Building-A-Nets: Robust Building Extraction From High-Resolution Remote Sensing Images With Adversarial Networks

Xiang Li, *Student Member, IEEE*, Xiaojing Yao, and Yi Fang, *Member, IEEE*

***Abstract*—With the proliferation of high-resolution remote sensing sensor and platforms, vast amounts of aerial image data are becoming easily accessed. High-resolution aerial images provide sufficient structural and texture information for image recognition while also raise new challenges for existing segmentation methods. In recent years, deep neural networks have gained much attention in remote sensing field and achieved remarkable performance for high-resolution remote sensing images segmentation. However, there still exists spatial inconsistency problems caused by independently pixelwise classification while ignoring high-order regularities. In this paper, we developed a novel deep adversarial network, named Building-A-Nets, that jointly trains a deep convolutional neural network (generator) and an adversarial discriminator network for the robust segmentation of building rooftops in remote sensing images. More specifically, the generator produces pixelwise image classification map using a fully convolutional DenseNet model, whereas the discriminator tends to enforce forms of high-order structural features learned from ground-truth label map. The generator and discriminator compete with each other in an adversarial learning process until the equivalence point is reached to produce the optimal segmentation map of building objects. Meanwhile, a soft weight coefficient is adopted to balance the operation of the pixelwise classification and high-order structural feature learning. Experimental results show that our Building-A-Net can successfully detect and rectify spatial inconsistency on aerial images while archiving superior performances compared to other state-of-the-art building extraction methods. Code is available at https://github.com/lixiang-ucas/Building-A-Nets.**

***Index Terms*—Adversarial network, building extraction, fully convolutional DenseNet (FC-DenseNet), remote sensing, structural feature learning.**

Manuscript received April 24, 2018; revised July 2, 2018; accepted August 9, 2018. Date of publication August 30, 2018; date of current version October 15, 2018. This work was supported in part by the Jiangsu Province Geographic Information Research Project (JSCHKY201720), in part by the National Science technology Support Plan Project of China under Grant 2015BAJ02B00, and in part by the China Scholarship Council (201704910704). *(Corresponding author: Yi Fang.)*

X. Li is with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China, and also with the Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: xl1845@nyu.edu).

X. Yao is with the Institute of Remote sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China (e-mail: yaoxj@radi.ac.cn).

Y. Fang is with the Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201 USA, and also with Department of Electrical and Computer Engineering, New York University Abu Dhabi, Abu Dhabi 129188, United Arab Emirates (e-mail: yfang@nyu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2018.2865187

I. Introduction

With the rapid development of sensor technology, tremendous amount of remote sensing data are being collected every day. These valuable data open up a new window for us to understand the earth surface. Among which, building rooftop is one of the most important types of terrestrial objects because it plays a vital role in a wide range of applications, such as urban planning, real-estate management, illegal building survey, geographic information systems, etc.

Researchers have proposed a variety of methods to conduct automatic building extraction on high-resolution remote sensing images. Remarkable progress comes from Mnih [1], who first introduced the convolutional neural networks (CNNs) for building extraction. In recent years, CNNs are driving advances in a wide variety of image recognition tasks, thanks to their powerful ability in effectively extracting high-level features without the involvement of human ingenuity in feature engineering, and quickly become prominent in remote sensing applications [2]. Researchers have adopted various CNN architectures for automatic building extraction. While early works mainly use patched-based CNNs [1], [3], recent progress has taken fully convolutional approaches [4], [5].

Though these CNN-based methods have achieved remarkable performance on some public building datasets, they usually cannot guarantee the spatial contiguity of building objects due to the independent label prediction of each pixel. To address this problem, a fully connected conditional random field (CRF) (dense CRF) [6] is commonly used as a postprocessing method to enforce spatial contiguity in the output score maps. Unlike the CNN models, which can only model unary potentials within a limited field of view, this methods can effectively account for all pairwise potentials on images by using a predefined second-order appearance kernel and smooth kernel. Moreover, higher order potentials, in recent progress, have also been observed to be useful and can be integrated into CNN-based segmentation models [7].

More recently, following the great success of generative adversarial networks (GANs), Luc *et al.* [8] proposed to train an adversarial network to encourages the segmentation network to generate label maps that cannot be distinguished from the ground truth. In this way, the joint distribution of all label variables at each pixel location can be assessed as a whole, and thus, can enforce forms of high-order consistency that cannot be

enforced by pixelwise classification or pairwise terms. However, traditional adversarial networks are known hard to train and face the danger of model collapse. This can lead to an optimization problem for segmentation network.

In this paper, we propose an adversarial training strategy for remote sensing images segmentation and use it for building extraction. To avoid the optimization problem of the traditional adversarial network, a novel stable adversarial network training strategy is used to train the model. The contributions of this paper are as follows.

1) To the best of our knowledge, this paper makes the first attempt to accommodate the adversarial training for building extraction on remote sensing images.
2) To ensure the stable learning of high-order structural features, we adopt an autoencoder network for adversarial learning instead of a binary classifier.
3) A soft weight coefficient is introduced to balance the process of the pixelwise classification and high-order structural feature learning, and the effect of different weight coefficients is discussed.

II. Related Work

A. Semantic Segmentation

Fully convolutional networks (FCNs) [9] are an important milestone in computer vision field and have become the base architecture for most of the semantic segmentation models. After the great success of FCN, recently proposed segmentation models are mostly in fully convolutional fashion and mainly designed by the following:

1) adopting more efficient basic encoder models for feature extraction [10], [11];
2) reconstructing novel decoder structure that can effectively recover the context information while integrating the high-level features in downsampling path [12], [13]; and
3) integrating structural features with some independent modules [14]–[16].

In this paper, our deep generator network and adversarial learning strategy can be viewed as one of (1) and (3), respectively.

On one hand, many researchers have reported an accuracy improvement when using more powerful encoders on different kinds of image recognition tasks [17]–[19]. Recently, densely connected convolutional network (DenseNet) introduces a new connective pattern for successive convolutional layers, and has shown superior results on scene classification tasks [19]. This encoder network presents good features of parameter efficiency, implicit deep supervision, and feature reuse. Based on this encoder network, Jégou *et al.* [11] proposed a fully convolutional DenseNet (FC-DenseNet) to deal with the problem of semantic segmentation and achieved state-of-the-art results on urban scene benchmark datasets. In this paper, we will use FC-DenseNet model as our base segmentation network.

On the other hand, traditional CNNs cannot guarantee label agreement between nearby pixels with the similar color or suppress small isolated regions [6]. Probabilistic graphical models, such as CRF [20], [21] or MRF [22]–[24], are commonly used to address this problem. Recent works have tried to train CNN models and graphical models jointly and the results have shown that integrating CRF with CNNs can lead to a better segmentation performance, especially along the boundaries of objects [6], [25]. Krähenbühl and Koltun [6] propose a fully connected CRF (named dense CRF) that establishes pairwise potentials on all pairs of pixels in the image. Following this idea, Zheng *et al.* [16] formulates the multistage mean-field approximate inference for the dense CRF as a recurrent neural network for end-to-end training. In comparison to these works, which mainly integrates pairwise terms with CNNs, in this paper, we explore an adversarial network to learn latent high-order structural features without explicitly constructing it. The learned features are integrated into segmentation network during network training. Once trained, our model can perform efficient prediction since it does not need this adversarial part in the inference stage.

B. Adversarial Learning

The adversarial learning was first proposed by Goodfellow *et al.* [26] to refine the generative model. Commonly, a GAN consists of two parts; a generator takes samples from a random Gaussian distribution and uses a deep network to produce images that are similar to real images, and a discriminator network is trained to distinguish whether an input comes from the generator or real ones. These two networks are trained alternatively until the equivalence point is reached where the generated images are indistinguishable from the real ones.

Following GAN, a set of methods was proposed to ensure the stable training of GANs and improve the visual quality of the generated images. Goodfellow *et al.* [26] proposed a deep convolutional version of GAN and a set of training tricks was used to ensure stable training. Zhao *et al.* [27] proposed to view the discriminator as an energy function, thus enables various loss functions other than binary classifier. Their results show that a GAN with an autoencoder architecture exhibits more stable behavior than regular GANs during training. Arjovsky *et al.* [28] figured out the optimization problem of original GAN caused by the binary classification loss function and proposed to use Wasserstein distance as adversarial loss function instead. More recently, Berthelot *et al.* [29] proposed to use a boundary equilibrium term to explicitly balance the training process of generator and discriminator while using an autoencoder network to generate loss distributions.

III. Method

To use GAN for semantic segmentation, we follow the training techniques proposed in [8]. Specifically, a base segmentation network is used to generate high accuracy prediction label maps, and an adversarial network is trained to distinguish the label maps produced by the segmentation network from the ground truth ones. These two networks act as generator and discriminator, respectively, and they compete with each other in an adversarial learning process until the equivalence point is reached to produce the optimal segmentation map of building objects. In this paper, instead of directly feeding the label maps only into our discriminator network, we multiply the label maps

with input aerial images in a pixelwise manner and feed the masked images into our adversarial network as inputs.

A. Segmentation Network

Traditional segmentation networks are commonly trained based on the pixelwise classification, in which each pixel is classified independently without considering the label consistency, and thus, cannot guarantee the object structural. To enable structural feature learning in our segmentation network, our generator network gathers losses from both pixelwise classification and the adversarial network. The former encourages our generator network to produce accurate class labels for each pixel location, and the latter encourages a potential high-order structural loss so as to refine the spatial inconsistencies of building objects. Here, we introduce a weight coefficient α to control the balance of these two losses, our revised generator objective loss function is given as follows:

$$\mathcal{L}_G = (1-\alpha)\cdot L_{\text{ae}}(x') + \alpha \cdot L_{\text{cls}}(G(z), y) \quad \text{for}\, \theta_G \tag{1}$$

where x' denotes the masked image produced by pixelwise multiplication of an input aerial image z and the predicted segmentation map $G(z)$, $L_{\text{cls}}(\cdot)$ and $L_{\text{ae}}(\cdot)$ stand for softmax cross-entropy loss in segmentation network and autoencoder reconstruction loss in discriminator network, defined by formulation (2) and (3), θ_G stands for the parameters of generator network

$$L_{\text{cls}}(G(z), y) = -\frac{1}{MN}\sum_{1}^{M}\sum_{1}^{N} y \cdot \log(G(z)) \tag{2}$$

$$L_{\text{ae}}(x') = \frac{1}{MN}\sum_{1}^{M}\sum_{1}^{N} |D(x') - x'| \tag{3}$$

where $D(x')$ denotes the reconstructed output of by feeding masked image x' into the autoencoder network and M and N indicate width and height of the input image.

One should note that, here, the binary classifier was replaced with an autoencoder network, to ensure stable training of the adversarial learning process, namely BEGAN rather than traditional GAN. In deep learning field, it is now pretty common to replace the binary classifier in discriminator with some other networks, like autoencoders, and this strategy works much better [29]. In this paper, the autoencoder network aims to match loss distributions using a loss derived from the Wasserstein distance rather than directly matching data distributions enforced by a binary classifier. Both reconstruction loss in autoencoder and discrimination loss in the binary classifier can be viewed as an implicit high-order regularize for the generator network (in this paper, the generator is a segmentation network). Although these adversarial networks do not explicitly conduct discrimination, commonly we still call them discriminators (one can view it as a discriminator for loss distribution). Interested readers can refer to the BEGAN paper [29] for details.

According to this definition of $\mathcal{L}_G$, a larger α value will encourage pixelwise classification to dominate the learning process of segmentation network. Especially, when α is set to 1, the high-order structural loss encouraged by discriminator will not be able to back propagate its gradients to the generator network, i.e., the adversarial network is disabled. We use this configuration as our baseline. On the contrary, a smaller α value will enforce our segmentation to pay more attention to the structural losses from the adversarial network while updating its parameters. Properly setting the α value can enable our model to find an optimum state to balance these two losses, and we will discuss this in details in our experimental part.

In this paper, we use a FC-DenseNet proposed in [11] as our base segmentation network. One would find it easy to replace it with some other end-to-end segmentation networks, such as FCN[9] or U-Net[30]. Following [11], our network is a typical encoder–decoder architecture with skip connections, composed of five dense blocks in downsampling path and five dense blocks in the upsampling path, and one bottleneck block connecting the downsampling path and the upsampling path, which is also a dense block. Hence, there are 11 dense blocks in total, and the growth rate for all convolutional layer in dense blocks is set to 12. The overall architecture of our segmentation network is shown in Fig. 1.

In our experiments, we build our segmentation networks based on FC-DenseNet architecture with two different network depths. The first one is a relative shallower network with 56 layers. Training and validation using this network would not take too much time, so we use this network to test the effect of adversarial learning. We build this network by stacking an equivalent of four convolutional layers in each dense block. The second one is a relative deeper network with 158 layers, this network is used to evaluate the performance of the Build-A-Nets model; and we build this network by giving each dense block with [9, 10, 12, 15, 17, 20, 17, 15, 12, 10, 9] convolutional layers.

B. Adversarial Network

Directly training traditional binary classification discriminator network may involve optimization difficulty for generator network because the discriminator tends to win too easily at the beginning of training [31], i.e., the discriminator would quickly come to convergence while the generator fails to get gradients to learn properly. Following the training techniques proposed in [29], a boundary equilibrium term γ is adopted to balance the competing learning process of generator and discriminator network. Supposing our goal is to encourage an equilibrium point where $E[\gamma \cdot L_{\text{ae}}(x)] = E[\mathcal{L}_G]$, we can define the objective loss function of discriminator network $\mathcal{L}_D$ as follows:

$$\mathcal{L}_D = L_{\text{ae}}(x) - k_t \cdot \mathcal{L}_G \quad \text{for}\, \theta_D \tag{4}$$

where x denotes the masked image produced by pixelwise multiplication of input aerial image z and corresponding label map y, $L_{ae}(x)$ denotes the reconstruction loss by feeding masked image x into the autoencoder network, formulated as (5), $\mathcal{L}_G$ denotes the generator loss, and the calculation of which is given in Section III-A, $k_t \in [0, 1]$ controls how much emphasis is put on generator loss $\mathcal{L}_G$ for each training step t, and we update k_t following (6), θ_{D} stands for the parameters of discriminator

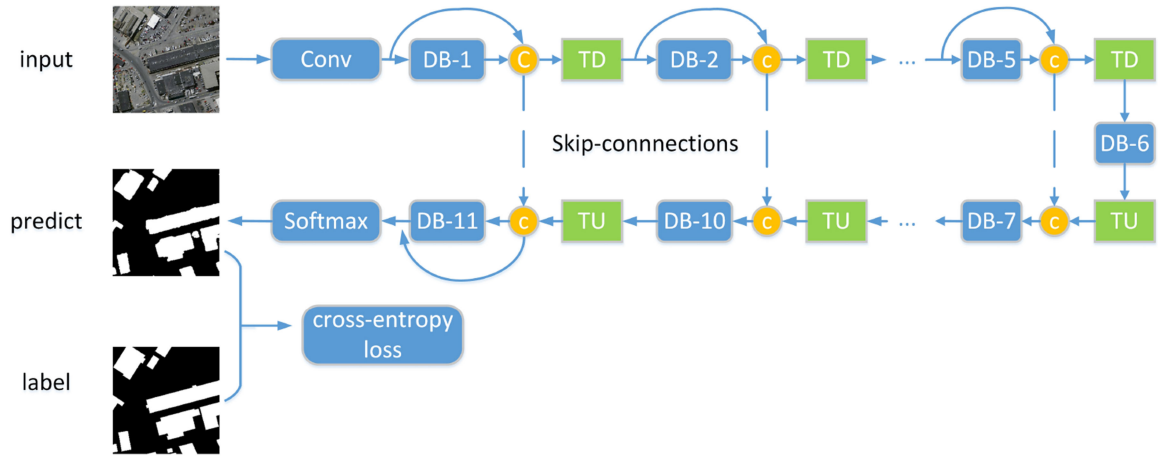


Fig. 1. Overall architecture of our segmentation network. DB indicates dense block, TD and TU stand for the transition down and transition up layer, respectively, and C stands for channelwise concatenation. Note that we skip DB-3 and DB-4 in downsampling path and DB-8 and DB-9 in the upsampling path and their corresponding transition layer.

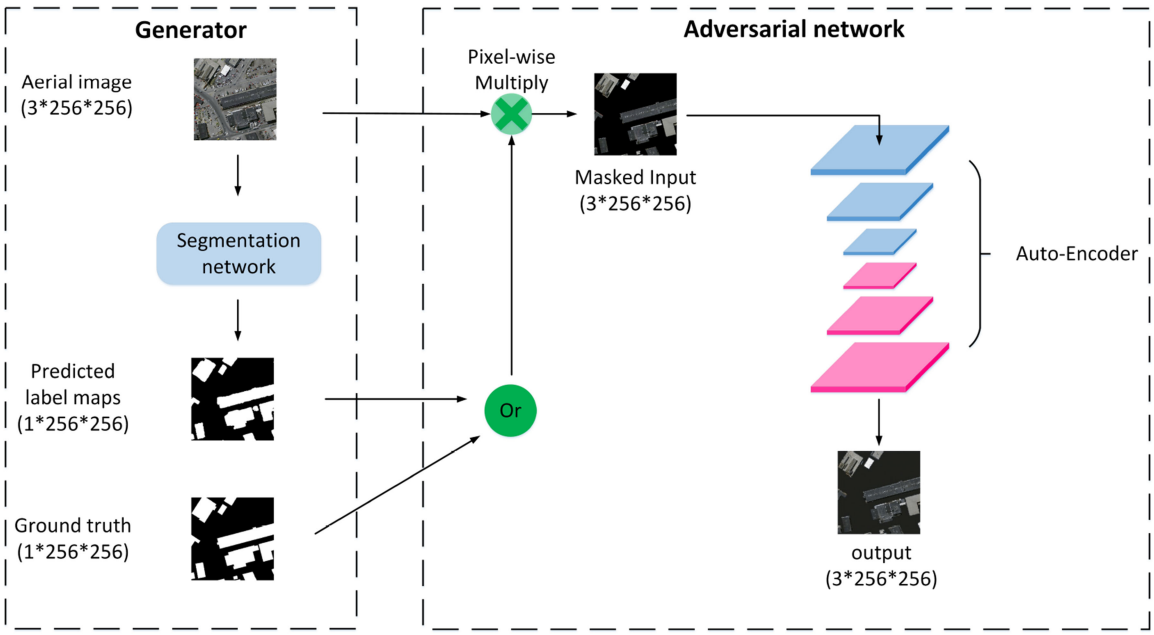


Fig. 2. Overview of our segmentation architecture with the adversarial network. Left: The segmentation network takes an aerial image as input and produces a pixelwise classification label map. Right: A label map, chosen from segmentation output or ground truth, is multiplied with their corresponding input aerial image to produce a masked image, and the adversarial network takes this masked image map as input and adopts an autoencoder network to reconstruct it.

network

$$L_{ae}(x) = \frac{1}{MN}\sum_{1}^{M}\sum_{1}^{N}|D(x) - x| \tag{5}$$

where $D(x)$ denotes the reconstructed output by feeding masked image x into the autoencoder network, M and N indicate width and height of input image

$$k_{t+1} = k_t + \lambda_k \cdot (\gamma \cdot L_{ae}(x) - \mathcal{L}_G) \quad \text{for each step } t \tag{6}$$

where the second term acts as a kind of balance loss used to update k_t, λ_k can be viewed as the learning rate for k_t. In our experiments, we initialize k_0 to 0 and set λ_k to 0.001. According to the proportional control theory, promisingly, above-mentioned learning strategies will lead to an equilibrium point where $E[\gamma \cdot L_{\text{ae}}(x)] = E[\mathcal{L}_G]$. In addition, a global convergence measure is used to determine whether our model has come to a convergence, formulated as

$$M_t = L_{\text{ae}}(x) + |\gamma \cdot L_{\text{ae}}(x) - \mathcal{L}_G|. \tag{7}$$

In this paper, we use a simple autoencoder as our discriminator network, as shown in the right part of Fig. 2, which consists of 3 convolutional layers and 3 deconvolutional (transposed convolution) layers with a stride of 2, and the channel number of each hidden layer is fixed to 128.

Algorithm 1: Training Procedure of Our Model.

1: Initialize all network parameters, set $k_0 = 0$.
2: **while** M_t does not convergence **do**
3: Sample mini-batch of m input aerial images $\{z^1, ..., z^m\}$ and corresponding ground-truth labels $\{y^1, ..., y^m\}$.
4: Generate the predicted label map $\{G(z^1), ..., G(z^m)\}$ by one forward pass of segmentation network G.
5: Generate the masked images x by pixelwise multiplication of input aerial image z and label map ($G(z)$ or y).
6: Feed the masked images x into discriminator D, and get the discriminator output $D(x)$ by one forward pass of discriminator network.
7: Calculate the objective loss $\mathcal{L}_G$ and $\mathcal{L}_D$ for generator and discriminator using equation (1) and (4).
8: Update the parameters of generator network θ_G with respect to $\mathcal{L}_G$.
9: Update the parameters of discriminator network θ_D with respect to $\mathcal{L}_D$.
10: Update k_t using (6).
11: Calculate convergence measure M_t using (7).
12: **end while**

C. Network Training

With carefully designed network architecture, our Building-A-Nets model can train in an end-to-end way rather than alternatively training generator and discriminator network. The training procedure follows Algorithm 1.

Moreover, it should be noted that the masked images usually involve a large area of background pixels, this would cause our adversarial network tend to produce all zero outputs. To address this problem, we zero out those background pixel losses in both ground-truth labels and output labels of the segmentation network. In this way, the discriminator network is forced to focus on reconstructing building pixels while ignoring those background pixels.

IV. Experiments

A. Dataset

1) Massachusetts: Massachusetts buildings dataset proposed by Mnih [1] consists of 151 aerial images of the Boston area. Each image has a size of 1500×1500 pixels and a spatial resolution of 1 m/pixel. The dataset is randomly split into three groups: training set (137 images), validation set (4 images), and test set(10 images) with no overlapping areas. In addition, for easy network training, each training and validation images were cropped into grid patches with a size of 256×256 pixels and those patches with a blank area more than 40 pixels were discarded. After scanning, 4386 training samples and 144 validation samples were generated, respectively. We randomly flip our training images horizontally and vertically and no other augmentation was used.

2) Inria: The Inria aerial image labeling dataset proposed in [32] consists of 360 orthorectified aerial images of size 5000×5000, with a spatial resolution of 0.3 m. It covers a large area of 810 km^2 in 10 different cities and incorporates various urban landscapes and settlements, which makes it valuable for evaluating the generalization power of different models. For a fair comparison, we split the dataset as described in [32], i.e., 155 images for training, 25 images for validation, and the remaining 180 images for testing. Similarly, we cropped the training and validation images into grid patches with a size of 256×256, gathering 62 000 and 10 000 image tiles for training and validation, respectively. No data augmentation was used in our experiments.

TABLE I
Test Accuracies on Massachusetts Test Set With Different α Values

α	Prec	Rec	F1	breakeven
1(*)	**97.72**	92.94	95.27	95.96
+crf	96.71	95.26	95.98	96.00
0.9	96.81	**95.65**	96.23	96.24
0.8	97.14	95.52	**96.32**	**96.40**
0.7	97.10	94.94	96.01	96.15
0.5	96.40	95.38	95.89	95.95

All values are calculated with relaxed parameter ρ set to three, and breakeven score indicates where precision equals recall. * indicates the baseline model when α equal to one and +crf indicates the baseline model with dense CRF.
Boldface indicates best evaluation performance.

B. Experiment Settings

We implement our Building-A-Nets model based on the TensorFlow library. Our segmentation network is trained from scratch using RMSProp algorithm [33], with an initial learning rate and exponential decay set to 1e-3 and 0.995, respectively. We use a weight decay of 1e-4 and a dropout rate of 0.2. Our adversarial network is optimized with Adam [34] algorithm with an initial learning rate set to 8e-5. We divide the learning rate of both networks by 2 every 5 epochs. A Nvidia K80 GPU with 12G memory is used to train our model, and it consumes around 8.7G memory while training. It takes about 31 h and 21 d to train our model on Massachusetts and Inria dataset, respectively.

C. Results on the Massachusetts Dataset

The following three common metrics are used to evaluate the performance of our proposed algorithm:

1) precision and recall scores;
2) breakeven score where precision equals recall; and
3) relaxed F1 score.

We use a relaxed version of these metrics following [1]–[3]. The relaxed recall denotes the fraction of true target pixels that are within ρ pixels of predicted target pixels, whereas the relaxed precision measures the fraction of predicted target pixels that are within ρ pixels of true target pixels. The slack parameter ρ is set to 3.

First, we test the performance of our model with different α values, results are shown in Table I. As can be seen, when the α value is closer to 1 (greater than 0.7), our Building-A-Nets model achieves better performance than the baseline model without the adversarial network. With α set to 0.8, our

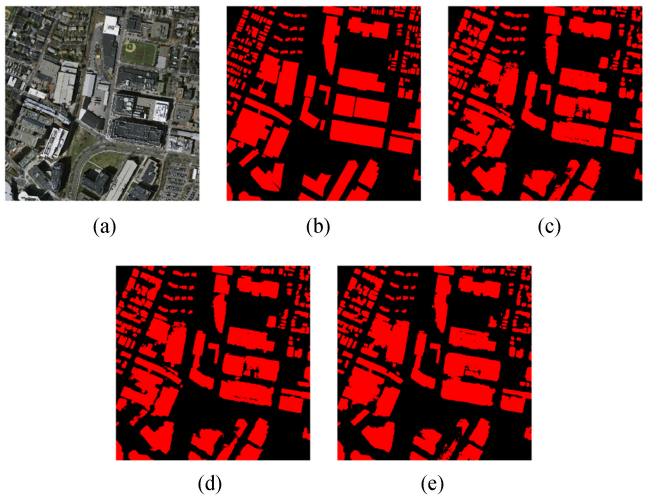


Fig. 3. Examples of building extraction results on Inria aerial image labeling dataset. (a) First column shows input aerial images. (b) Second column shows the ground-truth labels. (c) Third column shows our prediction results.

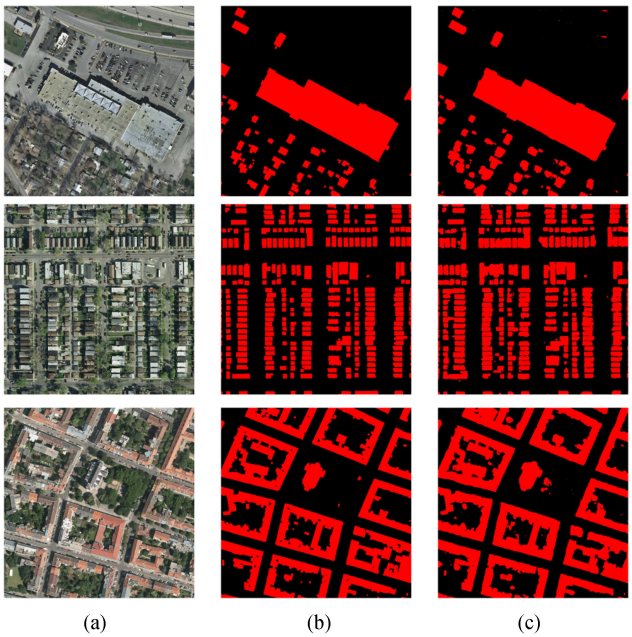


Fig. 4. Examples of building extraction results on the Massachusetts dataset. (a) Input aerial image. (b) Ground truth. (c) Baseline model. (d) Baseline model with dense CRF as postprocessing. (e) Ours.

model gets the best performance, and we will use this configuration in our successive experiments. Evaluation on test set shows a small but consistent improvement of 1.10% and 0.46% for F1 score and relaxed breakeven score ($\rho = 3$), respectively, whereas using dense CRF as postprocessing nearly gets no improvement over the baseline, with the F1 score and relaxed breakeven score ($\rho = 3$) improved by 0.75% and 0.02%, respectively. Fig. 3(c) shows some examples of the prediction result of our baseline model ($\alpha = 0$), Fig. 3(d) shows our baseline model with dense CRF as postprocessing, and Fig. 3(e) shows our proposed Building-A-Nets model with α set to 0.8. As shown in Fig. 4, our baseline model fails to maintain the spatial consistency of the building blocks, as indicated by unstable

TABLE II
TEST ACCURACY OF DIFFERENT MODELS ON THE MASSACHUSETTS DATASET

Model	Breakeven (ρ = 3)	Breakeven (ρ = 0)	Time (s)
Mnih-CNN [1]	92.71	76.61	8.7
Mnih-CNN+CRF [1]	92.82	76.38	26.6
Saito-multi-MA [3]	95.03	78.73	67.7
Saito-multi-MACIS [3]	95.09	78.72	67.8
HF-FCN [5]	96.43	84.24	1.07
Ours (56 layers)	96.40	83.17	**1.01**
Ours (158 layers)	**96.78**	**84.79**	4.38

Boldface indicates best evaluation performance.

building shapes, noises, and missing detection of many building pixels. Numerical results in Table I also prove this finding, where the baseline model gets a high precision score while not as good as for recall value. By adopting adversarial network to account for high-order structural constraints, our model successfully corrects most of the structural errors where our baseline model fails, whereas dense CRF takes no effect on these errors.

Then, with the decrease of α value, our model get better performance, this also indicates that the adversarial network plays a positive role in our model. However, when α value becomes too small, less than 0.5 here, the prediction performance degrades. This is probably because a too small α value will break out the balance between pixelwise classification and high-order structure constraints. When high-order structure information dominate the learning procedure of segmentation network, our model cannot guarantee accurate pixelwise classification, and structure information alone cannot lead to good segmentation.

Moreover, we compared the performance of our method with some state-of-art approaches, including Mnih-CNN+CRF [1], Saito-multi-MA&CIS [3], and HF-FCN [5]. As shown in Table II, our Building-A-Nets model with 56 layers gets superior performances than all comparing methods except the HF-FCN model. Compared with HF-FCN model, our proposed model get comparable accuracy on the relaxed breakeven score ($\rho = 3$) but fall a little behind on standard breakeven score ($\rho = 0$). This could be caused by our FC-DenseNet model, where 56 layers are not powerful enough for this segmentation task. So we conduct another experiment using a deeper segmentation network with 158 layers. Results show that this deeper model got an obvious improvement on all evaluation metrics, and the performance surpasses the state-of-the-art HF-FCN model obviously. To compare the prediction efficiency, we calculate the average processing time on ten test images on a single NVIDIA K80 GPU. Results in Table II show that our model with 158 layers is a lot faster than patch-based models, including Mnih-CNN+CRF and Saito-multi-MA&CIS, but slower than HF-FCN model. This is because our base segmentation network consisting of 158 convolutional layers is quite deeper than VGG-16 net used by HF-FCN model. For a compromise, one can also use our Building-A-Net model with 56 layers for better prediction efficiency while maintaining a suboptimal accuracy.

D. Results on the Inria Dataset

Intersection over Union (IoU) and global accuracy are commonly used to evaluate the performance on Inria aerial labeling

TABLE III
VALIDATION ACCURACY OF DIFFERENT NETWORK DEPTHS ON INRIA AERIAL IMAGE LABELING DATASET

Model	IoU	Acc.
FC-DenseNet (56 layers)	74.64	96.01
Ours (56 layers)	74.75	96.01
FC-DenseNet (158 layers)	77.11	96.45
Ours (158 layers)	**78.73**	**96.71**

Boldface indicates best evaluation performance.

TABLE IV
VALIDATION ACCURACY OF DIFFERENT MODELS ON INRIA AERIAL IMAGE LABELING DATASET

Model	IoU	Acc.	Time(s)
FCN [32]	53.82	92.79	-
MLP [32]	64.67	94.42	-
FCN with VGG-16 [36]	66.21	94.54	-
MLP with VGG-16 [36]	68.17	94.95	-
SegNet (Single-Loss) [36]	72.57	95.66	-
SegNet with mutli-task loss [36]	73.00	95.73	-
2-levels U-Nets + Aug. [37]	74.55	96.05	208.8
Ours (158 layers)	**78.73**	**96.71**	**150.5**

Boldface indicates best evaluation performance.

dataset. Here, we focus our experiments on IoU score because it can take into account both the false alarms and the missing detections, and are commonly used in computer vision field for segmentation evaluation. While global accuracy is not quite suitable for evaluation on Inria dataset because this dataset contains large areas of background pixels [35]. In this paper, we report the IoU score and global accuracy on the overall test set.

We conduct experiments on the Inria aerial image labeling dataset using the same configuration as the Massachusetts dataset, with α set to 0.8. First, we test the performance of different network depths. As shown in Table III, with the increase of network depth, our Building-A-Nets model gets remarkable improvement as indicated by IoU score and global accuracy. One can see that this improvement on the Inria building dataset is a lot more significant than on the Massachusetts dataset, this to some extent implies deeper networks are more suitable for modeling this large-scale dataset. Moreover, by integrating adversarial network, our Building-A-Nets model outperforms FC-DenseNet model with both depth configurations, while deeper models benefit more from the adversarial learning process. Fig. 5 shows some examples of prediction results of the proposed model with 158 layers in three challenging residential areas, covering buildings with various densities, scales, shapes, and surroundings. As can be seen, our proposed model presents a satisfying performance in these challenging areas.

Moreover, we compare the performance of our proposed model with the state-of-art approaches, including several extension of FCN, MLP, SegNet, and U-Net model proposed in [32], [36], and [37]. Table III lists the IoU score and global accuracy of different models on the validation set. As shown in Table IV, our Building-A-Nets model surpasses all comparing methods and achieves a remarkable improvement of 5.6% than the state-of-the-art method on IoU score. It should be noted that no data augmentation was used in our experiments while the state-of-the-art U-Nets [37] model boosts their performance from data augmentation. To compare model efficiency, we calculate the average processing time on the validation set. As shown in Table IV, our Building-A-Nets model only takes 208.8 s to process one 5000×5000 image tile, which is a lot faster than the state-of-the-art U-Nets method [37]. Detailed test performance can be found in the leaderboard at https://project.inria.fr/aerialimagelabeling/leaderboard/.

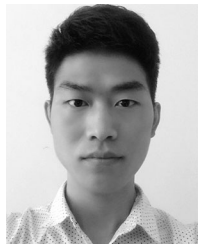
V. CONCLUSION

This paper presents a robust building extraction method for high-resolution remote sensing images. An adversarial network is introduced to learn structural losses and enforce higher order consistencies between segmentation maps and ground truth label maps. The learned structural losses along with the pixelwise classification loss are passed to segmentation network to update the network parameters, with soft weights to balance these two terms of losses. Results on the Massachusetts building dataset and the Inria aerial labeling dataset demonstrate that with adversarial learning, our model can effectively refine the spatial inconsistency of building blocks and boost the extraction performance. We achieved state-of-the-art performance on these two datasets.

REFERENCES

[1] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2013.
[2] 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 J. Photogrammetry Remote Sens.*, vol. 130, pp. 139–149, 2017.
[3] S. Saito, T. Yamashita, and Y. Aoki, "Multiple object extraction from aerial imagery with convolutional neural networks," *Electron. Imag.*, vol. 2016, no. 10, pp. 1–9, 2016.
[4] Z. Zhong, J. Li, W. Cui, and H. Jiang, "Fully convolutional networks for building and road extraction: Preliminary results," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 1591–1594.
[5] T. Zuo, J. Feng, and X. Chen, "HF-FCN: Hierarchically fused fully convolutional network for robust building extraction," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 291–302.
[6] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2011, pp. 109–117.
[7] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr, "Higher order conditional random fields in deep neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 524–540.
[8] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," in *Proc. NIPS Workshop Advers. Train.*, 2016.
[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[10] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation," *Cvpr*, vol. 1, no. 2, p. 5, 2017.
[11] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers Tiramisu: Fully convolutional denseNets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1175–1183.
[12] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1520–1528.
[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[14] F. Visin *et al.*, "ReSeg: A recurrent neural network based model for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2016, pp. 41–48.
[15] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, arXiv:1511.07122.

[16] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1529–1537.
[17] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[19] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, vol. 1, pp. 2261–2269.
[20] B. Triggs and J. J. Verbeek, "Scene segmentation with CRFs learned from partially labeled images," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2008, pp. 1553–1560.
[21] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 670–677.
[22] A. Barbu, "Training an active random field for real-time image denoising," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2451–2462, Nov. 2009.
[23] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 739–746.
[24] R. Mottaghi *et al.*, "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 891–898.
[25] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.
[26] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2014, pp. 2672–2680.
[27] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," 2016, arXiv:1609.03126.
[28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, arXiv:1701.07875.
[29] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017, arXiv:1703.10717.
[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.
[31] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2016, arXiv:1701.00160.
[32] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The Inria aerial image labeling benchmark," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2017, pp. 3226–3229.
[33] T. Tielemanm and G. Hinton, "COURSERA: Neural networks for machine learning," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. Lecture 6.5-RMSProp, 2012.
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, arXiv:1412.6980.
[35] G. Csurka, D. Larlus, and F. Perronnin, "What is a good evaluation measure for semantic segmentation?" in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 32.1–32.11.
[36] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," 2017, arXiv:1709.05932.
[37] A. Khalel and M. El-Saban, "Automatic pixelwise object labeling for aerial imagery using stacked u-nets," 2018, arXiv:1803.04953.



Xiang Li (S'18) received the B.S. degree in remote sensing science and technology from Wuhan University, Wuhan, China in 2014. He is currently working toward the Ph.D. degree at the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China.

His research interests include remote sensing image recognition, smart city data mining, machine learning, and computer vision.

Xiaojing Yao received the Ph.D. degree in cartography and geographical information system from the University of Chinese Academy of Sciences, Beijing, China, in 2017.

She is currently a Research Assistant with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China. Her research interests include spatial data mining, machine learning, and smart city applications.

Yi Fang (M'13) received the B.S. and M.S. degrees in biomedical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in mechanical engineering from Purdue University, West Lafayette, IN, USA, in 2011.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates. His research interests include three-dimensional computer vision and pattern recognition, large-scale visual computing, deep visual computing, deep cross-domain and cross-modality multimedia analysis, and computational structural biology.