Fast and Robust RGB-D Scene Labeling for Autonomous Driving

Manuel Jasch*, Thomas Weber, Matthias Rätsch
Reutlingen University, Reutlingen, Germany.

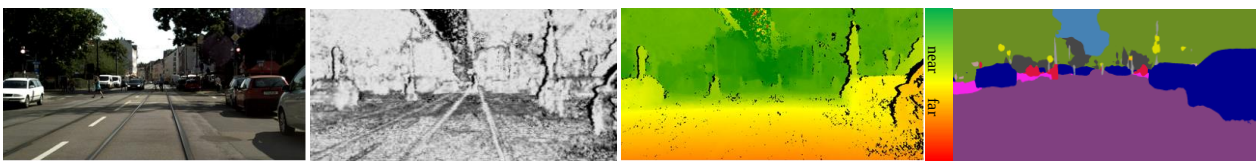
* Corresponding author. Email: info@manuel-jasch.de
Manuscript submitted April 19, 2017; accepted July 19, 2017.
doi: 10.17706/jcp.13.4.393-400

Abstract: For autonomously driving cars and intelligent vehicles it is crucial to understand the scene context including objects in the surrounding. A fundamental technique accomplishing this is scene labeling. That is, assigning a semantic class to each pixel in a scene image. This task is commonly tackled quite well by fully convolutional neural networks (FCN). Crucial factors are a small model size and a low execution time. This work presents the first method that exploits depth cues together with confidence estimates in a CNN. To this end, novel experimentally grounded network architecture is proposed to perform robust scene labeling that does not require costly preprocessing like CRFs or LSTMs as commonly used in related work. The effectiveness of this approach is demonstrated in an extensive evaluation on a challenging real-world dataset. The new architecture is highly optimized for high accuracy and low execution time.

Key words: CNN architecture, deep convolutional neural networks, depth information, semantic pixel-wise segmentation.

1. Introduction

In the automobile sector, understanding the scene context is important for autonomously driving cars and assistance systems. Scene labeling—assigning a semantic label to every pixel in the image—can therefore serve as foundation for higher level abstract applications [1]. Most works in the computer vision community focus on scene labeling on RGB images. In applications like autonomous cars however, additional depth information is often available, obtained e.g. using depth sensors, laser scanners, structure from motion or via stereo vision. This is the first paper, to the best knowledge of the authors, where both depth as well as confidence measurements are exploited for a deep learning based scene labeling system. Stereo vision is used in this paper, however the proposed method is independent of the actual method for obtaining those depth and confidence measures. Fig. 1 illustrates input and output of the presented method. The RGB image is provided for interpretability.



Original scene Confidence Depth Scene labeling

Fig. 1. Exemplary scene labeling output, where colors encode semantic classes. Depth and confidence, where white encodes high confidence, serve as input to the CNN.

Fully convolutional networks (FCN) [2] a specific variant of convolutional neural networks (CNN) have been applied to scene labeling with great success. These are trained end-to-end on manually annotated data to perform a classification of each individual pixel into a predefined set of classes, e.g. car, pedestrian or road. FCNs are optimized to form convolutional filters, like edge and blob detectors as well as to exploit contextual information. However, state-of-the-art FCNs like the GoogLeNet [3] tend to erroneously segment large objects into several individual classes [4]. One reason for that is the absence of the absolute scale of objects in the world, which is not available in one individual RGB image. Depth, on the other hand, contains exactly this missing ingredient. In order to exploit the depth information using FCNs, several problems have to be addressed. First, FCN architectures are well studied in the RGB domain, however this is not the case for depth data. In this work, we propose to use a specifically adapted variant of the Network-in-Network [5] architecture. Second, initialization of CNN parameters requires large amounts of costly labeled data which is not available for depth data. We address this problem by using parameters of a network trained on RGB data and by adapting them to the depth domain.

2. Related Work

The first group of related work is formed by the methods that were used to obtain depth measurements together with confidence estimates using a stereo vision system. The Cityscapes dataset comes with disparity values for each RGB image pixel. Disparity is the displacement of the same world point between the left and right stereo image, which is inverse to the depth. It is obtained from stereo images via semi global matching (SGM) [6]. This algorithm is highly optimized for execution time and power consumption on FPGAs [7]. The additional confidence estimates are obtained subsequently and reflect the certainty of individual depth measurements [8].

The second line of related work contains methods that use depth information in CNNs or specifically for scene labeling [9], [10]-[12]. In most cases, additional features are calculated from depth, e.g. plane detection [13], direction of gravity [14], height above ground, or orientation of gradient [9], [15]. In doing so, additional computational power is required. We rely directly on the outputs of the stereo algorithm without the need of post-processing of any kind. Despite CNNs, support vector machines (SVM) [16], [17] or conditional random fields (CRF) [10]-[12], [18] are commonly used. CNNs outperform SVMs in many tasks and applications. Further, our application specific real-time demands forbid the use of costly LSTMs [19] or CRFs [20]. Thus, we opt for pure CNN based depth and confidence processing: an efficient Network-in-Network [5]variant. Furthermore, we focus on the development of a system that successfully exploits depth jointly with confidence data. In the automobile context, the Cityscapes dataset [4], a real-world dataset containing a large number of manually labeled images in challenging inner-city driving scenarios, is well suited for our approach.

The main contributions of this paper are: (1) to the best knowledge of the authors, this is the first work leveraging disparity and confidence cuesin a CNN. (2) An innovative solution is presented enabling efficient and robust scene labeling with depth and confidence information in a lean architecture, ready for use in real-world applications. (3) This is achieved using a light-weight CNN architecture that does not require costly CRF or LTSM computation. (4) Convincing results are demonstrated in thorough evaluation on a challenging dataset.

3. Method

CNN architectures have been well studied and experimentally optimized in the computer vision community over many years in the recent past. Unfortunately, this is not the case for depth data as it is used in this paper, which we address to the fact that extremely large annotated datasets like ImageNet [21] do

not exist for depth data. Instead of designing a completely new architecture, we build on the assumption that depth and color data are related. In both, structures like blobs and edges need to be detected in order to recognize objects in the image. Accordingly, an existing rather simplistic and efficient network is used and fine-tuned, which was originally trained on RGB data: a Network in Network (NiN) [5].

A NiN consists of multiple network modules. Each is further composed of one convolutional layer with a kernel size larger than one that captures spatial information and one or multiple 1×1 convolutional kernels. Note that such a module is equivalent to a multi-layer perceptron (MLP). Finally, a global average pooling yields a classification score per image. For more details, the interested reader is referred to [5].The authors also propose and deploy a specific NiN architecture consisting of three network modules, each formed by a convolutional layer and a two layer MLP. We build on this architecture and the pre-trained weights on Image Net.

For our purposes we apply the following modifications to this network architecture. First, we need to address the fact that only a single depth channel serves as input instead of the three channel RGB input. Related work mainly eludes this by creating two artificial channels, e.g. angle of gradient and height above ground or by simply applying the depth channel three times. Instead, we fuse the weightsw_c for the three channels c of the first layer by summation $w_{new} = \sum_c w_c$ which is equivalent to using the depth input for all channels. However, this step removes redundant degree of freedom for the training of the network weights. Second, the network is modified to produce one label output per pixel in the image, i.e.it is transformed to a fully convolutional network [14]. The global average pooling layer is replaced by multiple de-convolutional layers that perform an up-sampling of the semantic labels to the original image resolution. Additional skip layers help to retrieve the fine details of the input, which typically get lost in the contracting part of the CNN. Finally, we observe that interpreting disparity images requires a lot contextual information. However, the network's contextual awareness is limited by the receptive field, which can be enlarged by increasing the number of pooling or spatial convolutional layers. Accordingly, we append an additional pooling layer, a NiN module with the corresponding skip layer to the end of the network's contracting part, which allows the network to capture more contextual information. The overall architecture, now consisting of four instead of three NiN modules, is illustrated in Fig. 2. The final output of the network is one label per input image pixel illustrated on the right. In many applications, confidence information is available in addition to the sole depth measurements. This can be leveraged using an additional input channel that is added to the network. The corresponding weights of the first layer are initialized equivalently, as described above.

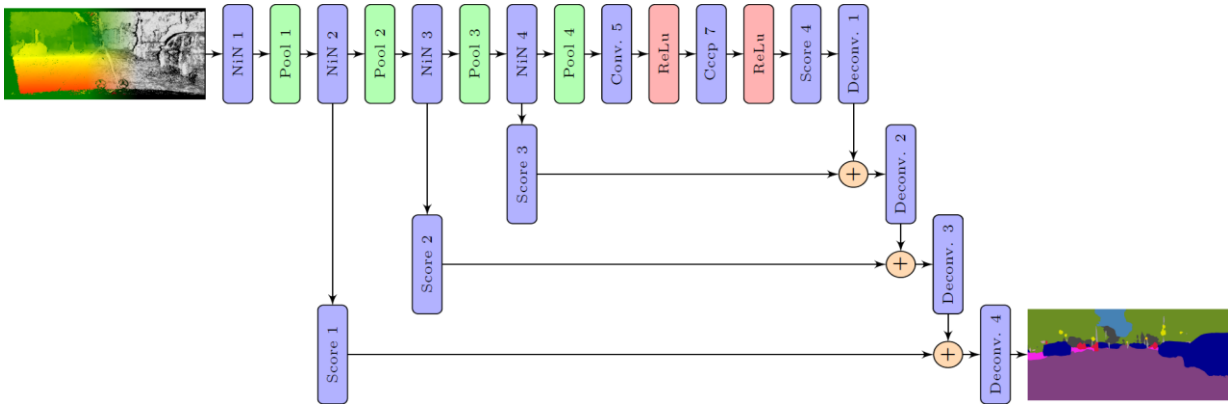


Fig. 2. Proposed architecture consisting of four NiN[5] modules. Three skip layers refine the output.

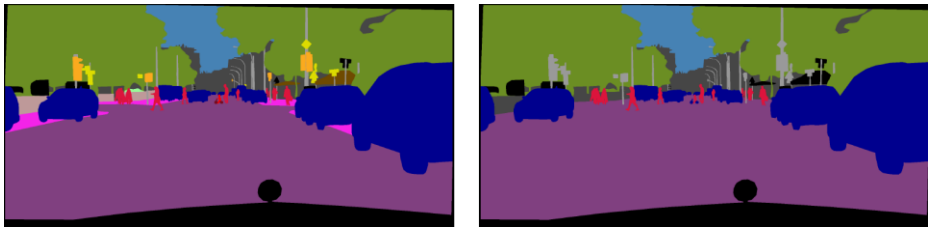
4. Experiments

In this Section an in-depth analysis of the proposed depth CNN is provided. The details of the training

procedure and parameterization are stated and qualitative results are discussed thoroughly.

4.1. Training and Dataset

Evaluation is carried out on the Cityscapes dataset, a highly complex dataset with challenging inner-city driving scenarios and dense annotations of 19 classes on 2975 images for training and 500 images for validation, which we used for testing. Based on depth information alone, it is extremely hard to distinguish some of these classes, e.g. bus, train and truck, or traffic sign and traffic light. For this reason, the Cityscapes dataset also provides 7 so called categories, which better reflect the needs of autonomous vehicles, as illustrated in Fig. 3. The label accuracy is evaluated in terms of Intersection over Union $IoU = TP / (FP + FN + TP)$ with TP, FP, and FN being the number of true positive, false positive and false negative pixel labels. In the experiments, results are reported on both tasks: the 19 classes *IoU class* as well as the 7 categories *IoU category*.



(a) Ground truth for 19 classes (b) Ground truth for 7 categories

Fig. 3. The corresponding ground-truth for Fig.1. Black areas are ignored.

4.2. Preprocessing

The depth information in the dataset has several characteristics, which might have negative impact on the label accuracy. First, it contains some invalid pixels encoded by a negative value in the disparity image obtained via stereo vision, cf. Section 2. In order to evaluate the impact of this invalid measurement, we compare to a preceding background *interpolation* [6]. Second, we use *disparities* as input to our network, as these meet the noise characteristics of standard stereo processing systems. However, disparity is a non-linear measure, which might harm the linear convolutional kernels in particular in the first layer. For comparison, disparity values d are transformed to *distance* z via $z = b \cdot f / d$, with baseline b and focal length f. Finally, the dataset is captured with different camera systems. Thus the camera calibration is inconsistent throughout the training images. In order to evaluate possible effects on the training, disparity data is *normalized* to $b_n = 22$ cm and $f_n = 2262$ pixels. Normalized disparity values d_n can be computed in accordance to $d_n = (b_n f_n d)/(bf)$.

Table 1. All Preprocessing Steps, such as Transformation to Distance from Camera, Normalization to a Standard Baseline and Interpolation of Invalid Disparity Estimates, Harm the Label Accuracy

	IoU class [%]	IoU category [%]
disparity	**39.6**	**67.6**
normalized	39.2	**67.6**
distance	36.1	65.8
interpolated	36.2	65.0

According to Table 1, the interpolation of invalid measurements leads to a relative performance drop of nearly ten percent, compared to the raw disparity input: the invalid measurements actually serve as feature for classification, e.g. in the sky or wall regions many invalid stereo estimates occur due to the low texture

in the image. Further, the disparity-to-depth-transformation has a similar negative impact on the classification performance, which might be due to the noise characteristics of the stereo method. Finally, the normalization of the camera system leads do a slight but less significant accuracy drop. Concluding, disparity values can be used directly from SGM algorithm output without the need of further costly preprocessing steps.

4.3. Additional Input Features

Related works commonly exploit different input features in addition to sole depth measurements, cf. Section 2. In contrast, we propose to use only the original stereo vision output as input and leave the rest to the end-to-end learning. For the purpose of comparison, the label accuracy of our network is also evaluated with the commonly used input feature height above ground. The height above ground y_w is computed from the known disparity $d_{u,v}$ at position u and v in the image, $\mathbf{x}_{u,v} = (u \quad v \quad 1)^{\mathrm{T}}$, the known intrinsic camera matrix K, baseline$b$and focal length f by transformation to camera coordinates $\mathbf{x}_{u,v} = (x_w \quad y_w \quad z_w)^{\mathrm{T}}$ via $\mathbf{x}_{u,v} = \mathbf{K} \cdot \mathbf{x}_{u,v}\, b_f / d_{u,v}$. Confidence values are used non-preprocessed and carry textural information, which is not available in the disparity domain. We evaluate the impact of the individual cues as well as their combinations.

The results in upper part of Table 2 indicate that depth is the strongest of the three compared cues. In combination, it becomes apparent that depth and height above ground contain highly correlated information. Thus the combination of both cues in the lower part of the Table does not improve the label accuracy. Instead, the additional input modality seems to harm the training process due to the additional degree of freedom during parameter optimization. The opposite holds for confidence and depth. Although confidence alone yields worse results compared to depth, their combination leads to a significant boost in performance. We address this to the fact that confidence and depth contain complementary information that the network can exploit.

Table 2. Results Using Different Input Features as well as Their Combinations. Depth and Confidence Information Complement Resulting in Improved Label Accuracy

Input Features	IoU class [%]	IoU category [%]
confidence (C)	37.2	67.1
height above ground (H)	36.7	66.7
disparity (D)	**39.6**	**67.6**
D+C	**43.8**	**71.6**
D+H	39.3	68.0
D+H+C	**43.7**	**71.9**

4.4. Architecture and Training Procedure

Network initialization is a major problem when working with CNNs. Learning the network's parameters requires a huge amount of data. In Table 3, label accuracy in terms of IoU is reported for different architectures and initialization strategies. Unfortunately, the training data of the Cityscapes dataset does not seem to be sufficient to train the full NiN *from scratch*, i.e. with random weight initialization. Significantly *reducing* the amount of *filters* per NiN module to one third also does not allow to train the NiN from scratch. Additionally *reducing* the number of channels in the *Score 4* layer results in bad classification scores. In our experiments, reduced counts of network parameters led to worsened performance, which we address to the extremely reduced network capacity. Instead we argue that disparity and color information are related, cf. Section 3, and initialize with learned weights on ImageNet, cf. Table 3 and Fig. 4.

Table 3. Learning from Scratch does not Succeed, however Initialization with Weights Trained on the ImageNet RGB Data Yields Satisfactory Results, Particularly on the Seven Categories

	IoU class [%]	IoU category [%]
from scratch	<5.0	<10.0
⅓ filters	<5.0	<10.0
reduced Score 4	29.4	60.9
ImageNet initialization	**39.6**	**67.6**

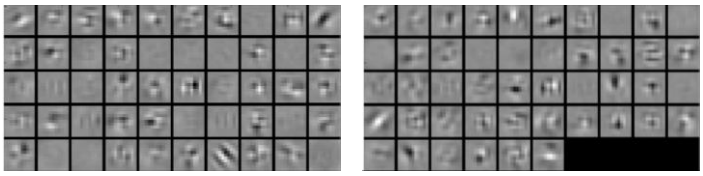


Fig. 4. The learned 96 convolutional filters with size11×11 in the first layer after fine-tuning. The similarity to the original filters indicates that depth and color information are highly related.

The Network Depth is a decisive factor for the network's capacity to approximate functions as well as its hardware demands, i.e. runtime and memory consumption. For applied systems like autonomous cars, we seek a good compromise of both factors. Therefore, label accuracy and run time is evaluated depending on the network depth, more precisely, the number of NiN modules. In accordance to Fig. 5 and Table 4, we opt for four NiN modules for our purposes, since the fourth layer yields significant improvements in terms of label accuracy without considerably harming the runtime.

Table 4. Absolute Runtime of Different Network Modules for up to Seven NiN Modules and the Score Layer. Overall Runtime Including Loss Layer and Up-Sampling is 36.0 ms Respectively 14.1 ms without

Network module	1	2	3	4	5	6	7	Score 4
Processing time [ms]	2.530	4.990	1.800	0.618	0.699	0.695	0.696	1.772
Relative execution time [%]	17.97	35.50	12.82	4.40	4.97	4.94	4.95	12.61

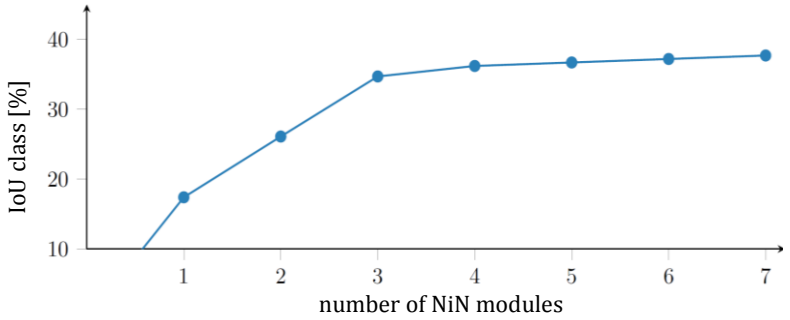


Fig. 5. Impact of the network complexity, i.e. the number NiN modules, on the label accuracy in terms of IoU.

5. Conclusion

This paper presented a novel method for processing depth together with confidence cues to perform scene labeling — assigning a semantic label to every pixel in the image. An efficient NiN architecture was transformed to a FCN and adapted to the depth and confidence domain, enabling the network to exploit a wide range of contextual information. The initialization problem was tackled using filters learned on RGB, which were fine-tuned for scene labeling on depth and confidence data. This method was superior to initialization from scratch, due to the insufficient availability of training data. Existing works neglect the

availably of confidence cues and commonly perform several preprocessing steps to boost performance. In thorough experiments was shown that such preprocessing steps do not improve the label accuracy. Furthermore, exploiting confidence cues as complementary cue led to significant improvements in terms of scene labeling accuracy. The proposed method is meant to form a building block in future work that combines the highly abstract depth features with those obtained from RGB images. Overall, a real-time and memory efficient FCN is presented that shows convincing scene labeling results and is readily applicable for many kinds of applications.

References

[1] Schneider, L., Cordts, M., Rehfeld, T., *et al.* (2016). *Semantic Stixels: Depth is not Enough.*

[2] Long, J., Shelhamer, E., & Darrell, T. (2014). Fully convolutional networks for semantic segmentation. *CVPR.*

[3] Szegedy, C., Liu, W., Jia, Y., & Sermanet, P. (2014). Going deeper with convolutions. *CVPR.*

[4] Cordts, M., Omran, M., Ramos, S., *et al.* (2015). The cityscapes dataset. *CVPR.*

[5] Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *Proceedings of ICLR.*

[6] Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *PAMI.*

[7] Gehrig, S. K., Eberli, F., & Meyer, T. (2009) A real-time low-power stereo vision engine using semi-global matching. *Proceedings of ICVS*.

[8] Pfeiffer, D., Gehrig, S., & Schneider, N. (2013). Exploiting the power of stereo confidences. *CVPR.*

[9] Höft, N., Schulz, H., & Behnke, S. (2014). Fast semantic segmentation of RGB-D scenes with GPU-accelerated deep neural networks. *KI.*

[10] Latecki, L. J. (2015). Semantic segmentation of RGBD images with mutex constraints. *Proceedings of ICCV.*

[11] Ren, X., Bo, L., & Fox, D. (2012). RGB-(D) scene labeling: Features and algorithms. *CVPR.*

[12] Krešo, I., Cauševic, D., Krapac, J., & Šegvic, S. (2016). Convolutional scale invariance for semantic segmentation. *GCPR.*

[13] Khan, S. H., Bennamoun, M., Sohel, F., Togneri, R., & Naseem, I. (2016). Integrating geometrical context for semantic labeling of indoor scenes using RGBD images. *IJCV.*

[14] Gupta, S., Arbelaez, P., & Malik, J. (2013). Perceptual organization and recognition of indoor scenes from RGB-D images. *CVPR.*

[15] Gupta, S., Girshick, R., & Arbel, P. (2014). Learning rich features from RGB-D images for object detection and segmentation. *ECCV.*

[16] Couprie, C., Farabet, C., Najman, L., & LeCun, Y. (2013). Indoor semantic segmentation using depth information. *Proceedings of ICLR.*

[17] Gupta, S., Girshick, R., Arbelaez, P., & Malik, J. (2014). Learning rich features from RGB-D images for object detection and segmentation. *LNCS.*

[18] Wang, P., Shen, X., Lin, Z., *et al.* (2015). Towards unified depth and semantic prediction from a single image. *CVPR.*

[19] Li, Z., Gan, Y., Liang, X., *et al.* (2016) RGB-D scene labeling with long short-term memorized fusion model. *CoRR.*

[20] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected CRFs. *Proceedings of ICLR.*

[21] Russakovsky, O., Deng, J., Su, H., *et al* (2015). ImageNet large scale visual recognition challenge. *IJCV.*

Manuel Jasch was born in Reutlingen, Germany, in 1991. He received the B.Eng. and the M.Sc. degree from the Reutlingen University, in 2015 and 2016, both in mechatronics. His research interests include artificial intelligence, machine learning, image understanding and autonomous driving.

Thomas Weber was born in Reutlingen, Germany, in 1990. He received his B.Eng. degree in 2014 and his M.Sc. degree in 2016, both from Reutlingen University, Germany, in mechatronics. He is currently pursuing the Ph.D. degree with Reutlingen Research Institute. His research interests include artificial intelligence, machine learning, image understanding, human robot collaboration and autonomous driving.

Matthias Rätsch was born in 1966, Neubrandenburg, Germany. He is a professor at the Reutlingen University for image understanding and interactive mobile robotics. In 2008, he received his Ph.D. degree in the graphics and vision research group (GraVis) at the University of Basel, Switzerland in 3DMM face analysis. His research interests are in the fields of image understanding, computational intelligence, autonomous robots and human robot collaboration. He is the head of the RoboCup team RT-Lions.