Supplementary Material for Sparsity Invariant CNNs

Jonas Uhrig[⋆,1,2] Nick Schneider[⋆,1,3] Lukas Schneider[1,4]
Uwe Franke[1] Thomas Brox[2] Andreas Geiger[4,5]

[1]Daimler R&D Sindelfingen [2]University of Freiburg
[3]KIT Karlsruhe [4]ETH Zürich [5]MPI Tübingen
{jonas.uhrig,nick.schneider}@daimler.com

1. Convergence Analysis

We find that Sparse Convolutions converge much faster than standard convolutions for most input-output-combinations, especially for those on Synthia with irregularly sparse depth input, as considered in Section 5.1 of the main paper. In Figure 1, we show the mean average error in meters on our validation subset of Synthia over the process of training with identical solver settings (Adam with momentum terms of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and delta $1\mathrm{e}{-8}$). We chose for each variant the maximal learning rate which still causes the network to converge (which turned out to be $1\mathrm{e}{-3}$ for all three variants). We find that Sparse Convolutions indeed train much faster and much smoother compared to both ConvNet variants, most likely caused by the explicit ignoring of invalid regions in the update step. Interestingly, the ConvNet variant with concatenated visibility mask in the input converges smoother than the variant with only sparse depth in the input, however, additionally incorporating visibility masks seems to reduce overall performance for the task of depth upsampling.

2. Semantic Segmentation

2.1. Detailed Results on Synthia

Relating to Section 5.3 of the main paper, we show in Table 1 the class-wise IoU for semantic labeling on 5% sparse input data and compare the three proposed VGG-like variants: Convolutions on depth only, convolutions on depth with concatenated visibility mask, and sparse convolutions using depth and visibility mask. We find that sparse convolutions learn to predict also less likely classes, while standard convolutions on such sparse data even struggle to get the most likely classes correct.

2.2. Semantic Segmentation on Real Depth Maps

Many recent datasets provide RGB and aligned depth information along with densely annotated semantic labels, such as Cityscapes [**?**] and SUN-RGBD [**?**]. Many state-of-the-art approaches incorporate depth as well as RGB information in order to achieve highest performance for the task of semantic segmentation [**?**]. As the provided depth maps are often not

Table 1: Evaluation of the class-level performance for pixel-level semantic labeling on our Synthia validation split subset (*'Cityscapes'*) after training on all Synthia *'Sequence'* subsets using the Intersection over Union (IoU) metric. All numbers are in percent and larger is better.

	sky	building	road	sidewalk	fence	vegetation	pole	car	traffic sign	pedestrian	bicycle	lanemarking	traffic light	**mean IoU**
VGG - Depth Only	27.1	30.3	25.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.4
VGG - Depth + Mask	20.9	27.8	14.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.9
VGG - Sparse Convolutions	**95.3**	**59.0**	**33.0**	**17.2**	**1.0**	**60.5**	**28.7**	**33.0**	**12.5**	**35.6**	**6.1**	**0.5**	**22.4**	**31.1**

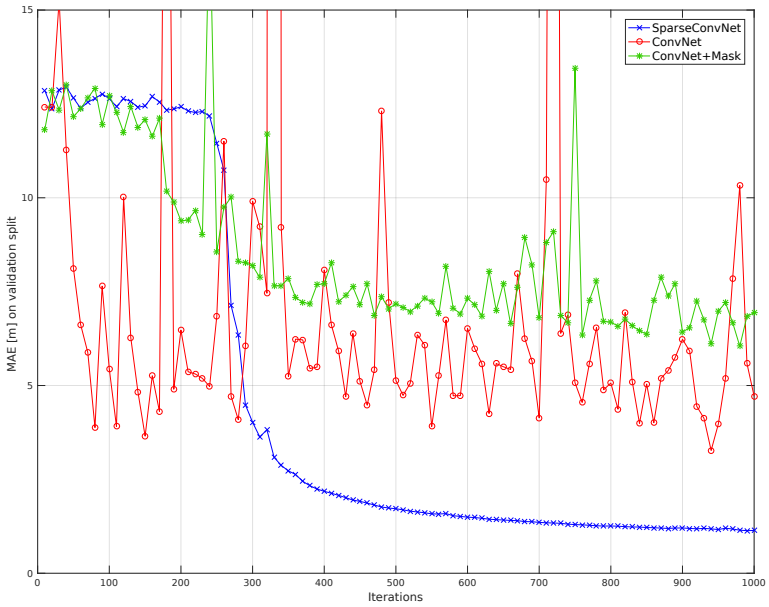

Figure 1: Convergence of the three considered network baselines from Section 5.1 of the main paper for the task of sparse depth upsampling on 5% dense input depth maps from our Synthia train subset.

Table 2: Performance comparison of different input and convolution variants for the task of semantic labeling on (sparse or filled) depth maps from the SUN-RGBD dataset [**?**]. All networks are trained from scratch on the training split using 37 classes, performance is evaluated on the test split as mean IoU, *c.f.* [**?**].

Convolution Type	Input Depth	Visibility Mask?	IoU [%]
Standard	Raw Depth	No	7.697
Standard	Filled Depth	No	10.442
Standard	Raw Depth	Concatenated	18.971
Standard	Filled Depth	Concatenated	18.636
Sparse	Raw Depth	Yes	**19.640**

completely dense, we propose to use sparse convolutions on the depth channel instead of filling depth maps artificially and applying dense convolutions afterwards.

We conduct experiments on SUN-RGBD with only depth maps as input to show the benefit of using sparse convolutions over traditional convolutions. As seen in Section 5.3 of the main paper, sparse convolutions help to incorporate missing depth information in the input for very sparse (5%) depth maps. In Table 2 we show the performance of a VGG16 (with half the amount of channels than usual) trained from scratch for the task of semantic labeling from (sparse) depth maps. We apply skip connections as used throughout literature [**?**, **?**] up to half the input resolution. We compare performance on the provided raw sparse depth maps (*raw*, *c.f.* Figure 3) as well as a dense depth map version obtained from a special inpainting approach using neighboring frames (*filled*) on the SUN-RGBD test dataset, as well as the used convolution type (sparse or standard). We find that sparse convolutions perform better than standard convolutions, on both raw and filled depth maps, no matter if a visibility map is concatenated to the input depth map or not. Like reported in [**?**], standard convolutions on the raw depth maps do perform very bad, however, we find that concatenating the visibility map to the input already doubles the achieved performance. A detailed class-wise performance analysis can be found in Table 3. Note that missing information in the input, like missing depth measurements in the SUN-RGBD dataset, does not always cause less information, which we discuss in the following section. This phenomenon boosts methods that explicitly learn convolutions on a visibility mask, such as the two standard convolution networks with concatenated visibility masks. Although we do not explicitly extract features of the visibility masks we still outperform the other convolution variants.

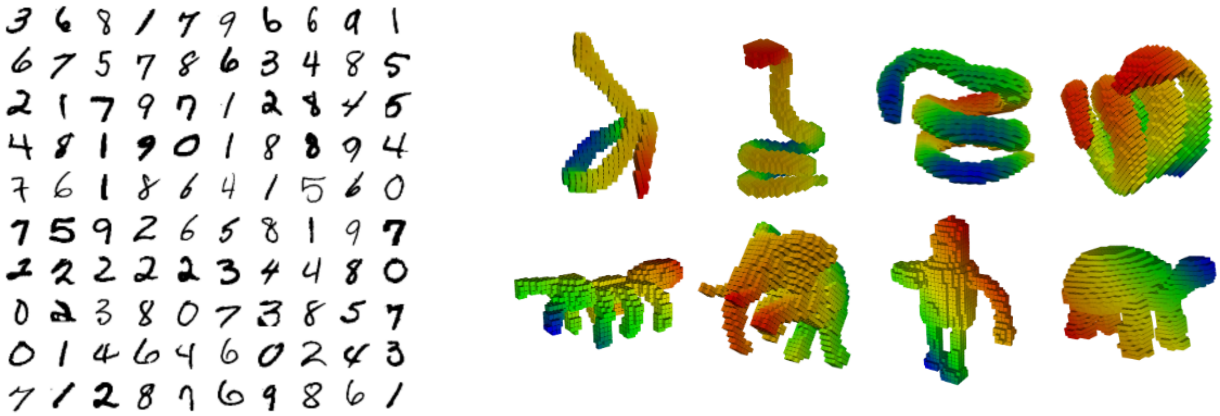

Figure 2: Missing data sometimes contains useful information as in the example of handwritten digit classification or 3D CAD model classification. Examples are taken from LeCun *et al*. [**?**] and Graham [**?**].

Figure 3: Active sensors such as ToF cameras might contain missing values because of strongly reflecting surfaces. However, the missing data clearly outlines the shape of certain objects and therefore gives a hint for semantic segmentation. This example is taken from the SUN-RGBD dataset [**?**].

2.3. Discussion: Missing data is not always missing information

In our experiments we recognized that missing data might sometimes be helpful for certain tasks. Let's consider *e.g*. digit classification [**?**] or shape recognition from 3D CAD models as depicted in Figure 2. For both cases the relation between invalid (background) and valid pixels/voxels is indispensable information for the classification. We want to stress that our approach does not tackle such cases. Instead it handles cases where unobserved data is irregularly distributed and does not contain additional information. Therefore, the missing data harms the results of the convolution.

Data from active sensors, such as Time-of-Flight (ToF) cameras used in the SUN-RGBD dataset, is often sparse as shown in Figure 3. However, the missing data might contain a pattern if *e.g*. only certain materials do reflect the emitted light. This might be the reason why the results in Table 2 show a significant improvement for standard convolutions if the visibility mask is concatenated. Our Sparse Convolution Network does not consider any missing data. Therefore, it might miss information encoded in the visibility mask. Although, the proposed method outperforms the naïve approaches, considering the valid mask explicitly will likely further improve the performance of our method.

Table 3: Evaluation of the class-level performance for pixel-level semantic labeling on Synthia Cityscapes subset after training on all Synthia Sequence subsets using the Intersection over Union (IoU) metric. All numbers are in percent and larger is better. Our sparse convolutions outperform the other variants on 18 classes, standard convolutions on filled depth with concatenated visibility mask outperform the others on 11 classes, and on 8 classes standard convolutions on raw depth with concatenated mask perform best.

	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	blinds	desk	shelves	curtain	dresser	pillow	mirror	floor mat	clothes	ceiling	books	fridge	tv	paper	towel	shower curtain	box	whiteboard	person	night stand	toilet	sink	lamp	bathtub	bag	**mean IoU**
Conv. Raw Depth	49.5	72.3	0.2	9.4	26.5	5.5	29.3	0.2	17.9	2.6	0.0	6.3	0.0	0.0	0.3	9.0	0.0	7.4	0.0	0.0	0.2	45.3	0.1	0.0	0.0	0.6	0.1	0.0	0.0	0.0	0.0	0.0	1.5	0.4	0.0	0.0	7.7
Conv. Filled Depth	53.1	76.1	8.7	19.6	34.5	8.5	34.5	0.3	9.1	10.4	0.5	12.3	0.0	0.0	0.0	27.6	0.0	14.0	0.1	0.0	1.3	48.9	5.1	0.0	0.0	0.1	0.8	0.0	0.0	0.0	0.0	0.0	2.1	8.4	7.6	2.9	0.1	10.4
Conv. Raw Depth, Mask concat.	59.4	80.2	28.7	53.3	49.0	37.3	**42.9**	2.7	21.7	**17.7**	7.3	**22.3**	0.0	5.2	0.9	**35.6**	11.4	21.4	14.0	0.0	**6.3**	34.8	10.0	**6.4**	4.6	0.1	**8.6**	0.0	4.7	12.1	**2.7**	0.1	35.0	30.2	9.2	23.3	2.7	19.0
Conv. Filled Depth, Mask concat.	59.9	**81.6**	**29.2**	52.5	**50.7**	**38.6**	42.6	0.6	15.3	16.9	**11.1**	17.0	0.1	0.5	0.2	20.5	**12.5**	11.3	16.2	0.0	4.0	38.0	**18.9**	4.3	5.4	0.0	5.5	0.0	3.6	**15.7**	0.0	**9.3**	32.9	27.4	**17.0**	**29.9**	0.6	18.6
Sparse Conv. Raw Depth	**60.1**	80.7	26.9	**54.2**	50.3	34.7	40.5	**9.3**	**22.0**	11.0	10.0	16.6	**4.0**	**8.5**	**3.0**	20.7	10.7	**23.2**	**17.9**	**0.0**	3.8	**44.5**	10.2	6.2	**6.9**	**2.5**	5.2	**4.6**	**5.0**	15.3	1.2	2.8	**42.9**	**31.6**	11.2	26.4	**3.0**	**19.6**

Table 4: Evaluation of differently generated depth map variants using the manually annotated ground truth disparity maps of 142 corresponding KITTI benchmark training images [**?**]. Best values per metric are highlighted. Cleaned Accumulation describes the output of our automated dataset generation without manual quality assurance, the extension '+ SGM' describes an additional cleaning step of our depth maps with SGM depth maps, applied mainly to remove outliers on dynamic objects. All metrics are computed in the disparity space.

	Density	MAE	RMSE	KITTI outliers	δ_i inlier rates δ_1	δ_2	δ_3
SGM	82.4%	1.07	2.80	4.52	97.00	98.67	99.19
Raw LiDaR	4.0%	0.35	2.62	1.62	98.64	99.00	99.27
Acc. LiDaR	30.2%	1.66	5.80	9.07	93.16	95.88	97.41
Cleaned Acc.	16.1%	**0.35**	**0.84**	**0.31**	**99.79**	**99.92**	**99.95**

3. Detailed Dataset Evaluation

Relating to Section 4.1 of the main paper, we manually extract regions in the image containing dynamic objects in order to compare our dataset's depth map accuracy for foreground and background separately. Various error metrics for the 142 KITTI images with corresponding raw sequences, where we differentiate between the overall average, *c.f.* Table 4, as well as foreground and background pixels, *c.f.* Tables 5 and 6.

We find that our generated depth maps have a higher accuracy than all other investigated depth maps. Compared to raw LiDaR, our generated depth maps are four times denser and contain five times less outliers in average. Even though we lose almost 50% of the density of the LiDaR accumulation through our cleaning procedure, we achieve almost 20 times less outliers on dynamic objects and even a similar boost also on the static environment. This might be explained through the different noise characteristics in critical regions, *e.g.* where LiDaR typically blurs in lateral direction on depth edges, SGM usually blurs in longitudinal direction. In comparison to the currently best published stereo algorithm on the KITTI 2015 stereo benchmark website [**?**], which achieves 2.48, 3.59, 2.67 KITTI outlier rates for background, foreground and all pixels (anonymous submission, checked on April 18th, 2017), the quality of our depth maps is in the range of 0.23, 2.99, 0.84. Therefore, besides boosting depth estimation from single images (as shown in Section 5), we hope to also boost learned stereo estimation approaches.

Table 5: Evaluation as in Table 4 but only for Foreground pixels.

Depth Map	MAE	RMSE	KITTI outliers	δ_i inlier rates δ_1	δ_2	δ_3
SGM	1.23	2.98	5.91	97.6	98.2	98.5
Raw LiDaR	3.72	10.02	17.36	84.29	86.11	88.56
Acc. LiDaR	7.73	12.01	59.73	55.67	73.73	83.04
Cleaned Acc.	**0.88**	**2.15**	**2.99**	**98.55**	**98.96**	**99.17**

Table 6: Evaluation as in Table 4 but only for Background pixels.

Depth Map	MAE	RMSE	KITTI outliers	δ_i inlier rates δ_1	δ_2	δ_3
SGM	1.05	2.77	4.36	96.93	98.72	99.27
Raw LiDaR	**0.22**	1.90	0.94	99.25	99.56	99.73
Acc. LiDaR	1.09	4.81	4.25	96.74	97.99	98.78
Cleaned Acc.	0.34	**0.77**	**0.23**	**99.83**	**99.94**	**99.97**

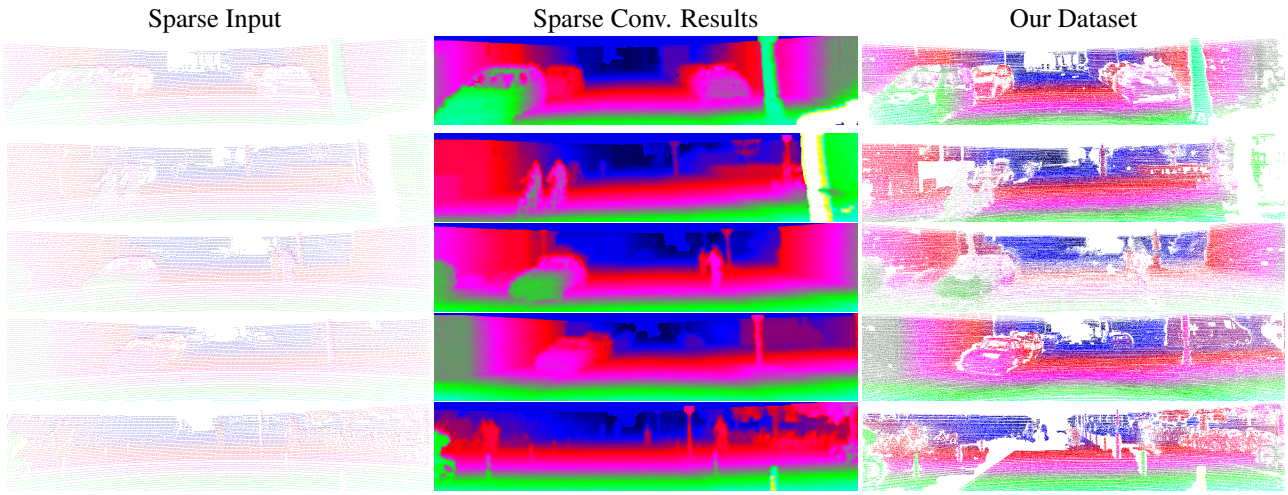

Figure 4: Further qualitative results of our depth upsampling approach on the KITTI dataset with corresponding sparse depth input and our generated dense depth map dataset.

4. Further Depth Upsampling Results

We show more results of our depth upsampling approach in Figure 4. The input data of the Velodyne HDL64 is sparse and randomly distributed when projected to the image. Our approach can handle fine structures while being smooth on flat surfaces. Sparse convolutions internally incorporate sparsity in the input and apply the learned convolutions only to those input pixels with valid depth measurements.

5. Boosting Single-Image Depth Prediction

As promised in Section 4 of the main paper, we conducted several experiments for a deep network predicting depth maps from a single RGB image, *e.g.* as done by [**?**, **?**, **?**] and many more. Due to the lack of training code and to keep this study independent of current research in loss and architecture design, we chose the well-known VGG16 architecture [**?**] with weights initialized on the ImageNet dataset [**?**] and vary only the used ground truth. For a fair comparison, we use the same amount of images and the same sequence frames for all experiments but adapt the depth maps: Our generated dataset (denser than raw LiDaR and even more accurate), sparse LiDaR scans (as used by most approaches for depth prediction on KITTI scenes), as well as depth maps from semi-global matching (SGM) [**?**], a common real-time stereo estimation approach, *c.f.* Table 7 (bottom). We evaluate the effect of training with the standard L1 and L2 losses, but do not find large performance differences, *c.f.* Table 7 (top). Also, we compare the difference between an inverse depth representation, as suggested in the literature [**?**, **?**], as well as an absolute metric representation, *c.f.* Table 7 (top). Surprisingly, we find that absolute depth values as ground truth representation outperform inverse depth values. We use the best setup (absolute depth with L2 loss due to faster convergence) to evaluate the performance on our test split, where our dataset outperforms the other most promising depth maps from raw LiDaR, *c.f.* Table 7 (bottom).

We find that our generated dataset produces visually more pleasant results and especially much less outliers in occluded regions, *c.f.* the car on the left for the second and last row of Figure 5. Also, our dense depth maps seem to help the networks to generalize better to unseen areas, such as the upper half of the image. We hope that our dataset will be used in the future to further boost performance for this challenging task.

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *arXiv.org*, 1411.4734, 2014.

[3] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Table 7: Evaluation of different depth ground truth and loss variants (top) used for training a VGG16 on single-image depth prediction. L1 and L2 loss achieve comparable performance, while absolute depth representation for training instead of inverse depth performs significantly better. We compare performance on our generated validation and test split, as well as 142 ground truth depth maps from KITTI 2015 [**?**] for the best performing setup with L2 loss on absolute depth (bottom).

Depth Maps	Loss	Inverse Depth?	MAE			RMSE		
			val	test	KITTI'15	val	test	KITTI'15
Our Dataset	L2	yes	2.980			6.748		
Our Dataset	L1	yes	2.146			4.743		
Our Dataset	L2	**no**	2.094			**3.634**		
Our Dataset	L1	**no**	**2.069**			3.670		
Our Dataset	L2	no	**2.094**	**1.913**	**1.655**	**3.634**	**3.266**	**3.275**
Raw LiDaR Scans	L2	no	2.184	1.940	1.790	3.942	3.297	3.610
SGM	L2	no	3.278	2.973	3.652	5.826	4.811	8.927

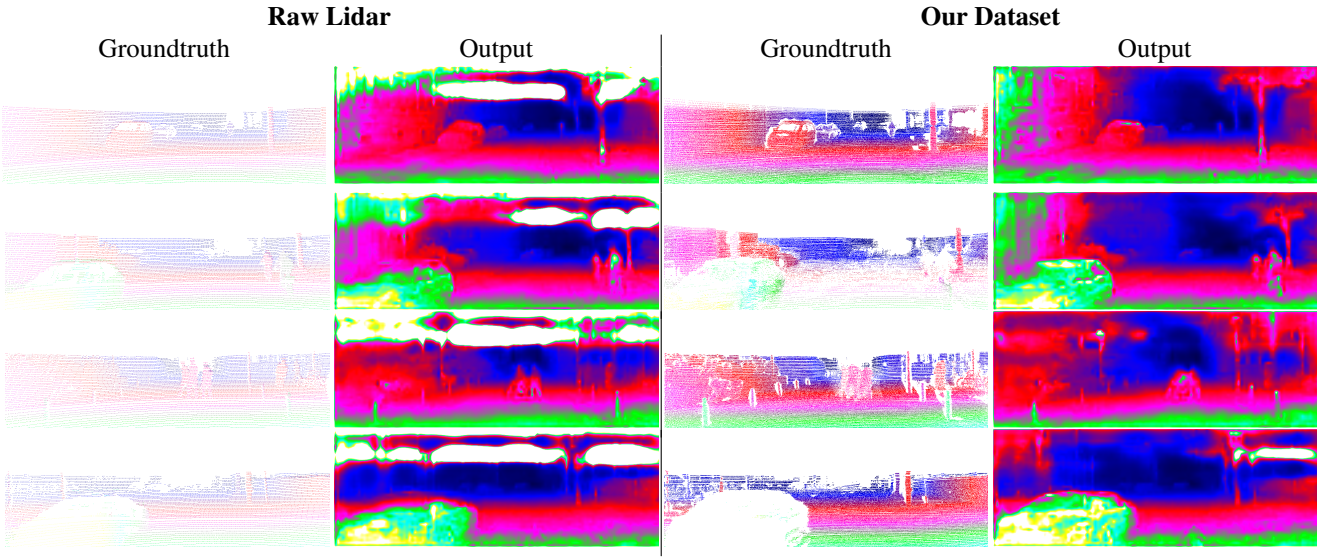

Figure 5: **Raw Lidar vs our dataset as training data for depth from mono:** Qualitative examples of the depth-from-mono CNN trained on our generated dense and outlier-cleaned dataset in contrast to the sparse raw LiDaR data. It becomes apparent that denser training data leads to improved results *e.g.* in the upper half of the image and at object boundaries (where most LiDaR outliers occur).

[4] B. Graham. Sparse 3d convolutional neural networks. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2015.

[5] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. ¡a href="https://hazirbas.github.io/projects/fusenet/" target="$_blank$" $> FuseNet : IncorporatingDepthintoSemanticSegmentationviaFusion - basedCNNArchitecture < /a > .InAsianConferenceonComputerVision$, 2016.$< ahref = "https : //github.com/tum - vision/fusenet"target = "_blank" >$ [code] $<$

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):328–341, 2008.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[10] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.

[11] F. Liu, C. Shen, G. Lin, and I. Reid. Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[14] S. Song, S. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[15] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.