Learning with Detail and Morphological Refinement for Satellite Image Analysis based on Convolutional Neural Network

Guan-Ting Lin* and Yung-I Yang*
Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, ROC
Department of Electrical Engineering, National Tsing Hua University, ROC
E-mail :{ilovevictor0424, blaou84312} @gmail.com
* = contribute equally to this work

***Abstract*—we present a multiple stages way to analyze the content of satellite images. Our methodology is divided into three major steps. First, Convolutional Neural Networks (CNNs) for semantic segmentation between buildings and nature scene. Then, the output semantic image would be refined in morphology way. In the last stage, Depth-First Search (DFS) algorithm is used for buildings counting. The experimental results show that refined images have smother boundaries. Base on the output images, we can count buildings using DFS algorithm accurately by refined image.**

I. INTRODUCTION

Satellite images are images collected by imaging satellites. We can use them for seeing all kinds of natural scenes and artificial structures [5]. The segmentation analysis of satellite images by image processing methods can distinguish between different objects, one common way for object recognition is deep learning techniques like Convolutional Neural Networks (CNN).

After image segmentation, our purpose is to find connected component in binary image for buildings counting by Depth-First Search (DFS) algorithm, and we want to evaluate how many buildings each satellite image has.

This research can be applied on various types of consumer electronics like smart phone or computer for intelligent assessment of image content.

II. PROPOSED METHOD

We combine the CNN with traditional and morphological method. We use CNN as a role of a predictor that classify every pixel into two classes with morphological refinement to make boundary smooth.

A. *Convolutional neural networks for semantic segmentation*

We adopt the architecture proposed by [1] which is a trainable encoder-decoder convolutional neural network for semantic pixel-wise segmentation, see Figure 1. The basis of [1] is mainly consist of VGG network [2], the encoder network is topologically identical to the former 13 convolutional layers in VGG16 and then cascaded with a trainable decoder. The function of the decoder is to upsample the aggregated features extracted by encoder to original input resolution feature maps for pixel-wise classification. The major differences between [1] and [3] are in the decode stage. [3] use convolution transpose to upsample the low-resolution feature maps into original resolution, which makes it memory and computational intensive, while [1] record the location information when performing max-pooling in the encoder stage and directly upsample the feature maps according to this information. By this way can reduce the number weights and inference time.

The Inria Aerial Image Labeling Dataset provides pixel level building and non-building annotations of satellite images in the size of 5000 by 5000.

Due to the GPU memory limitations, our training system can accept at large 500 by 500 images with mini-batch size of 1. As a result, we have to take effort on the dimensionality of training data. Naively resize the original ground-truth into a smaller size would cause degraded result due to loss of detailed information. To retain the detail information and downscale the dimensionality of images, we cut the original with 10 portions in each side. This would make one original 5000 by 5000 image becomes one-hundred images with the size of 500 by 500. This augmentation method not only makes the training system converge well but also enlarge the scale of dataset in 100 times.

Initialization and parameters: We re-initialize weights in the last convolutional layer released by [1] with our augmented database with base learning rate of 0.001 and batch size=3 with 20k iterations. After training with larger batch size, we scale down the batch size into 1 and smaller learning rate with 0.00001 and take 60k iterations to make model converge. Furthermore, we make the learning rate of the last convolution layer ten times compared to others.

B. Image Refinement with morphology and Buildings counting

Morphological opening is the combination of two basic operations of dilation and erosion in morphology. It will remove single white pixels and small white clusters, then make the boundaries of buildings in the image be smoother, see Figure 3.

We use DFS algorithm to perform building counting. This algorithm will scan the input image from left to right and top to bottom, for any pixels encountered that belongs to building, we add them to one stack for recording. In order to fill a connected component, we use square structuring element for 8-connected components.

For each pixel in the stack, we pop first location off the stack, and search for the nearest 8-connected pixels to this pixel. Once we find 8-connected pixels, we will check: 1) if 8-connected pixels in this list out of boundary of the image matrix, 2) if 8-connected pixels have been visited and 3) if 8-connected pixels equal to zero. If so, we delete that location of

8-connected pixels. Then go back to step that we pop first location off the stack, and check the requirement iteratively.

Until the stack is empty, we have found those pixels that are in an entire region, and we can form a matrix called Connected Component Image to record the data information

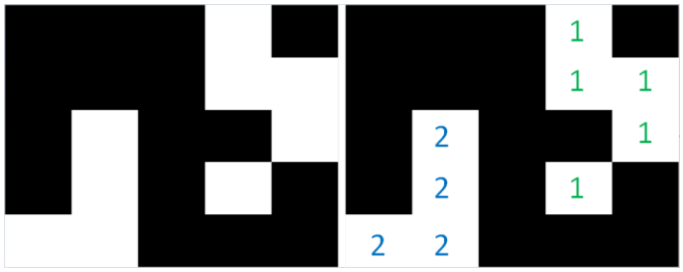

Figure 1. Input image(left) and connected component image(right). These two images both have pixels equal to zero(black) and equal to one(black). Different regions of connected components have different marked numbers on them.

III. EXPERIMENTS

A. *Learning with detail vs. naively resizing*

Some segmentation examples in the testing part of the databases are shown in Figure 2. We can see that naively resize the input resolution into a smaller one will lose the detailed features that makes network hard to learn makes the boundary of the buildings discontinuous, while model trained with our augmented database can get a smooth and integral output.

B. *Image Refinement*

We assume the smallest house size as kernel (in this project we set kernel as 9*9 pixels), and the refined image(RI) is the opening of input by kernel, is defined as in (1).

$$RI = (input \bigcirc kernel) = (input \ominus kernel) \oplus kernel \quad (1)$$

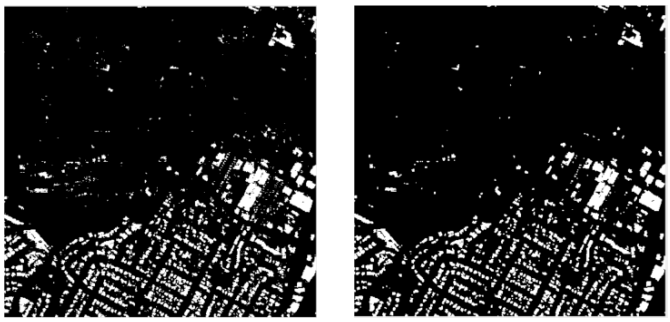

Figure 3. Comparison of Non-refinement and After-refinement. The left column is the original semantic image of CNN and the right column is our refinement result based on this image.

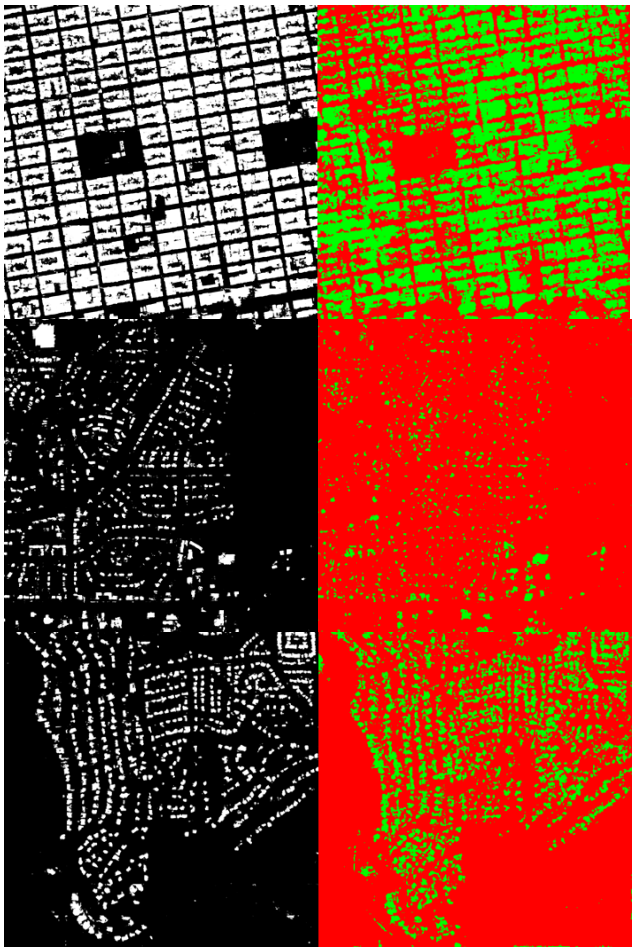

Figure 2. Comparison of our learning strategy and naïve resizing. The left column is the segmentation output of our method and the right column is the naïve one.

IV. CONCLUSIONS

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet. "A deep convolutional encoder-decoder architecture for image segmentation," arXiv preprint arXiv:1511.00561, 2015

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," arXiv preprint arXiv:1411.4038, 2015

[4] M.-S. Chen and K.G. Shin. "Depth-First Search Approach for Fault-Tolerant Routing in Hypercube Multicomputers," IEEE Computer Society, vol. 1, pp. 152-159, Apr. 1990.

[5] L.Soh, and C.Tsatsoulis. "Segmentation of satellite imagery of natural scenes using data mining," IEEE Transactions on Geoscience and Remote Sensing, vol. 37, pp. 1086-1099, Mar. 1999.