset shows a typical historic city with large building blocks. In order to grasp the global information of the building, the spacial resolution of the original image is reduced from 6000×6000 to 1500×1500. Each image in this dataset contains 5-channel information: red, green, yellow, DSM and nDSM. We split the dataset into training, validation and test sets in a proportion of $7 : 2 : 1$.

Data augmentation is made on the Vaihingen dataset and the Potsdam dataset. One reason is that methods using dataset $a)$ do not extend the data. Hence, to make a fair comparison with other methods, we also do not extend it. Another reason is that the data quantity of dataset $b)$ and $c)$ is not enough which may lead to inadequate training. Therefore, some measures of data augmentation are made in dataset $b)$ and $c)$ including data rotation and mirror flipping. Components of the datasets are listed in Table I. Meanwhile, some sampled patches of dataset $a)$, $b)$, $c)$ are shown in Fig. 4(xuejin:5).

B. Training Settings

HF-FCN is trained on dataset $a)$ firstly owing to large amounts of training data. The pre-trained model of VGG16

TABLE I
COMPOSITION OF DATASET

	Masschusetts	Vaihingen	Potsdam
Labeled images	151	16	24
GSD	1m	9cm	5cm
Bands	R,G,B	IR,R,G,DSM	IR,R,G,B,DSM
Training images	137	11	17
Training patches	75938	115088	85000
Training patch size	256×256	256×256	256×256
Validation images	4	3	4
validation patches	2500	28376	25000
Validation patch size	256×256	256×256	256×256
Test images	10	2	3

TABLE II
PARAMETERS FOR NETWORK TRAINING

	Massachusetts	Vaihigen	Potsdam
mini-batch size	18	15	15
initial learning rate	10^{-5}	10^{-6}	10^{-5}
test_interval	1000	1000	1000
training iteration	10000	10000	10000
momentum	0.9	0.9	0.9
clip_gradients	16000	10000	10000
weight_decay	0.02	0.005	0.005

Net and ResNet are used to finetune our HF-FCN. We use the stochastic gradient descent algorithm with the learning rate divided by 10 for each 8000 iterations to train our network. The drop-out ratio is set to 0.5 which avoids overfitting. When the HF-FCN converges on the dataset a), we transfer it to the other datasets. All experiments in this paper are performed using the deep learning framework Caffe and trained on a single NVIDIA Titan 12GB GPU. Besides, the hyper-parameters are listed in Table II (xuejin:III).

C. Evaluation Metrics

Several evaluation metrics are adopted in our work. For dataset a), the most common metrics are correctness (precision) and completeness (recall). The stantdard (ρ=0) and relaxed (ρ=3) precision and recall scores are used to evaluate the prediction results. Here the relaxed precision means the predicted pixels are within ρ pixels of a true pixel while the relaxed recall is the true pixels are within ρ pixels of a predicted pixel. Moreover, the time cost is used to measure the efficiency of our HF-FCN. For dataset b) and c), we use correctness, completeness and F1 score as evaluation metrics.

$$completeness = \frac{TP}{TP + FN}, \tag{8}$$

$$correctness = \frac{TP}{TP + FP}, \tag{9}$$

$$F1\_score = 2 \cdot \frac{completeness \cdot correctness}{completeness + correctness} \tag{10}$$

where TP indicates the true positives, FP implies the false positive, TN means the true negatives and FN refers to the false negatives.

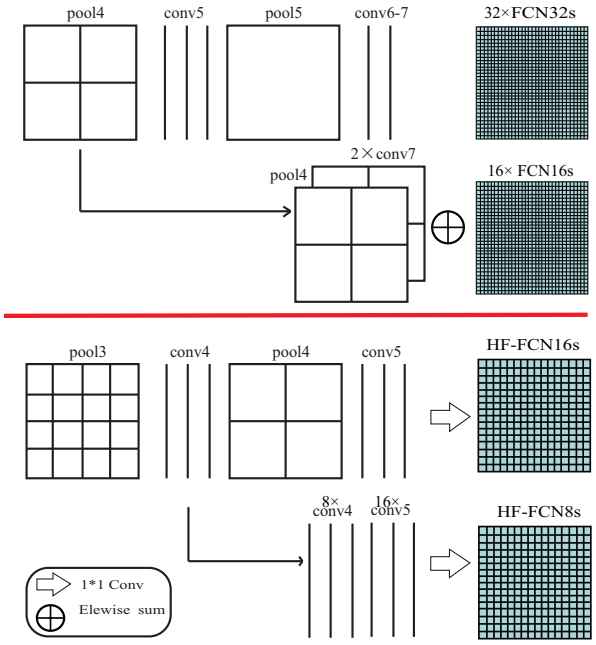


Fig. 5. HF-FCN variants. The feature maps generated from final group are fused into a coarse result, which is HF-FCN16s. The variant called HF-FCN8s concatenates the feature maps from the last 2 groups with the same fusion operation, and so on.

V. RESULTS AND DISCUSSION

In this section, the proposed method using dataset a), b), c) are compared to the recent non-deep-learning algorithms, such as Minh-CNN [9], Satio-multi [10] and Context [15]. Furthermore, it is also compared with some recent deep-learning based approaches, including FCN [18], SegNet [19], Deeplab [21] and U-Net [29].(xuejin:What else?) Moreover, for HF-FCN itself, we expect to investigate which kind of information extracted from feature extractors and how the effects of extracted information on the final prediction. Thus, some upsampled feature maps $\{U_k\}$ are presented. And several variants of HF-FCN which combine different up-sampling feature maps from Part 2 are proposed. They are shown in Fig. 6 and Fig. 5, respectively. In addition, we also tried different feature extractor networks, VGG16 Net and ResNet. The details are shown below. (xuejin:What about to change the backbone network?)

A. Massachusetts dataset

On the Massachusetts dataset, our method is compared to both the non-deep-learning algorithms and deep-learning based approaches. Table III(xuejin:4) present the quantitative analysis. A standard and relaxed precision and recall are amply to make a comparison. From the result, our method shows obvious superiority in terms of speed and precision. When comparing with SatiomultiMA&CIS [10], the standard and relaxed recall of our method are 5.5% and 1.3% higher than it. Meanwhile, the time cost is reduced from 67.84s to 1.07s and the speed is promoted about 63 times. Compared with U-Net [29], the speed is promoted about 3 times and recall is a little higher. These significant improvements demonstrate

TABLE III
CORRECTNESS AT BREAKEVEN OF HF-FCN V.S. [9] [10] [11] [18] [19] [29] [21]ON MASSACHUSETTS TEST SET. COST TIME IS COMPUTED IN THE SAME COMPUTER WITH A SINGLE NVIDIA TITAN 12GB GPU

	Recall (ρ = 3)	Recall (ρ = 0)	Time (s)
Mnih-CNN [9]	0.9271	0.7661	8.70
Mnih-CNN+CRF [9]	0.9282	0.7638	26.60
Satio-multi-MA [10]	0.9503	0.7873	67.72
Satio-multi-MA&CIS [10]	0.9509	0.7872	67.84
Alshehhi-GAP+seg [11]	0.955	–	–
FCN_4s [18]	0.839	0.6147	4.20
SegNet [19]	0.7710	0.5675	2.39
U-Net [29]	**0.9638**	**0.8357**	3.165
DeepLab_V2 [21]	0.9620	0.7575	**1.89**
HF-FCN(VGG16 Net)	**0.9643**	**0.8424**	1.07
HF-FCN(VGG+data aug)	–	–	–
HF-FCN(ResNet)	0.9588	0.8175	2.42
HF-FCN16s	0.9330	0.7233	0.85
HF-FCN8s	0.9643	0.8171	0.93
HF-FCN4s	0.9632	0.8394	**0.99**

that HF-FCN achieves best performance in effectiveness and efficiency.

Extensive comparisons are made between HF-FCN and other mainstream methods in semantic segmentation domain. The quantitative and visual results are shown in Table III and Fig. 9, respectively. On the charts, we can see that our method better performance in speed and precision. And the details and integrity of the building are well preserved by our method.

To explore which kinds of information extracted by hierarchical fusion operation in Part 2. Some upsampled feature maps $\{U_1, U_2, U_3, U_7, U_{10}, U_{13}\}$ are shown in Fig. 6. The U1_1 (U_1) in Fig. 6(b) means the upsampled feature maps from F1_1 (F_1) which are feature maps generated from conv1_1 in VGG16 Net. Due to small receptive field of conv1_1 and conv1_2, they extract low-level features like edges. And the U1_2 (U_2) looks like an over-segmentation which groups pixels with similar color or texture into a subregion. With the deepening of the network, in the U2_1 (U_3), as Fig. 6(d) shows, shape information is augmented. And from the U3_3 (U_7), we can see that regions with significantly varying appearance are merged into an integrated building by considering high-level features. In U4_3 (U_{10}) and U5_3 (U_{13}), more semantic information of rooftop is got, which can distinguish the rooftop and the roads with similar color and deal with the problem caused by shadow. The final prediction results are shown in Fig. 6(h).

Secondly, to explore the effects of the feature maps generated from each feature extract stage $\{F_k\}$ on the final result, variants of HF-FCN which are counterpart of FCN are designed. Fig. 5 shows the constrast diagram of variants of FCN and HF-FCN. Unlike FCN, a fusion operation rather than summation are leveraged to build our HF-FCN 16s, 8s and 4s. The performance of HF-FCN variants are shown in Fig. 7, Fig. 8 and Table III(xuejin:Figure 8, Figure 9 and Table V). From the disgrams, we can get the following conclusions easily. First, The prediction result obtained from the last layer gets a coarse result, which loses much of location information that are mainly encoded in the shallow feature maps. Second, the largest gap presented between HF-FCN16s and HF-FCN8s about 9% in recall rates, it may suggest that

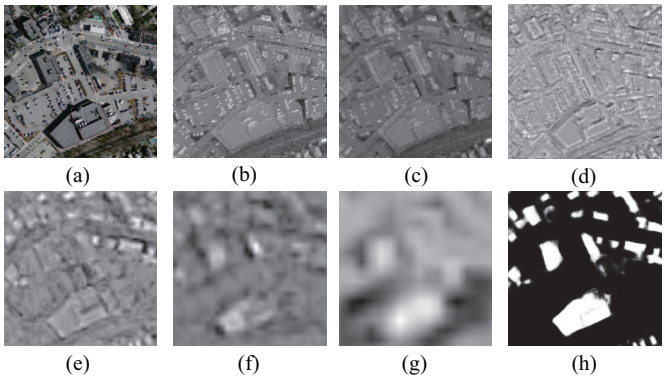


Fig. 6. (a) Input aerial image. (b-g) Feature maps of U1_1, U1_2, U2_2, U3_3, U4_3, U5_3, respectively. (h) Predicted label map. All the images are normalized to the range of $0 - 255$.

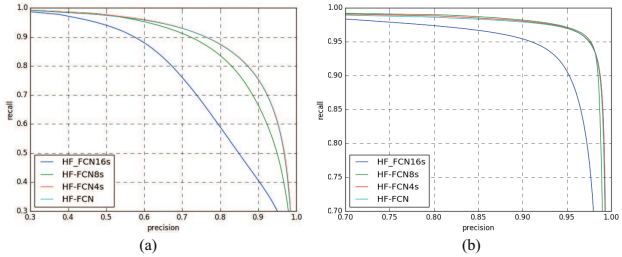


Fig. 7. The relaxed precision-recall curves from HF-FCN variants with two slack paramters. The biggest gap occurs between HF-FCN16s and HF-FCN8s, which indicates the most additional information coming from middle layers.

the most information supplement to the HF-FCN is got in middle layers. Third, the PR curves of HF-FCN4s and HF-FCN almost coincide. It illustrates the low-level information has little effect on the prediction results. Forth, with the addition of the shallow feature map, the network is more distinct for the segmentation of tiny buildings, which solves the problem of easy adhesion to adjacent buildings. Since, all the conv layers contained useful hierarchical information that is critical to the final prediction.

In the end, we want to prove that our fusion operations learn the connections between features. The connection weights of F1_1, F4_1 and Part 3 are shown in Fig. 10. The weights are not the same, which means that fusion operations have effect on feature combination. From the Fig. 10 (a) to Fig. 10 (c), the range of weights increases gradually. And from the Fig. 10 (c), we can arrive at the conclusion that the different layers have virous effects on the final result. For example, the U1_1 has little effect on the prediction while the U3_2 and U4_3 have bigger role on the final prediction. It also in accordance with our experimental results that middle layers provide more information. (xuejin:Weight for what? to fuse feature map? The distribution does not make too much sense.)

B. *Vaihingen dataset*

On Vaihingen dataset, three experiments are undertaken to explore the effects of different inputs, diverse variants and various methods. We utilize three kinds of combinations of

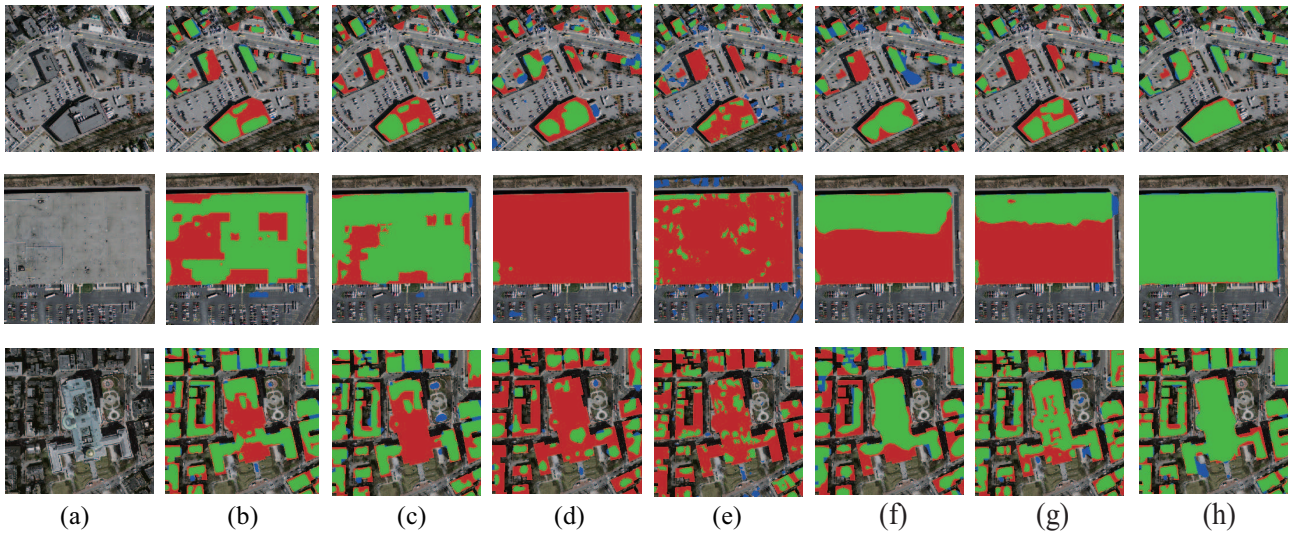


Fig. 9. (a) input images. (b) Results of Mnih-CNN+CRF. (c) Results of SatiomultiMA&CIS. (d) Results of FCN4s . (e) Results of SegNet. (f) Results of DeepLab_V2. (g) Results of U-Net. (h) Our results. TP are shown in green, FP are shown in blue and FN are in red.

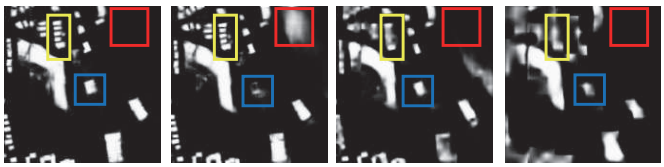


Fig. 8. Prediction results of HF-FCN, HF-FCN4s, HF-FCN8s and HF-FCN16s. The yellow box shows the continuous refinement of the tiny buildings. The red and blue boxes show the mutual promotion and contradiction between different layers.

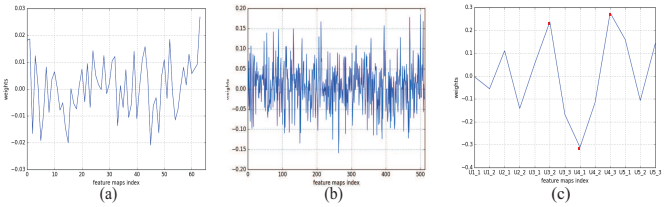


Fig. 10. (a) is weights learned by F1_1, (b) is weights learned by F4_1, (c) is weights learned by Part 3.

image channels as inputs. The inputs of the 3 channels are IR, R, G and adding the nDSM as the forth channel. Based on it, DSM is added and made up 5-channel input. We use three standards to make a more comprehensive evaluation. The evaluation results are shown in Table IV, which illustrates that 3-channel input performed better than the other. The Rec and Pre in Table IV means the recall and precision of prediction results. And F1 indicates the F1_score of results. The number in bold shows the best results in validation and test set. Corresponding visual results are shown in Fig. 13.

(xuejin:Do you compare with others?)

We compare with some other methods which use the same dataset. The detail comparison results are shown in Fig. 14. From a visual perspective, our method gets a much more refined roof region, both on continuity of labels and integrity

TABLE IV
PERFORMANCE COMPARISON OF THE RESULTS OF DIFFERENT INPUTS AND METHODS ON VAIHIGEN DATA SET. (XUEJIN:WHAT ARE THE NUMBERS IN THE IMG COLUMN?)

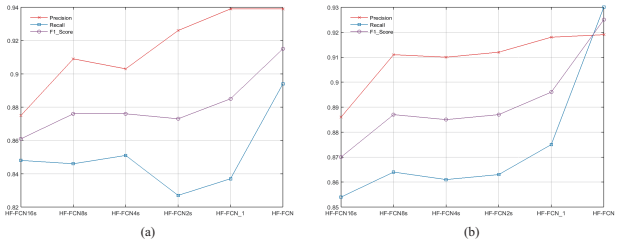
		Pre	Rec	F1
Val	3_in	0.939	0.894	**0.915**
	4_in	0.96	0.865	0.909
	5_in	0.939	0.880	0.907
Test	3_in	0.919	0.930	**0.925**
	4_in	0.907	0.872	0.888
	5_in	0.858	0.900	0.878
FCN_4s [18]		0.871	0.884	0.878
SegNet [19]		–	–	–
U-Net [29]		–	–	–
DeepLab_V2 [21]		0.926	0.881	0.903
HF-FCN16s		0.886	0.854	0.870
HF-FCN8s		0.911	0.864	0.887
HF-FCN4s		0.910	0.861	0.885

of structural.

The results of diverse variants are shown in Fig. 11. The HF-FCN_1 in Fig. 11 indicates that the last conv layer in Part 3 does not use the previous trained model to initialize. And HF-FCN means that the whole layers use the pre-trained model to initialize. From the curves, the performance of HF-FCN exceeds the variants and gets a excellent result. Additionally, using the pre-trained weights of Part 3 has a significance in the final results.

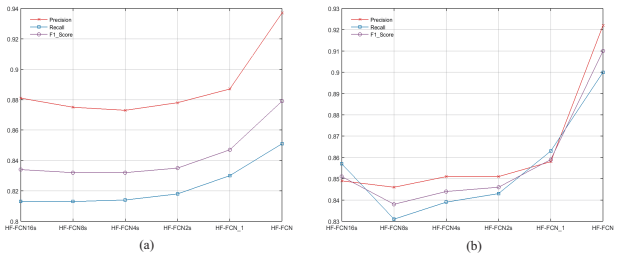
C. Potsdam dataset

The same experiments are implemented on Potsdam dataset. Firstly, we utilize DSM and IR information as extra inputs based on the RGB input. The specific quantitative evaluation and intuitive visual prediction results are shown in Table V and Fig. 15. In the validation process, the 4-channel input gets better overall performance. Meanwhile, the 5-channel input seems perform better in the course of testing. From the visual results, the 5-channel input network gets lower error detection rate which is shown on the image with small blue



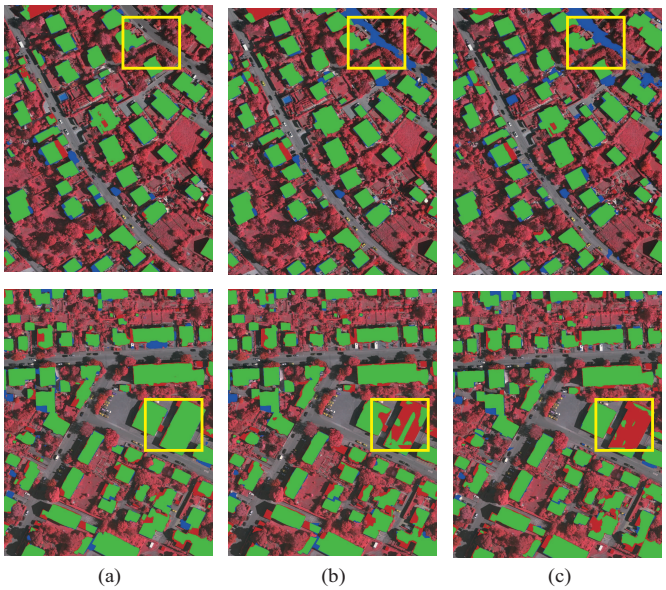
(a) (b)

Fig. 11. Results of HF-FCN variants on Vaihingen dataset. (a) (b) shows the precision, recall and F1_score of validation set and test set of Vaihingen dataset respectively.(xuejin:Bigger font)



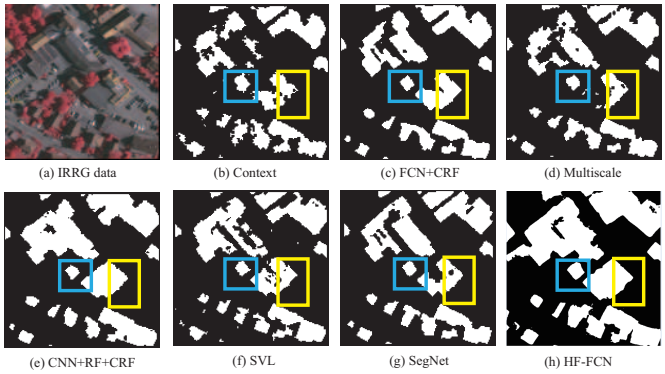
(a) (b)

Fig. 12. Results of HF-FCN variants on Potsdam dataset. (a) (b) shows the precision, recall and F1_score of validation set and test set of Potsdam dataset respectively.



(a) (b) (c)

Fig. 13. Prediction results on Vaihingen dataset. (a) (b) (c) shows results of the 3-channel input, 4-channel input and 5-channel input of Vaihingen dataset respectively. Here, TP are shown in green, FP are shown in blue and FN are in red.



(a) IRRG data (b) Context (c) FCN+CRF (d) Multiscale

(e) CNN+RF+CRF (f) SVL (g) SegNet (h) HF-FCN

Fig. 14. Results of different methods. (a) is input image, (b)(d)(g) are results of [15], (c) is result of [31], (f) is result of [32], (g) is our result. The blue and yellow frames show some details between these methods.

areas. And from the 3-channel input to 5-channel input, the F1 score increases from 0.879 to 0.891 on the validation set and increases 0.031 on the test set. It indicate that the other information of geographical feature have a certain effect on the final result.

We compare HF-FCN with other methods using the Potsdam dataset. Some qualitative results are shown in Fig. 16. From the figure, we can easily see that HF-FCN got more remarkable segmentation results. And edges and structure of buildings are preserved better.

As done on Vaihingen dataset, contrast experiments of HF-FCN variants are implemented. The performance curve of HF-FCN variants are shown in Fig. 12. The HF-FCN_1 in Fig. 12 indicates that the last conv layer in Part 3 does not use the previous trained model to initialize. And HF-FCN means that the whole layers use the pre-trained model to initialize. Initialization of parameters has a greater promotion on the final results.

TABLE V
PERFORMANCE COMPARISON OF THE RESULTS OF DIFFERENT INPUTS AND METHODS ON POTSDAM DATA SET

		Pre	Rec	F1
Val	3_in	0.937	0.851	0.879
	4_in	0.937	**0.872**	**0.894**
	5_in	**0.944**	0.864	0.891
Test	3_in	0.922	0.900	0.910
	4_in	0.937	0.935	0.936
	5_in	**0.940**	**0.943**	**0.941**
FCN_4s [18]		–	–	–
SegNet [19]		–	–	–
U-Net [29]		–	–	–
DeepLab_V2 [21]		–	–	–
HF-FCN16s		0.849	0.857	0.851
HF-FCN8s		0.846	0.831	0.838
HF-FCN4s		0.851	0.839	0.844

VI. APPLICATION

The segmentation results are further used to 3D building reconstruction. We make use of the depth map and generate the point cloud of remote sensing images. After that, the 3D building reconstruction methods could applied to the generated point cloud. In this paper, the approach proposed by zhou [33]

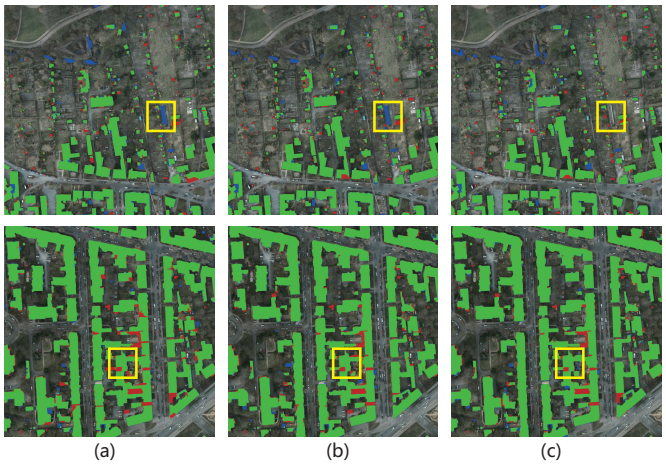


Fig. 15. Prediction results on potsdam dataset. (a) (b) (c) shows results of the 3-channel input, 4-channel input and 5-channel input of Vaihingen dataset respectively. Here, TP are shown in green, FP are shown in blue and FN are in red.

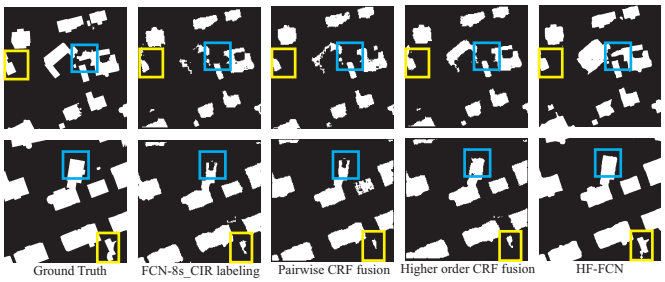


Fig. 16. Results of different methods. The second column is the results of using only the FCN with CIR. Pairwise CRF fusion shows the result of fusing FCN-8s_CIR with LiDAR data in a pairwise CRF. Higher-order CRF are used to generate the results shown in third column. Our results are shown in last colunm.

are used to generate the 3D models of buildings in the scene. Fig. 17 and Fig. 18 show the 3D models of Vaihingen and Potsdam dataset respectively. The details of a single building are also presented. From the figures, we can see the 3D models preserve the characteristics of buildings well whether the structure of the roof or the simplification of details.

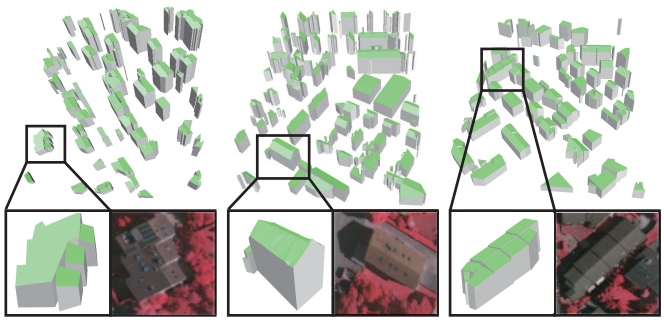


Fig. 17. The 3D modelling of Veihingen dataset. The single building model and its corresponding optical patch were shown together.

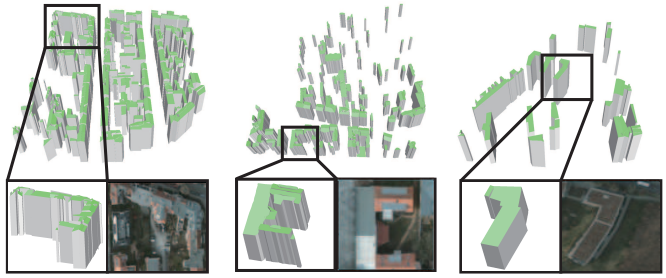


Fig. 18. The 3D modelling of Potsdam dataset. The single building model and its corresponding optical patch were shown together.

VII. Conclusion

In this paper, an efficient building detection approach is proposed and have a further application in building reconstruction. Using proposed feature fusion operations, a novel CNN architecture is presented for building extraction, named HF-FCN. Unlike previous non-deep-learning algorithms, we provide a end-to-end network for building extraction. And it is robust to the different scales of buildings and efficient for a large-scale remote sensing images. On the other hand, distinct from the previous deeplearning based methods, we utilize the multi-scale inherent information within the CNN and refine the details by a fusion manner stage by stage. In addition, an application of 3D building reconstruction depend on the segmentation results is implemented. Compared to the existing 3D reconstruction methods, our proposed approach greatly accelerates the part of building extraction. Finally, our study suggests that even with the powerful semantic expressive ability of CNNs and their good robustness to scale, it is still critical to address multi-scale problems utilizing hierarchical feature maps encoded in CNNs.

References

[1] A. Huertas and R. Nevatia, "Detecting buildings in aerial images," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 2, pp. 131–152, 1988.

[2] S. Noronha and R. Nevatia, "Detection and modeling of buildings from multiple aerial images," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 5, pp. 501–518, 2001.

[3] M. S. Nosrati and P. Saeedi, "A novel approach for polygonal rooftop detection in satellite/aerial imageries," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 1709–1712.

[4] M. Izadi and P. Saeedi, "Three-dimensional polygonal building model estimation from single satellite images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 6, pp. 2254–2272, 2012.

[5] J. Wang, X. Yang, X. Qin, X. Ye, and Q. Qin, "An efficient approach for automatic rectangular building extraction from very high resolution optical satellite imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 3, pp. 487–491, 2015.

[6] M. Cote and P. Saeedi, "Automatic rooftop extraction in nadir aerial imagery of suburban regions using corners and variational level set evolution," *IEEE transactions on geoscience and remote sensing*, vol. 51, no. 1, pp. 313–328, 2013.

[7] J. Peng, D. Zhang, and Y. Liu, "An improved snake model for building detection from urban aerial images," *Pattern Recognition Letters*, vol. 26, no. 5, pp. 587–595, 2005.

[8] B. Sirmacek and C. Unsalan, "Urban-area and building detection using sift keypoints and graph theory," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 4, pp. 1156–1167, 2009.

[9] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, University of Toronto (Canada), 2013.

[10] S. Saito, T. Yamashita, and Y. Aoki, "Multiple object extraction from aerial imagery with convolutional neural networks," *Electronic Imaging*, vol. 2016, no. 10, pp. 1–9, 2016.

[11] R. Alshehhi, P. R. Marpu, W. L. Woon, and M. Dalla Mura, "Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130, pp. 139–149, 2017.

[12] W. Zhao, S. Du, Q. Wang, and W. J. Emery, "Contextually guided very-high-resolution imagery classification with semantic segments," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 132, pp. 48–60, 2017.

[13] S. Paisitkriangkrai, J. Sherrah, P. Janney, V.-D. Hengel *et al.*, "Effective semantic pixel labelling with convolutional networks and conditional random fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 36–43.

[14] Y. Liu, S. Piramanayagam, S. T. Monteiro, and E. Saber, "Dense semantic labeling of very-high-resolution aerial imagery and lidar with fully-convolutional neural networks and higher-order crfs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 76–85.

[15] N. Audebert, A. Boulch, H. Randrianarivo, B. Le Saux, M. Ferecatu, S. Lefevre, and R. Marlet, "Deep learning for urban remote sensing," in *Urban Remote Sensing Event (JURSE), 2017 Joint*. IEEE, 2017, pp. 1–4.

[16] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Urban land cover classification with missing data using deep convolutional neural networks," *arXiv preprint arXiv:1709.07383*, 2017.

[17] Y. He, S. Mudur, and C. Poullis, "Multi-label pixelwise classification for reconstruction of large-scale urban areas," *arXiv preprint arXiv:1709.07368*, 2017.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[19] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for scene segmentation," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[20] J. Hoffman, S. Gupta, and T. Darrell, "Learning with side information through modality hallucination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 826–834.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[22] R. Vemulapalli, O. Tuzel, M.-Y. Liu, and R. Chellapa, "Gaussian conditional random field network for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3224–3233.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[26] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5987–5995.

[28] T. Zuo, J. Feng, and X. Chen, "Hf-fcn: Hierarchically fused fully convolutional network for robust building extraction," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 291–302.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[31] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla, "Semantic segmentation of aerial images with an ensemble of cnss," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016*, vol. 3, pp. 473–480, 2016.

[32] M. Gerke, "Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen)," 01 2015.

[33] Q.-Y. Zhou and U. Neumann, "2.5 d building modeling with topology control," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2489–2496.