# Single Image Depth Estimation Using a Multi-Scale Convolutional Neural Network

Xiaoyu Zhang, Mingxiang Fan, Mengzheng Pan, Lili Zhu

***Abstract*—Compared with stereo image depth estimation, finding depth relations from a single image is less straightforward. Moreover, the mapping between a single image and the depth map is inherently ambiguous, and requires both global and local information. In this project, we present a Multi-Scale Deep Convolutional Neural Network for single image depth estimation. The method we used in this project employed two deep network stacks: a coarse global prediction based on the entire image, and another to refines this prediction locally. This method is evaluated on the NYU-depth v2 dataset and compares with several previous works including network structures of AlexNet and ResNet.**

***Index Terms*—Convolutional Neural Network, Depth Estimation**

## I. Motivation

Depth prediction from RGB images is a crucial topic in robotics, virtual reality, and 3D modeling because it is beneficial for understanding geometric relations within a scene. In turn, such relations help extract more information from objects and their environment, usually leading to enhancements in current recognition projects, as well as speed up the development of further applications, such as 3D modeling, physics and support models, robotics, and potentially reasoning about occlusions.

Nowadays, although many researchers have done much research on estimating depth based on stereo images or motion, there has been relatively little on predicting depth from a single 2D image. However, in real-world practice, the monocular case arises more often and it is reasonable to expect a wide and convenient usage of single image depth estimation. For example, images distributed on the web and social media outlets, real estate listings, and shopping sites, are all monocular cases. Therefore, we decided to focus on depth prediction for monocular cases.

Depth prediction for the monocular image is more difficult than stereo ones. Provided accurate image correspondences, depth can be recovered deterministically in the stereo case. With stereoscopic images, depth can be computed from local correspondences and triangularization, while estimating the geometry relationship of the camera positions will help with the accuracy. By contrast, methods for inferring depth from a single image have involved image segmentation, texture variations, texture gradients, interposition and shading. It is barely possible to get good result without machine learning algorithms to automatically learn these complex tasks. Thus in our work, we performed supervised learning using convolutional neural network to achieve acceptable performance.

In this project, we present an approach for estimating depth from a single image. We directly regress on the depth using a neural network with two components: one that first estimates the global structure of the scene, then a second that refines it using local information. The network is trained using a loss that combines $l_2$ norm and scale-invariant error. We used raw data of NYU-Depth V2 [1], which is shown in Fig.1, to train our model, and validated our model on another dataset which was provided by Professor Matthew Johnson-Roberson.

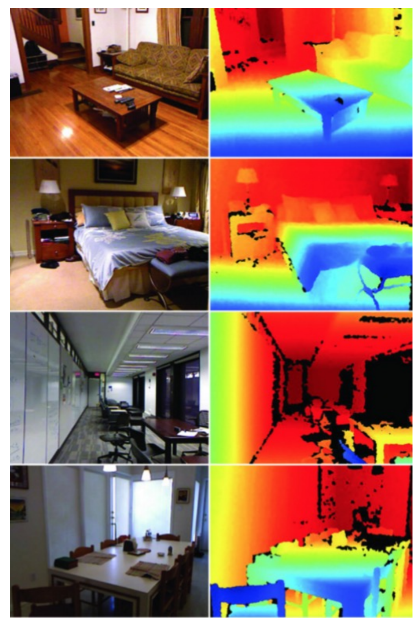

Fig. 1: Example of NYU dataset

## II. Previous Work

Stereo depth estimation has been a broadly used and extensively investigated approach of recovering depth information, using image pairs of the same scene to reconstruct 3D shapes. Scharstein et al. [2] have provided a survey and evaluation of many methods for 2-frame stereo correspondence. They use techniques like matching, aggregation and optimization to organize the system. Snavely et al. [3] apply a multiview stereo method to a creative application, matching across views of many uncalibrated consumer photographs of the same scene to create accurate 3D reconstructions of common landmarks.

In order to obtain better results and relax the need for careful camera alignment [4] [5] [6] [7], some machine learning

X. Zhang , M. Fan , M Pan and L Zhu were with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI, 48109, USA e-mail: zhxiaoyu@umich.edu

techniques can also be applied to stereo case. Konda et al. [4] predict depth from stereo sequences by training a factored autoencoder on image pathes. However, this approach is dependent on the local displacements of the stereo.

Estimating depth from a single image has made impressive progress these years. Many approaches has been tried and applied in this area. Linear regression and a Markov Random Field (MRF) is used by Saxena et al. [8] to predict depth from a set of image features and 3D model generation on Make3D [9] dataset. However, the system cannot give as good results in less controlled settings for its dependence on horizontal alignment of images. To improve their approach, they use superpixels to enforce the consistency between neighboring regions. Liu et al. [10], inspired by their work, solving the problem of depth estimation along with semantic segamentation, using predicted segmentation labels as constraints to achieve better 3D reconstruction. Hoiem et al. [11] also take use of categorizing of image regions into geometric structures(ground, sky, vertical) and use it to compose a simple 3D model of the scene.

Then, Ladicky et al. [12] integrate semantic object labels with monocular depth features. However, their system need handcrafted features and the use of superpixels to segment the image. Karsch et al. [13] provide a method to estimate depths of static backgrounds from single images with a kNN transfer mechanism based on SIFT Flow [14]. They improve their results with motion information to better estimate moving foreground subjects in videos. Although this achieves better alignment, it needs the entire dataset at runtime and involves an expensive alignment procedure.

On the contrary, Eigen et al. [15] presented a method that learns a set of network parameters which are easy to store and can be applied to real-time images. Their method estimate depth map by employing two deep network stacks as shown in Fig.2: coarse network makes a coarse global prediction based on the entire image, fine-scale network refines this prediction locally. In addition, AlexNet [16] and ResNet [17] are also two typical Convolutional Neural Networks that can be used for depth estimating.Their network structure are shown in Fig.3 and Fig.4.The ResNet explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We follow Eigen's approach, but we reduce one layer for each partial network. However, We achieve some accuracy as them with less computational complexity naturally.

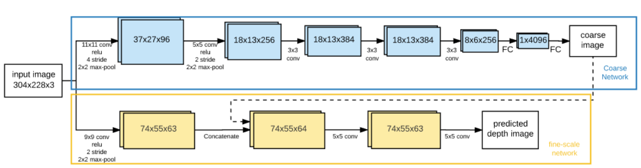


Fig. 2: Original network by Eigen et.al

## III. Problem Statement

Our work provides a solution to estimating pixel-wise depth information from single RGB images. In our problem, we have

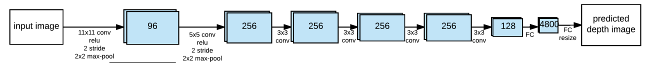


Fig. 3: Modified Alexnet network for depth estimation

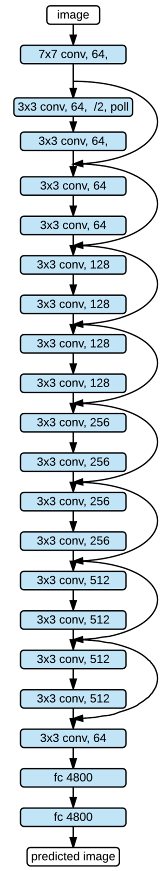


Fig. 4: ResNet-18 network for depth estimation

access to a typical RGB image and are asked to give a depth estimation for each pixel of the image. Then, we evaluate our method in both quantitative metric comparison and qualitative visualization comparison. More detail of both approach and evaluation will be in the following section.

The RGB-D data format of Kinect serves as a perfect training set of our problem, as the invalid depth information of pixels could be easily extracted and margined out, while data collected by RGB camera and Lidar could also be a good data set for our models, since we can use bilateral filter to get pixel-wise estimated depth information.

## IV. Approach

### A. Model Architecture

Our network contains two components to extract different information from two different scales, shown in Fig.5. The first one is a coarse-scale network, which predicts the depth of the scene at a global level. The second one is a fine-scale network, which focuses on refined details within local regions. As to coarse network, we used original images as input. As for the fine-scale network, we passed the original image through a convolution layer and combined it with the output of the coarse network as the first-layer image features. In this way, the local network has access to both origin image and global prediction, which enables the network to incorporate finer-scale details.

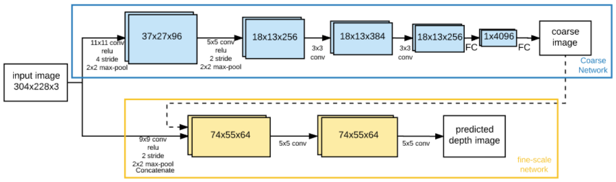


Fig. 5: Our Modified Eigen Network Structure

*1) Global Coarse-Scale Network:* The goal of the coarse-scale network is to predict the overall depth map structure using a global view of the scene. As shown in Fig.5, To get the entire image in their field of view contained, the upper layers of this network are fully connected. The lower and middle layers combine information from different parts of the image to a small spatial dimension via max-pooling operations. Therefore, the network can integrate a global understanding of the full scene to estimate the depth. By this way, we can make the best use of cues such as vanishing points, object locations, and room alignment, to understand single 2D images. By contrast, a local view, which is commonly used in stereo matching, is not enough to notice important features such as these.

As illustrated in Fig.5, there are four feature extraction layers of convolution and max-pooling, followed by two fully connected layers, in the global-coarse-scale network. The input, feature map and output sizes are also given in Fig.5. The final output is at $1/4$-resolution compared to the input, which is downsampled from the original images by a factor of 2, and corresponds to the input (as we describe later, we use convolutional layers with same output size as the input, which may introduce error on boarder areas. The error are relatively small and would not cause significant defects of the performance evaluation).

Because the spatial dimension of the output is larger than that of the topmost convolution feature map, we chose to make the top full layer to learn templates over the larger area before passing the prediction to the fine network, instead of limiting the output to the feature map size by upsampling. It performs better than the upsampled output of the extracted features prediction.

We used rectified linear units for activations at all hidden layers.

*2) Local Fine-Scale Network:* After we get the overall depth map structure using the coarse-scale network, we let image go through a second, fine-scale network to make local refinements, aligning the coarse prediction it receives with local details such as object and all edges.

This fine-scale network has three convolution layers and takes a pool step to edit the first layer edge features to get the refined output, which is a relatively high-resolution output at $1/4$ of the size of input images. The original image is fed in to the first layer of this network (first fine-scale network block in Fig.5 while the output of coarse network is fed in as an additional low-level feature map to the second layer, concatenated with the pooled output of first fine-scale layer (second fine-scale network block in Fig.5. Comparing to previous work from Eigen et.al [15], we found out that the affects of the output of the coarse layers should be emphasized, thus we eliminated the number of channels produced directly from origin input image. The rest layers gives same output size using zero-padded convolution. Since the depth of each pixels are positive, we simply use rectified linear activation function for all the activations of the hidden units.

Aiming at extract the detail depths information, compared to using the entire image pixels in coarse-scale network, the fine-scale network views only $49x49$ pixels of the input image, while the last three convolutional layers which deals with both coarse output and origin image actually take a $13x13$ area into consideration as generating the final depth result for each pixel.

*3) training:* We explored several tricks about training this network.

We trained the full network at once at first, and got acceptable result. while the output of the coarse layers expresses large-scale information, it's not easy for human to recognize and explain the correctness of our design of using multi-scale network to perform the task.

We then trained the coarse network using the resized ground truth of depth images, and as we later started training the fine-scale network, we did not backpropagate through the coarse network, maintaining the coarse-scale output fixed. We can see from Fig.6 that the output of the coarse network gives a good estimation of large scale depth estimation, while the total error of the overall output decreases as we apply this trick, which we will describe later.

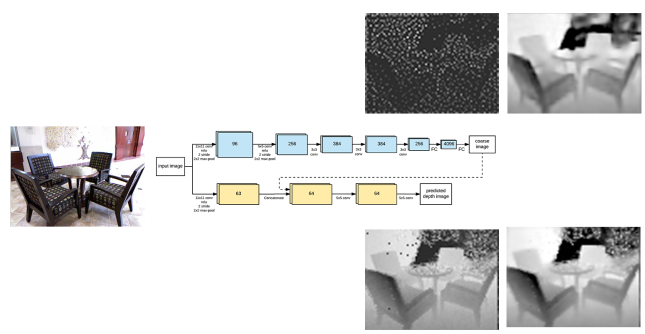

Fig. 6: coarse result with pretrained and without

## B. Scale-Invariant Error

We used a combination of $l_2$ error and scale-invariant error to measure the relationships between points in the scene, irrespective of the absolute global scale. We define the *scale-invariant mean squared error* as:

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

where $\hat{y}_i$ is predicted depth map, $y$ is ground truth, $i$ is the index of each n pixels. We should notice that since there exists unavailable depth data points in raw data from our dataset, only those pixels with available depth information are considered as computing the error. Setting $\lambda = 0$ reduces the loss to element-wise $l_2$ norm, while $\lambda = 1$ is the scale-invariant error exactly. In practice, we choose $\lambda = 0.5$ to achieve a balance between both errors.

The following equivalent forms provides an additional ways to view metric.

$$D(y, \hat{y}_i) == \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}(\sum_i d_i)^2$$
$$= \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}\sum_{i,j} d_i d_j$$

The equation above expresses the error by comparing relationships between pairs of pixels $i, j$ in the output: the difference between each pair of pixels should be a bit of different in prediction and ground truth for lower error. Our error function added a penalty $-\frac{\lambda}{n^2}\sum_{ij} d_i d_j$ to the original $l_2$ error, which credits when two pixels with same direction are estimated to be opposite. In conclusion, if a prediction has mistakes which is consistent with another, it is am imperfect prediction.

In addition to the combined scale-invariant error, we also measure the performance according to several error metrics as comparison.

### C. Training Loss

In addition to performance evaluation, our combined error function could also be used as training loss. We define training loss of each sample as the following equation:

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

where $d_i = \hat{y}_i - y_i$ and $\lambda \in [0, 1]$, $\log y$ is log of prediction. We can change $\lambda$ to modify the weight of elementwise $l_2$, while $\lambda = 1$ is the scale-invariant error exactly. We take the average, $\lambda = 0.5$, finding that this produces good absolute-scale predictions while slightly improving qualitative output.

## V. EXPERIMENTS

In this section, we give qualitative results of our models and quantitative metric evaluations. All experiments are implemented on a desktop computer with GTX 1080 graphic card, core i7 6700k CPU and 32GB memory.

Since our model was originally inspired by Eigen's Network, we compared our result with the network proposed in their paper. Besides, we also implemented a network based on ResNet-18 in Fig.4 and a network based on Alexnet in Fig.3, and compared our result with these networks. Although ResNet can be as deep as 152 layers, we only compared our method with ResNet-18 because they have similar model complexity.

TABLE I: metirc evalution results

| | **t < 1.25** | **Abs rel diff** | **RMSE** |
|---|---|---|---|
| **AlexNet** | 0.9070 | 0.0899 | 0.1051% |
| **ResNet18** | 0.8954 | 0.1101 | 0.1076% |
| **Eigen** | 0.8951 | 0.1033 | 0.1009% |
| **Eigen-modified** | 0.9010 | 0.0877 | 0.0920% |

### A. Dataset

We use NYU-Depth v2 Dataset [1] for this task. This data set is composed of video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect. We train our model on a subset of this NYU-Depth v2 dataset. Then ,we also validated our model on another dataset which was provided by Professor Matthew Johnson-Roberson.

### B. Evaluation Metrics

For ground truth depth images $y$ and predicted images $\hat{y}$, we evaluate our method on three different metrics:
Percentage of pixels with relative error (larger means better performance)

$$t = max(\frac{y}{\hat{y}}, \frac{\hat{y}}{y}) < 1.25$$

Absolute Relative Difference:

$$e = \|\frac{y - \hat{y}}{y}\|$$

Root Mean squared Error(RMSE):

$$e = \sqrt{\frac{1}{n}||y - \hat{y}||^2}$$

As shown in the Metric evaluation results Table.I, our modified-Eigen network is slightly better than the original Eigen network according to the evaluation, while surpassing the performance of Alexnet and resnet.

### C. Strengths

We can also see from Fig.7 that our modified Eigen network gives best prediction result, taking smooth and accuracy into account.

All the predicted results could sometimes cover the invalid pixels of the raw Kinect data, since the network can learn to predict the depth information from other parts of the dataset, which enables the neural networks to fix the raw depth image from kinect.

Our model uses less parameter than original Eigen network, and gets better performance, which indicates that our network structure is more suitable for predicting depths.

Besides, we also trained our model and original Eigen model on another dataset from Professor Matthew Johnson-Roberson from the University of Michigan. This dataset contains complex outdoor scenes from computer simulation. The predicted depth images and the original depth images are shown in Fig.8. As we calculate the error of our prediction versus Eigen's network, our work have better performance on this more complex dataset, which indicates that our network structure is more reasonable.

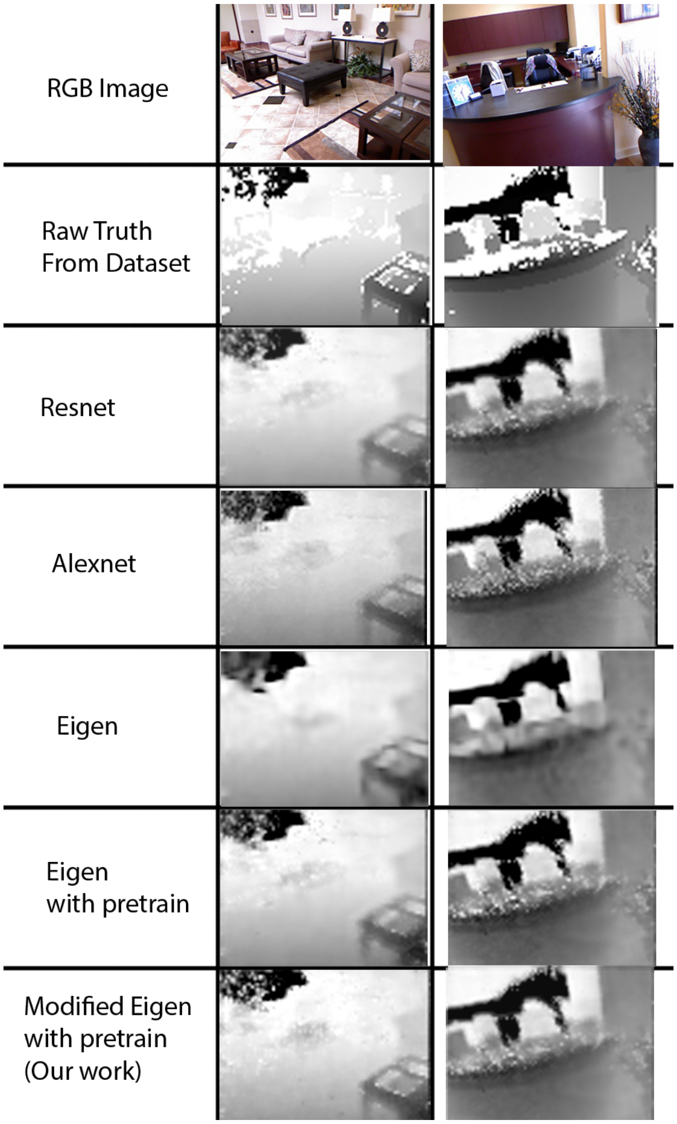


Fig. 7: Prediction results

### D. Weakness

Estimating depth from a single image is a challenge task for the ambiguity along with different complicated lighting and shading conditions of a single image. Also, different datasets may have different kinds of assumptions which may increase the difficulty of directly applying one model trained on one dataset to another dataset and give as good results. In our experiment, we train our model on a subset of the NYU-Depth V2 dataset, which breaks many assumptions from general cases. Besides, NYU-Depth V2 dataset consists mainly indoor images while there are datasets and real-life scenes that contains mainly outdoor images with complicated lighting conditions. Therefore, our model may have relatively poor performance on some other datasets.

Apart from applying to different datasets, our result still has spaces of improvement compared to ground truth depth image. Although we achieved better performance than some traditional convolutional neuron networks such as VGG, AlexNet, our result can still overlook some small details showed in ground truth map. On the other hand, because of using a fine-scale network to extract local details, sometimes the output also include some texture edges.

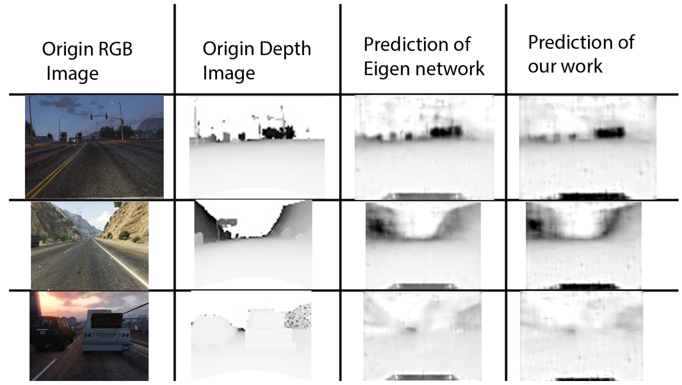


Fig. 8: prediction result on another dataset

Besides, because we have limited calculation power, only small-scale networks could be examined. To achieve state-of-the-art performance, we may need more complex structure for depth estimation, and our network structure may not be suitable to expand.

### E. Spaces of Improvement

To achieve a more state-of-the-art result, we can extend our method to add more scales to the system, such as successive finer-scaled local networks and extract the depth information which we cannot get from this model. We can also incorporate other 3D geometry information such as semantic segmentation and surface normal to help improve overall performance. Also, we can feed in our training system with images from different datasets which include complicated outdoor lighting conditions after taking proper preprocessing step to get better performance on different datasets.

## VI. Conclusion

Our system accomplishes depth prediction from single images through the use of two deep networks, one that estimates the global depth structure, and another that refines it locally at finer resolution. We achieve better results than some neural network structures from previous work with similar amount of parameters. In future work, we plan to extend our method to incorporate further 3D geometry information, such as surface normals. We'll also try to explore the performance of multi-scale neural networks on different types of tasks, and investigate some more complex network structures with stronger hardwares, if possible.

## Acknowledgment

This project is funded through the generous donation from parents of the team members in the form of tuition fee and computers(contribution shows in Fig.9). This project is advised by Prof.Pilanci from University of Michigan, and we got precious advises from the GSIs of EECS545. We would like to thank them for their love, advice and support.

| | Mengzhen Pan | Lili Zhu | Mingxiang Fan | Xiaoyu Zhang |
|---|---|---|---|---|
| Pre-processing of datasets | | | √ | |
| Network structure design | | √ | | √ |
| Tuning of hyper-parameters | √ | | √ | |
| Training and testing | √ | √ | √ | √ |
| Evaluate result | √ | | | |
| Paperworks | | √ | √ | √ |

Fig. 9: contribution table

## REFERENCES


[1] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[3] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM transactions on graphics (TOG)*, vol. 25, pp. 835–846, ACM, 2006.

[4] K. Konda and R. Memisevic, "Unsupervised learning of depth and motion," *arXiv preprint arXiv:1312.3429*, 2013.

[5] R. Memisevic and C. Conrad, "Stereopsis via deep learning," in *NIPS Workshop on Deep Learning*, vol. 1, p. 2, 2011.

[6] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun, "Continuous markov random fields for robust stereo estimation," *Computer Vision–ECCV 2012*, pp. 45–58, 2012.

[7] F. H. Sinz, J. Q. Candela, G. H. Bakır, C. E. Rasmussen, and M. O. Franz, "Learning depth from stereo," in *Joint Pattern Recognition Symposium*, pp. 245–252, Springer, 2004.

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

[9] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-d scene structure from a single still image," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[10] 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*, pp. 1253–1260, IEEE, 2010.

[11] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM transactions on graphics (TOG)*, vol. 24, no. 3, pp. 577–584, 2005.

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

[13] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from video using non-parametric sampling," in *European Conference on Computer Vision*, pp. 775–788, Springer, 2012.

[14] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "Sift flow: Dense correspondence across different scenes," in *European conference on computer vision*, pp. 28–42, Springer, 2008.

[15] 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*, pp. 2366–2374, 2014.

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

[17] 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*, pp. 770–778, 2016.