Segmentation-Guided Self-Supervised Monocular Depth Estimation

Hakan Sivuk
Bilkent University
Ankara, Turkey
www.github.com/hakansivuk/SegDepth

Abstract

Depth estimation is an important topic of computer vision, especially in autonomous driving and robotics. Sensors that can sense the depth are used in several projects but their high prices make them less accessible. There are also self-supervised techniques for this task, however, the number of available ground truth depth data is low. Both these facts have increased popularity of monocular depth estimation techniques that do not require ground truth data and exploits data itself as a supervision signal during the training.

Some of the research on this area focus on improving the current loss functions used for this task. For this aim, complex architectures and new designed loss functions are used. In this paper, we showed that adding a segmentation network to the overall architecture can improve the training performance especially for finding the global minimum and better generalizing performance without causing an extra inference time during the test.

1. Introduction

Depth estimation is a computer vision task which has been studied in recent years. LIDAR sensors can provide accurate estimations by targeting an object with a laser light and measuring the distance according to its return time. However, these sensors are quite expensive and its price affects its usage negatively. With the increasing popularity of deep learning, supervised depth estimation techniques have been used widely. However, there is no much ground truth dataset available for this task.

As a result of these difficulties, self-supervised methods have been increased. Stereo depth estimation networks may infer depth values by comparing two frames from stereo cameras. Also, using monocular videos is sufficient to train a depth estimation model. But, an additional pose estimation model is required to get a supervision signal by using two consecutive frames. Although there are several possibilities for this task, estimating depth from videos, or stereo cameras is an ill-posed problem which means there are more than one possibilities that are correct according to supervision signals. Apart from that, the reprojections loss, generally used loss function, has difficulties to find the global minimum.

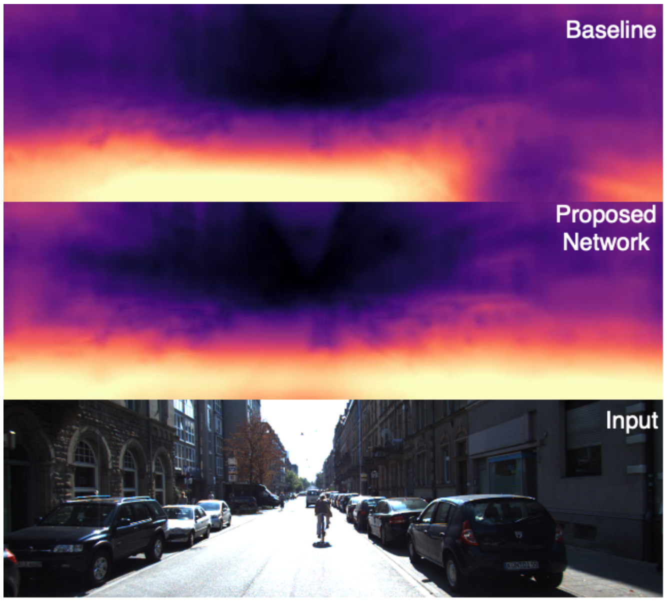

Figure 1. A sample inference. At the top, inference of the baseline (Monodepth2). At the middle, inference the proposed network. At the bottom, input image.

There are several techniques attempting to improve these problems by using complex architectures, new designed loss functions, etc. We used MonoDepth2 as a baseline network and proposed an improved network with an additional semantic segmentation network. By adding U-net segmentation network, we tried to exploit the close relationship between depth estimation and semantic segmentation tasks. Indeed, according to results, there are improvement comparing to baseline network.

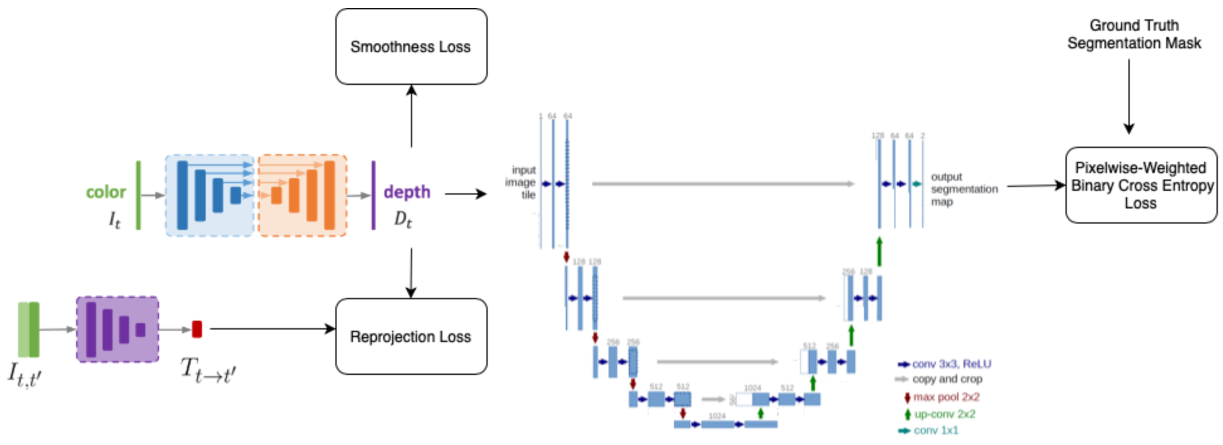

Figure 2. The overall network. As a result of depth encoder-decoder network and pose estimation encoder network, disparity maps at different scales and pose estimation with respect to the target image is created respectively. Disparity map at the highest resolution (the same with the input 640x192) is passed to the U-net architecture and segmentation mask is generated as an output. It basically separates objects from background. Finally, as a result of pixel-wise weighted binary cross entropy loss, segmentation loss value is calculated.

2. Related Work

In this section, some studies that are prior or related to our work. Among monocular depth estimation studies, there are different techniques for loss functions, network architectures. Besides, works that combine segmentation and depth estimation tasks are also mentioned as a part of literature review.

2.1. Self-Supervised Depth Estimation

As it is mentioned in the previous sections, a lack of availability of ground truth depth data has shifted some of the researchers to Self-Supervised Depth Estimation. Monocular Depth Estimation is a branch of Self-Supervised Depth Estimation which requires a monocular data and uses temporal frames as supervision signals. Most of the studies use videos, which can be accessed easily. To close the gap between Supervised Depth Estimation and Self-Supervised Depth Estimation, these studies focus on different issues. As [15] tried to use different loss spaces and loss functions, [2, 7] added a feature encoder-decoder network for better results by learning feature representations of images. To overcome expensive computation for high resolution images and loss of details in low resolution images, [10] implemented a dual network contains a high resolution and a low resolution network. Attention, which is one of the most popular deep learning techniques recently, is integrated into this task by [1, 14]

All of these studies, attempted to use different loss spaces, complex architectures. In that regard, [4, 8] drawn an attention with its simple improvements and SOTA results. [4] They tried to address 3 problems. The first is the occluded pixel problem which means the pixels that are not visible in some of the source images and visible in the target image. Since these pixels are not matched for the source and target images, even if the network predicts depth values correctly, it would result in high error penalties for these pixels. The second problem is the violation of the moving camera, static scene assumption. For example when the car is stopped or some objects in the scene are moving, these assumptions break down. As a result, some infinite depth holes may appear in the test time especially for the objects seen usually as moving in the training time. Finally, most of the networks use multi-scale depth prediction and image reconstruction to prevent the training objective from getting stuck in a local minimum. This multi-scale approach finds total loss as a combination of loss values at each scale. The third problem is that this approach may result in holes in large low-texture regions in depth maps.

To prevent these problems they offer three simple improvements:

(i) instead of averaging photometric error over the source images, taking the minimum,

(ii) creating binary masks to ignore pixels violating the assumption. The pixels whose reprojection error for warped image is larger than reprojection error for unwarped source image do not contribute to the loss due to this masking method,

(iii) instead of combining the loss at each scale directly, first upsample each low resolution depth map to the image resolution and compute the loss value at this high resolution. In that way, depth maps from each

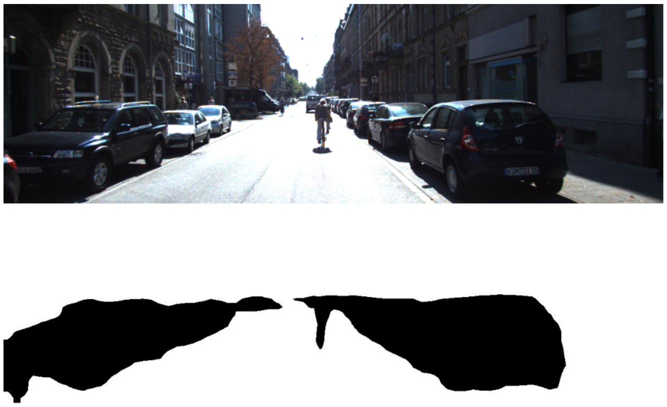

Figure 3. A segmentation mask generated by DeepLabv3. Areas covered with black show foreground objects and remaining parts are background. It basically separates objects from background.

scale have the same objective which is reconstructing the target image accurately.

They used photometric reprojection loss with taking minimum over the source images and edge-aware smoothness loss. Finally, the total loss is a combination of per-pixel photometric error and edge-aware smoothness loss. Due to their simplicity and effective results, these techniques are also used within the many studies yielding state of the art results.

2.2. Semantic Segmentation

Semantic segmentation task can be seen related to depth estimation if a segmentation area is considered as a set of points with similar depth values. There are several studies exploiting this close relationship. [18] uses depth and segmentation together for a better learning performance.

Depth information is used for semantic segmentation in different studies. [5, 6, 13] use point clouds to improve semantic segmentation performance. Similarly, [16, 19] use RGB-D data for better performance.

When we look at studies that exploit semantic segmentation as a booster for depth estimation tasks, there are fewer works. As [12, 11] use segmentation masks as one of the structure information and guide learning process with this structure information, [9] improves depth estimation with a semantic segmentation based sampling.

3. Proposed Approach

3.1. Baseline Network

In this study, Monodepth2 is used as baseline network because of simple but effective improvements. Simplicity of its novelties allows us to implement additional novelties easily. As it can be seen from 2 disparity maps at different scales are created through an encoder-decoder depth network. Also, pose estimations are generated by passing source and target frames to pose encoder network. After that instead of combining the loss at each different scale, they up-sample disparity maps to the original resolution and calculate minimum re-projection loss values at this scale. In addition to that, edge-aware smoothness loss value is calculated through generated depth maps.

Network's input dimension is 3x192x640 and it generates depth maps at different scale but the highest output dimension is 1x192x640.

They used re-projection loss as many other self-supervised depth estimation studies. Normally re-projection error is summed over source images. Reprojection loss is calculated for target image and its estimated pose with respect to source image.

$$L_p = \sum_{t'} pe(I_t, I_{t'\rightarrow t}),$$

$$I_{t'\rightarrow t} = I_{t'}\Big\langle proj(D_t, T_{t\rightarrow t'}, K)\Big\rangle$$

$$pe(I_a, I_b) = \frac{\alpha}{2}(1 - \mathrm{SSIM}(I_a, I_b)) + (1-\alpha)\|I_a - I_b\|_1,$$

However, instead of summing over the source images, they take the minimum re-projection loss among the source images. In addition to that, they also use edge-aware smoothness loss function for more smooth transitions between different depth regions.

$$L_p = \min_{t'} pe(I_t, I_{t'\rightarrow t}).$$

$$L_s = |\partial_x d_t^*| \, e^{-|\partial_x I_t|} + |\partial_y d_t^*| \, e^{-|\partial_y I_t|}$$

3.2. Segmentation Network

In this paper, U-Net segmentation network [17] is used as an addition to depth estimation network for better supervision signals. In its original paper, it takes 3x572x572 input images and generates nx572x572 where n is the number of segmentation classes. In this paper it takes 3x192x640 and generates 2x192x640 segmentation outputs with probabilities.

In the architecture, both down and up-conv layers are followed by RELU layer. Network first down-size input images with down-conv layers and then up-size with up-conv layers to the starting resolution.

$$\textbf{Abs Rel}: \frac{1}{|D|}\sum_{d\in D}|d^*-d|/d^* \quad \textbf{RMSE}: \sqrt{\frac{1}{|D|}\sum_{d\in D}||d^*-d||^2}$$

$$\textbf{Sq Rel}: \frac{1}{|D|}\sum_{d\in D}||d^*-d||^2/d^* \quad \textbf{RMSE log}: \sqrt{\frac{1}{|D|}\sum_{d\in D}||logd^*-logd||^2}$$

$$\boldsymbol{\delta_t}: \frac{1}{|D|}|\{d\in D|\, max(\frac{d^*}{d},\frac{d}{d^*}) \ < 1.25^t\}| \times 100\%$$

Figure 4. Evaluation metrics which are used to evaluate our work.

As a loss function, we used pixel-wise weighted cross entropy loss function as it is described in the paper. Weight map is generated according to distance to depth borders. Then, this weight map is used to calculate pixel-wise cross entropy loss with softmax function. In that way, points closer to the borders are more important than other points in the image. Softmax function is denoted with

$$p_k\left(x\right)$$

, as it can be seen below.

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right)$$

$$E = \sum_{\mathbf{x}\in\Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x}))$$

The main goal of adding this module to the network is using the close relationship between segmentation and depth estimation tasks as supervision signal. It is expected that accurate depth estimations yield accurate semantic segmentation masks. Therefore, pushing network to give accurate segmentation masks also improves depth estimation results during the training process.

Another important aspect of this novelty is that segmentation network does not cause extra inference time during the test because there is no need to that network for only estimating the depth. We use this network only during the training process to increase learning performance.

For creating ground truth data, DeepLabv3 [3] is used. All images in dataset is passed to the network and generated segmentation masks are saved as .npy file. Normally, the original model has 21 classes with background class. But in our case, we only have 2 classes which are background and foreground objects. Therefore, the output masks are arranged according to our case. A sample segmentation ground truth data can be seen from 3

3.3. Implementation Details

Final loss function of the overall network is

$$L = \mu L_p + \lambda L_s + \alpha L_b$$

where

$$L_b$$

is the segmentation loss which is calculated with the output of U-net architecture and ground truth segmentation masks. In our implementation,

$$\lambda = 1e-3, \alpha = 5e-4$$

As a start point, we initialized depth network encoders and segmentation network encoder with weights pre-trained on ImageNet. It provides faster convergence compared to training from scratch and yields better results.

We used NVIDIA Geforce 1050-TI and it took 7 hours approximately to train our proposed network.

Original resolution of the dataset images is 1242x375 but they are used as 640x192 during the training.

4. Results

4.1. Evaluation Metrics

You can see the evaluation metrics we used to evaluate our work from 4. d and d* represent predicted and ground truth disparity maps. Absolute relative error is calculated taking the mean of relative disparity map errors. Square relative error is calculated similarly but square of disparity difference is used. RMSE and RMSE log are the root of mean square disparity difference and log disparity difference. Finally, there are three accuracy metrics which differ in correctness thresholds. Ground truth and predicted disparity maps are compared and ones whose proportions are less than a particular threshold are considered as true.

4.2. KITTI Eigen Split

In that paper, we used a subset of KITTI Eigen Split, since the whole data is very hard to train and keep in the storage. Details about this subset can be found in GitHub page of the project. Particularly, there are 3949 training samples and 451 validation samples.

To compare our network with the baseline network, we first found hyperparameters that give the best result for baseline network. After that, we train our network with these hyperparameters and compare the results. As it can

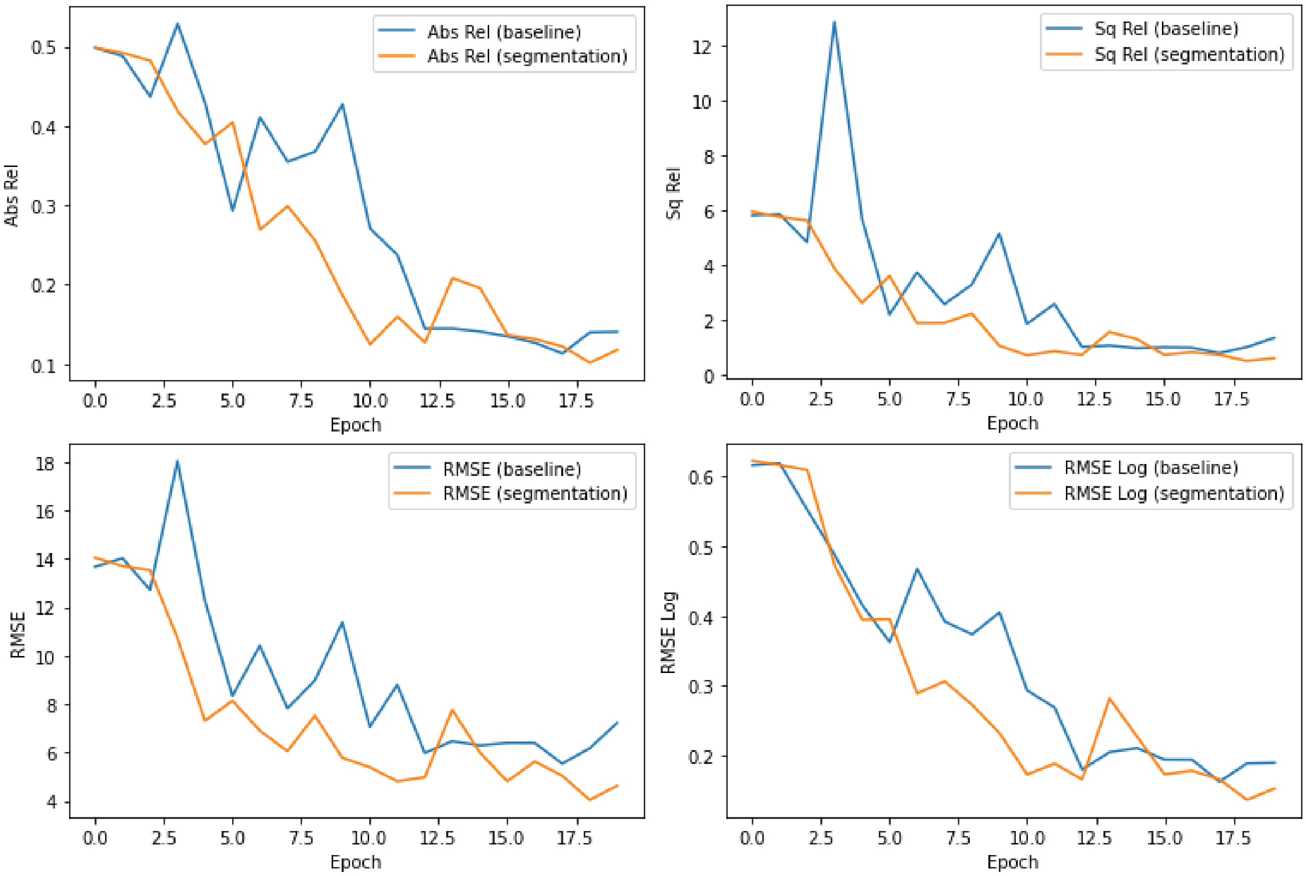

Figure 5. Evaluation metric values over 20 epochs. As it can be seen, our network converges faster and yields better final metric values.

Method	Abs Rel	Sq Rel	RMSE	RMSE Log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou [Ref]	0.183	1.595	6.709	0.270	0.734	0.902	0.959
SfMLearner [Ref]	0.208	1.768	6.958	0.283	0.678	0.885	0.957
Vid2Depth [Ref]	0.163	1.240	6.220	0.250	0.762	0.916	0.968
DNC [Ref]	0.182	1.481	6.501	0.267	0.725	0.906	0.963
DualNet [Ref]	0.121	0.837	4.945	0.197	0.853	0.955	0.982
SuperDepth [Ref]	0.116	1.055	-	0.209	0.853	0.948	0.977
Yang	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian	0.163	1.240	6.220	0.250	0.762	0.916	0.968
GeoNet	0.155	1.269	5.857	0.233	0.793	0.931	0.973
DF-Net	0.150	1.124	5.507	0.223	0.806	0.933	0.973
LEGO	0.162	1.352	6.276	0.252	0.783	0.921	0.969
Struct2depth	0.141	1.036	5.291	0.215	0.816	0.945	0.979
Monodepth2	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Monodepth2 (Trained on subset)	0.288	2.311	7.432	0.340	0.511	0.833	0.933
Monodepth2 + Seg (Trained on subset)	0.276	2.151	7.530	0.333	0.533	0.833	0.945

Table 1. Evaluation metric results for eigen split. Comparing to state of the art results, our network yields a poor performance. But, comparing the baseline network trained on the same subset, it gives better results.

be seen from 5, our network outperformed the baseline network for all four evaluation metrics. Firstly, it converges faster to the optimum point. Secondly, it gets close to the optimum point better than the baseline network.

Apart from that, we also compare our network with other state of the art networks on Eigen split test data. As it can

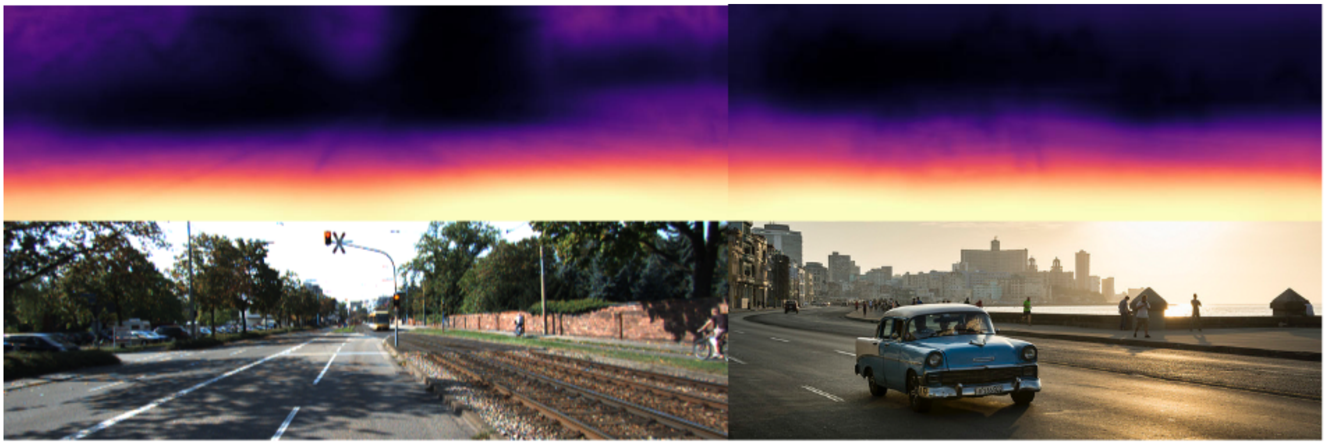

Figure 6. Some failure cases. Especially for the images that the number of foreground objects is low, our network may not improve the performance comparing to the baseline network

be seen from 1, our model does not yield comparable results with these networks. It is expected because our network is trained on subset of Eigen split which has much less training samples than the original dataset. However, when we compare our network with the baseline network trained on the same subset, our model again outperformed the baseline model.

4.3. Failure Cases

Although it gave better results than the baseline network trained on the same subset, our network gave poor results for some images due to the fact that it saw few data during the training process. As it can be seen from 6, especially for the images in which the number of foreground objects is low, our network may not improve the performance comparing to the baseline network.

5. Conclusion

In this paper, we showed that segmentation can be used as a booster supervision signal for depth estimation task. We use its close relationship with depth estimation by adding segmentation loss values to the final loss value of the network. Although we use DeepLabv3 that is trained on COCO2017 dataset for creating ground truth segmentation masks, we got promising results, especially comparing to the baseline network. As future work, more reliable and detailed segmentation masks should be used as ground truth data.

Another problem of this study is the extra computational time for segmentation network during the training. It makes training process 3 times slower. To handle this problem, more lightweight segmentation networks can be used. www.github.com/hakansivuk/SegDepth

References

[1] Gustavo Carneiro Adrian Johnston. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. *CVPR 2020*. 2

[2] Zhixiang Duan Chang Shu, Kun Yu and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. *ECCV 2020*. 2

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. 2018. 4

[4] Michael Firman Gabriel Brostow Clement Godard, Oisin Mac Aodha. 2

[5] Jianfei Cai Jianmin Zheng Jun Xiao Haiyong Jiang, Feilong Yan. End-to-end 3d point cloud instance segmentation without detection. *CVPR 2020*. 3

[6] Hao Wang Tzu-Yi HUNG Hanyu Shi, Guosheng Lin and Zhenhua Wang. Spsequencenet: Semantic segmentation network on 4d point clouds. *CVPR 2020*. 3

[7] Simon Hadfield Jaime Spencer, Richard Bowden. Defeat-net: General monocular depth via simultaneous unsupervised representation learning. *CVPR 2020*. 2

[8] Gabriel J. Brostow-Daniyar Turmukhambetov Jaime Watson, Michael Firman. Self-supervised monocular depth hints. *ICCV 2019*. 2

[9] Yibing Song Rynson Lau Jianbo Jiao, Ying Cao. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. *ECCV 2018*. 3

[10] Yuwang Wang Kaihuai Qin Junsheng Zhou and Wenjun Zeng. Unsupervised high-resolution depth learning from videos with dual networks. *ICCV 2019*. 2

[11] Oliver Wang Long Mai Zhe Lin Ke Xian, Jianming Zhang and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. *CVPR 2020*. 3

[12] Jia Zheng Jingyi Yu Junfei Zhang Rui Tang Shugong Xu Shenghua Gao Lei Jin, Yanyu Xu. Geometric structure based and regularized depth estimation from 360○ indoor imagery. *CVPR 2020*. 3

[13] Shaoshuai Shi Shu Liu Chi-Wing Fu Jiaya Jia Li Jiang, Hengshuang Zhao. Pointgroup: Dual-set point grouping for 3d instance segmentation. *CVPR 2020*. 3

[14] Oliver Wang Zhe Lin Lijun Wang, Jianming Zhang and Huchuan Lu. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. *CVPR 2020*. 2

[15] Yifan Wang Huchuan Lu Lijun Wang, Jianming Zhang and Xiang Ruan. Cliffnet for monocular depth estimation with hierarchical embedding loss. *ECCV 2020*. 2

[16] Yongri Piao Zhengkun Rong Huchuan Lu Miao Zhang, Weisong Ren. Select, supplement and focus for rgb-d saliency detection. *CVPR 2020*. 3

[17] Thomas Brox Olaf Ronneberger, Philipp Fischer. U-net: Convolutional networks for biomedical image segmentation. *MICCAI 2015*. 3

[18] Xiaoming Liu Shengjie Zhu, Garrick Brazil. The edge of depth: Explicit constraints between segmentation and depth. *CVPR 2020*. 3

[19] Chunyan Xu Zequn Jie Xiang Li Jian Yang Zhenyu Zhang, Zhen Cui. Joint task-recursive learning for semantic segmentation and depth estimation. *ECCV 2018*. 3