Deep Learning Semantic Segmentation for Satellite Imagery

Haoran Zhang

April 22, 2017

Date Started: October 10, 2016
Mentor: Roy Yang

1 Background

Satellite images which taken with ultra high-resolution cameras always have big size. And it is hard to annotate the image manually for analyzing. Then the algorithm about how to extract buildings from the images have been a research focus for many years. Extracting buildings from endless images is helpful for urban planning and management.

2 Previous Art

In the past, researchers used conventional computer vision algorithms to deal with building extraction. Do image enhancement then classify the buildings with combining the texture, tone, context, and shape by fuzzy decision tree. Then delineate the roofs' line by hough transformation [15]. But this method cannot recognize the circular, ring, C, S edges of buildings well. Then researchers developed their method by SVM finding the building patch first then delineate rectangular building, circular building by Circular Hough Transform, and S or C by refining them [20].

3 Learning by Convolution Neural Network

These attempts are limited by the difficulties of designing features manually which cannot adjust to too many patterns. There are a lot of buildings in more complicated shape not merely include S or C. In the real world, our human beings' discrimination is predicated upon our knowledge and context analysis which cannot be implemented by trivial algorithm without learning.

With the development of GPU and deep neural network, researchers take advantages of the state of the art techniques to construct new extractors. Deep

neural network has the capability to learn knowledge from huge flow of data both semantically and spatially. And the semantic problem is the most difficult one in the past like that it is hard for SVM to tell parking lots from buildings. However, the state of the art Inception-ResNet-v2 [22] can even tell Alaskan Malamute from Siberian Husky with high confidence. So it is better off leveraging the learning ability of DNN to process satellite images.

Specifically, in convolutional neural network, parameters can not only learn features layers by layers automatically but also can it learn invariance to rotation, scale, translation [11].

Traditional convolution kernels are designed in a prior way for creating textons [1], but in CNNs [13], the parameters in the kernels are treated as learnable layers which can learn hierarchies of features [12]. Take an image as a example. An image is a (height x width x channel) input matrix, feed into the first layer. The first layer represents the details like edges or curves, the second layer detect motifs assembling by the edges , the third layer learn the combinations of the motifs until the last layer can tell the difference between cats and dogs. All parameters in the layers are learnt by back propagation algorithm [7].

Moreover, the kernel size is a tricky hyperparameter. There are (1 x 1) first introduced by Network in Network [14], (3 x 3), (5 x 5), (7 x 7), (m x 1) [24], (m x n), Multi scale convolution, atrous hole. Google Brain use reinforcement learning [27] to search better topological structure for CNN, and find that using smaller kernels in shallower layers, (m x 1) or (m x n) in middle layers and bigger kernels in deeper layers is good. GoogLeNet [23] [24] [22] use an Inception structure containing different scale of kernels which is helpful to cope with the large-scale variance especially in remote imagery [17]. Experiments shows that the learner can learn more efficient features by this way. Furthermore, ResNet [9] use skip connection to train a even 1000 layers neural networks and solve the gradient explosion and vanishing by identity map [10].

Researchers find that it is more effective and efficient to achieve better performance by deepen the networks rather than merely widen them. And some experiments [2] [25] show the significance that the the neural network for image processing have to be both deep and convolutional.

However, the more semantic meaning it learn, the more spatial location it forget. Traditional CNN classifier which only present a softmax probability to a particular object works amazingly great but forget all of the spatial information. So it cannot be use for semantic segmentation straightforwardly. Fully Convolutional Networks [16] is the first one to solve semantic segmentation by using a end-to-end deep neural networks and combine the information of both semanteme and location.

4 Transfer learning and Semantic Segmentation

One of the strategies of FCN [16] is to transform the fully connected layers into convolution layers so that the prior NN architecture can produce a heat map which turn semantic segmentation into a classification problem. In other

words, reinterpret the configuration of that layers, such as (25088 x 4096) to (512 x 7 x 7 x 4096). This strategy enable the neural network to leverage prior arts so that it only need to fine-tune the fully convolutional layer. The result of fcn shows in Figure 2.

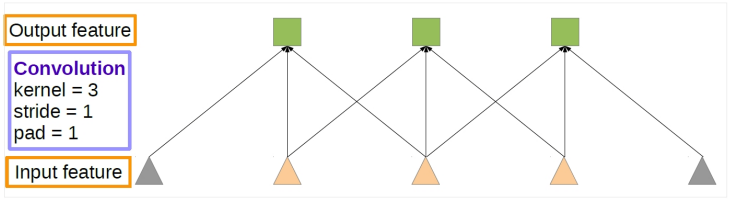
That is also a type of transfer learning which can explain the reason why it works with less data set and without training the shallower layers.

Actually, the pretrained model has learnt parameters from large scale data like ImageNet. These parameters learn transferable features which can be seen in different task. So it is wasteful to train a semantic segmentation from scratch. Moreover, training from scratch with little data tends to overfitting, especially when the segmented train image is hard to annotate.

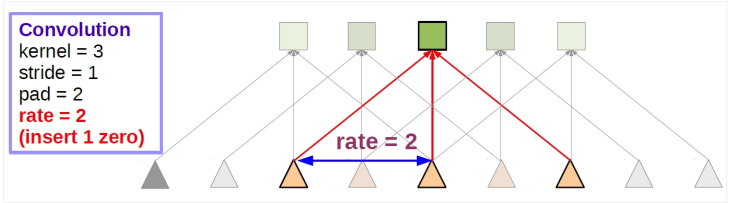
In order to deal with the trade-off between semantic and spatial. FCN uses skip training to fulfill different depth layers' features. This also inspires successors' works.

Then DeepLab-v1 [3] use more elegant atrous convolution to preserve more spatial information without retraining these layers. This powerful tool compensates the drawback of FCN whose heat map is too coarse to do dense prediction well. It allows convolution neural network to enlarge the receptive field for producing denser heat map. At the end of the network, deeplab-v1 also append a Conditional Random Field layer to enhance the ultimate performance.

The mechanism of atrous hole shows in Figure1.



(a) Sparse feature extraction



(b) Dense feature extraction

Figure 1: atrous hole

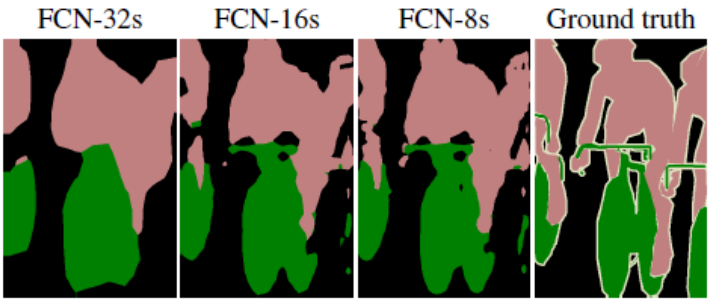


Figure 2: fully convolutional network

5 Data Preprocessing

To train semantic segmentation in a neural network way, the primary prerequisite is to annotate the data set in a supervising learning format, namely, to annotate the original image semantic block in different labels. Then encode it into one-hot dealing with the linearly non-separable problem.

Aiming to automatically draw building polygons in satellite images. This project propose a solution by constructing a end-to-end neural network.

This project support a script to annotate the image by hand in polygons. Users will not spend too much time on annotating simple object. The result shows in Figure 3.

Figure 3: mannuly annotated image

For training the neural network, the data downloads from kaggle dstl with 10 categories. This project only consider buildings.

Buildings are described in the form of polygons and multipolygons, which are simply a list of polygons.

Our data set only use 10 images for training. In order to gain fully training of the networks, we also do some data preprocessing steps.

Figure 4: dstl

5.1 Data Cropping and Augmentation

The shape of the original image is about (3348 x 3403 x 3), then crop it with stride 100, then every image (100 x 100 x channel),10890 images totally.

The images are not precise (3348 x 3403 x 3), so naturally the cropping can start from top left, top right, lower left, lower right, which turn 10890 to 43560.

Then remove all the smaller images without buildings only for simplicity. Shuffle the whole dataset and split it in train:validation:test = 8:1:1. Namely, 14568:1821:1821.

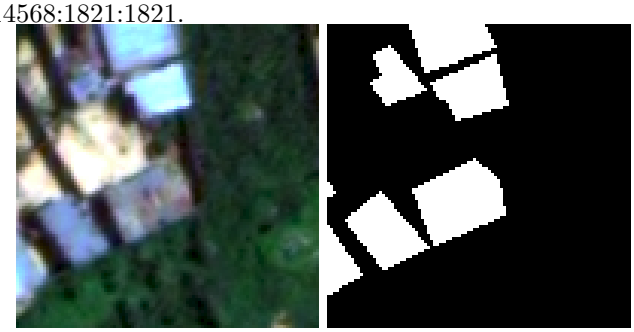


Figure 5: cropped image

5.2 Prepare for train

The network constructs on the tensorflow platform. For ease of fast iteration, we encode it into binary format, namely tfrecords. Then standardize the whole data with subtracting the mean value of pixel for each channel, namely, mean normalization. It is not necessary to do variance normalization, because different features always share similar variances to each other in natural images [18].

6 Method

The prototype of the whole project is based on DeepLab-v1 and implemented on tensorflow.This section will show the structure of the project.

6.1 Weight initialization

In the past, researchers preferred to initialize the bias to be 0 and weight by uniform distribution:

$$W_{ij} \sim U\left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right] \tag{1}$$

And neural network training will not fall into local optima as people thought with low-dimension angle in the past. In fact, The higher the dimension, the less the local optimal solution, the more the saddle point, the higher the difficulty of optimization, the lower the optimization error[5]. With carefully observing how neural networks saturate, Xavier initializer[6] brings substantially faster convergence:

$$W_{ij} \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right] \tag{2}$$

6.2 model

To begin with, we use a pretrained VGG-19[21] neural network, then reinterpret the fully connected layers to convolution layers like FCN does, followed by three transpose convolutions layers for upsampling.

In the near future, we will replace the last six convolution layers to atrous convolution layers aiming to denser prediction.

The whole configuration is:

conv	relu	conv	relu	pool
conv	relu	conv	relu	pool
conv	relu	conv	relu	conv
relu	conv	relu	pool	
conv	relu	conv	relu	conv
relu	conv	relu	pool	
conv	relu	conv	relu	conv
relu	conv	relu	avg_pool	

Table 1: pretrained VGG-19

then train new layers on the top of it:

transpose conv	relu	transpose conv	relu	transpose conv	relu

Table 2: new layers

6.3 fuse loss optimizer

In the upsampling process, fuse operation combine different depth information, and it is very easy to implement, add the result of transpose convolution layers to shallower pooling layers, then pass to next transpose convolution layers.

Inspired by Mask RCNN[8], this binary classification task only use sigmoid and cross entropy instead of other normalized function.

By trying different optimizer, the NN converge really faster with Adam optimizer but slower when using stochastic gradient descent with weight decay.

Hyper parameter: batch size = 2,learning rate = 1e-4, dropout = 0.85.

7 Result

Although dstl dataset is completely different from ImageNet, it has gained amazingly great result on it.

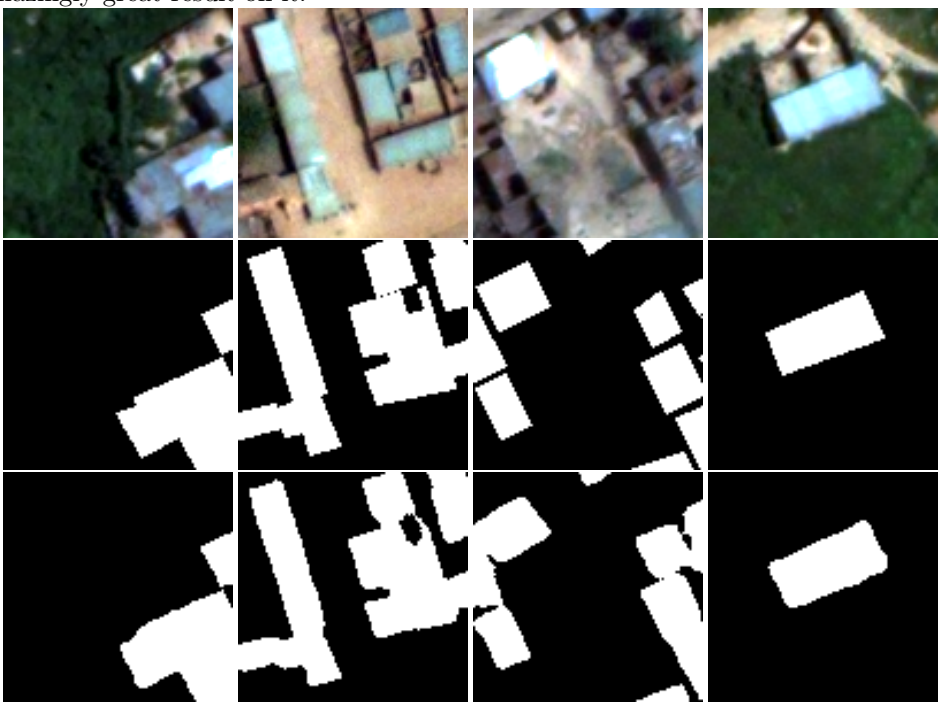


Figure 6: top:input image. middle:ground truth. bottom:prediction

But there is still some problems: It is hard to detect adjacent small objects

like Figure 8 right.And it will ignore the buildings when there is only an angle of it appeared in the image like Figure 8 middle.

Figure 7: bad prediction. top:input image. middle:ground truth. bottom:prediction

Moreover, there are some prediction which are even better than the ground truth. These prediction depict more details of the buildings than the ground truth.

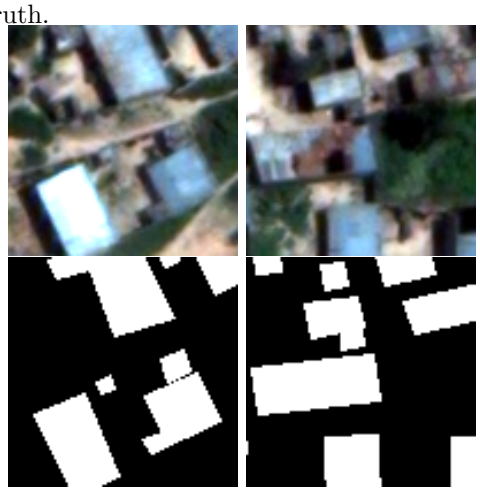


Figure 8: Great prediction. top:input image. middle:ground truth. bottom:prediction

8 Future works

Future works will more concentrate on remote sensing image task itself and to solve the problems found so far.

Nest step is to replace deeper conv with atrous for bigger feature map. This technology has been widely used in many kinds of pixel-wise prediction task include semantic segmentation.

It is helpful to replace the backbone network to the state of the art one. Even in different types of transfer learning task, better backbone always means better performance.

For small building segmentation, this prototype model use none of them, so it is hopeful to get a better small object result by leveraging them. And many of them is easy to implement and of low computational complexity.

More specifically, YOLOv2 [19] simply add a passthrough layer, the shallow feature map (resolution of 26 * 26) connected to the deep feature map. By linking high and low resolution feature map, it work great on small object and naturally to transfer to semantic segmentation task. And deeplabv2 introduce atrous pyramid to combine multi scale information.

More complex ideas is to train paralleled network for complementary task. Other researchers trained HNED [26] network for edge detection then combine semantic one and edged one by domain transformation. Or train a 3 channel network include classification, object detection and mask [8].

Then I want to implement a counting network in order to differentiate every building instead of a big block. But this puts a higher demand to the dataset.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[2] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[5] Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Mathematics*, 111(6 Pt 1):2475–2485, 2014.

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask r-cnn. 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[15] Zhengjun Liu, Shiyong Cui, and Qin Yan. Building extraction from high resolution satellite imagery based on multi-scale image segmentation and model matching. In *Earth Observation and Remote Sensing Applications, 2008. EORSA 2008. International Workshop on*, pages 1–7. IEEE, 2008.

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[17] Zhong Ma, Zhuping Wang, Congxin Liu, and Xiangzeng Liu. Satellite imagery classification based on deep convolution network. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(6):1113–1117, 2016.

[18] Andrew Ng, Jiquan Ngiam, Chuan Yu Foo, Yifan Mai, and Caroline Suen. Ufldl tutorial pca whitening, 2012.

[19] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. 2016.

[20] D Koc San and M Turker. Building extraction from high resolution satellite images using hough transform. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, 38(Part 8), 2010.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[23] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[25] Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*, 2016.

[26] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. pages 1395–1403, 2015.

[27] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.