DAG of convolutional networks for semantic labeling

Alexandre Boulch

ONERA, *The French Aerospace Lab*, F-91761 Palaiseau, France

Abstract

This paper presents a method for labeling of urban area orthoimages. It is based on two levels of classification using deep neural networks. The objective is to define which part of the image can be considered as difficult and perform specific labeling on these areas. We evaluate the method on the ISPRS 2D semantic labeling challenge.

1 Introduction

Semantic labeling of urban areas a key step for applications such as 3D city modeling, physical simulations (risk of traffic jams, transportation time estimation...), change detection or geographical database update. Aerial photography gives a wide and rich view of a urban area as most of the city structure are recognizable (roads, buildings, vegetation...). Pixel labeling of such images is a very interesting and challenging task: they present difficulties coming from occlusions due to the view angle of the camera or a great variety of shapes (among buildings, cars and miscellaneous structures).

The possibility of using deep neural networks [5, 2, 9, 7] in the context of urban labeling [8] is now possible due to the availability of large and labeled dataset.

This technical report presents a method for semantic labeling of orthorectified aerial images. This method makes an intensive use of deep neural networks (section 2.2). We first create patches using superpixels that will feed a directed acyclic graph (DAG) of neural networks. The classification itself is performed by a DAG of classification networks. The first level is a one versus all classification for each class. Depending on the results of this step, the final classification is obtained by a different network. We train and evaluate the method on the ISPRS 2D semantic labeling challenge.

2 Method

2.1 Input data

The first stage of this classification method is to produce images patches to feed neural networks. The data provided in the benchmark is a high resolution color image (HR) and digital surface model (DSM). Our tree takes a RGB input. We produce a composite image with 3 channels : the red channel of the HR image, the HR image as gray level, the normalized DSM (nDSM) generated from the DSM in [4].

As the classification step takes images as input, we generate patches from the composite image. In order to take into account the frontiers in the original image, we compute superpixels with the SLIC algorithm [1]. The parameters of the superpixel extraction are tuned in order to get around 100 pixels by superpixel.

The patches are 128 × 128 subimages. One patch is generated for each superpixel. The figure 1 presents the patch creation process.

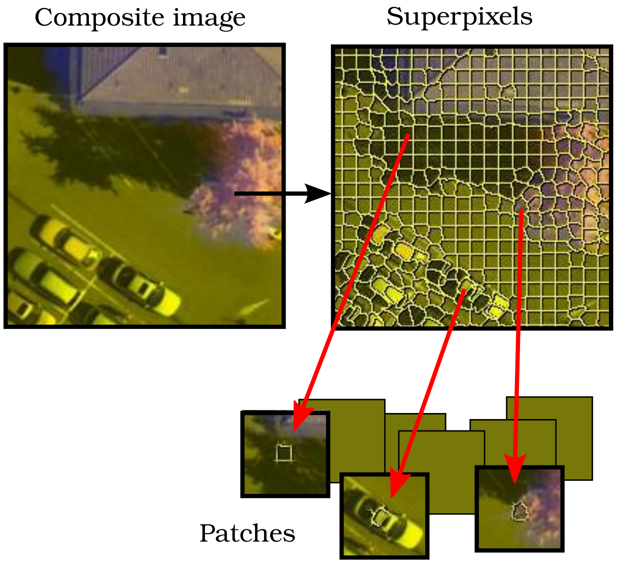

Figure 1: Patch generation from composite image

2.2 Classification framework

The classification part is a two steps algorithm that intends to mimic the behaviour of human for some difficult classification.

Let's take an example. We have an image of an object and we want to determine what is this object. If it is not a difficult image, we can directly give the label, that is the easy case. Now, if is a hard example: noisy, occluded or very distorted or as an example a minautor. A way to reduce the uncertainty of our labeling is to first determine a list of possible labels. That could be {human, bull}. In a second step determine the label, given the reduce list of possibilities. Is the minotaur a human or a bull ?

Our classification for urban data is based on that principle. The figure 2 gives an overview of the method. As in the example there are two steps. First, the patch feeds K classifier 1vsAll, where K is the number of possible classes. Each classifier Class$_k$ answers the question *"is the patch of class C_k ?"* From these answer, we build a vector $X = (x_1, \ldots, x_K) \in \{0,1\}^K$, where $x_i = 1$ if the patch can belong to class i, 0 otherwise.

If $\sum x_i = 1$, only one class is kept and the label l is:

$$l = \underset{i}{\operatorname{argmax}}\, x_i \tag{1}$$

that is the easy case of the example, we know directly the label of our image.

For the other cases, there is a second step, a specific classifier has been trained for the different X. The image is given to the corresponding network and we get the final label.

This approach, particularly for the first step of the classification, shares common points with the training of multiclass SVMs [6]. One way to train multiclass SMV is to train K 1vsAll SVMs. The label is then given by the SVM with the greatest margin.

2.3 Classifiers

Each classifier is a AlexNet [7] network. It contains eight learned layers : five convolutions and three fully-connected. It was proven very efficient on ImageNet [10] and is very flexible and adaptive.

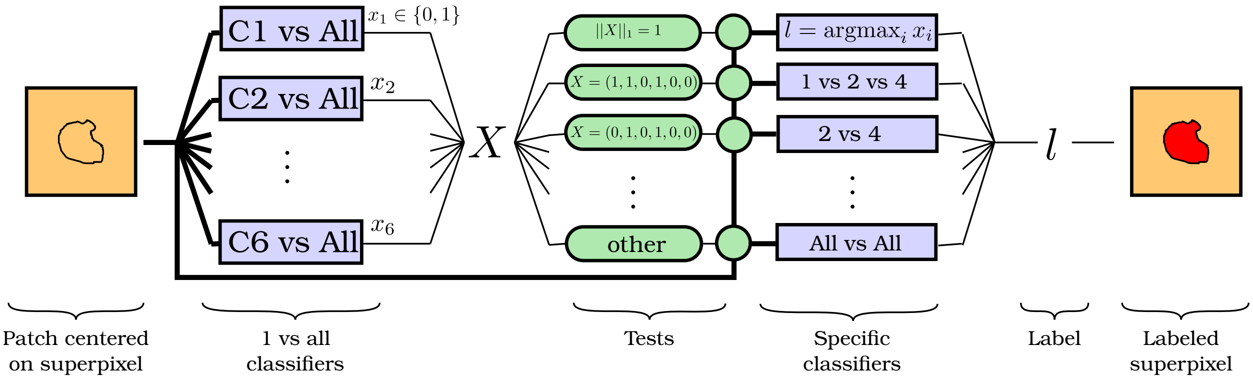

Figure 2: Classification framework

In the first step of the classification, each Class$_k$ classifier is trained on 150000 patches randomly chosen in training set. The X vectors are computed on all the training set. Then, the second set of classifiers are trained on all the corresponding patches.

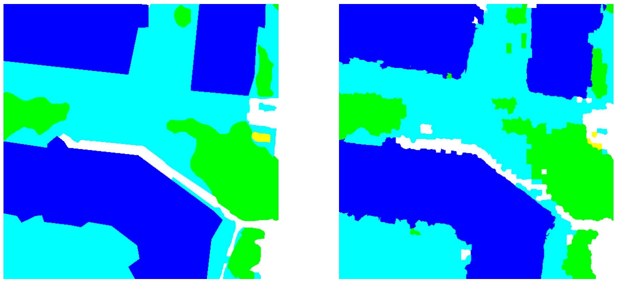

Figure 3: Ground truth versus estimated labels.

2.4 Experimentations

We do not provide quantitative score on the training set because we did not create a validation set. Particularly, the second classification step uses all the available data for training.

Figure 3 presents the ground truth and the estimated labels on a detail of the tile *area1* of ISPRS dataset. The main differences are around the frontiers between labels. The estimated labels are less regular, mainly due the superpixel segmentation.

For a better sense of the performance of the proposed method, the figure 4 shows these differences. The blue and green pixels are well labeled, the red and orange are not. Blue and red pixels are the pixels which labels were estimated after first classification step, they were considered as easy. Note that the red are this point completely lost for the algorithm. Orange and green pixels are labeled during the second step.

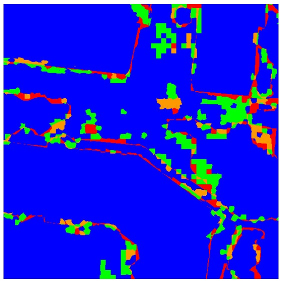

Figure 4: Difference between ground truth and estimated labels.

3 Conclusion

We presented a method for urban classification from aerial orthoimages. We used a two level classification scheme. The first layer computes the possible labels using one against all trained convolutional networks. The second layer performs a specialized classification.

Acknowledgment The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [3] : http://www. ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[3] M. Cramer. The DGPF test on digital aerial cam- era evaluation, overview and test design. *Photogrammetrie, Fernerkundung, Geoinformation*, 82:2:73, 2010.

[4] M. Gerke. Use of the stair vision library within the ISPRS 2d semantic labeling benchmark (Vaihingen). *Technical report, ITC, Univ. of Twente*, 2015.

[5] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[6] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] Adrien Lagrange, Bertrand Le Saux, Anne Beaupère, Alexandre Boulch, Adrien Chan-Hon-Tong, Stéphane Herbin, Hicham Randrianarivo, and Marin Ferecatu. Benchmarking classification of earth-observation data: from learning explicit features to convolutional networks. *Proc. of IGARSS'2015, Milano, Italy*, 2015.

[9] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.