# Optional Depth Module for MaskRCNN

**Lovish Chum, Jianjin Xu, Zhaoyang Wang**
Department Computer Science
Columbia University
{lc3454,jx2386,zw2585}@columbia.edu

**Abstract:** Instance segmentation is one of the most important perception tasks in computer vision. We present an approach to *optionally* use the depth given along an image to aid the performance on this task. We observe that depth information incorporated through Spatially-Adaptive (DE)normalization (SPADE) results in significant improvement on the task on NYUv2 dataset. Additionally, we observe that the use of ODM (Optional Depth Module) helps to prevent the degradation of performance even when the depth data is unavailable to the network. Code which reproduces the experimental results is available at https://github.com/AtlantixJJ/OptionalDepthPathway

**Keywords:** Depth, Instance Segmentation,

## 1 Introduction

Instance segmentation is a challenging task. Compared to semantic segmentation, the goal is to classify each pixel into object categories along with differentiating the object instances. Depth seems to provide additional information for this task, as distinct objects in any given scene are located at different spatial positions.

That said, a major issue preventing the use of depth in instance segmentation task is the *lack of paired* RGB and depth data along with instance labels. Although existing RGB datasets with instance labels can be supplemented with the estimated depth, the accuracy of the system built is limited by the performance of the underlying estimation algorithm. To mitigate this, we formulate the problem as instance segmentation contingent on the *optional* availability of depth information.

We show that a incorporating an Optional Depth Module (ODM) to MaskRCNN successfully allows for RGB or RGB-D input during both the training or test time. Our method, SPADE- MaskRCNN, is an extension of MaskRCNN [1] and uses a combination of Spatial Adaptive Normalization (SPADE) [2] and Batchnorm [3].

We evaluate our method on NYUv2 dataset. SPADE-MaskRCNN outperforms all the baselines in the presence of depth data. Besides, the performance of our method only has *insignificant* drop in accuracy when the depth is missing in the input instance. This distinguishes our method from fusion-based techniques such as FuseNet[4] or UpNet[5], which are unable to perform inference when a certain modality is missing. Moreover, our model can also be trained with missing modalities, which indicates that our model can be trained using more data than the fusion-based methods.

Although in this work we only experimented with depth as an optional modality, the idea can be used to devise frameworks which are robust to the availability of any modalities. This makes it particularly relevant to be used for fault-tolerance robotics where one of the sensors on the agent fails in the midst of a task. The contribution of this works includes:

- Optional Depth Module (ODM), a flexible framework to learn from optionally available modality.
- SPADE-MaskRCNN, a new state-of-the-art on instance segmentation with depth on NYUv2.

The rest of the paper is organized as follows. Section 2 briefly describes literature pertinent to our method and task. Our approach towards the problem is described in Section 3. Section 4 covers the details about the dataset used, training methodology and evaluation methods. Finally, we conclude and provide future directions in Section 5.

## 2 Related Works

Mask RCNN [1] is a standard instance segmentation framework built upon Faster R-CNN [6]. He et. al. [7] explores the possibility of a partially supervised training schema with sometimes missing masks. Liu et. al.[8] further boost the information flow in the Mask RCNN framework by pooling features and shortening distance between feature levels. All of the above methods are limited to only use RGB image and haven't explored using depth to aid the perception tasks such as semantic or instance segmentation.

Majority of work attempting to integrate depth focused on semantic segmentation. FuseNet [4] fuses the RGB and depth features with a encoder-decoder style network. UpNet [5] explores early and late fusion in both multi-modal and multi-spectral images. Adapnet++ [9] proposes a self-supervised model that adapts and learns the optimal fusion of features. Our work is different from these as we are working on instance segmentation by optionally using depth, rather than making its presence mandatory in the dataset.

With the extra depth information, it is natural to consider the task of segmentation in 3D. However, most 3D segmentation methods work with point cloud data [10, 11], which has different structure with depth map. Also, their result is segmentation of point cloud, which is not directly comparable to 2D instance segmentation. 3D MaskRCNN [12] is able to reconstruct objects parametrized by PCA basis, but it does not use depth modality.

## 3 Method

Our method borrows the backbone from MaskRCNN framework. However, to incorporate the Optional Depth Module (ODM) during training and testing, we use SPADE along with batch normalization (BN). In the case when depth is unavailable, BN is used to train or infer on a given data point. However, if the depth information is available at either time, SPADE layer replaces the BN layer. The rest of the architecture is common for both the cases. This makes the ODM adapt to the availability of depth.

In the subsections below, we briefly describe SPADE and ODM. This is followed by a summary of the baselines and training scheme of the network.

### 3.1 Spatially-Adaptive (DE)normalization (SPADE)

SPADE[2] has been used for image generation conditional on the semantic segmentation map. This prompted us to use SPADE for optional enhancement for conditional training and inference using depth map.

In conventional batch normalization, the input feature map is normalized with batch mean and variance and then de-normalized with learned scale and offset constant.

$$BN(x) = \gamma \frac{x - \mu}{\sigma} + \beta$$

where $x$ is input feature map, $\mu$, $\sigma$ is batch mean and variance, $\gamma$ and $\beta$ is scale and offset parameter respectively. Note that they are vectors having the same dimension as feature map depth.

SPADE is an extension of batch normalization, where the scale and offset is mapped from a semantic segmentation map

$$SPADE(x, c) = \gamma(c) \frac{x - \mu}{\sigma} + \beta(c)$$

.

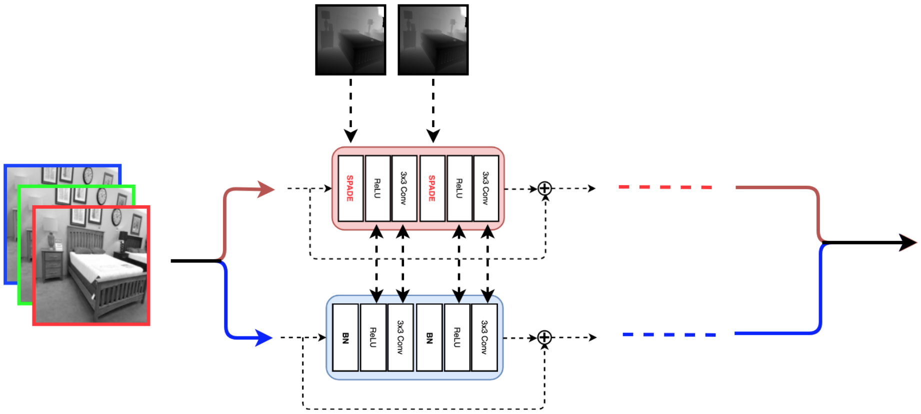


Figure 1: The pathway of giving RGB input or RGBD input in a residual block. All other layers share weights except that BN is replaced with optional depth module. Red path shows training/inference when the depth is available. Blue path shows the same when the depth is unavailable.

In Park. et. al [2] SPADE helps the generator learns a more precise mapping from semantic segmentation to the generated image. A possible reason is that it helps preserve semantic information across the layers as compared to giving semantic segmentation map in the input only.

We find that our problem is similar in the sense what we also want to condition our output on depth. Simply providing the depth information to the first layer is bound to have little effect after a series of transformations. Therefore, it makes sense to inject depth information throughout the backbone using SPADE.

### 3.2 Optional Depth Module (ODM)

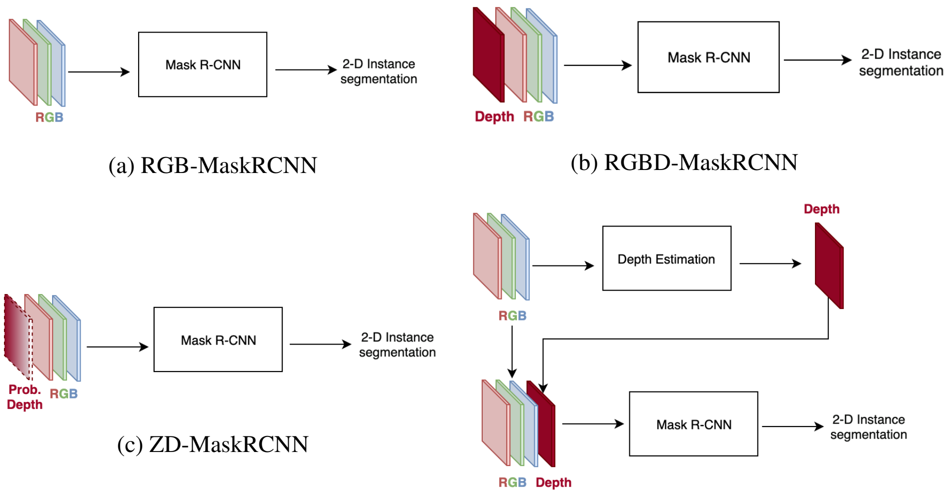
The Optional Depth Module (ODM) consists of a modified Resnet basicblock. During training, if the batch of images contains depth information, the network uses SPADE layer to update weights for each of these blocks. In the absence of depth information, the network falls-back to BN. Figure 1 depicts the two pathways through a ResNet block of MaskRCNN backbone.

During testing, a similar procedure is followed. Note that we expect the network to perform at the same level even without depth information during inference because it has been trained in a similar manner. Thus, we can say that ODM provides the network robustness to the lack of modality.

### 3.3 Baselines

We consider the following baselines for evaluation.

- **RGB-MaskRCNN**: Original MaskRCNN trained on RGB data. Ideally, our method should match this baseline when only optional depth modules are trained and outperform when the whole network is trained with mixed data.
- **RGBD-MaskRCNN**: MaskRCNN with the first layer of convolution modified to 4 channels to accept RGBD data, and trained on RGBD data. The weights of the $4^{th}$ channel in the first layer are randomly initialized. This is a trivial method to inject depth. Our method should outperform this approach when depth data is given.
- **ZD-MaskRCNN**: The same architecture with RGBD-MaskRCNN but trained on mixed data. When depth is not available, we fill the channel with zeros. This works as a naive method to work with optional depth. Our method should be able to outperform this when tested on both RGB and RGB-D data.
- **ED-MaskRCNN**: Same as ZD-MaskRCNN except that when no depth is available, we estimate the depth (ED) using Densedepth [13]. The estimated depth may not be accurate


RGB
Mask R-CNN
2-D Instance segmentation
Depth
RGB
Mask R-CNN
2-D Instance segmentation
Prob. Depth
RGB
Mask R-CNN
2-D Instance segmentation
RGB
Depth Estimation
Depth
RGB
Depth
Mask R-CNN
2-D Instance segmentation

(a) RGB-MaskRCNN

(b) RGBD-MaskRCNN

(c) ZD-MaskRCNN

(d) ED-MaskRCNN

Figure 2: Baselines

but will have more similar distribution with real depth comparing to ZD-MaskRCNN. Our method should also be able to outperform it on both types of data.

Figure 2 shows the wireframe diagram for the baselines. We expect SPADE to outperform all the baseline models.

### 3.4 Training

We train our model in the following options.

- I. Take a model pre-trained on a large visual dataset such as ImageNet, COCO, etc.
- II. Train with RGB data.
- II'. Train with RGBD data.
- III. Train the parameters of Optional Depth Module (ODM) with RGBD data.
- IV. Fine-tune the whole network with mixed data, i.e. the mixture of RGB-only data and RGB-D data.

Currently, we simulate the mixed data by randomly dropping the depth data. In future work, we intend to train with a mixture of COCO, Cityscapes and NYUv2.

RGB-MaskRCNN will go through options I+II. RGBD-, ZD-, ED-MaskRCNN will go through options I+II'. SPADE-MaskRCNN will go through options I+II+III+IV.

## 4 Experiment

### 4.1 Dataset

We use NYUv2 dataset for both training and evaluation. NYUv2 dataset provides depth collected by Kinect. It has 1,449 images and instance segmentation pairs, from which 795 images and 654 images are in training and test set respectively. We select 26 classes out of provided 40 classes with the following rules:

1. It is conventional object (not stuff such as "wall", "floor", "otherproperty").
2. Have at least 100 instances in training split.

| Option | Method | Testing | AP |
|---|---|---|---|
| I+II | RGB-MaskRCNN | RGB | **0.3088** |
| I+II' | RGBD-MaskRCNN | RGB-D | **0.3025** |
| I+II+III | SPADE (Fix) | RGB | 0.3076 |
| I+II+III | SPADE (Fix) | RGB-D | 0.0373 |
| I+II+III+IV | SPADE-MaskRCNN | RGB | **0.3321** |
| I+II+III+IV | SPADE-MaskRCNN | RGB-D | **0.3280** |

Table 1: Instance **segmentation** precision

| Option | Method | Testing | AP |
|---|---|---|---|
| I+II | RGB-MaskRCNN | RGB | **0.3618** |
| I+II' | RGBD-MaskRCNN | RGB-D | **0.3236** |
| I+II' | ZD-MaskRCNN | RGB | **0.3925** |
| I+II' | ZD-MaskRCNN | RGB-D | **0.3901** |
| I+II+III | SPADE (Fix) | RGB | 0.3618 |
| I+II+III | SPADE (Fix) | RGB-D | 0.1808 |
| I+II+III+IV | SPADE-MaskRCNN | RGB | **0.3424** |
| I+II+III+IV | SPADE-MaskRCNN | RGB-D | **0.3596** |

Table 2: Instance **segmentation** precision (New)

### 4.2 Training details

We use ResNet-50 and FPN as a backbone. All the training is carried out using original resolution $480 \times 640$ with batch size of 4. Color jittering and vertical flipping are used in data augmentation. Depth data is scaled to the limit $[-1, 1]$.

For option II, the network is trained for 18000 iterations, which is equivalent to 90 epochs. The learning rate starts with a base of $2.5 \times 10^{-3}$. Linear warm-up is carried out for 500 iterations starting from $\frac{1}{3}$ of base learning rate. The learning rate then follows a multi-step scheduling policy of multiplying 0.1 on step 12000 and 16000. Option II' is the same with option II except that RGB-D data is used.

For option III, the network is trained for 1000 iterations. The base learning rate is $10^{-4}$ and also follows the same warm-up policy as option II. At step 500 and 800, the learning rate is multiplied with 0.1.

For option IV, the network is trained for 9000 iterations, which is equivalent to 45 epochs. The probability of dropping depth is $p_d = 0.5$. The base learning rate is $2 \times 10^{-3}$.

### 4.3 Evaluation

We evaluate models on RGB-only data or RGB-D data of NYUv2 dataset. RGB-MaskRCNN is evaluated on RGB-only data and RGBD-MaskRCNN is evaluated on RGBD data. ZD-, ED- and SPADE-MaskRCNN are evaluated on both RGB-only data and RGBD data. Average Precision is selected as major evaluation metrics, which is the same as in He et. al [1].

Old quantitative results are shown in Table 1, which are the results obtained before the presentation. After that we found images are normalized to [-127, 127] scale, which is wrong. After re-running the experiments, we obtained the new results in Table 2.

The accuracy of RGB-MaskRCNN is the best, and RGBD-MaskRCNN is the worst. This shows that adding depth will not trivially improve the result. SPADE-MaskRCNN with depth works secondly well, but still the accuracy is a little bit lower than RGB-MaskRCNN, which indicates that the current fusion method needs to be improved. Even though SPADE-MaskRCNN cannot outperform RGB-MaskRCNN, it outperforms RGBD-MaskRCNN and SPADE-MaskRCNN without depth, which means SPADE-MaskRCNN is a better fusion technique than trivially add depth and also is able to obtain advantage using depth over RGB-only data. Figure 3 show shows results from our model compared to RGB-MaskRCNN.

## 5 Conclusion and future work

In this work, we propose a SPADE-based optional depth module to make MaskRCNN capable of accepting depth as an optional modality both during training and testing. We train and evaluate SPADE-MaskRCNN on NYUv2 dataset and compared it to 4 baselines: RGB-, RGBD-, ZD-, ED-

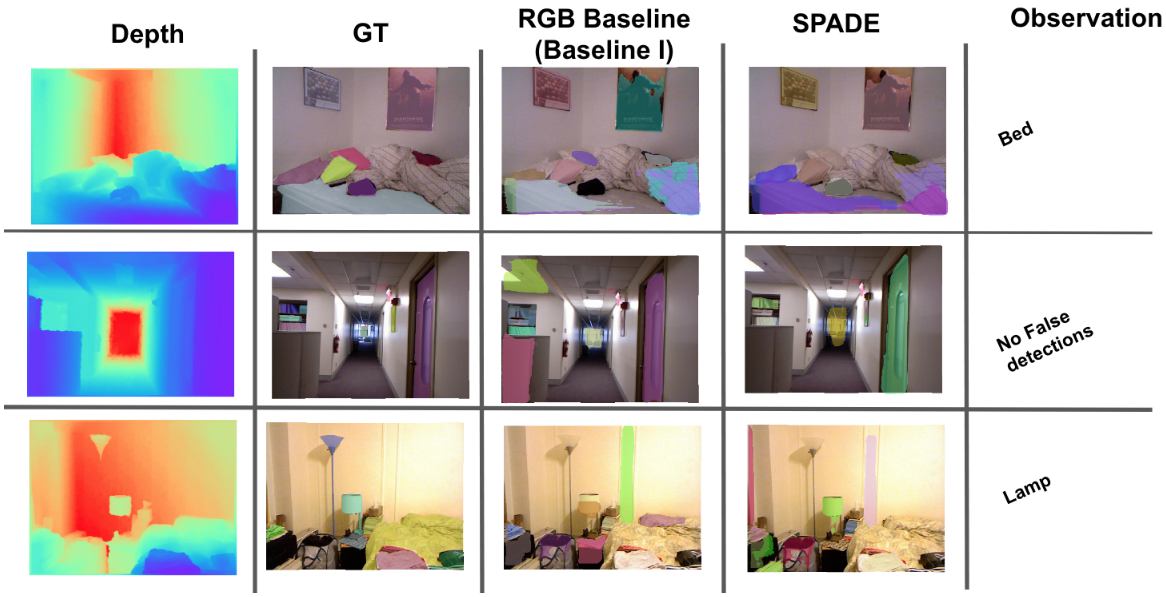


Figure 3: Qualitative comparison between RGB-MaskRCNN and SPADE MaskRCNN. Our approach uses depth to improve the instance segmentation performance for the above instances.

MaskRCNN. SPADE-MaskRCNN outperforms all other models. This supports that SPADE is good for learning representation for optional modality.

In future work, we want to prove that SPADE-MaskRCNN is able to train on large scale RGB data. The simulated mixed data in this work will be replaced with real mixed data of COCO and NYUv2.

## 6 Contribution

The major contribution of this work is listed in the following Table 3.

| Jianjin Xu | Lovish Chum | Zhaoyang Wang |
|---|---|---|
| Idea proposal and proposal slide | Record keeping | Literature review |
| NYUv2 dataset preparation | RGB and RGBD Baseline preparation | |
| Evaluation algorithm implementation | Re-runing experiments | |
| SPADE-MaskRCNN | ZD-MaskRCNN | ED-MaskRCNN |
| Demo video preparation | | Demo video preparation |
| Final presentation slides | Final presentation slides | |
| Final report (draft) | Final report (revision) | Final report (literature review) |

Table 3: The contribution of this work

**Acknowledgments**

We thank Prof. Shuran Song and all the students enrolled in the course. We learned a lot about the recent trends in computer vision and robotics through this course.

## References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[2] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[3] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[4] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, 2016.

[5] A. Valada, G. Oliveira, T. Brox, and W. Burgard. Towards robust semantic segmentation using deep fusion. In *Robotics: Science and Systems (RSS 2016) Workshop, Are the Sceptics Right? Limits and Potentials of Deep Learning in Robotics*, 2016.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[7] R. Hu, P. Dollár, K. He, T. Darrell, and R. B. Girshick. Learning to segment every thing. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2017.

[8] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[9] A. Valada, R. Mohan, and W. Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *arXiv preprint arXiv:1808.03833*, 2018.

[10] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and A. Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *ArXiv*, abs/1906.01140, 2019.

[11] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2017.

[12] A. Kundu, Y. Li, and J. M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3559–3568, 2018.

[13] I. Alhashim and P. Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.