23rd European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, April 2015.

Depth and Height Aware Semantic RGB-D Perception with Convolutional Neural Networks

Hannes Schulz, Nico Höft, and Sven Behnke

University Bonn – Computer Science VI, Autonomous Intelligent Systems
Friedrich-Ebert-Allee 144, 53113 Bonn – Germany

Abstract. Convolutional neural networks are popular for image labeling tasks, because of built-in translation invariance. They do not adopt well to scale changes, however, and cannot easily adjust to classes which regularly appear in certain scene regions. This is especially true when the network is applied in a sliding window. When depth data is available, we can address both problems. We propose to adjust the size of processed windows to the depth and to supply inferred height above ground to the network, which significantly improves object-class segmentation results on the NYU depth dataset.

1 Introduction

An important property of convolutional neural networks (CNN) is their invariance to translations. This invariance property is created by sharing weights between all image locations, and by pooling over nearby locations. For many tasks in computer vision, translation invariance is desirable, but some object classes (eg. sea, sky) are more likely to appear in certain scene regions. Furthermore, depending on the distance to the camera, the same objects can also appear at different scales, but scale invariance is missing in the CNN architecture. There are two commonly applied solutions to the problem. We might present the input to the network at multiple scales during training, such that the network can learn the invariance. This option requires large models and sufficient data, since we need to learn the same task independently at all scales. If object annotations are available, we can scale objects to a uniform size before presenting them to the CNN. To produce a prediction for a novel image, the network has to be applied to every image position at multiple scales, and the results must be combined. This approach is called sliding windows.

In object-class segmentation, where every pixel is labeled with the class of the object it belongs to, there are typically no object annotations. Thus, the sliding window approach is not feasible.

In this paper, we propose a third option which relies on the availability of depth for the image pixels (RGB-D). The required dense depth estimates are produced e.g. by affordable consumer cameras such as the Microsoft Kinect, which has become popular in computer vision [1, 2] and robotics [3]. We propose to use depth information in CNNs as follows:

1. We train the network on image patches with a size chosen proportional to the depth of the patch center. Since training is scale invariant, we can afford smaller models and make more efficient use of training data.

2. For novel images, we propose a sampling scheme which covers the image with overlapping patches of depth-adjusted size. Thus, closer image regions are processed at a large scale, while far-away regions are processed at a small scale. This automatic adjustment is more efficient than sliding windows, because the scale is chosen automatically. In contrast to a multi-scale sliding window, our scale adjustments are continuous.
3. Finally, we propose to use height above ground as an additional input to the CNN. Height is an important clue and quite distinct from the distance from the camera. E.g., floor pixels might occur at any distance from the camera, but they always have zero height.

We evaluate our method on the NYU Depth v2 dataset, which contains indoor scenes annotated according to their object class and find that both height annotation and depth normalization significantly improve CNN performance.

2 Related Work

Depth normalization of features has been proposed in the context of random forests [1, 4] by Stückler et al. [5]. The binary features in their work consist of region average differences, where both region sizes and distances to the query pixel are scaled with the depth. In this work, we scale image patches, not features. While this requires more computation during preprocessing, it allows for more expressive features.

Hermans et al. [6] and Stückler et al. [7] use random forests as a baseline and aggregate video information over time with self localization and mapping (SLAM). Here, we focus on single image prediction, which is comparable to random forest learning.

Using height for indoor scene object-class segmentation was introduced by Müller and Behnke [8]. The authors use the output of a random forest, merge the predictions within superpixels and learn a conditional random field (CRF) which has access to the average superpixel height. In contrast to their work, we incorporate height into the base classifier, which we then use to directly improve the unary term of their CRF.

Couprie et al. [9] and Höft et al. [10] train CNNs for object-class segmentation using depth information, with very different approaches. Couprie et al. [9] train three CNNs with shared weights on three scales. The upsampled results are then combined to yield output maps corresponding to object-class labels. Thus, in contrast to our proposed method, the image is always trained and evaluated on all three scales. The label probabilities are then averaged within superpixels of an oversegmentation. Superpixel averaging is compatible with our approach and further improves our performance.

Höft et al. [10] also use a CNN with a multi-scale approach, but treat scales differently. Larger scales have access to predictions from smaller scales and can modify them. Treating scales differently can be justified by the fact that in indoor scenes, certain objects (dressers, beds) are typically much larger than others (vases, television sets), and need more context to be recognized. Note that

Table 1: Network architecture used for this paper.

Layer	# Parameters	Filter Size	Stride	#Maps	Map Size
Input	–	–	–	8	64×64
Conv1	12,576	7×7	1	32	64×64
Pool1	–	2×2	2	32	32×32
Conv2	50,208	7×7	1	32	32×32
Pool2	–	2×2	2	32	16×16
Conv3	6,304	7×7	1	4	16×16

while in this work, we use only one scale for every patch, it is also possible to use a multi-scale approach where all scales are depth-adjusted simultaneously.

3 Methods

Network Architecture We use the simple feed forward convolutional architecture shown in Table 1, with interleaved convolutional max-pooling layers, and rectification (ReLU) non-linearities. With less than 70,000 parameters in total, it is a very small network (cf. [9, 10]). While performance might improve with size and better regularization (i.e. dropout), we would like to emphasize that depth and height awareness allows to reduce the number of parameters significantly.

Covering Windows We choose patch sizes s in the original image inversely proportional to the depth $d(\mathbf{x}_c)$ of a selected patch center $\mathbf{x}_c$, with $s = \gamma/d(\mathbf{x}_c)$. The parameter γ is set to $300\,\mathrm{px\,m}$ throughout this paper, such that enough context is provided to the network, and receptive field sizes are similar to the scale of the random forest features by Stückler et al. [7]. The patch is then scaled to the input dimension of the CNN with bilinear interpolation. If parts of the patch are outside the original image, we extend it by reflection on the border. Due to irregular patch sizes, a sliding window approach with fixed strides would sample too densely in regions with shallow depth or too coarsely in far-away regions. Instead, we simply ensure that the patches cover the image. We sequentially sample patch centers $\mathbf{x}_c$ from a multinomial distribution, with probabilities

$$p(\mathbf{x}_c) \propto \begin{cases} 0 & \text{if } \mathbf{x}_c \in \bigcup_{w \in W} w \\ d(\mathbf{x}_c) & \text{else,} \end{cases} \tag{1}$$

where W is the set of patches sampled so far. Depth-proportional probabilities encourage that far regions are covered by their corresponding small patches before near regions are covered by large patches.

When predicting, we use bilinear interpolation to upsample the network output to the original patch size and accumulate the predictions for all image patches. We use radially decreasing weights $r(\|\mathbf{x} - \mathbf{x}_c\|)$ in the accumulation, since the depth normalization is strictly valid only for the patch center.

Input Features We use eight input maps: The raw RGB channels, four containing a simplified histogram of oriented depth [10], and one map for the height. The height map is computed by extracting normals in the depth images, finding ten clusters with k-means and determining the cluster that is most vertical. All points are projected to this normal and the height of the lowest point is subtracted. From all input maps, we subtract the dataset mean and ensure that maps have similar variances.

3.1 Training Procedure

During training, we select patch centers $\mathbf{x}_c$ randomly, determine their size, and add small distortions (rotations of up to 5°, scalings of up to 5%, and horizontal flipping). CNN weights are initialized randomly from $\mathcal{U}(-0.01, 0.01)$. We use a batch size of 128 and an initial learning rate of 0.001, with a momentum of 0.9 and exponentially decreasing learning rate schedule. We optimize pixel-wise weighted multinomial logistic loss over the output maps, with weights

$$w(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \text{ is not annotated} \\ 0 & \text{if } \mathbf{x} \text{ is outside the original image} \\ r(\|\mathbf{x} - \mathbf{x}_c\|)/p(c(\mathbf{x})) & \text{else,} \end{cases} \tag{2}$$

where $p(c(\mathbf{x}))$ is the prior probability of the class $\mathbf{x}$ is annotated with.

4 Experiments

We train our network for object-class segmentation on indoor scenes on the NYU Depth v2 dataset [11], containing detailed annotations of 1449 RGB-D images split into 795 training and 654 testing images. We focus on the four semantic structural classes *floor, structure, furniture,* and *prop.* An additional *void* class resembles regions not annotated and is excluded from evaluation.

Our results are summarized in Table 2. We compared our method with other state-of-the art neural networks as well as methods which, for comparability, do not use extensive post-processing through CRFs or aggregation over time. We trained four models: a baseline method only using covering windows (CW), two with added depth normalization and height (CW+DN and CW+H, respectively), and a combined model (CW+DN+H). When training without depth normalization, we use the average patch size found by the depth normalization (135 px). We find that our combined model improves significantly over the other methods in terms of class average and pixel accuracies. The height feature contributes more to the overall improvement than depth normalization, but both ideas seem to complement each other.

Finally, our predictions can be used as input to high-level methods, such as super-pixel averaging (CW+DN+H+SP) and conditional random fields (CW+DN +H+CRF). We use method and implementation of Müller and Behnke [8], and find that class and pixel average accuracies improve by more than one percentage point when using our CNN predictions in place of their globally optimized random-forest predictions.

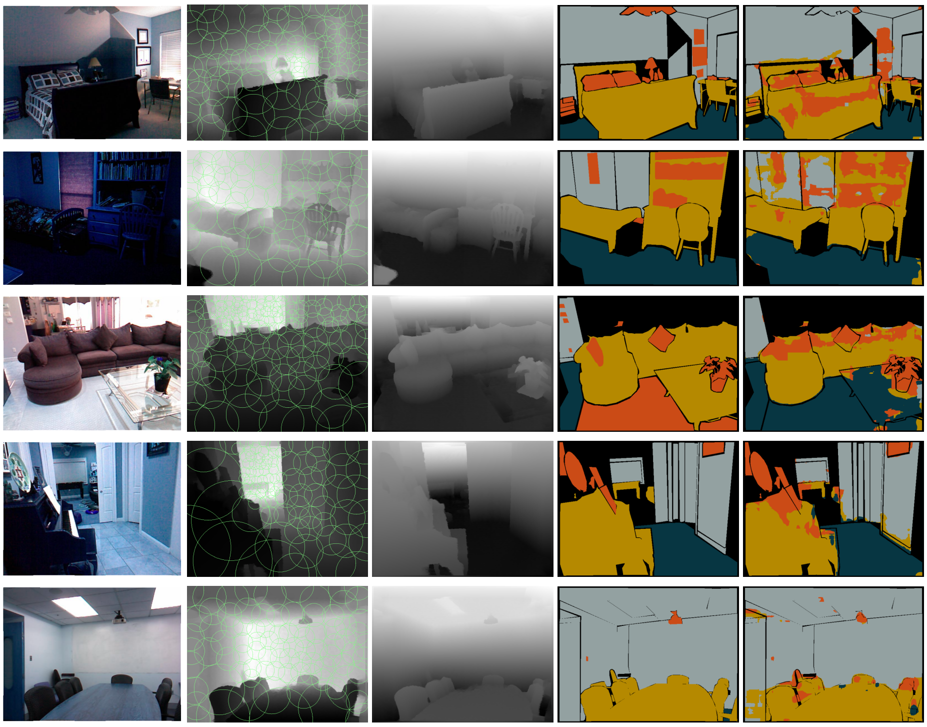

Fig. 1: Sample segmentations from test set using the CW+DN+H model. Left to right: Original image, depth with patches denoted by circles, height above ground, ground truth and prediction.

Table 2: Results on NYU Depth v2 dataset

Method	floor	struct	furnit	prop	Class Avg.	Pixel Acc.
CW	84.6	70.3	58.7	52.9	66.6	65.4
CW+DN	87.7	70.8	57.0	53.6	67.3	65.5
CW+H	78.4	74.5	55.6	62.7	67.8	66.5
CW+DN+H	93.7	72.5	61.7	55.5	70.9	70.5
CW+DN+H+SP	91.8	74.1	59.4	63.4	72.2	71.9
CW+DN+H+CRF	93.5	80.2	66.4	54.9	**73.7**	**73.4**
Müller et al.[8]	94.9	78.9	71.1	42.7	71.9	72.3
Random Forest [8]	90.8	81.6	67.9	19.9	65.1	68.3
Couprie et al.[9]	87.3	86.1	45.3	35.5	63.6	64.5
Höft et al.[10]	77.9	65.4	55.9	49.9	62.3	62.0
Silberman [12]	68	59	70	42	59.7	58.6

CW is covering windows, H is height above ground, DN is depth normalized patch sizes. SP is averaged within superpixels and SVM-reweighted. CRF is a conditional random field over superpixels [8]. Structure class numbers are optimized for class accuracy.

Sample segmentations, as well as patch size and height visualizations, are shown in Fig. 1. Note that the network sometimes overgeneralizes (carpet is labeled as *floor*, not *prop* in row 3), but generally identifies *floor*/not-*floor* well even in images where no or little floor is visible and our simple height extraction algorithm fails (row 5).

5 Conclusion

We proposed two extensions for convolutional neural networks which exploit depth information: i) covering windows which are scaled by the depth of their center and ii) height-above-ground input maps. Our evaluation on the NYU Depth v2 dataset shows that the proposed approach can outperform other neural network and random forest methods. In future work, we plan to extend our method with multi-scale depth-normalized processing.

References

[1] J Shotton et al. “Real-time human pose recognition in parts from single depth images”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2011.

[2] L Bo, X Ren, and D Fox. “Unsupervised feature learning for RGB-D based object recognition”. In: *Int. Symp. on Experimental Robotics*. 2013, pp. 387–402.

[3] RA Newcombe et al. “KinectFusion: Real-time dense surface mapping and tracking”. In: *Int. Symp. on Mixed and Augmented Reality (ISMAR)*. 2011.

[4] V Lepetit, P Lagger, and P Fua. “Randomized trees for real-time keypoint recognition”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2005.

[5] J Stückler, N Birešev, and S Behnke. “Semantic Mapping Using Object-Class Segmentation of RGB-D Images”. In: *Intelligent Robots and Systems (IROS)*. 2012.

[6] A Hermans, G Floros, and B Leibe. “Dense 3D Semantic Mapping of Indoor Scenes from RGB-D Images”. In: *International Conference on Robotics and Automation (ICRA)*. 2014.

[7] J Stückler, B Waldvogel, H Schulz, and S Behnke. “Dense real-time mapping of object-class semantics from RGB-D video”. In: *Journal of Real-Time Image Processing* (2013).

[8] AC Müller and S Behnke. “Learning Depth-Sensitive Conditional Random Fields for Semantic Segmentation of RGB-D Images”. In: *International Conference on Robotics and Automation (ICRA)*. 2014.

[9] C Couprie, C Farabet, L Najman, and Y LeCun. “Indoor Semantic Segmentation using depth information”. In: *CoRR* (2013). arXiv: `abs/1301.3572 [cs]`.

[10] N Höft, H Schulz, and S Behnke. “Fast Semantic Segmentation of RGB-D Scenes with GPU-Accelerated Deep Neural Networks”. In: *German Conference on Artificial Intelligence (KI)*. 2014.

[11] N Silberman and R Fergus. “Indoor scene segmentation using a structured light sensor”. In: *Computer Vision Workshops (ICCV Workshops)*. 2011.

[12] N Silberman, D Hoiem, P Kohli, and R Fergus. “Indoor segmentation and support inference from RGBD images”. In: *European Conference on Computer Vision (ECCV)*. 2012.