Perceptual Alignment for High-Quality Monocular Depth Estimation

Arjun Ashok
PSG College of Technology
arjun.ashok.psg@gmail.com

Abstract

Estimating the depth from a single RGB image is an ill-posed problem that requires both global and local information. This problem plays an important role in various applications including autonomous driving and scene understanding. Here, we leverage an end-to-end trainable encoder-decoder architecture with interleaved skip connections to tackle this problem. We then propose a method that aligns high-level representations using the perceptual loss, and demonstrate its advantages over using only low-level per-pixel loss functions for this task. We then show that our model produces comparable quantitative and qualitative results on the NYU Depth v2 dataset benchmark.[1].

1. Introduction

Depth estimation is a fundamental problem in computer vision, which helps in providing richer representations of objects and their environments. It is often used to improve other classic computer vision tasks such as 3D modelling, robotics, video surveillance, semantic segmentation[4], human pose estimation[39] and autonomous driving[10].

The problem can be tackled using two different approaches, either by using stereo images or by using monocular images. While many successful methods for stereo depth estimation have been proposed, the problem of estimating the depth from a single RGB image often arises in practice, when direct depth sensing equipment is not available. This problem is difficult, as a single 2-D RGB image can correspond to an infinite number of real-world scenes. It is also a computationally difficult problem, as the global properties of the scene such texture variation or defocus information should be captured[17].

Most of the traditional methods for depth estimation rely on the assumption of having observations of the scene, either in space or time (e.g., stereo or multi-view, structure from motion) [36][37]. Recently, convolutional neural networks have significantly improved the performance of monocular depth estimation methods[17]. Here, we propose an encoder-decoder architecture that captures the implicit relation between the RGB image and the depth values. For optimization, in addition to a combination of three standard per-pixel loss functions, we extract high-level features from a pretrained network to assess the quality of the depth maps. By evaluating on the benchmark NYU Depth v2 dataset [31], we show that our model achieves considerable improvements in single-image depth map estimation.

[1]Code is made publicly available at this https url

2. Related work

2.1. Traditional methods

Classic methods relied heavily on strong assumptions about the scene geometry, and relied on carefully handcrafted features and probabilistic graphical models which exploit the geometry information. Saxena *et al*. [35] introduced a model which used a discriminatively-trained Markov Random Field (MRF) to estimate the depth. Later, Liu *et al*. [26] used a joint approach of combining semantic segmentation along with the depth estimates to get better results. Ladicky *et al*. [21] used a classification based approach for joint depth estimation and semantic segmentation. Other approaches rely on camera motion[33], variation in illumination[42] or variation in focus[38] to estimate the depth from a single-view image.

This problem has also been tackled by using feature-based mapping between an RGB Image and a repository of RGB-D images to find the nearest neighbours, which are then warped and combined to produce the final depth map. Konrad *et al*. [19] use cross-bilateral filtering to smooth the mean retrieved depth map. Karsch *et al*. [16] use a global optimization scheme after warping using SIFT flow [26]. Liu *et al*. [29] formulate the optimization problem as a Conditional Random Field (CRF) with continuous and discrete variable potentials. It is important to note that these approaches rely on the presumption that similarities between regions in the RGB images imply similar depth cues.

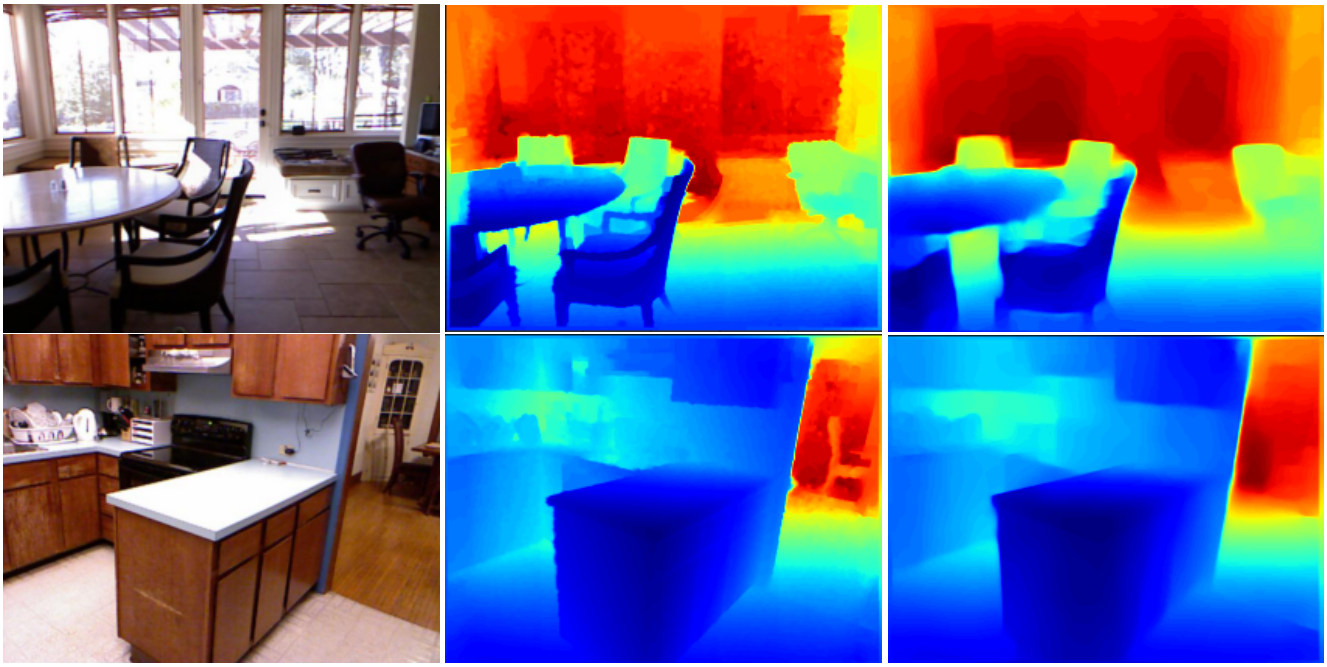

Figure 1. From left to right: Input RGB images, ground truth depth maps, and our estimated depth maps

2.2. Deep learning based methods

Since the success of Alexnet[20] on the Imagenet dataset[3], many approaches based on CNNs have been proposed. Eigen *et al*. [5] were the first to approach this problem using CNNs. They used two networks, where one network made a coarse prediction of the depth map, and the other refines the prediction locally. Further, in [4], three stacks of CNN are used to additionally predict surface normals and labels together with depth. Roy *et al*. [34] combined random forests[25] and CNNs, using very shallow architectures at each tree node, thus limiting the need for big data. Methods combining graphical models and CNNs have also been proposed[27][28][17].

Since the introduction of fully-convolutional networks [30] which allow arbitrary-sized inputs and return spatial outputs, they have been used to tackle the depth estimation problem [2][24]. Laina *et al*. [22] introduced efficient residual up-sampling blocks which improved the depth maps. Godard *et al*. [9] used an unsupervised learning approach. Alhashim *et al*. [1] used transfer learning to estimate high-quality depth maps. In this work, we use a network with an architecture similar to that in previous works. Our main contribution is applying the perceptual loss introduced by Johnson *et al*. [15] to this problem, with a trained network's feature maps at carefully selected layers being used to improve the quality of the depth maps.

3. The Approach

3.1. Network architecture

Our model consists of an encoder network which downsamples the input RGB image multiple times, and a decoder network which iteratively upsamples the downsampled input back to the original shape. The intermediate layers of the encoder and the decoder are connected by means of skip-connections. Figure 2 depicts our model's architecture. The encoder and decoder networks are described below.

3.1.1 The Encoder

In general, the encoder can be any network which downsamples the input to half the size at every level. We found that the Densenet-121[13] architecture works better than other architectures. The Densenet architecture is composed of a series of dense blocks, which consists of multiple layers in which for each layer, the feature-maps of all preceding layers are used as inputs.

Consequently, the l^{th} layer receives the concatenation of the feature-maps of all the preceding layers $x_0, x_1, ..., x_{l-1}$ as input:

$$x_l = H_l([x_0, x_1, ..., x_{l-1}])$$

H_l is a composite function[12] consisting of Batch-normalization[14], ReLU activation[8] and a 3 x 3 convolution. Each dense block contains a different number of these layers. The number of channels k in each layer x_i of a dense block is the same and is known as the growth rate

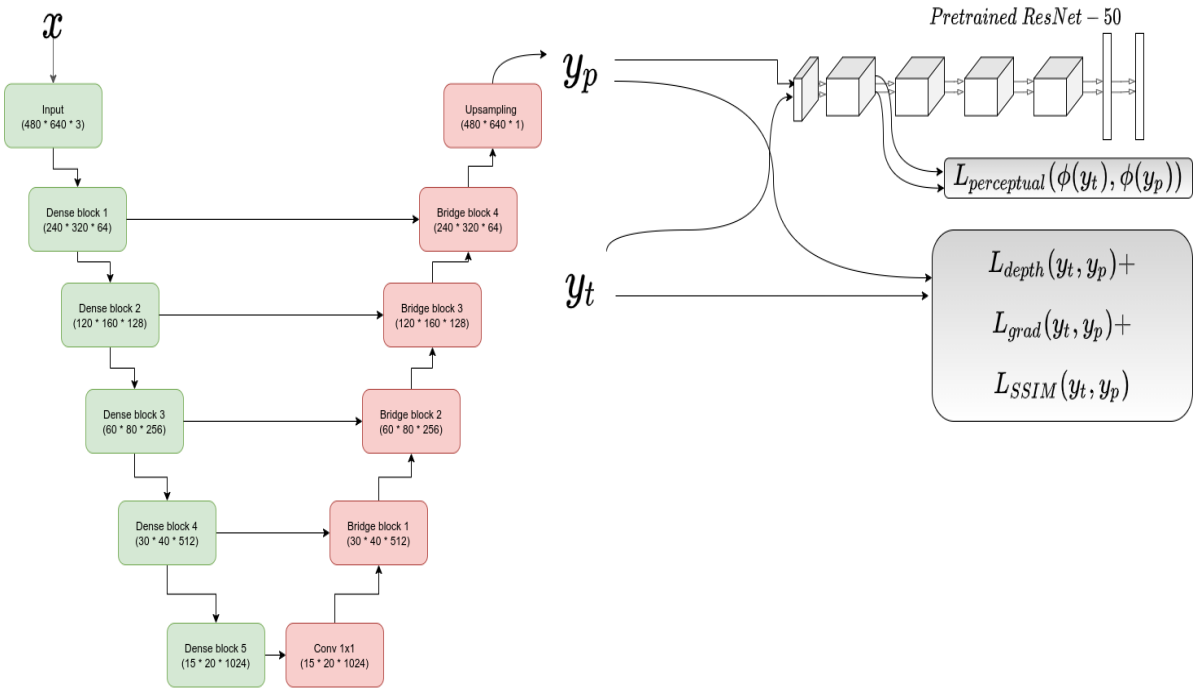

Figure 2. **Model Architecture**. The numbers in the brackets of each block denote the dimensions of the output of the block. Figure 3 gives an example of the dense block. Figure 4 gives the bridge block.

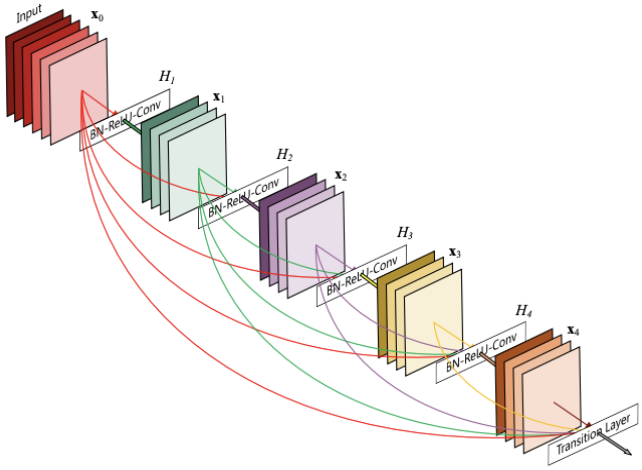

Figure 3. **An example of the dense block** with 5 layers and growth rate of 4. Taken from [13].

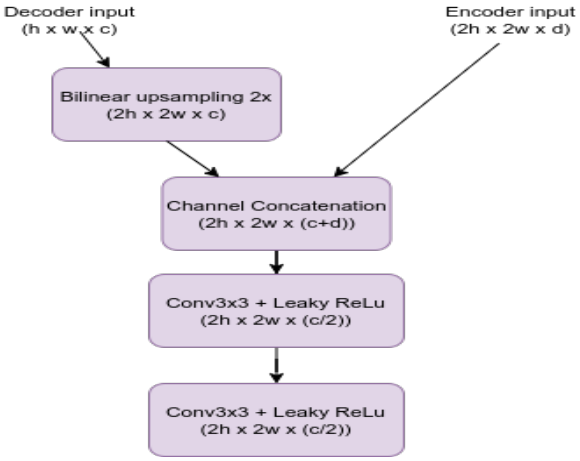

Figure 4. **The bridge block**

of the dense block. The last layer of every dense block is the transition layer which consists of a batch normalization and an 1 x 1 convolution followed by 2 x 2 average pooling, thus downsampling the input. These dense blocks are stacked on top of each other, thus downsampling the input at every level after a series of densely-connected convolutions. Figure 3 depicts a dense block consisting of $l = 5$ layers, with a growth rate of $k = 4$.

3.1.2 The Decoder

The input to the decoder is the encoder's output followed by a 1 x 1 convolution. The decoder consists of a series of bridge blocks, which takes two inputs: 1. The output of the previous block of the decoder & 2. The output of the corresponding dense-block of the encoder, a skip-connection. Each bridge block first upsamples input 1 to match the dimensions of input 2, followed by the concatenation of the channels of the inputs and two 3 x 3 convolutions each followed the Leaky-ReLU activation. The last bridge block does not have a skip-connection, instead just upsamples the previous bridge block to match the dimensions of the input RGB image. This output denotes the predicted depth map. Figure 4 depicts the bridge block.

3.2. Optimization Objectives

For training our network, we define the loss function L between y_t, the true depth map and y_p, the predicted depth map, as

$$L(y_t, y_p) = \lambda * L_{depth}(y_t, y_p) + L_{grad}(y_t, y_p) + L_{SSIM}(y_t, y_p) + \gamma * L_{perceptual}(\phi(y_t), \phi(y_p))$$

The first term L_{depth} is the point-wise L_1 loss, defined as

$$L_{depth}(y_t, y_p) = \frac{1}{n}\sum_{k=0}^{n} |y_{t_k} - y_{p_k}|$$

This loss has a weightage of λ, which is a hyperparameter. In our experiments, we found that $\lambda = 0.1$ works well.

The second term L_{grad} is the point-wise gradient loss, defined as

$$L_{grad}(y_t, y_p) = \frac{1}{n}\sum_{k=0}^{n} |g_x(y_{t_k} - y_{p_k})| + |g_y(y_{t_k} - y_{p_k})|$$

where g_x and g_y are difference between the x and y components of the depth map. This loss ensures a smooth depth map to be output by the network.

The third term L_{SSIM} is the structural dis-similarity loss, which uses the structural similarity term(SSIM)[40], a commonly used metric for image reconstruction tasks. The $SSIM$ metric is based on perceptual quality, and considers image degradation as perceived change in structural information. Internally, the $SSIM(x, y)$ term consists of comparisons with respect to the luminance, constrast and structure. As the structural similarity term($SSIM$) has an upper-bound of one, the loss L_{SSIM} is defined as

$$L_{SSIM}(y_t, y_p) = \frac{1 - SSIM(y_t, y_p)}{2}$$

The fourth term $L_{perceptual}$ is the feature reconstruction loss, inspired by Johnson *et al*. [15]. Rather than encouraging the pixels of the depth map y_p to match exactly with y_t, we instead encourage them to have similar feature representations as computed by a pretrained loss network ϕ. In our case, we choose the pretrained network ϕ to be Resnet-50 [12], and we choose the feature representations of the first residual block of size 56 x 56 x 256. The lower layers tend to localize well, and thus reconstructing lower layers' representation results in better localization of depth in the output of the main network. We use the standard L_1 loss to minimize the difference between the feature representation of the predicted depth map $a_p = \phi(y_p)$ and the feature representation of the true depth map $a_t = \phi(y_t)$ as

$$L_{perceptual}(y_t, y_p) = \frac{1}{n}\sum_{k=0}^{n} |a_{t_k} - a_{p_k}|$$

This loss has a weightage of γ, which is a hyperparameter. We found that setting $\gamma = 0.5$ worked well.

4. Experiments

4.1. Dataset

The NYU Depth v2 dataset[31] is a dataset composed of images of size 480 x 640 and their corresponding depth maps for various indoor scenes. We use a subset of the dataset comprising 50K training samples across 249 scenes, and 654 test samples across 215 scenes[5]. The missing depth values are filled using the inpainting method proposed in [23]. The training data is augmentated using two policies. One is the horizontal flip with a probability of 0.5, and the other is the RGB channel swap with a probability of 0.25. At test time, the final depth map is computed by averaging the prediction of the input image and the prediction of the horizontally mirrored input image.

4.2. Implementation Details

The model is implemented using the Pytorch framework[32]. The encoder network is pretrained on the Imagenet dataset[3]. The decoder is initialized following [7]. The total number of parameters in the network is approximately 15M parameters. We use stochastic gradient descent(SGD) with the Adam optimizer [18] with a learning rate of $3e-4$ and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We train on one NVIDIA Tesla K80 GPU with 12GB of RAM, with a batch size of 8. Training is done for $80,000$ iterations, taking 9 hours to complete.

4.3. Evaluation

In this section, we give qualitative results of our models as well as quantitative metric evaluations.

4.3.1 Quantitative Evaluation

From images with a ground truth depth of y_t and a predicted depth of y_p, we use 6 different metrics to quantify the per-

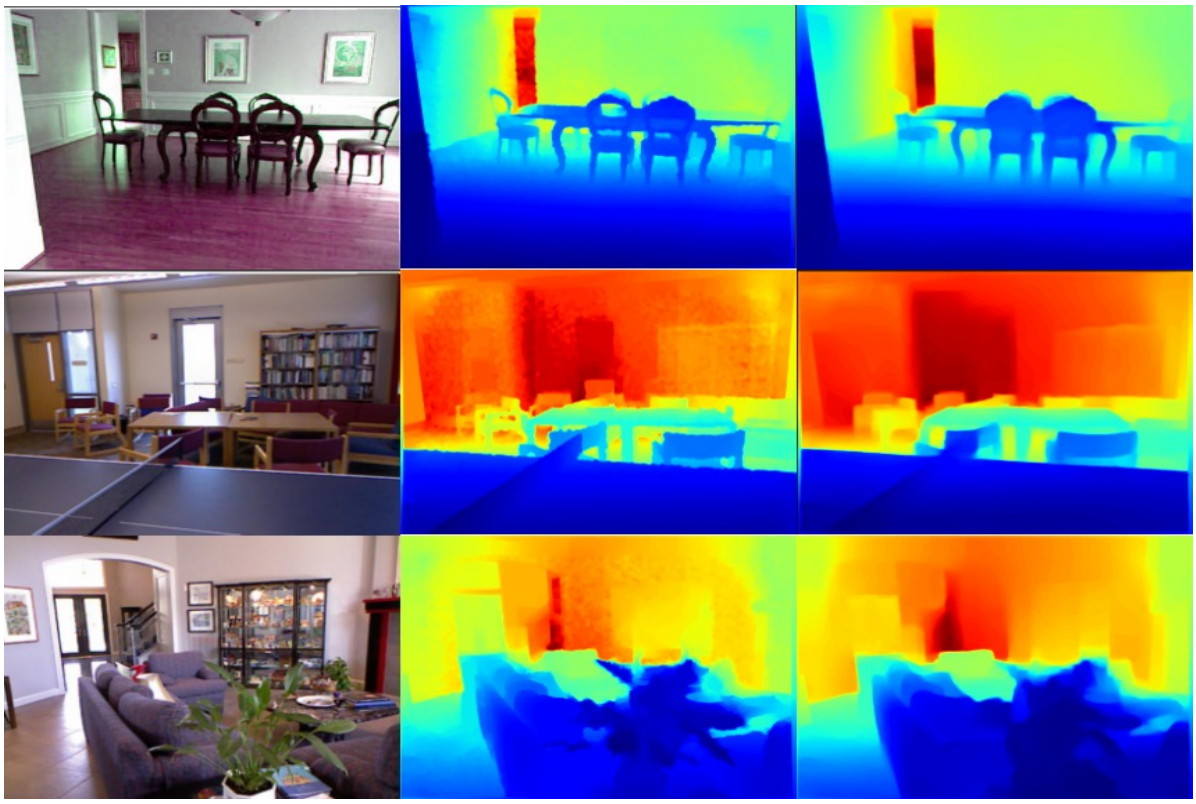

Figure 5. **Qualitative Results**. From left to right: input RGB image, ground truth depth map, estimated depth map.

Method	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	$rel \downarrow$	$rms \downarrow$	$log_{10} \downarrow$
Eigen *et al*. [5]	0.769	0.950	0.988	0.158	0.641	-
Laina *et al*. [22]	0.811	0.953	0.988	0.127	0.573	0.055
MS-CRF [41]	0.811	0.954	0.987	0.121	0.586	0.052
Hao *et al*. [11]	0.841	0.966	0.991	0.127	0.555	0.053
Fu *et al*. [6]	0.828	0.965	0.992	**0.115**	0.509	**0.051**
Alhashim *et al*. [1]	0.846	0.974	0.994	0.123	**0.465**	0.053
Ours	**0.852**	**0.976**	**0.995**	0.122	0.500	0.053

Table 1. Comparison of quantitative results on the NYU Depth v2 dataset. The reported numbers are from the corresponding original papers.

formance of our network:

- Percentage of pixels with relative error $t = max(\frac{y_t}{y_p}, \frac{y_p}{y_t})$ less than δ, with $\delta = 1.25^1, 1.25^2, 1.25^3$
- Absolute relative difference: $\frac{|y_t - y_p|}{y_t}$
- Root mean squared error: $\sqrt{\frac{1}{n}(y_t - t_p)^2}$
- log_{10} error: $|log_{10}(y_t) - log_{10}(y_p)|$

Table 1 compares our results with the previously proposed methods. It can be seen that our model is comparable and that quantitatively better than the previously proposed methods.

4.3.2 Qualitative Results

Figure 5 gives the qualitative results of our model on 3 test images. As seen in the samples displayed, the model has succeeded in capturing most of the object boundaries, and has estimated the depth map correctly, to a certain extent. In the case of the third image, it can be seen that the edges of the plant have not been ignored and that the boundaries are captured well.

Loss function	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	$rel \downarrow$	$rms \downarrow$	$log_{10} \downarrow$
$L_{depth} + L_{grad}$	0.823	0.966	0.989	0.129	0.507	0.054
$L_{depth} + L_{SSIM}$	0.818	0.958	0.988	0.135	0.510	0.056
$L_{depth} + L_{grad} + L_{SSIM}$	0.846	0.974	0.994	0.123	**0.465**	0.053
$L_{depth} + L_{grad} + L_{SSIM} + L_{perceptual}$	**0.852**	**0.976**	**0.995**	**0.122**	0.500	**0.053**

Table 2. Loss function comparison

4.4. Ablation studies

4.4.1 Comparison of Different Combinations of Loss Functions

We experimented with different combinations of the three loss functions to find out how much influence each loss function has. Our results are given in table 2. It can be sent that only either of L_{grad} or L_{SSIM} along with L_{depth} does not yield the best results. Only the combination of three loss functions along with the perceptual loss yields the best results on all the evaluation metrics.

5. Conclusion

Depth prediction from monocular images plays an essential role in many practical applications and is challenging because of the inherent ambiguity. We combine the benefits of using low-level and high-level representations for better estimation of depth maps, and show that our model achieves considerable gains compared to previous works on the NYU Depth v2 Dataset, both quantitatively and qualitatively. Our framework can be further extended by using careful combinations of low-level and high-level optimization objectives and training with deeper networks. In the future, we would like to focus on moving towards self-supervised learning of these systems.

References

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

[2] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2017.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] 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.

[5] 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.

[6] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[7] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[8] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[9] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[10] 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.

[11] Z. Hao, Y. Li, S. You, and F. Lu. Detail preserving depth estimation from a single image using attention guided networks. In *2018 International Conference on 3D Vision (3DV)*, pages 304–313. IEEE, 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

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

[15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

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

[17] F. Khan, S. Salahuddin, and H. Javidnia. Deep learning-based monocular depth estimation methods—a state-of-the-art review. *Sensors*, 20(8):2272, 2020.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[20] 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.

[21] 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.

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

[23] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694. 2004.

[24] 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.

[25] A. Liaw, M. Wiener, et al. Classification and regression by randomforest.

[26] 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, 2010.

[27] 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.

[28] 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, 2015.

[29] 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.

[30] 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.

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

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[33] G. Qian and R. Chellappa. Structure from motion using sequential monte carlo methods. *International Journal of Computer Vision*, 59(1):5–31, 2004.

[34] 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.

[35] 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.

[36] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[37] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, 2003.

[38] 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.

[39] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 103–110. IEEE, 2012.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[41] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017.

[42] 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.