Depth Estimation from Single Image Using CNN-Residual Network

Xiaobai Ma
maxiaoba@stanford.edu

Zhenglin Geng
zhenglin@stanford.edu

Zhi Bie
zhib@stanford.edu

Abstract

In this project, we tackle the problem of depth estimation from single image. The mapping between a single image and the depth map is inherently ambiguous, and requires both global and local information. We employ a fully convolutional architecture, which first extracts image feature by pretrained ResNet-50 network. We do transfer learning by replacing the fully connected layer of ResNet-50 with up-sampling blocks to recover the size of depth map. The up-sampling block combines residual learning concept. This CNN-Residual network can be trained en-to-end, and runs real time on images with enough computing power.

We demonstrate that our method of doing up-sampling by CNN-Residual network yields better result than fully connected layer, because it avoids overfitting. We also compare our model with pure CNN network and illustrates the effectiveness of tansfer learning. We also show the influence of different loss functions during training. The results are shown by comparing qualitative visualization and quantitative metrics.

1. Introduction

As a fundamental problem in computer vision, depth estimation shows the geometric relations within a scene. These relations help provide richer representations of objects and their environment, often leading to improvements in existing recognition tasks, as well as enabling many further applications such as 3D modeling, physics and support models [28], autonomous driving, video surveillance, robotics [4], and potentially reasoning about occlusions.

Many successful techniques for depth estimation from stereo images has been proposed. Provided accurate image correspondences, depth can be recovered deterministically in the stereo case.

While estimating depth from based on stereo images or motion has been explored extensively, depth estimation from monocular image often arises in practice, such as better understandings of the many images distributed on the web and social media outlets, real estate listing, etc, which include both indoor and outdoor examples. Hence, it is not only a challenging task to develop a computer vision system capable of estimating depth maps by exploiting monocular cues, but also a necessary one in scenarios where direct depth sensing is not available. Moreover, the depth information is well-known to improve many computer vision tasks with respect to the RGB-only counterpart, such as in recognition [22] and semantic segmentation [2].

Estimating depths from a single monocular image is a well-known ill-posed problem, since one captured RGB image may correspond to infinite number of real world scenes, and no reliable visual cue can be obtained. Several works have tried to tackle this problem. Structure-from-Motion method leverages camera motion to estimate camera poses through different temporal intervals and, in turn, estimate depth via triangulation from pairs of consecutive views. Other works use variations in illumination [34] and focus [31] as assumptions.

Recently, Convolutional Neural Networks (CNNs) have been employed to learn an implicit relation between color pixels and depth. We implemented an end-to-end trainable CNN architecture combined with residual network to learn a mapping between color image pixel intensity with corresponding depth map.

2. Related work

2.1. Classic methods

In the single-view depth estimation problem, most works rely on camera motion (Structure-from-Motion method [21]), variation in illumination (Shape-from-Shading [34]) or variation in focus (Shapre-from-Defocus [31]).

Without such information, single RGB image depth estimation has also been investigated. Classic methods rely on strong assumptions about the scene geometry, relied on hand-crafted features and probabilistic graphical models which exploits horizontal alignment of images or other geometric information. For example, Saxena et al. [26] predicted depth from a set of image features using linear regression and a MRF, and later extend their work into the Make3D system for 3D model generation [27]. However, the system relies on horizontal alignment of images, and suffers in less controlled settings. Inspired by this work, Liu et al. [15] combine the task of semantic segmentation with depth estimation, where predicted labels are used as additional constraints to facilitate the optimization task. Ladicky et al. [10] instead jointly predict labels and depths in a classification approach. Hoiem et al. [6] do not predict depth explicitly, but instead categorize image regions into geometric structures and then compose a simple 3D model of the scene.

2.2. Feature-based mapping methods

A second type of related works perform feature-based matching between a given RGB image and the images of a RGB-D repository in order to find the nearest neighbors, the retrieved depth counterparts are then warped and combined to produce the final depth map. Karsch et al. [7] perform warping using SIFT Flow [16], followed by a global optimization scheme, whereas Konrad et al. [8] compute a median over the retrieved depth maps followed by cross-bilateral filtering for smoothing. Instead of warping the candidates, Liu et al. [19], formulate the optimization problem as a Conditional Random Field (CRF) with continuous and discrete variable potentials. Notably, these approaches rely on the assumption that similarities between regions in the RGB images imply also similar depth cues.

2.3. CNN based methods

Recently, CNN based depth estimation methods begin to dominate. Since the task is closely related to semantic labeling, most works have built upon the most successful architectures of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [25], often initializing their networks with AlexNet [9] or the deeper VGG [30]. Eigen et al. [3] are the first to use CNN for single image depth estimation. The authors addressed the task by employing two deep network stacks. The first network makes a global coarse depth prediction for the whole image, and the second refines this prediction locally. This idea is later extended in [2], where three stacks of CNN are used to additionally predict surface normals and labels together with depth. Unlike the deep learning structures used in [3, 2], Roy et al. [24] combined CNN with regression forests [14], using very shallow architectures at each tree node, thus limiting the need for big data.

Another direction for improving the quality of the predicted depth maps has been the combined use of CNNs and graphical models. Liu et al. [17] propose to learn the unary and pairwise potentials during CNN training in the form of a conditional random field (CRF) loss and achieved state-of-the-art result without using geometric priors. This idea makes sense because the depth values are continuous [18]. Li et al. [13] and Wang et al. [32] use hierarchical CRFs to refine their patch-wise CNN predictions from superpixel down to pixel level.

2.4. Fully convolutional networks

The deep learning based methods mentioned above increase the accuracy on standard benchmark datasets considerably, and are the best in the state of the art. Meanwhile, researchers are trying to improve CNN method accuracy further. Recent work has shown that fully convolutional networks(FCNs) [20] is a desirable choice for dense prediction problems due to its ability of taking arbitrarily sized inputs and returning spatial outputs. [1] uses FCN and adopts CRF as post-processing. Besides classical convolutional layers, [12] uses dilated convolutions as an efficient way to expand the receptive field of the neuron without increasing the parameters for depth estimation; [23] uses transpose convolution for up-sampling the feature map and output for image segmentation.

Laina et al. [11] proposed a fully connected network, which removes the fully connected layers and replaced with efficient residual up-sampling blocks. We closely follow this work in this project. We reimplemented the architecture in PyTorch, and compared the performance of this method with pure CNN.

3. Methods

We experimented with three CNN based methods. We will illustrate each one in the next subsections, and compare their performance in the experiment section.

3.1. CNN+FC

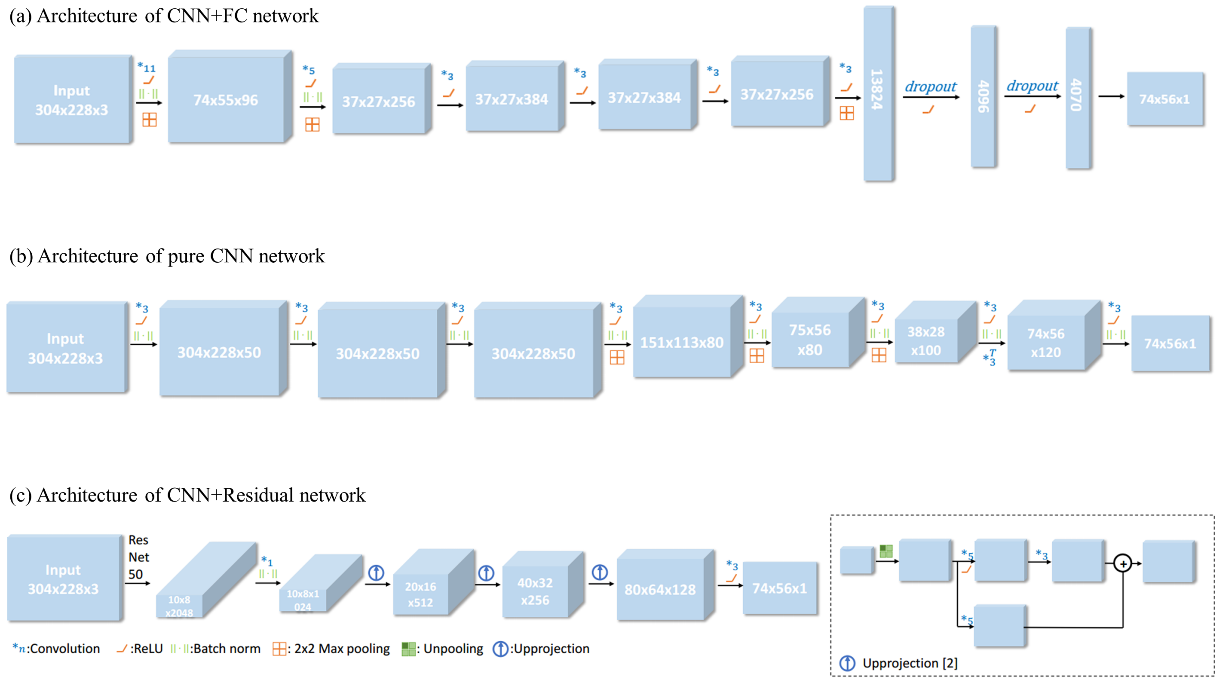
The first architecture follows the work in [3], where the authors used coarse and fine CNN networks to do depth estimation. We basically reimplemented the structure of the coarse network in the paper.

As shown in Figure 1(a), the network consists of two parts, convolution layers and fully connected layers.The input RGB image first goes through convolution layers with $11 \times 11, 5 \times 5, 3 \times 3$ filters. Batch normalization, ReLU layers and 2×2 max-pooling layers follow the convolution layers. After 6 convolution layers, the data goes through 2 fully connected layers, and the final output is resized to be the size of the ground truth depth map. Dropout is used after the first fully connected layer to avoid overfitting.

The total number of parameters of all the convolution layers are 27232. In contrast, the number of parameters of the two fully connected layers are 73293824, which is astronomical. As shown in the experiment part, this model can overfit thousands of images, but fails to get reasonable result on validation set. Even adding dropout layers won't fix the problem.

3.2. Pure CNN

In order to fix the overfitting issue, we replace the fully connected layers with convolution layers instead. This


(a) Architecture of CNN+FC network

(b) Architecture of pure CNN network

(c) Architecture of CNN+Residual network

Figure 1. Architecture of three networks used in this project.

heavily reduces the number of parameters used, and when the number of convolution layers are proper, can achieve or outperform the CNN+FC network.

As shown in Figure 1(b), this network consists of 8 convolution layers. Since convolution layers take into account both global and local information, we expect this network to have a better behavior. As demonstrated in class, big convolution filters can be replaced with more layers of convolution with smaller size filters, which reduces the total number of parameters to train and can obtain similar results. We replaced the 11×11 and 5×5 filters with 3×3 filters. Each convolution layer is followed by batch normalization to facilitate training process, and then followed by ReLU activation layer. 2×2 max-pooling is used to downsample the original image to obtain the size of depth map.

We tried a variation of this architecture as well. When we read through the papers that use CNN based method for depth estimation, we found that the original image is usually downsampled to have a small height and width feature, and then upsampled to have the size of the final depth map. So we changed the last two layers of this network. The second last layer is followed by a 2×2 max-pooling layer, and the last layer is replaced with a transposed convolution layer to upsample the input. This architecture is referred to as CNN-transpose in the experiment section.

3.3. CNN+Residual

Our third and most promising architecture follows the work in [11]. The essence is that we do transfer learning with extracted image features, and the transfer learning involves not only training the fully connected layer as we do in classification tasks, but also convolution and up projection to construct a depth map.

The structure is shown in Figure 1(c). The input RGB image is fed as input to the pretrained ResNet-50 network. We take the extracted image feature before fully connected layer, which has the dimension of $10 \times 8 \times 2048$. In order to obtain depth map with higher resolution, we need to do upsampling. As illustrated in [33], unpooling layers increase the spatial resolution of feature maps by performing the inverse operation of pooling. The up projection block used in this network consists of unpooling layers with convolution layers, which is shown in the lower right in Figure 1. An input feature with size $H \times W \times C$ first goes through an unpooling layer, which doubles the feature size to $2H \times 2W \times C$ by mapping each entry to the upper left corner of a 2×2 kernel. The expanded entries are filled with zeros or the averaged value of its two nearest neighbors. The expanded data is then fed into a 5×5 convolution layer, which is guaranteed to be applied to more than one non-zero elements at each location. This is then followed by ReLU activation and a 3×3 convolution. In addition to this up convolution architecture, a *projection* connection is built from lower resolution feature map to the block output. To keep the dimension consistent, an up-convolution (unpooling and 5×5 convolution) is also added on this branch.

4. Dataset

We use NYU Depth Dataset V2 [29] for this task. This dataset is composed of 4K indoor scenes, taken as video sequences using a Microsoft Kinect camera. Because the depth and RGB cameras operate at different variable frame rates, we associate each depth image with its closest RGB image in time, and throw away frames where one RGB image is associated with more than one depth. We use the camera projections provided with the dataset to align RGB and depth pairs; pixels with no depth value are left missing and are masked out. To remove any invalid regions caused by windows, open doorways and specular surfaces we also mask out depths equal to the minimum or maximum recorded for each image.

KITTI is another popular dataset frequently used in depth estimation projects. Since our calculation resources are limited, we will compare the feasibility of our algorithm based on the training result on NYU dataset only.

5. Experiments

In this section, we give qualitative results of our models as well as quantitative metric evaluations. We also compare the performance of different loss functions on this task. All experiments are implemented in PyTorch.

5.1. Visualization of depth output

5.1.1 CNN+FC

We set up a simple version of CNN+FC architecture and overfit on a small dataset. In our implementation, the coarse part of the CNN architecture in [3] is used and batch normalization is added right after each convolution layer. Dropout is added after the FC layer with the probability of 0.5. Mean square error function is used as our loss function on a per pixel basis. Adam is chosen as our optimizer, with the learning rate set to be $1e-3$. L2 regularization is added with the weight decay to be $1e-4$.

3590 images are fed to our training process and 399 images are used for validation. Batch size is set to 32 during training. Preliminary results are shown in 2. On the left is the loss over time. From this graph, we are clearly overfitting our data: the training loss keeps decreasing while the validation loss reduces to the point of 1300 then stops decreasing. On the right are some examples from our training sets. The first row is the input images. The second row is the ground truth. The third row is the prediction on our training data, which, again, is the result of overfitting our data.

5.1.2 Pure CNN

This fully convolutional network consists of 8 convolution layers, and batch normalization and ReLU activation follows each convolution layer. We removed fully connected layers in this model to avoid overfitting. Although we reduce the number of parameters to train drastically, the memory usage of convolution layers are more, and we reduced the batch size to 8 to avoid 'out of memory' error. Learning rate is set to $1e-3$. We used Adam optimizer in this task.

The results of pure CNN network and CNN-transposed network are shown in Figure 3. These are trained on 300 images over 20 epochs. As we tried to train with thousands of images, the average training loss is difficult to converge, but the training and validation loss drop in the same pace.

The CNN-transposed network uses transposed convolution to upsample and obtain the depth map. As we can see from the figure, both methods produce visually reasonable depth prediction, but artifacts exist. The depth map of transposed convolution model is strided. This implies this upsampling method is not optimal. In the convolution model with only downsampling convolution layers, the predicted depth maps are smooth, but we can see that the depth maps resemble more of the original images than the depth images. More training cycles may resolve this problem.

5.1.3 CNN+Residual

The CNN+Residual model is the architecture proposed by [11] which uses resNet50 [5] without the last fully-connect layer and pooling layer as feature extractor, and then uses *upprojection* blocks to upsample the extracted feature. Due to the size of the network, training of this network takes a long time. We trained this network on a smaller dataset with 500 images. We varied the unpooling layer in the *upprojection* and compare the results in 5.4. We also tested the influence of using pretrained parameters in 5.5.

5.2. Metrics evaluation

For images with ground truth depth value y and predicted value $\hat{y}$, we use three different metrics to quantify the performance of different network architecture: percentage of pixels with relative error $t = \max\{\frac{y}{\hat{y}}, \frac{\hat{y}}{y}\}$ less than 1.25, absolute relative difference $\frac{|y-\hat{y}|}{y}$ and root mean squared error $\sqrt{\frac{1}{n}(\hat{y}-y)^2}$. We compare the performance in 1. CNN+ResNet yields the best performance. This may because of the good feature extraction done by the pretrained ResNet50.

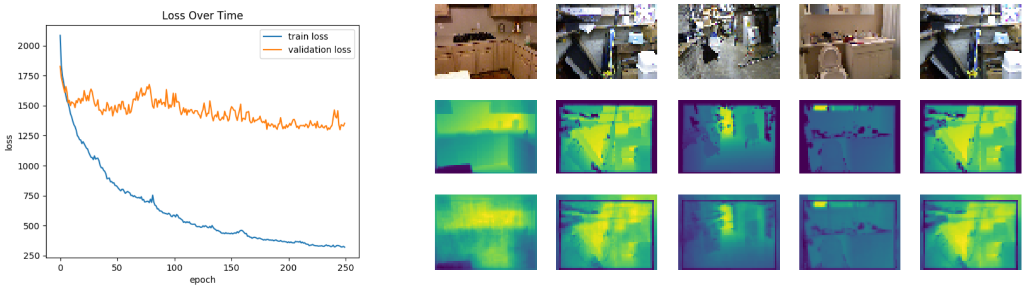


Figure 2. Training results of CNN+FC network

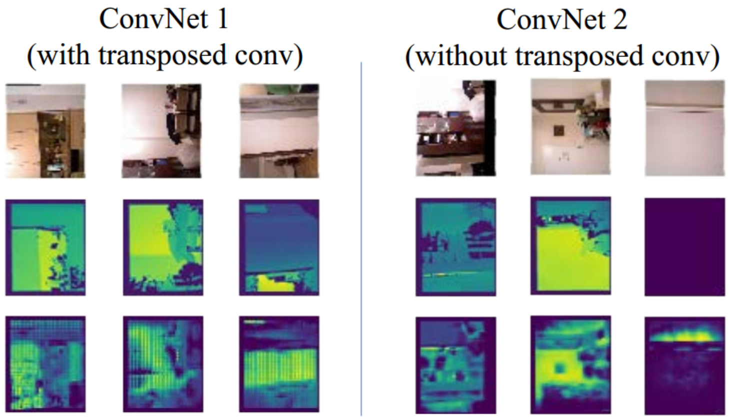


Figure 3. Training results of CNN network

	$t < 1.25$	Abs rel diff	RMSE
CNN+FC	0.307	3.4e5	1.12
CNN	0.195	1.2e6	1.22
CNN+trans	0.143	3.2e5	1.23
CNN+Res	0.634	2.35e4	1.74

Table 1. Metric evaluation results

5.3. Comparison of loss functions

For a image of n pixels with the true depth y and predicted depth $\hat{y}$, the loss function $B(\hat{y}, y)$ is defined in three different ways:

5.3.1 Mean squared error

$$B(\hat{y}, y) = \frac{1}{n}(\hat{y} - y)^2$$

5.3.2 Scale-invariant mean squared error

$$B(\hat{y}, y) = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n}(\sum_i d_i)^2$$

where $d_i = logy_i - log\hat{y}_i$, $\lambda \in [0, 1]$

5.3.3 Berhu loss

$$B(\hat{y}, y) = \frac{1}{n}\begin{cases} |\hat{y} - y| & \text{if } |\hat{y} - y| \leq c \\ \frac{(\hat{y}-y)^2+c^2}{2c} & \text{otherwise} \end{cases}$$

where $c = \frac{1}{5}max_i(|\hat{y}_i - y_i|)$ for each image.

5.3.4 Comparison

We compare the performance of different loss function using fully convolution network without transpose convolution in 2. Network using RMSE loss yields the highest percentage with 1.25 error threshold. Meanwhile scale-invariant loss results in the smallest absolute relative difference and Berhu loss gives rise to the smallest RMSE.

	$t < 1.25$	Abs rel diff	RMSE
RMSE	0.380	4.52e6	1.23
Scale-invariant	0.195	1.21e6	1.22
Berhu	0.352	1.76e6	0.98

Table 2. Metric evaluation of different loss functions

5.4. Comparison of Different Unpooling Methods

In CNN+Residual model,we compared two different unpooling methods in upprojection block. One is to use zero to fill the empty entries, another is to use the average value of the two nearest neighbor to fill the empty entry (average unpooling). We compared the two methods by training the model on a small dataset of 25 images with 120 epochs. The results are shown in Table 3 and Figure 4

	$t < 1.25$	Abs rel diff	RMSE
Normal Unpooling	0.0537	1.28e6	6.18
Average Unpooling	0.0557	1.27e6	6.37

Table 3. Metric evaluation of different unpooling methods

From Figure 4, we could clearly see that the generated depth maps of normal unpooling methods have many grid patterns which are likely due to the different portion of zeros received in the downstream CNN filters, where the outputs of average unpooling are much more smooth. The performance of the two different unpooling methods are similiar. Probably more training epochs would make them have a larger difference.

5.5. Influence of Transfer Learning

There are pretrained parameters of CNN+Residual model on NYU dataset, which enables transfer learning. To test the influence of initial parameters, we trained the CNN+Residual model with two different initial parameters. For both models, we used pretrained resNet50 [5]. Then for the upprojection blocks, we initialized one with small random numbers and the other with pretrained parameters. We trained two models on a dataset of 500 images with 20 epochs. The results are shown in Table 4 and Figure 5.

	$t < 1.25$	Abs rel diff	RMSE
Random Initialization	0.634	2.35e4	1.74
Transfer Learning	0.729	1.43e6	0.734

Table 4. Metric evaluation of different initialization

From Figure 5 we could see that, for same number of epochs, the random initialized model performs much worse than the transfer learning one. There are obvious grid pattern in the random initialized one. The metric evaluation of the transfer learning model is also better. These results show the benefit of using pretrained models and doing transfer learning.

6. Conclusion

Depth prediction using monocular image plays an essential role in many practical applications and is challenging because of the inherent ambiguity. In this project, we approach this problem using CNN and compare the performance of different CNN architecture on NYU Depth Dataset V2. CNN with fully connected layer like the one used in [3] is powerful but can easily overfit on the dataset because of the large number of parameters in fully connected layer. This motivates us to use only convolutional layers and stack more layers to increase the receptive field. This architecture yields acceptable result while reduces a large number of model parameters. We also try [18] which is a CNN architecture using transfer learning on the ResNet [5] and are able to get reasonable results on the validation set.

References

[1] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *arXiv preprint arXiv:1605.02305*, 2016.

[2] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[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*, pages 2366–2374, 2014.

[4] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[6] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM transactions on graphics (TOG)*, 24(3):577–584, 2005.

[7] K. Karsch, C. Liu, and S. Kang. Depth extraction from video using non-parametric sampling. *Computer Vision–ECCV 2012*, pages 775–788, 2012.

[8] J. Konrad, M. Wang, and P. Ishwar. 2d-to-3d image conversion by learning depth from examples. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 16–22. IEEE, 2012.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–96, 2014.

[11] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016.

[12] B. Li, Y. Dai, H. Chen, and M. He. Single image depth estimation by dilated deep residual convolutional neural network and soft-weight-sum inference. *arXiv preprint arXiv:1705.00534*, 2017.

[13] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

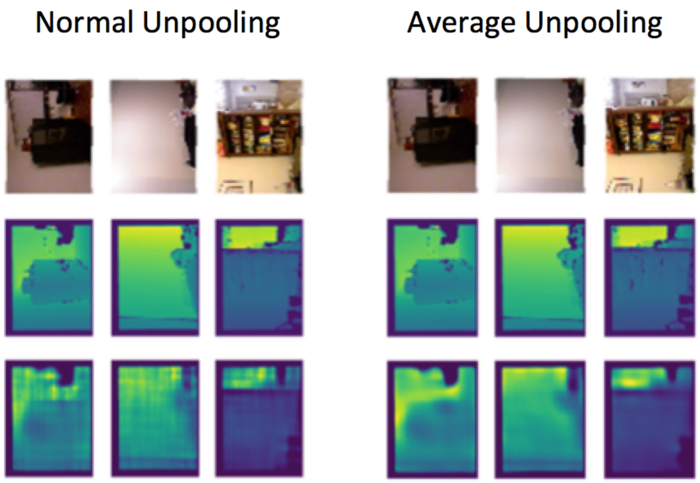


Figure 4. Visualization of different unpooling methods

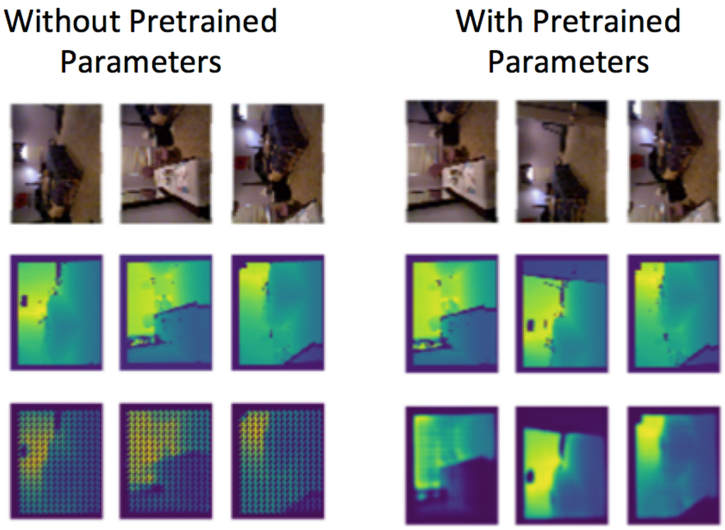


Figure 5. Visualization of different initialization

[14] A. Liaw and M. Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[15] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1253–1260. IEEE, 2010.

[16] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2011.

[17] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[18] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2016.

[19] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2014.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[21] G. Qian and R. Chellappa. Structure from motion using sequential monte carlo methods. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 614–621. IEEE, 2001.

[22] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2759–2766. IEEE, 2012.

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[24] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5506–5514, 2016.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[26] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.

[27] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2009.

[28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *Computer Vision–ECCV 2012*, pages 746–760, 2012.

[29] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *Computer Vision–ECCV 2012*, pages 746–760, 2012.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] S. Suwajanakorn, C. Hernandez, and S. M. Seitz. Depth from focus with your mobile phone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3497–3506, 2015.

[32] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2015.

[33] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2018–2025. IEEE, 2011.

[34] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.