SEMANTIC SEGMENTATION IN DEPTH DATA : A COMPARATIVE EVALUATION OF IMAGE AND POINT CLOUD BASED METHODS

Jigyasa Singh Katrolia[1*], *Lars Krämer*[2*], *Jason Rambach*[1], *Bruno Mirbach*[1], *Didier Stricker*[1]

[1]DFKI, Germany [2]Technische Universität Kaiserslautern, Germany

ABSTRACT

The problem of semantic segmentation from depth images can be addressed by segmenting directly in the image domain or at 3D point cloud level. In this paper, we attempt for the first time to provide a study and experimental comparison of the two approaches. Through experiments on three datasets, namely SUN RGB-D, NYUdV2 and TICaM, we extensively compare various semantic segmentation algorithms, the input to which includes images and point clouds derived from them. Based on this, we offer analysis of the performance and computational cost of these algorithms that can provide guidelines on when each method should be preferred.

Index Terms— scene segmentation, depth image, point cloud

1. INTRODUCTION

Conventionally, deep learning based semantic segmentation networks, methods that assign a class label to each pixel in an image, have been applied to RGB images [1, 2, 3], largely ignoring depth information. Depth modality is essential in several scenarios where the requirement is to perceive the 3D structure of a scene. While several RGB-D methods for segmentation already exist [4, 5, 6], there are very few methods relying exclusively on depth images [7]. This suggests that it is generally overlooked that in real world many situations often arise where only depth data is available either by design or as a consequence of other limitations. Using only depth information can be advantageous in several indoor applications, for example, smart building, in-car TICaM monitoring or robotic applications [8, 9]. Depth images preserve privacy as subjects cannot be identified, they are more robust to illumination and color variations, and allow natural background removal. Additionally, it is easier to generate realistic synthetic depth data for training machine learning systems than it is to generate RGB data. Due to this, [7] achieved comparable performance to state-of-the-art RGB-D systems on NYUdV2 dataset even when trained only on depth images. For real-time applications, depth data lends itself well to training light-weight models. For all these reasons, it becomes important to compare state-of-the-art semantic segmentation methods that can be applied to depth data in various formats (2D depth map, point cloud, etc.). Such a comparison is meaningful when the only input modality available is depth data due to one of the aforementioned reasons or when training a parallel depth-only stream of an RGB-D network for late-stage fusion of predictions.

There are different ways to represent depth information, ranging from two dimensional representation like images captured by a depth camera to three dimensional representations like meshes, voxels and point clouds captured with a 3D scanner. Corresponding to each of these representations exist semantic segmentation methods that are designed to leverage the properties of these representations to extract semantic labels for each pixel in the scene. In this paper, we present the first experimental comparison of semantic segmentation from depth images in different depth representations like 2D depth images and 3D point clouds, and analyze their performance with respect to various factors to answer the following questions: (a) which representation of depth data gives best performance on scene segmentation and (b) under which conditions should one prefer one representation over the other? This is in contrast to previous surveys that have focused on specific data representations and related segmentation methods [10, 11].

We select two representative image based semantic segmentation methods, namely FCN [1], and DeepLab [2] and two point cloud based segmentation methods PointNet [12] and PointNet++ [13], and through experiments compare the usability of images and single view point clouds for the task of semantic segmentation. We focus on performing a fair comparison by using point clouds directly derived from depth images (single-view point cloud) and by evaluating run time, memory footprint and network training time. Three datasets are selected for this purpose: SUN RGB-D [14], NYUdV2 [15] and TICaM [16].

This work was partially funded within the Electronic Components and Systems for European Leadership (ECSEL) Joint Undertaking in collaboration with the European Union's H2020 Framework Program and Federal Ministry of Education and Research of the Federal Republic of Germany (BMBF), under grant agreement 16ESE0424 / GA826600 (VIZTA).

* Equal contribution

2. SEGMENTATION APPROACHES

2.1. Image based semantic segmentation methods

Classically, deep learning based semantic segmentation methods have been applied to RGB images because of the ease of acquisition of ground truth annotations and intuitive usage of CNNs on images due to their spatially invariant nature. Popular Deep Convolutional Neural Networks or DCNNs meant for image classification like VGG [17] and ResNet [18] have been modified into a fully convolutional form to adapt them for the task of semantic segmentation [1, 2, 3]. FCN [1] championed the idea of using only convolutional layers in conjunction with fusing lower layer features with higher level features through upsampling via transposed convolutions. DeepLab [2] substituted the transposed convolutions with a range of atrous convolutions applied with different rates to capture a bigger receptive field as well as objects at different scales.

2.2. Point cloud based semantic segmentation methods

The seminal work of PointNet [12] addressed the use of deep learning architectures for point cloud classification and segmentation while respecting the limitations and properties of point cloud representation like absence of point order, invariance of point cloud semantics to rigid transformations and physical world meaning of distance between points. It uses a series of Multi-Layer Perceptrons (MLPs) to learn a feature representation of points followed by a max-pooling operation to learn an order invariant global signature of the point cloud. Combining both local and global point cloud features they were able to achieve good results on point cloud segmentation. PointNet++ [13] went one step further to emulate the concept of multiple levels of abstraction associated with CNNs. They used PointNet as a feature descriptor to extract features from multiple overlapping neighbourhoods of points, where each neighbourhood could vary in scale depending on the point density.

3. EXPERIMENTAL SETUP

Datasets. We evaluate the segmentation methods on three datasets, SUN RGB-D [14], NYUdV2 [15] and TICaM [16]. SUN RGB-D and NYUdV2 are RGB-D datasets of indoor scene images whereas TICaM contains depth images captured with a Kinect AZURE camera that is fitted inside a driving simulator [19] where the rear view mirror would be in a real car. Figure 1 shows the data capturing setup and a sample depth image from TICaM. We train for segmentation on only three classes for TICaM namely person, object and background. We use preprocessed depth images from SUN RGB-D and NYUdV2 where the noisy depth images have been inpainted to fill holes. However, we do not fill the holes in TICaM images in order to include a dataset with raw noisy depth images in our evaluation.

Fig. 1. **Left:** A sample image from the TICaM dataset, **Right:** The driving simulator setup used to capture TICaM dataset images.

Image segmentation implementation. We refer to the implementations of DeepLab and FCN provided by PyTorch [20, 21] to train the image segmentation models. We employ ResNet-101 [18] as the backbone for both the networks and initialize them with pretrained weights on COCO dataset. We use a batch size of 2 and SGD with momentum of 0.9 [22] and an initial learning rate of 0.005. The learning rate strategy is same as in [2] (poly learning rate policy with power 0.9).

Point cloud segmentation implementation. We generate single-view point clouds from the depth images and use them to train PointNet and PointNet++ networks as adapted from [23]. We train both models for 128 epochs with batch size of 4. We use Adam optimizer with initial learning rate of 0.001 and momentum of 0.1. The learning rate is decreased by a factor of 0.7 after every 10 epochs.

4. EVALUATION AND RESULTS

4.1. Performance Metrics

We report the mean Intersection-over-Union (IoU) and mean Precision performance of the four segmentation methods on the three selected datasets in Table 1. All models are evaluated on a GeForce GTX 1070. Note that the point cloud methods presented here are trained with randomly selected 4096 points. It is notable that on challenging NYUdV2 and SUN RGB-D datasets image segmentation methods vastly outperform point cloud based segmentation methods with DeepLab superseding FCN. For comparatively simpler TICaM dataset, PointNet++ leads in IoU and is comparable to other image segmentation methods overall. It maybe interesting to remember here that the TICaM dataset has a lot of missing depth values in the image.

An apparent shortcoming of point cloud based methods is that they provide point labels for only a subset of points randomly sampled from the point cloud unlike image based methods that provide dense pixel labels. For a one-to-one

Table 1. mean IoU and mean Precision of compared methods.

	DeepLab		FCN		PointNet		PointNet++	
	IoU	Precision	IoU	Precision	IoU	Precision	IoU	Precision
NYUdV2	**31.46**	**48.81**	29.06	46.97	9.18	15.92	14.53	25.52
SUN RGB-D	**27.55**	45.54	21.27	**45.60**	9.04	19.16	12.98	30.82
TICaM	85.35	94.60	81.49	**96.74**	75.07	84.16	**87.32**	91.19

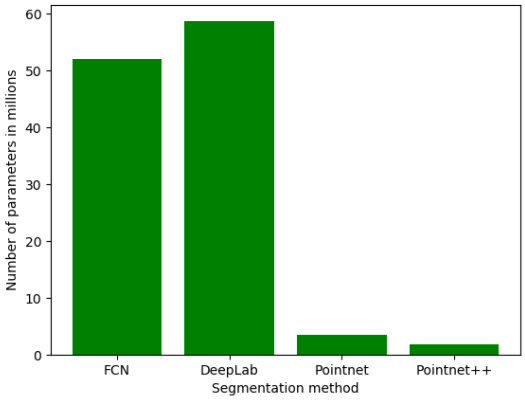


Fig. 2. Number of model parameters in millions.

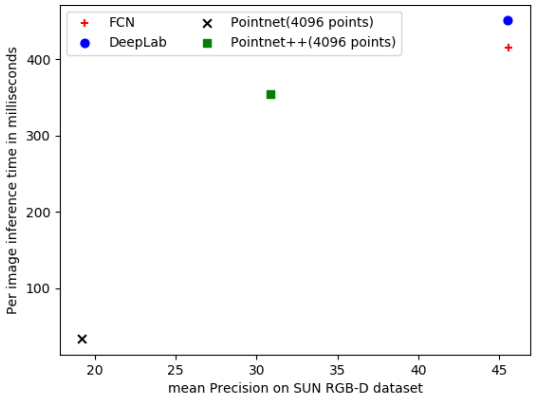
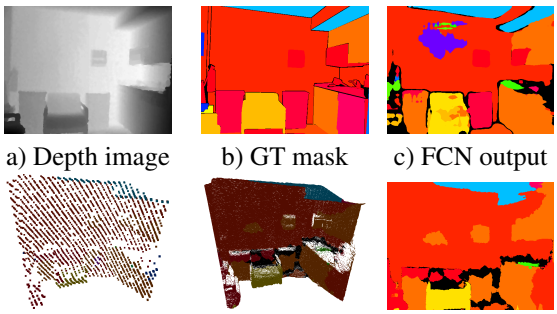


Fig. 3. mean Precision vs. per image inference time trade-off on SUN RGB-D dataset.



a) Depth image b) GT mask c) FCN output

d) PointNet++ output e) Upsampled f) Image projection

Fig. 4. Dense mask predictions from image based and point cloud based methods.

comparison between dense mask predictions, we use K-Nearest Neighbour algorithm to upsample sparse point cloud segmentation output to get labels for each point in order to have a better comparison between the image based and point cloud based methods. Figure 4e shows the upsampled version of PointNet++ prediction 4d, whereas Figure 4f shows it in the image for comparison with 4c. In Table 3, we can see that although the upsampling strategy is naive, it does not affect the performance of point cloud methods greatly.

4.2. Space-time complexity

For practical considerations, we also compare the number of model parameters and per image inference time for each of the segmentation methods. Figure 2 shows the number of model parameters in millions. We can observe that point cloud methods have significantly less parameters than image based methods. We also compare in Figure 3 per image inference time in milliseconds against the precision metric on SUN RGB-D dataset. Overall we can see that point cloud methods can be a good option when resources are limited. In the next section we compare in detail the performance of point cloud methods trained with varying number of points.

4.3. Resolution of input

We compared different versions of PointNet and PointNet++ on each of the three datasets where we vary the number of input points to the two networks. All points are randomly sampled. It is clear from Table 2 that for a negligible increase in inference time PointNet offers little increase in performance as the number of points is increased. However, for PointNet++, significant improvement can be noticed with inference time also growing, albeit less than proportionally.

4.4. Effect of frequency and size of class mask

We also analyze the effect of size of a class object and its frequency in the training dataset on the precision of a method for that class. We use the term 'frequency' to signify the number of images which contain that class in a dataset. Whereas

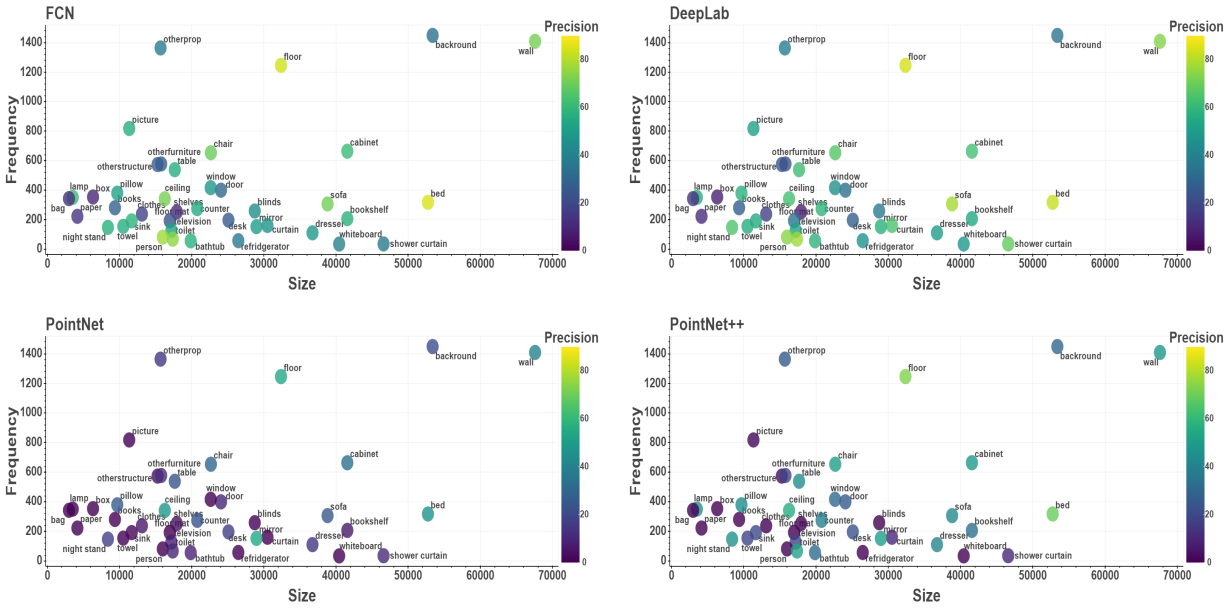


Fig. 5. Precision of FCN (top-left), DeepLab (top-right), PointNet (bottom-left) and PointNet++ (bottom-right) on different classes of NYUdV2 against the size and frequency of that class.

Table 2. Inference time and mean precision of PointNet and PointNet++ on all test datasets when trained with 1024, 2048 and 4096 points.

		Inference time (ms)			Precision		
		1024	**2048**	**4096**	**1024**	**2048**	**4096**
SUN RGB-D	**PointNet**	33	34	34	17.78	19.82	19.16
	PointNet++	312	333	354	21.49	30.02	30.86
NYUdV2	**PointNet**	27	27	28	16.01	16.43	15.92
	PointNet++	312	337	357	20.23	23.81	25.52
TICaM	**PointNet**	23	23	23	70.08	83.32	84.16
	PointNet++	317	347	356	88.57	90.61	91.19

Table 3. mean Precision and IoU of PointNet and PointNet++ on NYUdV2 dataset when evaluated on 4096 subsampled points and entire point cloud.

	N = 4096		All points	
	Precision	**IoU**	**Precision**	**IoU**
PN	15.92	9.18	18.38	9.09
PN++	25.52	14.53	25.14	14.04

'size' refers to the total number of pixels labelled as belonging to a class across the dataset divided by the frequency of that class. In Figure 5 we compare the precision of FCN, DeepLab, PointNet and PointNet++ on each of the 40 class of the NYUdV2 dataset and see how per-class precision varies with the size of that class object and its number of instances in the training data. It can be seen that most of the performance loss of point cloud based methods can be accounted for by attributing them to both small and infrequent object classes.

5. CONCLUSION

In this work, we have evaluated and compared two representatives each of image based and point cloud based semantic segmentation methods on depth images. Coming to the questions that we set out to answer, we can say that doing segmentation in image space yields better results than point cloud based methods if speed is not a priority and the dataset is quite challenging with many classes. For simpler datasets, both image based and point cloud based methods have comparable performance. PointNet++ offers an acceptable combination of runtime and performance even when run with 4096 points. Lastly, we can say that the lower mean precision of point cloud based methods can be mostly attributed to small and infrequent object classes in training data.

6. REFERENCES

[1] Evan Shelhamer, J. Long, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[2] Liang-Chieh Chen, G. Papandreou, I. Kokkinos, Kevin Murphy, and A. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[4] Saurabh Gupta, Ross Girshick, Pablo Arbelaez, and Jitendra Malik, "Learning rich features from rgb-d images for object detection and segmentation," 2014.

[5] Caner Hazırbaş, Lingni Ma, Csaba Domokos, and Daniel Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," 2016.

[6] Weiyue Wang and U. Neumann, "Depth-aware cnn for rgb-d segmentation," *ArXiv*, vol. abs/1803.06791, 2018.

[7] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla, "Understanding realworld indoor scenes with synthetic data," 2016.

[8] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017.

[9] Jigyasa Katrolia, Lars Krämer, Jason Rambach, B. Mirbach, and Didier Stricker, "An adversarial training based framework for depth domain adaptation," in *VISAPP 2021 - 16th International Conference on Computer Vision Theory and Applications*, 2020.

[10] Jiaying Zhang, Xiaoli Zhao, Zheng Chen, and Zhejun Lu, "A review of deep learning-based semantic segmentation for point cloud," *IEEE Access*, 2019.

[11] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[12] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, NIPS'17.

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

[15] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[16] Jigyasa Katrolia, Bruno Mirbach, Ahmed El-Sherif, Hartmut Feld, Jason R. Rambach, and Didier Stricker, "Ticam: A time-of-flight in-car cabin monitoring dataset," *ArXiv*, vol. abs/2103.11719, 2021.

[17] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 2014.

[18] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] Hartmut Feld, Bruno Mirbach, Jigyasa Singh Katrolia, Mohamed Selim, Oliver Wasenmüller, and Didier Stricker, "Dfki cabin simulator: A test platform for visual in-cabin monitoring functions," in *Proceedings of the 6th Commercial Vehicle Technology Symposium - CVT*, 2020.

[20] "Deeplab-resnet101 implementation," `https://github.com/pytorch/vision/blob/master/torchvision/models/segmentation/deeplabv3.py"`.

[21] "Fcn," `https://github.com/pytorch/vision/blob/master/torchvision/models/segmentation/fcn.py"`.

[22] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks : the official journal of the International Neural Network Society*, 1999.

[23] "Pointet and pointnet++ pytorch implementations," `https://github.com/yanx27/Pointnet_Pointnet2_pytorch`.