

# SDNet: Semantically Guided Depth Estimation Network

Matthias Ochs<sup>1</sup>, Adrian Kretz<sup>1</sup>, and Rudolf Mester<sup>1,2</sup>

<sup>1</sup> VSI Lab, Goethe University, Frankfurt am Main, Germany

<sup>2</sup> Norwegian Open AI Lab, CS Dept. (IDI), NTNU Trondheim, Norway

**Abstract.** Autonomous vehicles and robots require a full scene understanding of the environment to interact with it. Such a perception typically incorporates pixel-wise knowledge of the depths and semantic labels for each image from a video sensor. Recent learning-based methods estimate both types of information independently using two separate CNNs. In this paper, we propose a model that is able to predict both outputs simultaneously, which leads to improved results and even reduced computational costs compared to independent estimation of depth and semantics. We also empirically prove that the CNN is capable of learning more meaningful and semantically richer features. Furthermore, our SDNet estimates the depth based on ordinal classification. On the basis of these two enhancements, our proposed method achieves state-of-the-art results in semantic segmentation and depth estimation from single monocular input images on two challenging datasets.

## 1 Introduction

Many computer vision applications rely on or at least can increase their performance, if pixel-wise depth information is available in addition to the input image. Based on this depth information, geometric relationships in the environment can be explained better and can support, for instance, the scene understanding in robotics, autonomous driving, image editing or 3D modeling/reconstruction.

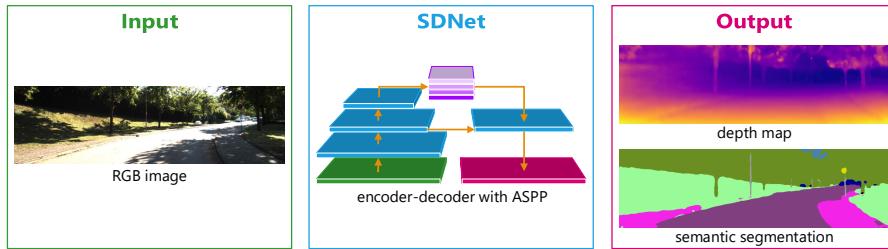


Fig. 1: Our proposed SDNet estimates pixel-wise depth and semantic labels from a single monocular input image based on a modified DeepLabv3+ architecture.

Unfortunately, estimating the depth from a single monocular image is an ill-posed problem, which cannot be solved without further knowledge. In this paper, we propose a deep learning-based approach, where this additional information is learned in a supervised manner from the geometric cues of many training samples by a CNN. The network architecture of this CNN is inspired by the DeepLabv3+ model from [2] and extended by two subnets to simultaneously predict the depth and semantic labels. Due to this combined prediction, our net learns semantically richer and more stable features in the Atrous Spatial Pyramid Pooling (ASPP) module, which results in an improved estimation of both. Following the insights of [6] and having in mind that depths follow a natural order, we regard the depth estimation as an ordinal classification problem and not as a regression-based one. This ensures that the neural network converges faster and that the resulting depth maps are more precise and can retrieve more details of fine structures and small objects. In the evaluation section, we show that SDNet yields state-of-the-art results on two datasets while still being able to generalize to other datasets as well.

## 2 Related Work

In this section, we review classical as well as deep learning based methods for solving the depth estimation problem. In most classical approaches [24,25,20], hand-crafted features are extracted from the monocular images, which are then used to predict the depth map by optimizing a probabilistic model like, a Markov random field (MRF) or conditional random field (CRF). Ladicky et al.[15] estimate the depth based on different canonical views and show that semantic knowledge helps to improve the prediction. This is verified for stereo-matching methods by [26,31], too. In this paper, we empirically prove that this concept also holds for depth estimation from a single monocular image with a CNN.

Besides these classical techniques, many recent approaches make use of Convolutional Neural Networks (CNNs). These techniques can be divided into two categories: supervised and unsupervised learning methods. Supervised learning techniques include models which are trained on stereo images, but which can infer depth maps on monocular images. [12] propose an approach which follows this paradigm. They use the correlations of CNN feature maps of stereo images and derive the unary matching costs. The depth maps are then computed by using a CRF. But there are also CNNs [28,21,23,8], which try to find stereo correspondences to estimate the disparity/depth.

The previously mentioned approaches are trained in a supervised manner. [14] introduce a semi-supervised approach, where the loss function includes an additional unsupervised term. This term captures the stereo alignment error of two images given an approximate depth map. The approaches of [7,30] and [9] use modifications of this alignment error to train networks in a completely unsupervised way. Since ground truth depth maps are not necessary with these approaches, it is possible to make use of much larger training databases. These

unsupervised techniques tend to produce much smoother and thus more inaccurate depth map than supervised techniques, though.

Fu et al.[6] introduced DORN, which is a CNN trained in a supervised way and achieves current state-of-the-art results. Instead of using regression to estimate the depth of each pixel, the depths are discretized and ordinal classification is used. We also use a similar idea for our depth estimation.

### 3 Approach

In this section, we introduce our proposed model, which classifies the depth based on ordinal depth classes while simultaneously inferring the semantic labels for each pixel, too. This has the advantage that more meaningful features are learned in the final layers of the encoder, from which both the semantic segmentation and the depth estimation can benefit. This allows the CNN to describe, detect and classify objects more accurately.

#### 3.1 Architecture

The network architecture of our model is depicted in figure 2. It is similar to the DeepLabv3+ model from [2], which represents the current state of the art for semantic segmentation.

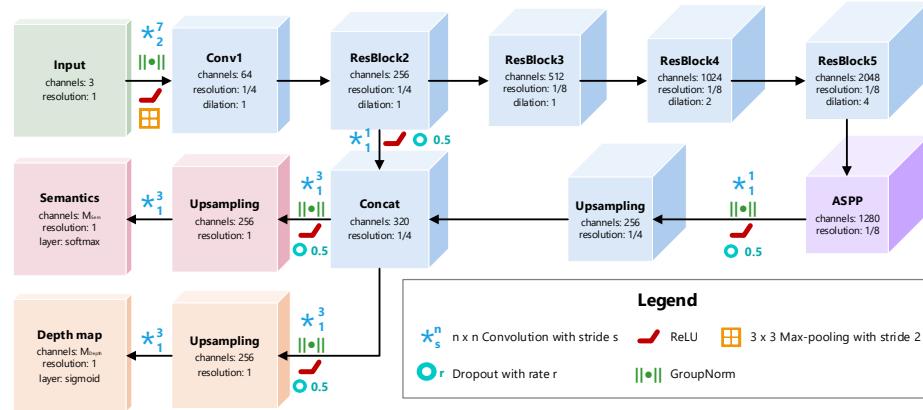


Fig. 2: Encoder-decoder architecture of SDNet with ASPP module. The decoder consists of two subnets, which predict the pixel-wise depths and semantic labels.

Theoretically, every encoder, e.g. ResNets or InceptionNets, can be used to extract feature maps which serve as input for the ASPP module. Due to limited computational capacity and real-time requirements, we use a ResNet-50 in this work. The ResNet is configured as proposed by [2] (dilation rate, multi-

grid, output stride of 8, atrous convolution, etc.). With the combination of this encoder and the ASPP module, SDNet is capable of extracting features on multiple scales, which improves the results significantly. The ASPP module is also adopted from DeepLabv3+, except that we use group normalization [29] instead of batch normalization, which applies to all normalization layers in the CNN. Furthermore, we add dropout layers, as shown in figure 2. Compared to DORN [6], our ASPP module is *fully-convolutional* and thus our CNN is not dependent on a predefined input resolution.

Unlike DeepLabv3+, our decoder consists of two subnets. First, the decoder interpolates the semantically meaningful feature maps from the ASPP module which are then concatenated with feature maps from the second ResBlock. These feature maps provide additional structural information and are able to recognize fine structures in the image. The resulting feature maps serve as input for our two subnets. In the first subnet, the semantic labels are estimated. The second subnet determines the depth. Both output layers of the two subnets compute class probabilities, which is done by a softmax function for the semantics and by a sigmoid layer for the depths.

### 3.2 Discretization

For ordinal regression, the continuous depth must be discretized so that a particular class  $c$  can be assigned to each depth  $d$ . Normally, the depths  $d \in [d_{\min}, d_{\max}]$  are divided linearly into classes between the minimum and maximum depth. However, this partitioning has the disadvantage that an erroneous estimation of a depth class at shallow depths causes a larger relative estimation error compared to a faulty estimated large depth.

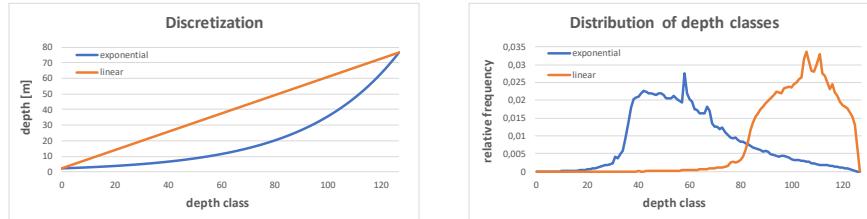


Fig. 3: The left plot shows the exponential and linear discretization of the depths with  $M_{\text{depth}} = 128$  in the interval [2 m, 80 m]. In the right histogram, the relative frequency of these exponential and linear discretized depth classes are shown over all existing depths of the KITTI depth prediction training dataset.

This relative error can be reduced by either defining the classes by their inverse depths  $\rho = \frac{1}{d}$ , or by choosing a more complex mapping function, e.g., [6]

chose a logarithmic mapping function for discretization. In this work, we split the inverse depth into discrete classes  $c_i$  using an exponential function

$$c_i = \rho_{\min} \cdot \exp \left( \log \left( \frac{\rho_{\max}}{\rho_{\min}} \right) \cdot \frac{i - 1}{M_d - 1} \right), \quad i \in \{1, \dots, M_d\} \quad (1)$$

where  $M_d$  corresponds to the number of depth classes. In the left plot of figure 3, both the linear and the exponential mapping functions are shown. On the basis of this graph, one can see that due to the exponential character, more discretization levels or classes are available for small and medium depths than for larger ones. Thus, the relative prediction errors of an estimator should be correspondingly smaller. In the adjacent histogram, the relative frequencies of these two depth classes are shown using the depths found in the KITTI depth prediction training data. A uniform distribution of these classes would be ideal for training a CNN. By a suitable choice of the mapping function, such as the proposed exponential, the depth distribution can be approximated to be a uniform one.

In contrast to the semantic classes, the depth classes follow an ordinal relation. This means that a larger depth class  $c_i$  is always included in the smaller ones:  $\{c_1 \preceq c_2 \preceq \dots \preceq c_{M_d}\}$ . By this concept, an inverse depth  $\rho$  which actually belongs to the class  $c_i$ , also belongs to the classes  $c_1, \dots, c_{i-1}$ . This ordered classification was introduced for neural networks by [3] and [22]. [6] and we use this idea, and show that an ordinal classification is advantageous over a standard classification or regression based estimation of depth values.

### 3.3 Output & Loss

As described in Section 3.1, SDNet returns the result of the semantic segmentation and depth estimation in two separate subnets. The output and the loss of the semantic subnet follow the standard multi-class classification for each pixel  $k \in 1, \dots, K$ , where  $K$  is the number of pixels in the image. Hence, the semantic subnet outputs logits  $\mathbf{y}_k^s \in \mathbb{R}^{M_s}$ , where  $M_s$  is equal to the number of semantic classes. The logits  $\mathbf{y}_k$  are then converted into probabilities  $\mathbf{p}_k^s$  by using the softmax function. As training loss, we use the multi-class cross entropy with  $\mathbf{t}_k^s \in \{0, 1\}^{M_s}$  as the one-hot-encoded vector representing the ground truth:

$$L_{k,\text{sem}}(\boldsymbol{\theta}; \mathbf{p}_k^s, \mathbf{t}_k^s) = -\frac{1}{M_s} \cdot \sum_i t_{k,i}^s \log p_{k,i}^s, \quad (2)$$

where  $\boldsymbol{\theta}$  are the trainable parameters in the CNN.

This vanilla multi-class classification cannot be used for the ordered depth classes, because a single pixel can belong to multiple labels (multi-label classification), whereas each depth class is not dependent on the others. Thus, it is regarded as a binary classification problem for each class  $c_i$  and pixel  $k$ . The depth logits  $\mathbf{y}_k^d \in \mathbb{R}^{M_d}$  are transformed by the sigmoid function to probabilities  $\mathbf{p}_k^d$ , where  $p_{i,k}^d \in [0, 1]$  indicates whether the pixel  $k$  belongs to class  $c_i$  or not.

Then, SDNet calculates the following multi-hot-encoded vector for each pixel:

$$\begin{aligned} \mathbf{z}_k^d &= [\sigma(p_{k,1}^d), \dots, \sigma(p_{k,M_d}^d)]^T, \quad \text{with} \\ \sigma(x) &= \begin{cases} 1, & \text{if } x \geq 0.5 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (3)$$

To preserve the ordered relation of the classes, the components of the classified vector  $\mathbf{y}_i$  are accumulated until the first entry of the vector  $\mathbf{z}_k^d$  is 0. This sum then corresponds to the class of the pixel:

$$\begin{aligned} c_k &= \sum_i \eta(\mathbf{z}_k^d, i), \quad \text{with} \\ \eta(\mathbf{z}_k^d, i) &= \begin{cases} 1, & \text{if } i = 1 \text{ and } \mathbf{z}_{k,i}^d = 1 \\ 1, & \text{if } \eta(\mathbf{z}_k^d, i-1) = 1 \text{ and } \mathbf{z}_{k,i}^d = 1 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (4)$$

To train the depth part of SDNet, we choose the following binary cross entropy loss and the ground truth vector  $\mathbf{t}_k^d$  is formulated as a multi-hot-encoded vector for class  $c_k$ :

$$L_{k,\text{depth}}(\boldsymbol{\theta}; \mathbf{p}_k^d, \mathbf{t}_k^d) = -\frac{1}{M_d} \cdot \sum_i t_{k,i}^d \log p_{k,i}^d + (1 - t_{k,i}^d) \log(1 - p_{k,i}^d). \quad (5)$$

Thus, the total loss is the mean of two individual losses over all pixels in the batch, with  $\lambda$  acting as a coupling constant:

$$L(\boldsymbol{\theta}; \mathbf{p}_k, \mathbf{t}_k) = \frac{1}{K} \sum_k L_{k,\text{depth}}(\boldsymbol{\theta}; \mathbf{p}_k^d, \mathbf{t}_k^d) + \lambda \cdot L_{k,\text{sem}}(\boldsymbol{\theta}; \mathbf{p}_k^s, \mathbf{t}_k^s). \quad (6)$$

To weight both loss terms similarly,  $\lambda = 10$  was empirically determined. We have also analyzed additional loss terms, such as regularization of the predicted depth map by its gradient as proposed by [14] and [9], or the absolute sum of the depth values from [32]. Another possible improvement would be the use of class balancing factors or the focal loss from [18]. However, all of these improvements have not produced any additional positive effects, so we disregarded them.

## 4 Evaluation & Experiments

This section describes the training of SDNet in more detail. Both, the predicted depth and semantic segmentation are evaluated and achieve state-of-the-art results in the KITTI depth prediction (KD) benchmark [27] and the KITTI pixel-level semantic segmentation (KS) benchmark [1]. In addition, we demonstrate the generalization of SDNet to other data on Cityscapes (CS) [4] and compare our semantic segmentation results with a modified DeepLabv3+ model. We use the same evaluation metrics as in [5] and the official benchmarks.

Both the evaluation and ablation studies show that a joint estimation of depths and semantic labels is preferable to an independent estimation, because it yields superior results. This observation indicates that a CNN is learning semantically more meaningful features. Using these features, the CNN can detect and recognize objects better, which in turn improves both predictions. Additionally, the experiments also show that depths can be estimated more precisely using classes rather than a regression-based approach.

#### 4.1 Training Protocol

To train the SDNet, we used the three datasets mentioned above in the following order. First, we train on the 3000 samples of fine-annotated training data from CS, which contain both semantic information and depth maps from SGBM [11]. These depth maps are relatively dense compared to LIDAR measurements from KITTI, but they are also less accurate and exhibit the typical stereo artifacts. Thus, SDNet was also trained with the 85898 samples of training data from KD, in which LIDAR measurements were accumulated over several frames to get denser depth maps. There is no semantic information available for this data, so only the depth part of SDNet could be trained with KD. However, KITTI published a training dataset (KS) with 200 images containing semantic annotations. LIDAR data also exists for this dataset, so that all necessary data is available for a complete training of SDNet.

Due to the different resolutions and the limited GPU memory, we randomly cropped the images to  $768 \times 352$  pixel at each epoch. Furthermore, we used random flipping and the standard color jittering as data augmentation techniques. We initialized the encoder with pre-trained weights from ImageNet and froze the parameters of the first and the second ResNet block during training. We used Adam as optimizer and a batch size of 4. The weight-decay factor and dropout rate was set to 0.0005 and 0.5 respectively. The best results were achieved when using 128 depth classes to split the depth interval of [2m, 80m].

In the first training phase, SDNet was trained on CS and KD with 50.000 iterations each and an initial learning rate of 0.0001. For training on KD,  $\lambda$  was set to 0 because there is no semantic data and thus only the depth subnet should be trained. In all other cases  $\lambda = 10$  was chosen. The second training phase consists of fine-tuning on KS, where both subnets are optimized using learning rate of 0.00001 and 3000 iterations.

During deployment, there is no need to divide the input images, because SDNet is fully-convolutional. Thus, the output can be estimated on arbitrary sized input images. Similarly, as suggested by [9], SDNet also predicts the output of the flipped input image. The final prediction is then computed by merging these two outputs, which makes the result more robust to occlusion and outliers.

#### 4.2 KITTI Dataset

The depth estimation and semantic segmentation were evaluated by several evaluations within KD and KS, and compared to state-of-the-art methods. Before

KITTI published its benchmark for the monocular depth estimation in 2018, it was common to evaluate the results using an unofficial dataset suggested by [5].

Method	ARD ↓	SRD ↓	RMSE ↓	RMSE <sub>Log</sub> ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Fu et al.[6]	0.072	0.307	2.727	0.120	0.932	0.985	0.995
<b>SDNet</b>	<b>0.079</b>	<b>0.504</b>	<b>3.700</b>	<b>0.167</b>	<b>0.921</b>	<b>0.968</b>	<b>0.983</b>
Yang et al.[32]	0.097	0.734	4.442	0.187	0.888	0.958	0.980
Kuznetsov et al.[14]	0.113	0.741	4.621	0.189	0.862	0.960	0.986
Godard et al.[9]	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Eigen et al.[5]	0.190	1.515	7.156	0.270	0.692	0.899	0.967
Liu et al.[19]	0.217	1.841	6.986	0.289	0.647	0.882	0.961
Saxena et al.[25]	0.280	3.012	8.734	0.361	0.601	0.820	0.926
Fu et al.[6]	0.071	0.268	2.271	0.116	0.936	0.985	0.995
<b>SDNet</b>	<b>0.075</b>	<b>0.430</b>	<b>3.199</b>	<b>0.163</b>	<b>0.926</b>	<b>0.970</b>	<b>0.984</b>
Yang et al.[32]	0.092	0.547	3.390	0.177	0.898	0.962	0.982
Kuznetsov et al.[14]	0.108	0.595	3.518	0.179	0.875	0.964	0.988
Godard et al.[9]	0.108	0.657	3.729	0.194	0.873	0.954	0.979
Garg et al.[7]	0.169	1.080	5.104	0.273	0.740	0.904	0.962

Table 1: Evaluation and comparison with other approaches on the test split by Eigen et al.[5]. In the upper part of this table, the depth is evaluated in range of [0m; 80m] whereas the maximum depth was reduced to 50 m in the lower part.

The Eigen test dataset consists of 697 images, which are selected from 29 different sequences of the KITTI raw dataset. The evaluation in table 1 proves that SDNet achieves very good results in all metrics and is only outperformed by DORN. Moreover, the qualitative comparison of the predicted depth maps in figure 4 shows that SDNet can reproduce the depths more accurately and with more details compared to other methods, whose depth maps usually look more blurred. Further examples of depth estimation on KS can be found in figure 6.

Besides to this unofficial analysis, the estimated depths of SDNet were also evaluated on the official KD benchmark, which consists of 500 test images. The images depict different driving situations, so that the network has to generalize in order to perform well in this benchmark. The results are shown in table 2, where only previously published methods have been listed and, again, SDNet achieves state-of-the-art results. Based on the D1 error images in figure 5, which shows the deviation to the ground truth for some test images, it can be recognized that the predicted depth of SDNet is erroneous especially at large depths. One reason for this could be that the maximum depth class has been set to 80 m. This also explains the relatively small gap to DORN at the Eigen evaluation, in which the maximum depth is limited to 50 m or 80 m.

Additionally, the predicted semantic segmentation of SDNet was also evaluated on the KS benchmark, which consists of 200 images. The results obtained

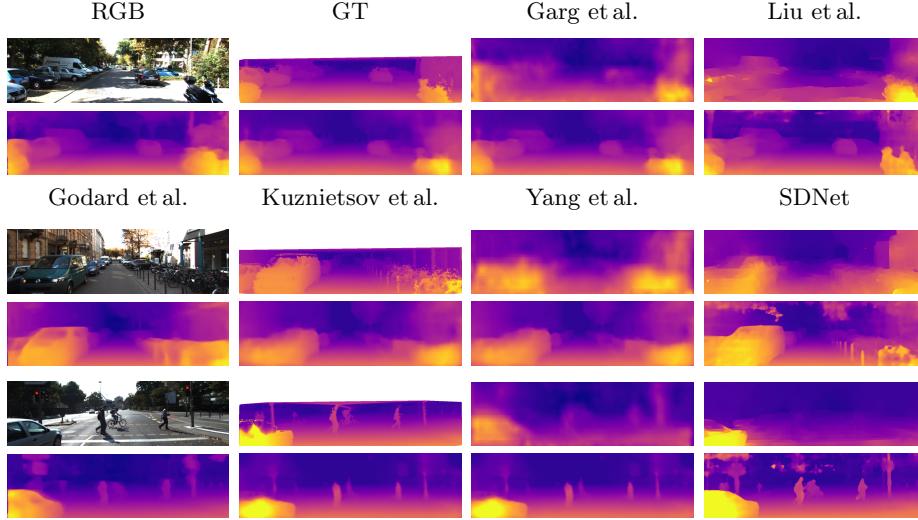


Fig. 4: Qualitative comparison with other state-of-the-art methods on KITTI Eigen test dataset. The ground truth depth maps are interpolated from the sparse LIDAR measurements for better visualization. Our approach can retrieve the depths for small objects and fine structures better than other methods.

in this benchmark can be found in table 3. Although, the focus of SDNet is the estimation of depth and in consequence the semantic segmentation has not been purposefully optimized and has only been used to learn more meaningful features, SDNet also achieves state-of-the-art results in this benchmark.

In figure 6, sample results are shown on this test dataset. It can be seen that the results of semantic segmentation and depth estimation correlate directly with each other. For example, if a pixel is assigned to a wrong semantic class, then mostly the depth prediction is also incorrect. If, on the other hand, the depth or semantic class has been correctly classified, then in most cases the other output is correct as well.

### 4.3 Cityscapes Dataset

Both outputs of SDNet were also evaluated on CS to show the generalization capability of SDNet. Unfortunately, there is no benchmark for depth estimation, which is why the results could only be assessed qualitatively. In figure 7, the estimated depth maps of SDNet and [14] are exemplarily shown for CS. One can clearly see that the SDNet depths are more detailed compared to Kuznetsov et al.. For example, the depth for the persons in front of the Brandenburg Gate and the gate itself are retrieved well by SDNet in contrast to [14], where the depths are noticeably more blurry. Aside from the depths, the semantic segmentation of SDNet is also shown in this figure. Qualitatively, the predicted

Method	SiLog [ $100 \cdot \log(m)$ ]	SRD in %	ARD in %	iRMSE [1/km]
DORN [6]	11.67	2.21	9.04	12.23
VGG16-UNet [10]	13.41	2.86	10.06	15.06
DABC [17]	14.49	4.08	12.72	15.53
<b>SDNet</b>	14.68	3.90	12.31	15.96
APMoE [13]	14.74	3.88	11.74	15.63
CSWE [16]	14.85	3.48	11.84	16.38
DHGRL [33]	15.47	4.04	12.52	15.72

Table 2: Official results of the KITTI depth prediction benchmark [27] (as of May 01, 2019). This table includes only already published methods. If several results of one method are ranked, then only the best is taken.

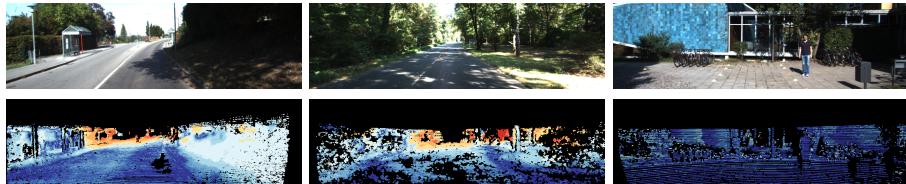


Fig. 5: D1 error of the SDNet depth estimate from the ground truth on three exemplary images from the KITTI Depth Prediction test dataset. The errors were computed by KITTI and blue represents a small deviation and red a large one.

semantics yield correct and good results, with a few exceptions. This can also be shown by the quantitative results on CS (see supplementary materials).

#### 4.4 Ablation Studies

We also perform ablation studies on the KD validation data to empirically prove that our contributions/enhancements result in verifiable improvements. These experiments can be divided into three parts: i.e. loss function, discretization levels and architecture. The results are shown in table 4. As a baseline configuration for SDNet, a ResNet-50 with group normalization layers is chosen as encoder and the proposed ordinal BCE loss is used as loss function from equation 5. Thus, the CNN is trained only on the depths and the semantic loss is completely ignored. The depth interval [2m; 80m] is divided into  $M_d = 128$  classes.

The performance of the network can be increased, if the semantic loss is added to the loss. Hence, the network can learn superior features that do not solely depend on the depth, but also on the semantics. As a result, as shown here empirically, a better performance can be achieved. In addition, the ordered classification of the depths is superior to a regression approach with MAE.

The depth estimation achieves the best results, when dividing the depth interval into 128 classes. Although the accuracy is larger for less classes, the

Method	IoU class	iIoU class	IoU cat.	iIoU cat.
SegStereo [31]	59.10	28.00	81.31	60.26
<b>SDNet</b>	51.14	17.74	79.62	50.45
APMoE [13]	47.96	17.86	78.11	49.17

Table 3: Official results of the KITTI pixel-level semantic segmentation benchmark [1] (as of May 01, 2019). This list includes only already published methods.

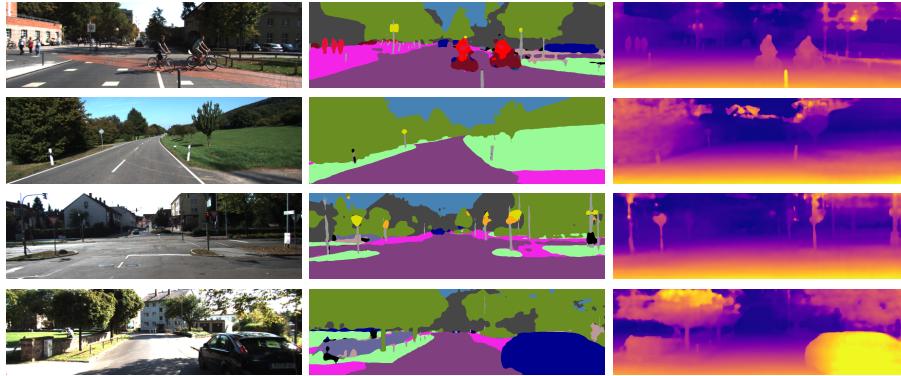


Fig. 6: Qualitative results of semantic segmentation and depth estimation using examples from KS test dataset.

prediction results get worse, because the discretization error increases with fewer number of classes. On this basis, 128 classes have proven to be the best trade-off.

Replacing the ResNet-50 by a ResNet-101 as an encoder, was supposed to increase performance, because even better features can be extracted. However, the evaluation shows that exactly the opposite was the case. The reasons for this could be that larger networks generally are harder to train. Therefore, adjustments or fine-tuning had to be made in the training protocol, which was not done (e.g. additional warm-up phase or adjustment of the learning rate). Additionally, the change of the normalization method was also evaluated. By default, batch normalization layers are used in the ResNets, but these proved disadvantageous for small batch sizes. Therefore, these layers have been replaced by group normalization layers, which had a positive effect on all metrics.

## 5 Summary & Conclusion

In this paper, we propose the novel SDNet for the simultaneous prediction of pixel-wise depths and semantic labels from a single monocular image. The architecture of SDNet is based on the DeepLabv3+ model, which we have extended. The depths are determined by ordered classes rather than the classic regression-based approach. In comparison to other methods, the input image is also seg-

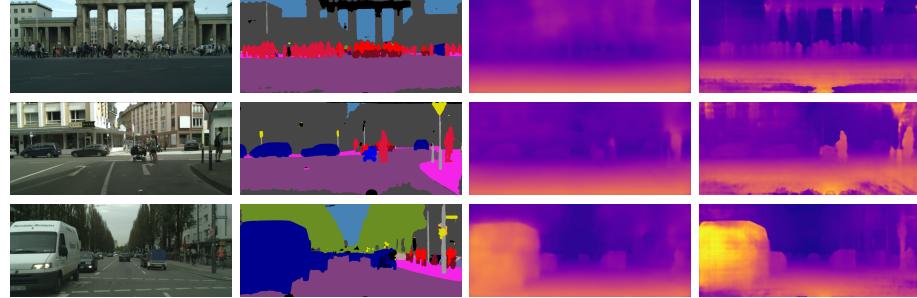


Fig. 7: Qualitative results of the semantic segmentation and the depth estimation based on exemplary examples from the CS test dataset. The first column depict the input image and the second the predicted semantics. In the last two columns, the depth maps of [14] (third column) and SDNet (last column) are shown.

Method	SiLog [ $100 \cdot \log(m)$ ]	ARD in %	RMSE [m]	iRMSE [1/km]	Accuracy in %
sem. loss ( $\lambda = 10$ )	14.83	9.39	3.11	12.17	20.10
MAE	18.80	12.11	4.03	15.30	-
$M_{\text{depth}} = 96$	16.04	10.88	3.30	13.28	22.50
$M_{\text{depth}} = 160$	15.57	10.06	3.23	12.87	15.50
BatchNorm	15.84	10.31	3.42	12.77	18.95
ResNet-101	16.25	10.89	3.44	13.12	18.24
baseline	15.38	9.92	3.30	12.55	20.50

Table 4: Ablation studies on the KD validation dataset for different configurations. As baseline, we use SDNet without semantic loss ( $\lambda = 0$ ).

mented semantically. These two modifications turn out to be beneficial as the CNN learns semantically richer features, resulting in significant better results.

In a further stage, a new error or uncertainty measure could be developed on basis of the output, because prediction errors occur mostly in both predictions. This knowledge could be used for such a measure. Another improvement of SDNet would be that a better encoder than ResNet-50 is used. But this implies also that hyper-parameters must be tuned during training, which by itself could be an improvement. Nevertheless, SDNet yields state-of-the-art results for both semantic segmentation and depth estimation on various datasets.

**Acknowledgement:** This project (HA project no. 626/18-49) is financed with funds of LOEWE Landes-Offensive zur Entwicklung Wissenschaftlich-kognitiver Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## References

1. Alhaija, H., Mustikovela, S., Mescheder, L., Geiger, A., Rother, C.: Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)* **126**(9), 961–972 (2018)
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision (ECCV). pp. 833–851 (2018)
3. Cheng, J., Wang, Z., Pollastri, G.: A neural network approach to ordinal regression. In: International Joint Conference on Neural Networks. pp. 1279–1284 (2008)
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3213–3223 (2016)
5. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems (NIPS). pp. 2366–2374 (2014)
6. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2002–2011 (2018)
7. Garg, R., Kumar BG, V., Carneiro, G., Reid, I.: Unsupervised CNN for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision (ECCV). pp. 740–756 (2016)
8. Gidaris, S., Komodakis, N.: Detect, replace, refine: Deep structured prediction for pixel wise labeling. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7187–7196 (2017)
9. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6602–6611 (2017)
10. Guo, X., Li, H., Yi, S., Ren, J., Wang, X.: Learning monocular depth by distilling cross-domain stereo networks. In: European Conference on Computer Vision (ECCV). pp. 484–500 (2018)
11. Hirschmller, H.: Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **30**(2), 328–341 (2008)
12. Knobelreiter, P., Reinbacher, C., Shekhovtsov, A., Pock, T.: End-to-end training of hybrid CNN-CRF models for stereo. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1456–1465 (2017)
13. Kong, S., Fowlkes, C.: Pixel-wise attentional gating for parsimonious pixel labeling. In: Winter Conference on Applications of Computer Vision (WACV) (2019)
14. Kuznietsov, Y., Stckler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2215–2223 (2017)
15. Ladick, L., Shi, J., Pollefeys, M.: Pulling things out of perspective. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 89–96 (2014)
16. Li, B., Dai, Y., He, M.: Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition* **83**, 328 – 339 (2018)
17. Li, R., Xian, K., Shen, C., Cao, Z., Lu, H., Hang, L.: Deep attention-based classification network for robust depth prediction. CoRR **arXiv**, 1807.03959 [cs.CV] (2018)

18. Lin, T., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2018)
19. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **38**(10), 2024–2039 (2016)
20. Liu, M., Salzmann, M., He, X.: Discrete-continuous depth estimation from a single image. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 716–723 (2014)
21. Mayer, N., Ilg, E., Husser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4040–4048 (2016)
22. Niu, Z., Zhou, M., Wang, L., Gao, X., Hua, G.: Ordinal regression with multiple output cnn for age estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4920–4928 (2016)
23. Pang, J., Sun, W., SJ. Ren, J., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: International Conference on Computer Vision (ICCV) - Workshop. pp. 887–895 (2017)
24. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Advances in Neural Information Processing Systems (NIPS). pp. 1161 – 1168 (2005)
25. Saxena, A., Sun, M., Ng, A.Y.: Make3D: Learning 3D scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **31**(5), 824–840 (2009)
26. Schneider, L., Cordts, M., Rehfeld, T., Pfeiffer, D., Enzweiler, M., Franke, U., Pollefeys, M., Roth, S.: Semantic stixels: Depth is not enough. In: Intelligent Vehicles Symposium (IV). pp. 110–117 (2016)
27. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant CNNs. In: International Conference on 3D Vision (3DV) (2017)
28. Žbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research (JMLR)* **17**, 2287–2318 (2016)
29. Wu, Y., He, K.: Group normalization. In: European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
30. Xie, J., Girshick, R., Farhadi, A.: Deep3D: fully automatic 2D-to-3D video conversion with deep convolutional neural networks. In: European Conference on Computer Vision (ECCV). pp. 842–857 (2016)
31. Yang, G., Zhao, H., Shi, J., Deng, Z., Jia, J.: Segstereo: Exploiting semantic information for disparity estimation. In: European Conference on Computer Vision (ECCV). pp. 660–676 (2018)
32. Yang, N., Wang, R., Stückler, J., Cremers, D.: Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In: European Conference on Computer Vision (ECCV). pp. 835–852 (2018)
33. Zhang, Z., Xu, C., Yang, J., Tai, Y., Chen, L.: Deep hierarchical guidance and regularization learning for end-to-end depth estimation. *Pattern Recognition* **83**, 430 – 442 (2018)