Convolutional Patch Networks with Spatial Prior for Road Detection and Urban Scene Understanding

Clemens-Alexander Brust, Sven Sickert, Marcel Simon, Erik Rodner and Joachim Denzler
Computer Vision Group, Friedrich Schiller University of Jena, Jena, Germany

Keywords: Convolutional Neural Networks, Patch Classification, Road Detection, Semantic Segmentation, Scene Understanding.

Abstract: Classifying single image patches is important in many different applications, such as road detection or scene understanding. In this paper, we present convolutional patch networks, which are convolutional networks learned to distinguish different image patches and which can be used for pixel-wise labeling. We also show how to incorporate spatial information of the patch as an input to the network, which allows for learning spatial priors for certain categories jointly with an appearance model. In particular, we focus on road detection and urban scene understanding, two application areas where we are able to achieve state-of-the-art results on the KITTI as well as on the LabelMeFacade dataset.
Furthermore, our paper offers a guideline for people working in the area and desperately wandering through all the painstaking details that render training CNs on image patches extremely difficult.

1 INTRODUCTION

In the last two years, the revival of convolutional (neural) networks (CN) (LeCun et al., 1989) has led to a breakthrough in computer vision and visual recognition. Especially the field of object recognition and detection made a huge step forward with respect to the final recognition performance as can be seen by the success on the large-scale image classification dataset ImageNet (Krizhevsky et al., 2012). This breakthrough was possible mainly due to two reasons: (1) large-scale training data and (2) huge parallelization to speed up the learning process. In general, an essential advantage of CNs is the automatic learning of *task-specific* representations of the input data, which was previously often hand-designed.

While the majority of works focuses on applying these techniques for object classification tasks, there is another field where CNs can be really useful: semantic segmentation. It is the task of assigning a class label to each pixel in an image. This is why it is also referred to as pixel-wise labeling. Previous works already showed how to use CNs in this area, *e.g* for road detection (Alvarez et al., 2012; Masci et al., 2013). However, the architectural choices and many critical implementation details have not been discussed and studied, although they are crucial for a high recognition performance. In our work, we therefore also give a brief list of guidelines for CN training and discuss several aspects important to get pixel-wise labeling with CNs running.

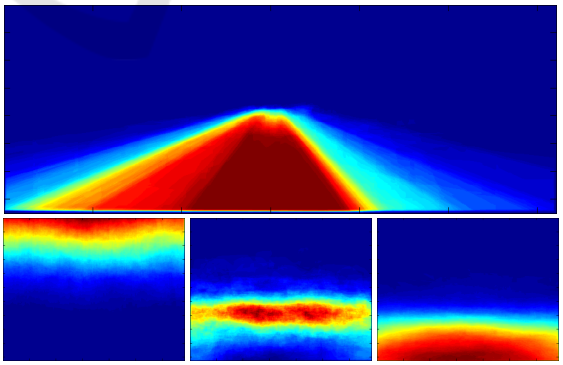

Figure 1: Illustration of spatial bias for categories in road detection and urban scene understanding: (top) class *road* of KITTI road challenge (Geiger et al., 2012), (bottom) classes *sky*, *car* and *road* of LabelMeFacade (Fröhlich et al., 2012). Warmer colors indicate higher probabilities (best viewed in color).

Furthermore, we show how to learn spatial priors during CN training, because some classes appear more frequently in some areas of an image (see Fig. 1). In general, predicting the label of a single pixel requires a large receptive field to incorporate as much context information as possible. However, the high

Brust C., Sickert S., Simon M., Rodner E. and Denzler J..
Convolutional Patch Networks with Spatial Prior for Road Detection and Urban Scene Understanding.
DOI: 10.5220/0005355105100517
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 510-517
ISBN: 978-989-758-090-1

input dimensionality would cause a huge CN model with too many parameters for learning it robustly while given only a small amount of training data. We avoid this by incorporating absolute position information in the fully connected layers of the CN.

In this paper, we use CN pixel-wise labeling for the tasks of road detection and urban scene understanding. In road detection, the pixels of an image are classified into *road* and *non-road* parts, which is an essential task for autonomous driving. The challenge is the huge variability of road scenes, because of changing light conditions, surface changes, and occlusions. For a qualitative and quantitative evaluation, we use the road estimation challenge of the popular KITTI Vision Benchmark Suite (Geiger et al., 2012). Urban scene understanding goes one step further by increasing the number of categories that need to be distinguished, such as buildings, cars, sidewalks, etc. We obtain state-of-the-art performance in both domains.

In the following, we first give a brief overview of the application of CNs for pixel-wise labeling. Section 3 introduces our CN models and shows how to learn spatial priors. Experiments are discussed and evaluated in Section 4. A summary in Section 5 concludes this paper.

2 RELATED WORK

Semantic segmentation was and is an active research area with numerous publications. We will present only those with relevant techniques (convolutional neural networks or randomized decision trees) or a similar scope of applications (road detection or urban scene understanding).

Semantic Segmentation with CNs. The work of (Couprie et al., 2014) presents an approach for semantic segmentation with RGB-D images. The main idea of their work is a multi-scale CN comprised of multiple CNs for different scales of the images, which are all linked to the fully connected layers at the end. In contrast to their work, our approach incorporates the spatial prior information as an important cue and a possibility to learn a bias of the position of an object in the image (Torralba, 2003).

Instead of performing semantic segmentation by classifying image patches, (Gupta et al., 2014) builds on algorithms for unsupervised region proposal generation. Each of the proposed regions is then classified with an SVM that makes use of features learned by CN using depth and geometric features as well as a CN trained on RGB image patches. Similarly, (Hariharan et al., 2014) also classifies object proposals and combines a CN for detection and a CN for classifying regions. In contrast to these works, we perform pixel-wise labeling and are therefore not limited to a few proposals generated by another algorithm.

Road Detection. Following up on their work with slow feature analysis (Kühnl et al., 2011), the authors of (Kühnl et al., 2012) propose spatial ray features to find boundaries of the road. The former work serves as a source for base classifiers which model road, boundary, and lanes. Especially the last one is very important for the method, since ray features are extracted from classifier outputs. In contrast to our work, this method strongly depends on the availability of lane markings in the scene.

Another work in the field aims at the problem of changing light conditions in street scenes. In (Alvarez and Lopez, 2011), the authors compute illumination invariant images to segment the road even if the image is highly cluttered due to shadows. Seeds are placed in the bottom part of the illumination invariant image where the road is supposed to be situated. All pixels that have a similar appearance to the seeds will then be classified as *road*. Since we learn local image filters with the CNs, we do not have to explicitly model illumination invariance in our approach but learn all variations from the given dataset.

Similar to our approach, (Alvarez et al., 2012) applied CNs for the task of road detection. However, their work focuses more on the transfer of labels learned from a general image database which has more images to learn from. Furthermore, they propose a texture descriptor which makes use of different color representations of the image. Finally, general information acquired from road scenes and information extracted from a small area of the current image are combined in a Naive Bayes framework to classify the image.

Urban Scene Understanding. While road detection is a binary classification scenario, the task of urban scene understanding is to distinguish multiple classes like *car*, *building* and *sky*.

In (Fröhlich et al., 2012) so-called iterative context forests are used to classify images in a pixel-wise or region-wise manner. The method is derived from the well known random decision forests with the advantage that classification results of one level of the tree can be used in the next level as additional features. The authors of (Scharwaechter et al., 2013) also aim for the classification of regions. They combine appearance features of gray scale images and

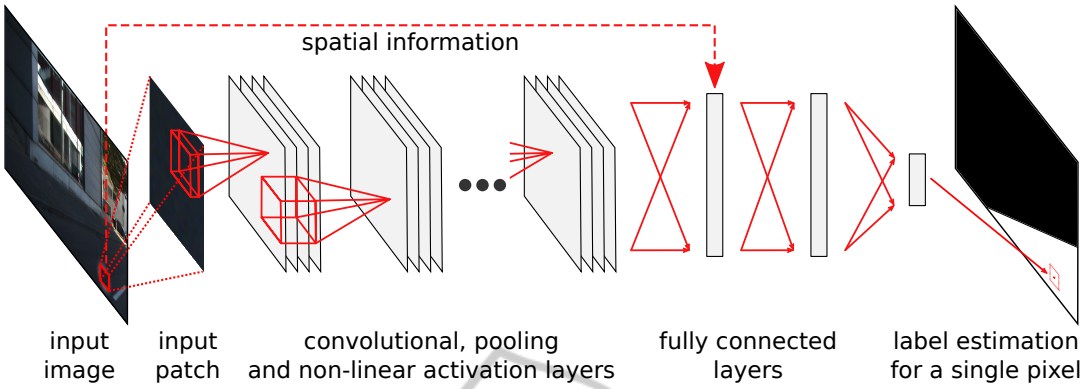

Figure 2: Example of our convolutional patch network: in addition to the visual features we also incorporate the absolute position information in the fully connected layers. The concrete architectures are given in the experimental section.

depth cues which are derived from dense disparity maps. They incorporate a medium-level environment model in order to obtain meaningful region hypotheses. Then, a multi-cue bag-of-features pipeline is used to classify these regions into object classes.

There are other works that incorporate additional sources of information other than images from a single camera. In (Zhang et al., 2010) dense depth maps are used to compute view-independent 3D-features, *i.e. surface normal* and *height above ground*. In contrast, the authors of (Kang et al., 2011) make use of an additional near-infrared channel. They use hierarchical bag-of-textons in order to learn spatial context from the image. However, as these methods are closely tied to a database that provides such information, we propose a more generic approach.

3 CONVOLUTIONAL PATCH NETWORKS

Convolutional (neural) networks (CNs) (LeCun et al., 1989) belong to a family of methods, usually referred to as "deep learning" approaches, especially in the popular literature. The main idea is that the whole classification pipeline consists of one combined and jointly trained model. Most recent deep learning architectures for vision are based on a single CN. CNs are feed forward neural networks, which concatenate several layers of different types with convolutional layers playing a key role.

3.1 Architecture and CN Training

The generic architecture of our CNs is visualized in a simplified manner in Fig. 2. The input for our network is always a single image patch extracted around a single pixel we need to classify. Therefore, we use the name *Convolutional Patch Network* for the method.

The network itself is structured in multiple layers. Each convolutional layer convolves the output of the previous layer with multiple learned filter masks. Afterwards, the outputs are optionally combined with a maximum operation in a spatial window applied to the result of each convolution, which is known as max-pooling layer. This is followed by an element-wise non-linear activation function, such as the hyperbolic tangent or the rectified linear unit used in (Krizhevsky et al., 2012).

The last layers are fully connected layers and multiply the input with a matrix of learned parameters followed again by a non-linear activation function. The output of the network are scores for each of the learned categories or in the case of binary classification one score related to the likelihood of the positive class. We do not provide a detailed explanation of the layers, since this is described in many other papers and tutorials (LeCun et al., 2001). In summary, we can think about a CN as one huge model $f(x;\theta)$ that tries to map an image through different layers to a useful output. The model is parameterized by θ, which includes the weights in the fully connected layers as well as the weights of the convolution masks.

All parameters θ of the CN are learned by minimizing the error of the network output for an example x_i compared the given ground-truth label y_i:

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} w_i \cdot L(f(x_i;\theta), y_i) \quad . \qquad (1)$$

In this setting L is a loss function and in our case we use the quadratic loss.

Optimization is done with stochastic gradient descent using momentum and mini-batches of 48 training examples (Krizhevsky et al., 2012). The learning rate and all other hyperparameters are optimized on a validation set.

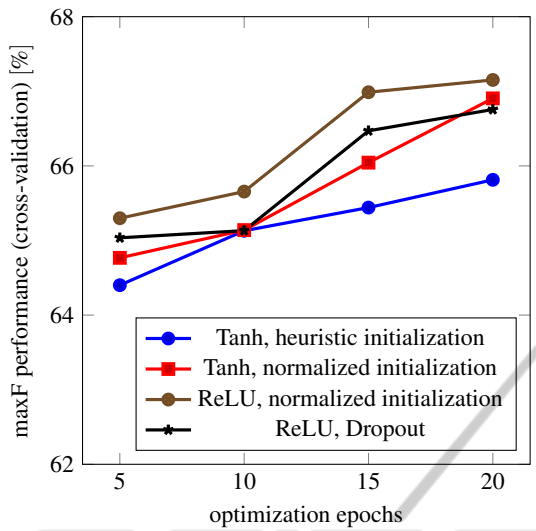

Figure 3: Performance for road detection with respect to the number of optimization epochs. The performance is measured by 10 times cross-validation on the KITTI training set.

3.2 Incorporating Spatial Priors

As already motivated in the introduction and in Fig. 1, predicting the category by using only the information from a limited local receptive field can be challenging and in some cases impossible. Therefore, the work of (Couprie et al., 2014) proposes a multi-scale CN approach to incorporate information from receptive fields with different sizes. In contrast, we exploit a very common property in scene understanding. The absolute position of certain categories in the image is not uniformly distributed. Therefore, the position of a patch in the image can be a powerful cue. This is especially true for road detection as also validated by (Fritsch et al., 2013).

Due to this reason, we provide the normalized position of a patch as an additional input to the CN. In particular, the $x \in [0,1]$ and $y \in [0,1]$ coordinates are added as inputs to one of the fully connected layers. This can be viewed as having a smaller CN, which provides a feature representation of the visual information contained in the patch, and a standard multiple layer neural network, which uses these features in addition to the position information to perform classification. Whereas incorporating the position information is a common trick in semantic segmentation, with (Fröhlich et al., 2012) being only one example, combining these priors with CN feature learning has not been exploited before.

3.3 Software Framework

We implemented a new open source CN framework specifically designed for semantic segmentation, which will be made publicly available.[1] The source code was designed from scratch in C++11 aiming at multi-core architectures and not necessarily strictly depending on GPU capabilities. An important feature of the framework is the large flexibility with respect to possible CN architectures. For example, every layer can be connected to an auxiliary input layer, which is important in our case to allow for the incorporation of position information or to incorporate the weight of a training example in the loss layer.

The framework does not depend on external libraries, 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. Convolutions are in general realized by transforming them into matrix-vector products, which requires some additional memory overhead but leads to a significant speedup as also empirically validated by (Chellapilla et al., 2006). For fast testing, the complete forward propagation through the network can also be accelerated by utilizing a device that computes OpenCL.

3.4 Important Details and Implementation Hints

Implementing our own framework allowed us to have influence on every aspect of the convolutional network in order to apply it for the task of semantic segmentation. Thereby, we made some important observations concerning parameter initialization and optimization techniques.

Initialization of Network Parameters. The training of networks with many layers poses a particular challenge because of the vanishing gradient issue (Glorot and Bengio, 2010). A repeated multiplication of the derivatives produces smaller and smaller values. This quickly leads to numerical problems in deep networks, particularly when using single-precision floating point calculations.

Usually, the weights in a layer with n inputs are initialized randomly by sampling from a uniform distribution in $[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$, which we refer to as *heuristic initialization*. However, the authors of (Glorot and

[1]This work was supported by Nvidia with a hardware donation.

Bengio, 2010) analyze the effect of vanishing gradients in detail and they derive an improved initialization scheme, known as *normalized initialization*, which has an important impact on the learning performance. This can be seen in Figure 3, where we plot the cross-validation accuracy of our road detection application after different numbers of optimization epochs (an epoch are 10000 iterations with mini-batches). As can be seen, the normalized initialization leads to a better performance of the network after a few epochs.

Benefit of Dropout and ReLU for Smaller Networks. Dropout as a regularizer is a means to prevent a network from overfitting (Hinton et al., 2012), which happens likely due to the large model complexity of the networks. However, the small convolutional net used in our approach for the task of road detection does not benefit from dropout as can be seen in Figure 3. Dropout has been shown to reduce error rates significantly in larger CN architectures (Krizhevsky et al., 2012). Furthermore, Figure 3 also reveals that using rectified linear units (Krizhevsky et al., 2012) as nonlinear activations is beneficial for the task of road detection.

Task-specific Weighting of Training Examples. If a recognition approach with a high performance with respect to a task-specific performance measure is required, one should optimize with a learning objective that comes as close as possible to the final performance measure. This hint might sound simple but we give two examples in the following where this is extremely important to boost the performance.

For the KITTI Vision road detection benchmark, performance is measured in the birds-eye view, while data is presented in ego view. The authors of (Fritsch et al., 2013) claim that the vehicle control usually happens in 2D space and therefore road detection should also be done in this space. A wrong classified pixel near the horizon in ego view represents a whole bunch of pixels in the birds-eye view. To compensate for this, we need to choose weights w_i for the training examples proportional to the size of the pixels after transformation in the birds-eye view.

In urban scene understanding, we are faced with a highly imbalanced multi-class problem, since pixels labeled as building are more common than pixels labeled as door, for example. Therefore, performance is usually measured in terms of accuracy (percentage of correctly labeled pixels) and average recognition rate (average of the class-wise accuracy) (Fröhlich et al., 2012). To focus on the average recognition rate, we weight examples according to their number of examples in the training set, *i.e.* $w_i = n_{y_i}^{-1}$.

4 EXPERIMENTS

Our experiments in semantic segmentation are evaluated on two applications: road detection and urban scene understanding, which are both challenging due to the high variation of possible appearances within the classes.

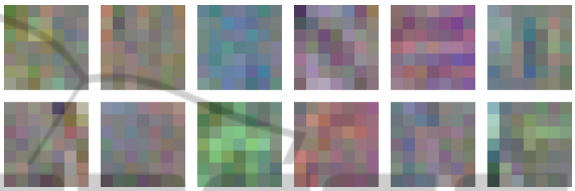

Figure 4: Convolution masks of the first layer found during learning on the KITTI road detection dataset.

4.1 Road Detection

For the task of road detection, we have to differentiate between road and non-road patches and therefore it is a typical binary classification problem. In recent years, the most commonly used road scene challenge is the KITTI Vision benchmark (Geiger et al., 2012). This dataset features a multi-camera setup, optical flow vectors, odometry data, object annotations, and GPS data.

There is a specific benchmark for road detection with 600 annotated images where road and non-road parts are labeled in the image. The dataset consists of three different urban road settings: single-lane roads with markings (UM), single-lane roads without markings (UU) and multi-lane roads with markings (UMM). There are challenges for road detection and ego-lane detection. For this dataset, we follow the evaluation protocol given in (Geiger et al., 2012) and report the F1-measure as a binary classification metric which makes use of precision and recall such that both have the same weight.

CN Architecture. We use the CN architecture listed in Table 1 for road detection, where we classify patches of size 28×28 extracted at each pixel location. This architecture was optimized using ten-fold cross validation. An important architectural choice is the incorporation of the absolute position of the patch as an input in layer 8. This allows for learning a spatial prior of the road category. Furthermore, it is interesting to note that the first layer applies convolution masks of a rather small size of 7×7. These

Figure 5: Qualitative results on the KITTI road detection dataset. Images show original input-data with labeled *road* pixels as green overlay. Although a spatial prior is used in our approach we are able to segment road scenes with curves or occlusions well. This figure is best viewed in color.

Table 1: CN architectures used for road detection (road det.) and urban scene understanding (urban sun.) along with their respective parameters. The number of outputs is denoted as o. The parameters for a convolutional layer are given by $w \times h \times n$, where n refers to the number of spatial filters used, each of them with a size of $w \times h$. For pooling layers, the parameters determine the spatial window for which the maximum operation is performed.

#	Type of layer	Road det.	Urban sun.
1	convolutional layer	$7 \times 7 \times 12$	$7 \times 7 \times 16$
2	maximum pooling	2×2	2×2
3	non-linear	ReLU	tanh
4	convolutional layer	$5 \times 5 \times 6$	$5 \times 5 \times 12$
5	non-linear	ReLU	tanh
6	fully connected layer	$o = 48$	$o = 64$
7	non-linear	ReLU	tanh
8	fully connected layer **+spatial prior**	$o = 192$	$o = 192$
9	non-linear	ReLU	tanh
10	fully connected layer	$o = 1$	$o = 8$
11	non-linear	tanh	sigmoid

Table 2: Results on the KITTI road detection dataset for methods that only use the camera input image for prediction.

Method	MaxF
CN approach of (Alvarez et al., 2012)	73.97%
Spatial prior only (Fritsch et al., 2013)	82.53%
Spray features (Kühnl et al., 2012)	88.22%
Our approach without spatial prior	76.34%
Our approach with spatial prior	86.50%

masks are visualized in Fig. 4 and depict certain textural and color elements that seem to be informative when distinguishing between road and non-road image patches.

Evaluation. The quantitative results of our road detection approach are given in Table 2. We compare with the method of (Kühnl et al., 2012), which obtains the current best result on the dataset, and the CN method of (Alvarez et al., 2012). As can be seen, we outperform the previous CN method by a large margin of over 10%. This can be mainly contributed to the spatial prior we learn with the CN. How important position information is for this dataset can be seen by the performance of the baseline algorithm of (Kühnl et al., 2012) which uses only the pixel position during testing without any appearance features.

As the qualitative results of our approach in Figure 5 show, we are able to segment curves although a spatial prior is incorporated. This is due to the weighting of these information which is automatically learned in the training. When allowing a very high weight for the spatial prior we would obtain similar results to the baseline algorithm.

Note that we do not make use of any additional information other than a single camera view. Some competitors in the challenge incorporate data of the second camera of the stereo setup or make use of the 3D point clouds from velodyne laser scanner. Their results are not reported here but are given on the KITTI website. At the time of submission we are on place 4 of 22 in the ranking[2].

4.2 Urban Scene Understanding

For urban scene understanding, each pixel is classified into one of K classes. Our experiments are based on the LabelMeFacade dataset (Fröhlich et al., 2012), which consist of 945 images. The classes that need to be differentiated are: *building*, *window*, *sidewalk*, *car*, *road*, *vegetation* and *sky*. Furthermore, 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 (Fröhlich et al., 2012) and use the overall recognition rate (ORR, plain accuracy) and the average recognition rate (ARR) which is the mean of class-wise accuracies.

CN Architecture. For urban scene understanding, we use the CN architecture reported in the right column of Table 1, which was also optimized with 50 training examples and 50 validation examples randomly selected from the LabelMeFacade dataset. As in the previous experiment, we extract patches of size

[2]http://www.cvlibs.net/datasets/kitti/eval_road.php, our method is named CN24

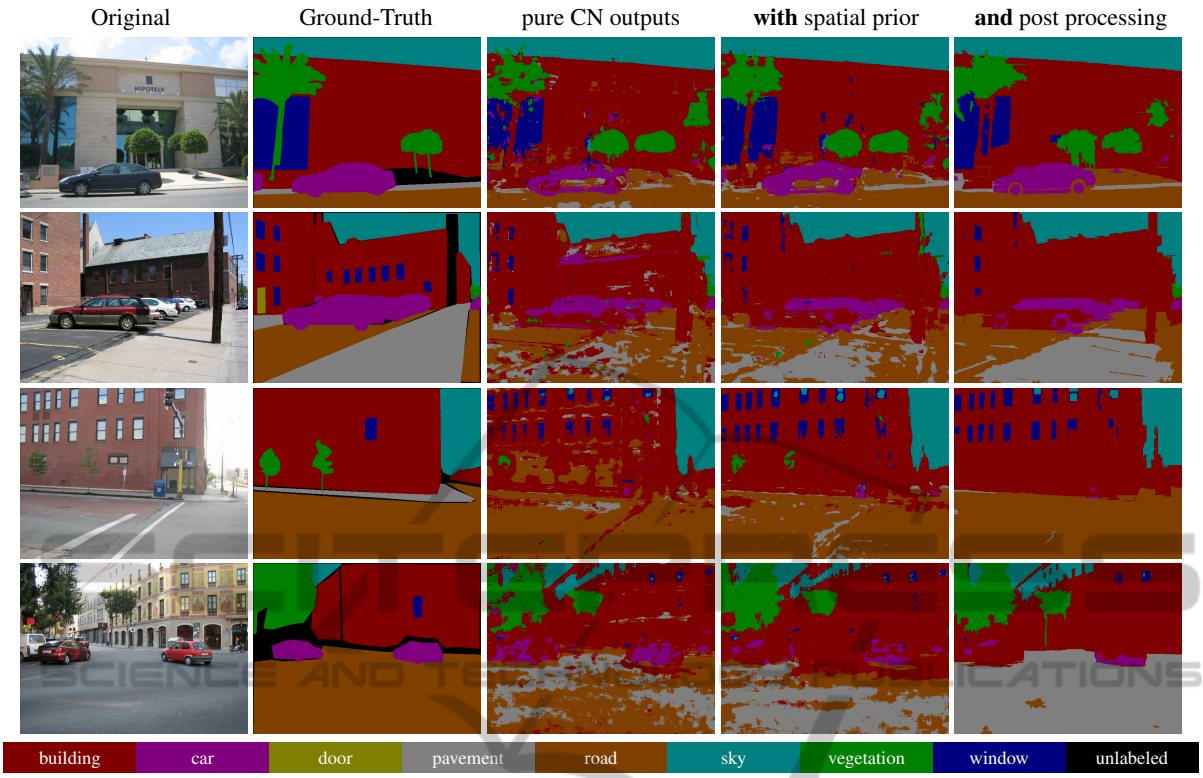

Figure 6: Qualitative results for the LabelMeFacade dataset. We show results of our approach with and without adding pixel positions as information for the learning procedure. As can be seen in the first three rows, these additional information improve the results (*road* vs. *building*). However, in some cases spatial priors do not help (*pavement* vs. *road*).

Table 3: Results on LabelMeFacade in comparison to previous work. We report overall and average recognition rates for different networks. The *weighted* CN was optimized with inverse class frequency weights.

Method	**ORR**	**ARR**
RDF+SIFT (Fröhlich et al., 2010)	49.06%	44.08%
ICF (Fröhlich et al., 2012)	67.33%	56.61%
RDF-MAP (Nowozin, 2014)	71.28%	-
Our approach		
pure CN outputs	67.87%	42.89%
+spatial prior	72.21%	47.74%
+post processing	74.33%	47.77%
+weighting	63.41%	58.98%

28×28 at each pixel location. In contrast to the CN for road detection, we used the hyperbolic tangent function as a non-linearity because it requires fewer neurons to express anti-symmetric behavior.

Evaluation. As can be seen in Table 3, we are able to achieve state-of-the-art performance on this dataset. The spatial prior significantly helps again to boost the performance. Furthermore, we can directly see that the weighting with respect to class frequencies has a significant impact on the ARR and the ORR performance as already discussed in Sect. 3.4.

Four segmentation examples are shown in Figure 6 in comparison to the given ground-truth labels. The authors of (Fröhlich et al., 2012) use a post-processing step by fusing their results with an unsupervised segmentation of the original images. The probability outputs of the classifier and the segments are combined to ensure a consistent label within regions. Since the output of our approach is scattered due to the pixel-wise labeling (column 4 and 5), we also added this post-processing step. We make use of the graph-based segmentation approach of (Felzenszwalb and Huttenlocher, 2004) with parameters $k = 550$ and $\sigma = 0.5$. As can be seen in the last column the results are improved with respect to object boundaries. However, this procedure can also lead to large regions with a wrong labeling (row 4). Instead of *road* the whole lower part of the image is classified as *pavement*. Both classes have a very similar appearance in most of the images.

5 CONCLUSIONS

In this paper, we showed how convolutional patch net-

works can be used for the task of semantic segmentation. Our approach performs classification of image patches at each pixel position. We analyzed different popular network architectures along with different techniques to improve the training. Furthermore, we demonstrated how spatial prior information like pixel positions can be incorporated into the learning process leading to a significant performance gain.

For evaluation, we used two different application scenarios: road detection and urban scene understanding. We were able to achieve very good results in the road detection challenge of the popular KITTI Vision Benchmark Suite. In this scenario we outperformed several competitors, even those that use stereo images or laser data.

For a second set of experiments, we used the dataset LabelMeFacade of (Fröhlich et al., 2010) which is a multi-class classification task and shows very diverse urban scenes. We were again able to achieve state-of-the-art results. Future work will focus on speeding up the prediction phase, since we currently need around 30*s* for each image to infer the label at each position.

REFERENCES

Alvarez, J. M., Gevers, T., LeCun, Y., and Lopez, A. M. (2012). Road scene segmentation from a single image. In *European Conference on Computer Vision (ECCV)*, pages 376–389.

Alvarez, J. M. and Lopez, A. M. (2011). Road detection based on illuminant invariance. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):184–193.

Chellapilla, K., Puri, S., Simard, P., et al. (2006). High performance convolutional neural networks for document processing. In *Tenth International Workshop on Frontiers in Handwriting Recognition*.

Couprie, C., Farabet, C., Najman, L., and LeCun, Y. (2014). Convolutional nets and watershed cuts for real-time semantic labeling of rgbd videos. *Journal of Machine Learning Research (JMLR)*, 15:3489–3511.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):1–26.

Fritsch, J., Kühnl, T., and Geiger, A. (2013). A new performance measure and evaluation benchmark for road detection algorithms. In *IEEE International Conference on Intelligent Transportation Systems*, pages 1693–1700.

Fröhlich, B., Rodner, E., and Denzler, J. (2010). A fast approach for pixelwise labeling of facade images. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 7, pages 3029–3032.

Fröhlich, B., Rodner, E., and Denzler, J. (2012). Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach. In *Asian Conference on Computer Vision (ACCV)*, pages 218–231.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 249–256.

Gupta, S., Girshick, R., Arbeláez, P., and Malik, J. (2014). Learning rich features from RGB-D images for object detection and segmentation. In *European Conference on Computer Vision (ECCV)*.

Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. (2014). Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)*.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Kang, Y., Yamaguchi, K., Naito, T., and Ninomiya, Y. (2011). Multiband image segmentation and object recognition for understanding road scenes. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1423–1433.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105.

Kühnl, T., Kummert, F., and Fritsch, J. (2011). Monocular road segmentation using slow feature analysis. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 800–806.

Kühnl, T., Kummert, F., and Fritsch, J. (2012). Spatial ray features for real-time ego-lane extraction. In *IEEE Conference on Intelligent Transportation Systems*, pages 288–293.

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

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (2001). Gradient-based learning applied to document recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press.

Masci, J., Giusti, A., Ciresan, D. C., Fricout, G., and Schmidhuber, J. (2013). A fast learning algorithm for image segmentation with max-pooling convolutional networks. *arXiv preprint arXiv:1302.1690*.

Nowozin, S. (2014). Optimal decisions from probabilistic models: the intersection-over-union case. In *Computer Vision and Pattern Recognition (CVPR)*.

Scharwaechter, T., Enzweiler, M., Franke, U., and Roth, S. (2013). Efficient multi-cue scene segmentation. In *German Conference on Pattern Recognition (GCPR)*, Lecture Notes in Computer Science, pages 435–445.

Torralba, A. (2003). Contextual priming for object detection. *International Journal of Computer Vision (IJCV)*, 53(2):169–191.

Zhang, C., Wang, L., and Yang, R. (2010). Semantic segmentation of urban scenes using dense depth maps. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *European Conference on Computer Vision (ECCV)*, pages 708–721.