Convolutional Neural Networks for Semantic Labeling

Adrien Lagrange and Bertrand Le Saux
Onera – The French Aerospace Lab, F-91761 Palaiseau, France

July 27, 2015

Abstract

In this paper, we describe the approach for semantic labeling of aerial images we use in our entries to the ISPRS 2D semantic labeling challenge. The proposed method performs super-pixel segmentation, trains deep Convolutional Neural Networks (CNNs) in order to generate features which are then used as inputs of a Support-Vector Machine. We validate the method on multisource data: photo with infra-red channel and corresponding elevation.

1 Introduction

The study of urban centers using Earth-Observation (EO) data has a lot of potential users and applications, from urban management to flow monitoring, and in the meantime offers great challenges: numerous and diverse semantic classes, occlusions or bizarre geometries due to the image-capture angle and the ortho-rectification. Semantic labeling consists in automatically building maps of geolocalized semantic classes. With the advent of large, labeled dataset it is now possible to train deep networks for example to detect roads using restricted Boltzmann machines and cross-entropy-based neural networks [7] or classify land cover in hyperspectral data based on convolutional networks [10].

Our approach, that we describe precisely in section 2 benefits from 4 assets: superpixel segmentation of the images for introducing spatial constraints, multiscale and multisource data preparation for representing what appears at a given location, deep convolutional networks for processing data patches and extracting intermediate-level features and finally support-vector machine (SVM) for data fusion and final classification. With respect to the works of [9] and [5], we do train the intermediate layers of the CNNs on the training data in a feed-forward manner. The work of [8] uses also CNNs and is the most similar to ours. It differs in the fact that we do not use conditional random fields for post-regularization, relying on the contrary on pre-processing with the superpixels for providing the precise spatial structure of observed area. We also use only CNNs, instead of combining them with Random Forest classifiers to improve the performance. Moreover, we support the idea that by using larger multi-scale information, we are more able to model the context around objects in the image.

2 Our approach for the ISPRS semantic labeling challenge

2.1 Approach

The main feature of our approach is the use of CNNs topped with SVMs for classification. They get patches of pixels extracted from the images as inputs at both training and classification times. Before describing the CNNs, we first explain how we generate these inputs.

Superpixel segmentation We first segment ortho-images using the SLIC (Simple Linear Iterative Clustering [1]) method with the implementation of the VLFeat toolbox [12]. This allows to generate coherent regions at sub-object level.

Patches used to feed the CNNs will then be extracted around the superpixel centroïd, and the class estimated by the algorithm will be assigned to the whole superpixel.

Multiple scale and multi source data Our CNNs use $32 * 32$-sized 3-channel patches as input. For each superpixel, we generate a first $32 * 32$ patch at full resolution (which is roughly the size of a car) and a $124 * 124$ patch (which is roughly the size of a house or a car in context) that we downsize to $32 * 32$. We also build a composite image using the Digital Surface Model (DSM) from the original benchmark data, the normalized DSM (nDSM) provided with one of the baselines of the benchmark [3] and a Normalized Difference Vegetation Index (NDVI) computed using the Infrared (I) and Red (R) channels of the ortho-photo according to the formula: $NDVI = (I - R)/(I + R)$. From this composite image we extract $32 * 32$ patches. As a result, for each location defined by a superpixel, we get 3 patches at multiple scale / multiple data (ms/md).

Convolutional Neural Networks We used two different network architectures that have already been proven efficient on image datasets such as CIFAR [4]: LeNet and Network-in-Network (NiN), as implemented in the MatConvNet library [11].

- The LeNet network is made of three convolutional layers each followed by a relu and pooling layers, then one more convolutional layer followed only by a relu, and finally two fully-connected layers and a softmax.
- The NiN network [6] implements a more complex structure which include three convolutional layers each followed by two fully-connected layers and then a final fully-connected layer and a softmax.

Although we could have used these networks as is, we chose to use them as feature extractors generated by the layer before the softmax one. For the NiN architecture, we had to add a second fully-connected layer at this stage to be able to generate usable vector outputs. The network parameters are learned using patches extracted from the training set, along with their respective class. We used mean substraction, contrast augmentation and data whitening for preparing the network inputs. For each type of network, we train three CNNs in parallel: one for each scale and one for patches from the composite image.

Support-Vector Machine The final classifier is a linear SVM trained after performing a grid search to tune the SVM parameters. More precisely, we train six SVMs corresponding to our six classes to proceed as a one-vs-all manner. Each SVM generates a soft-score map and from these six maps, we apply a simple max operation to select the predicted class. We form the inputs of the SVM classifier by concatenating the intermediate-layer features generated by the CNNs. Thus, the SVM performs both the data fusion of various networks (i.e. various data) and the classification.

2.2 Entries in the benchmark

Table 1: Description of the benchmark entries: various CNNs (LeNet or NiN) for various input data ($32 * 32$ orthophoto patches or multiple scale / multiple data (ms/md).

Entry name	*Network-type*	*Inputs*	*Description*
ONE-1	NiN	32x32	1 NiN network
ONE-2	LeNet	ms/md	3 LeNet networks
ONE-3	NiN	ms/md	3 NiN networks
ONE-4	LeNet+NiN	ms/md	3 LeNet networks and 3 NiN networks

We submit 4 entries to the benchmark, which are summarized in table 1. We have a single-scale NiN network, the two flavors of CNN (LeNet and NiN) on multiscale, multiple data, and a last entry that tries to make the best of both architectures by combining them. Table 2 shows the performances obtained by these approaches using cross-validation on the part of the dataset for which a ground-truth was provided. It shows competitive results with the previous entries of the benchmark, and a small advantage to NiN-based classifiers when cars (i.e. small objects) are considered.

Table 2: F1 measures, overall accuracy and Cohen's Kappa coefficient of NiN-network or LeNet-network intermediate-layer features and SVM for various input data.

Network	*Inputs*	Imp. surf	Building	Low veg.	Tree	Car	Overall acc.	kappa
NiN	32x32	87.00	88.84	78.40	90.40	59.62	86.30	81.76
LeNet	ms/md	91.41	94.61	83.04	91.25	58.94	90.07	86.69
NiN	ms/md	90.84	93.12	83.51	91.35	71.80	89.82	86.43
LeNet+NiN	ms/md	90.94	93.03	83.43	91.37	71.97	89.85	86.46

3 Concluding remarks

We presented an approach for semantic labeling that combines superpixel segmentation for discovering the underlying structure of the image and convolutional networks topped with SVM for intermediate-level feature extraction, classification and data fusion. This framework is quite straightforward and can accommodate various kinds of input data and output classes: thus it can be a competitive solution to generic semantic labeling.

Acknowledgment

The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [2] : `http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html`.

References

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

[2] M. Cramer. The dgpf test on digital aerial camera evaluation – overview and test design. *Photogrammetrie – Fernerkundung – Geoinformation*, 2:73–82, 2010.

[3] M. Gerke. Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen). Technical report, ITC, Univ. of Twente, 2015.

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Univ. of Toronto, 2009.

[5] 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. In *Proc. of IGARSS'2015*, Milano, Italy, 2015.

[6] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[7] V. Mnih and G. Hinton. Learning to detect roads in high-resolution aerial images. In *Proc. of Eur. Conf. Comp. Vis.*, 2010.

[8] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. Van-Den Hengel. Effective semantic pixel labelling with convolutional networks and conditional random fields. In *Proc. of CVPRw/Earth-Vision*, Boston, MA, 2015.

[9] O. Penatti, K. Nogueira, and J. Dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Proc. of CVPRw/Earth-Vision*, Boston, MA, 2015.

[10] A. Romero, C. Gatta, and G. Camps-Valls. Unsupervised deep feature extraction of hyperspectral images. In *Proc. of WHISPERS*, 2014.

[11] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. *CoRR*, abs/1412.4564, 2014.

[12] Andrea Vedaldi and Brian Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.