DAGM Workshop on New Challenges in Neural Computation, Frankfurt, August 2011

Object-Class Segmentation using Deep Convolutional Neural Networks

Hannes Schulz and Sven Behnke
University of Bonn, Computer Science VI,
Autonomous Intelligent Systems Group
Friedrich-Ebert-Allee 144, 53113 Bonn, Germany
{schulz,behnke}@ais.uni-bonn.de

Abstract

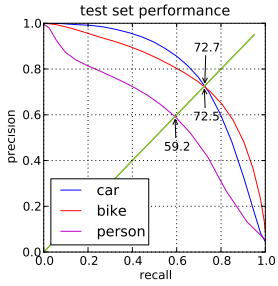
After successes at image classification, segmentation is the next step towards image understanding for neural networks. We propose a convolutional network architecture that outperforms current methods on the challenging INRIA-Graz02 dataset with regards to accuracy and speed.

1 Introduction

Neural networks have long history of usage for image classification, e. g. on MNIST [1], NORB [2], and Caltech [3]. For these datasets, neural networks rank among the top competitors [4]. Despite the success, we should note that these image classification tasks are quite artificial. Typically, it is assumed that the object of interest is centered and at a fixed scale, i. e. that the segmentation problem has been solved. Natural scenes rarely contain a single object or object class. Such images need to be analyzed on various scales and positions for objects of different categories. Object detection and object-class segmentation are thus the logical step towards general image understanding. In this work, we propose variations of the convolutional network for object-class segmentation. We show that with HOG and color input, intermediate outputs and squared epsilon-insensitive loss error function, we can achieve state-of-the-art accuracy on the INRIA Graz-02 (IG02, [5]) dataset. Due to the efficient reuse of information during convolution as well as a fast GPU implementation, we achieve a framerate of about 10 fps during recall.

2 Related Work

In the deep learning community, research on real images has largely focused on object detection (as opposed to segmentation). For example, using extensive dataset augmentation, pretraining of a sparse encoder, bootstrapping, Kavukcuoglu et al. [6] perform comparably well on the INRIA pedestrian dataset. Licence plates and faces are blurred in Google Street View using a convolutional neural network as part of a larger pipeline. Both techniques are applied in a sliding window, that is, the probability of a pixel being member of a class is determined independently



	PR-EER (%)		
	Car	Bike	Person
Ours	**72.7**	**72.5**	59.2
CRF[8]	72.2	72.2	**66.3**
LIN[9]	62.9	71.9	58.6

Figure 1: Precision/Recall on the IG02 dataset.

for every pixel and scale. We propose to use a convolutional architecture with multi-scale input, resulting in efficient reuse of data structures. Jain et al. [7] proposed convolutional architectures and cost functions to detect boundaries prior to segmentation. We acknowledge that this can improve segmentation results at the borders, but we believe that this should be a second step after finding object or object-class hypothesis. Most current approaches start with an oversegmentation of the image, e. g. Fulkerson et al. [8] classify superpixels based on histograms of features in their neighborhood. Superpixels are often expensive to compute and potentially introduce errors that are hard to correct later. Finally, Aldavert et al. [9] use a handtuned integral linear classifier cascade to achieve close to very good performance. However, we achieve better accuracy at a higher framerate.

3 Methods

Preprocessing We use eight square feature maps as input. Three maps are the whitened color channels, five maps represent histogram of oriented gradients (HOG180) features. The whitening kernel is derived from 5×5 random patches of the training set. HOG features are calculated at twice the map resolution and then subsampled. We perform these operations at three scales, with resolution decreasing by a factor of two. The teacher, i. e. an image where each pixel is marked with the class it belongs to, is split into one map per class where pixels are 1 when they are in the class and are 0 otherwise. Finally, the teacher is smoothed and downsampled for each scale.

Network Architecture For each scale s, we have input maps m_{si}, two convolutions resulting in maps m_{s1}, m_{s2} and one (intermediate) output layer o_s. The activities of o_s are determined by m_{s1} and fed to m_{s2} with additional convolutions. Between scales, we use maximum pooling to gain some spatial invariance. At each output layer, we measure the pixelwise class error using the squared epsilon-insensitive loss function $E(x, \hat{x}) = \max(0, |x - \hat{x}| - \varepsilon)^2$, where we fix $\varepsilon = 0.2$. This loss function does not punish small deviations from the target value and essentially acts as a regularizer which plays well with the final thresholding.

The error is backpropagated through the network in the usual way, see e. g. [10]. Errors of intermediate output are scaled by a factor of 0.1. With six

Figure 2: Sample test set object-class segmentations. Left: original image, center: ground truth segmentation, right: our segmentation. The colors red, green, blue represent cars, bikes and persons, respectively. White represents values at or below the EER thresholds. Large objects, such as on lower right, still have potential for improvement.

hidden layers, the network can be regarded as a "deep" network.

Training We update the weights with the accumulated errors after each epoch using the RPROP [11] algorithm with standard settings, which avoids the need to cross-validate a learning rate. All operations except preprocessing are performed on GPU using the CUV library [12].

4 Results

We test our architecture on the challenging INRIA Graz-02 dataset [5]. The dataset contains images of bikes, cars and persons covering an extremely wide range of pose, scale and lighting. We use the training/testing splits suggested on the dataset website, resulting in (after horizontal mirroring) 958 training and 479 testing images. The images are scaled to 172×172 and squared by horizontal or vertical centering and mirroring into non-occupied space. We use 32 maps on all layers, and filters of size 7×7. Error is measured as in [8] using precision-recall at equal-error rate (PR-EER), at input resolution. After 2000 weight updates, we find that in two categories we outperform state-of-the-art (see Fig. 1). We did not observe overtraining, which we attribute to the regularizing effect of the squared epsilon-insensitive loss. Some selected segmentations are depicted in Fig. 2. While our method generally performs well on small to medium scales, there is still room for improvement in the precise estimation of currently blurred boundaries. We further observe difficulties in images with e. g. large persons (lower right). Without pre-processing, we are able to process 28 fps, assuming current GPU HOG implementations for preprocessing, we estimate an estimated 10 fps for the trained network.

5 Conclusion

In this paper, we showed that convolutional networks can achieve state-of-the-art performance in object-class segmentation with regards to accuracy as well as speed. We plan to improve our results further using conditional random fields (CRFs) for post-processing.

References

[1] Y. LeCun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[2] Y. LeCun, F. Huang, and L. Bottou. "Learning methods for generic object recognition with invariance to pose and lighting". In: *CVPR*. 2004, pp. 97–104.

[3] L. Fei-Fei, R. Fergus, and P. Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories". In: *CVIU* 106.1 (2007), pp. 59–70.

[4] D. Ciresan et al. "High-Performance Neural Networks for Visual Object Classification". In: *CoRR* abs/1102.0183 (2011).

[5] M. Marszatek and C. Schmid. "Accurate object localization with shape masks". In: *CVPR*. 2007.

[6] K. Kavukcuoglu et al. "Learning Convolutional Feature Hierarchies for Visual Recognition". In: *NIPS 23*. Ed. by J. Lafferty et al. 2010, pp. 1090–1098.

[7] V. Jain et al. "Boundary learning by optimization with topological constraints". In: *CVPR*. 2010, pp. 2488–2495.

[8] B. Fulkerson, A. Vedaldi, and S. Soatto. "Class segmentation and object localization with superpixel neighborhoods". In: *ICCV*. 2009, pp. 670–677.

[9] D Aldavert et al. "Fast and robust object segmentation with the Integral Linear Classifier". In: *CVPR*. 2010, pp. 1046–1053.

[10] D. Scherer, A. Müller, and S. Behnke. "Evaluation of pooling operations in convolutional architectures for object recognition". In: *ICANN*. 2010, pp. 92–101.

[11] M. Riedmiller and H. Braun. "A direct adaptive method for faster backpropagation learning: The RPROP algorithm". In: *Neural Networks, 1993., IEEE*. 1993, pp. 586–591.

[12] H. Schulz, A. Müller, and S. Behnke. "Exploiting local structure in Boltzmann machines". In: *Neurocomputing* 74.9 (2011), pp. 1411–1417.