Efficient Convolutional Patch Networks for Scene Understanding

Clemens-Alexander Brust, Sven Sickert, Marcel Simon, Erik Rodner and Joachim Denzler
Computer Vision Group, Friedrich Schiller University of Jena, Jena, Germany
Project page and open source code: http://cvjena.github.io/cn24/

Abstract

In this paper, we present convolutional patch networks, which are convolutional (neural) networks (CNN) learned to distinguish different image patches and which can be used for pixel-wise labeling. We show how to easily learn spatial priors for certain categories jointly with their appearance. Experiments for urban scene understanding demonstrate state-of-the-art results on the LabelMeFacade dataset. Our approach is implemented as a new CNN framework especially designed for semantic segmentation with fully-convolutional architectures.

In the last years, the revival of convolutional (neural) networks (CNN) [5] has led to a breakthrough in computer vision and visual recognition. While the majority of works focuses on applying these techniques for object classification tasks, there is another field where CNNs can be really useful: semantic segmentation, *i.e.*, assigning a class label to each pixel in an image.

In this paper, we show how to learn spatial priors during CNN training, because some classes appear more frequently in some areas of an image. In general, predicting the label of a single pixel requires a large receptive field to incorporate as much context information as possible. We avoid this by incorporating absolute position information in a layer of the CNN as additional input. Urban scene understanding features a number of categories that need to be distinguished, such as buildings, cars, sidewalks, etc. We obtain state-of-the-art performance in this domain on the LabelMeFacade dataset [4].

Architecture and CNN training Convolutional (neural) networks (CNNs) [5] are feed forward neural networks, which concatenate several layers of different types with convolutional layers playing a key role. The main idea is that the whole classification pipeline consists of one model, which can be jointly optimized during training.

The goal of our network is to predict the object category for every single pixel in an image. The CNN architecture is completely described in [1]. However, in addition to [1], we implemented a fully-convolutional version [6] of it which is mathematically equivalent, but allows for fast prediction. We still train the network in a patch-wise manner, since preliminary experiments showed that training the network in a fully-convolutional manner (batches for gradient computation are comprised of full images only) resulted in slower (wall time) convergence and ultimately a less accurate network, in contrast to the results of [6] on other datasets. With image-based gradient batches, the model only learned to distinguish between the four most common classes. This may be due to our relatively small dataset resulting in a reduced randomization during optimization, altough we try to introduce more randomness by using spatial loss sampling as detailed in [6].

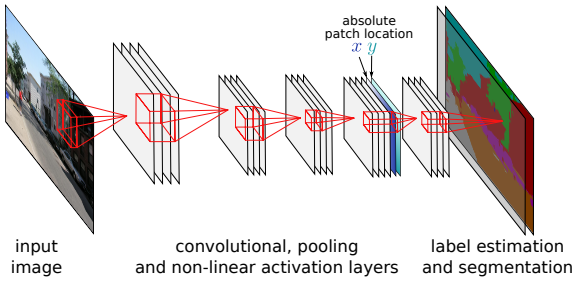

Figure 2. Basic outline of our CNN architecture. The x and y feature maps allow for learning a spatial prior.

Incorporating spatial information Predicting the category by only using the information from a limited local receptive field can be challenging and in some cases impossible. We exploit that the absolute position of certain categories in the image is an important contextual cue.

We provide the normalized position of a patch as an additional input to the CNN. In particular, the $x \in [0, 1]$ and $y \in [0, 1]$ coordinates are added as additional feature maps to one of the layers (Figure 2). Whereas incorporating the position information is a common and simple trick in semantic segmentation, with [4] being only one example, combining these priors with CNN feature learning has not been exploited before.

New CNN library: CN24 We implemented a new open source CNN library specifically designed for semantic segmentation [1], which is publicly available. An important

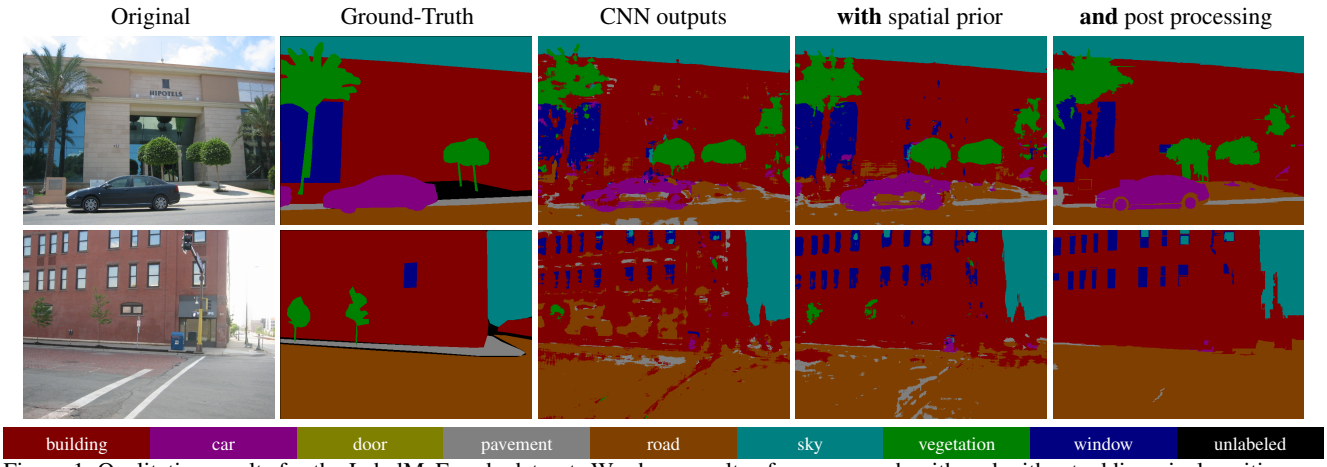

Figure 1. Qualitative results for the LabelMeFacade dataset. We show results of our approach with and without adding pixel positions as information for the learning procedure.

Method	ORR	ARR
RDF+SIFT [3]	49.06%	44.08%
ICF [4]	67.33%	56.61%
RDF-MAP [7]	71.28%	-
Our approach		
CNN outputs (fully convolutional training)	58.17%	29.48%
CNN outputs (patchwise training)	67.87%	42.89%
+spatial prior	72.21%	47.74%
+post processing	**74.33%**	47.77%
+weighting	63.41%	**58.98%**

Table 1. Results on LabelMeFacade in comparison to previous work. We report overall and average recognition rates.

feature of the framework is the large flexibility with respect to possible CNN architectures making it ideal for research purposes. For example, every layer can be connected to an auxiliary input layer or architectures can be freely specified as a directed graph. The framework does not depend on external libraries a priori, which makes it practical, especially for fast prototyping and heterogeneous environments. However, *OpenCL* or fast BLAS libraries such as ATLAS, ACML, or Intel MKL can be used to speed up convolutions and other algebraic operations significantly.

Experiments and evaluation Our experiments are based on the LabelMeFacade dataset [4], which consist of 945 images. The classes that need to be differentiated are: *building*, *window*, *sidewalk*, *car*, *road*, *vegetation* and *sky*. There is an additional background class named *unlabeled*, which we only use to exclude pixels from the training data. Since this is a multi-class classification problem, we are following [4] and use the overall recognition rate (ORR, plain accuracy) and the average recognition rate (ARR) which is the mean of class-wise accuracies.

Table 1 shows that we are able to achieve state-of-the-art performance on this dataset. The spatial prior significantly helps to boost the performance. To improve the slightly noisy CNN output, we added a post-processing step using the graph-based segmentation approach of [2]. To boost the average recognition rate, we also experimented with weighting each example with the inverse class frequency, which is denoted in the table with "weighting". As can be seen, care has to be taken with respect to choosing the right learning objective.

Conclusions We briefly showed how convolutional patch networks can be used for the task of semantic segmentation. Furthermore, we demonstrated how spatial prior information like pixel positions can be incorporated into the learning process leading to a significant performance gain. We were able to achieve state-of-the-art results on LabelMeFacade [3] which is a multi-class classification task and shows very diverse urban scenes. A longer version of this paper [1] includes experiments on the KITTI road segmentation benchmark but without fully-convolutional architectures.

References

[1] C.-A. Brust, S. Sickert, M. Simon, E. Rodner, and J. Denzler. Convolutional patch networks with spatial prior for road detection and urban scene understanding. In *VISAPP*, 2015. 1, 2

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):1–26, 2004. 2

[3] B. Fröhlich, E. Rodner, and J. Denzler. A fast approach for pixelwise labeling of facade images. In *ICPR*, volume 7, pages 3029–3032, 2010. 2

[4] B. Fröhlich, E. Rodner, and J. Denzler. Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach. In *ACCV*, pages 218–231, 2012. 1, 2

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 1

[6] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[7] S. Nowozin. Optimal decisions from probabilistic models: the intersection-over-union case. In *CVPR*, pages 548–555, 2014. 2