Monocular Depth Estimation Using Multi Scale Neural Network And Feature Fusion

Abhinav Sagar[*]
Vellore Institute of Technology
Vellore, Tamil Nadu, India
abhinavsagar4@gmail.com

Abstract

Depth estimation from monocular images is a challenging problem in computer vision. In this paper, we tackle this problem using a novel network architecture using multi scale feature fusion. Our network uses two different blocks, first which uses different filter sizes for convolution and merges all the individual feature maps. The second block uses dilated convolutions in place of fully connected layers thus reducing computations and increasing the receptive field. We present a new loss function for training the network which uses a depth regression term, SSIM loss term and a multinomial logistic loss term combined. We train and test our network on Make 3D dataset, NYU Depth V2 dataset and Kitti dataset using standard evaluation metrics for depth estimation comprised of RMSE loss and SILog loss. Our network outperforms previous state of the art methods with lesser parameters.

1 Introduction

Deep learning powered by neural networks has been successful in a range of problems in computer vision. Making autonomous Driving a reality requires solving the perception problem. There are a lot of sub-tasks involved like object detection, instance segmentation, depth estimation, scene understanding etc. Neural Networks tries to mimic the human brain by learning from the data without being explicitly programmed (Goodfellow et al., 2016). In this work, we tackle the depth estimation problem especially in the context of autonomous driving.

Depth estimation is an important but complex problem in computer vision. This requires learning a function which calculates the depth map from the input image. Humans have this ability naturally as their brain is able to understand the scene by making use of information from lighting, shading, perspective vision and presence of objects at various sizes (Godard et al., 2017). For humans it is pretty easy to infer the distance at which objects are present from a single image, however the task is quite challenging for a computer (Laina et al., 2016).

Stereo cameras have been traditionally used in Simultaneous Localization and Mapping (SLAM) based systems which has access to depth maps. However using monocular camera offers benefits like low power consumption, light weight and cheap. Hence this approach seems like a better alternative. In the literature, depth estimation has been mostly tackled using stereo cameras (Rajagopalan et al., 2004). Depth estimation from a single image or monocular camera has been lately tackled using a range of convolutional network architectures (Eigen et al., 2014), (Laina et al., 2016) and (Liu et al., 2015b). The problem have been cast as a regression one which uses a Mean Square Error(MSE) in log space as the loss function.

[*] Website of author - https://abhinavsagar.github.io/

Preprint. Under review.

2 Related Work

Early works on depth estimation were mostly based on stereo images using geometry based algorithms. Supervised learning was used to learn depth from monocular cues in 2D images (Saxena et al., 2008). A lot of work has been done using handcrafted techniques for feature extraction (Rajagopalan et al., 2004). However these methods can only capture local information. Depth estimation has been tackled using image classification networks as feature extractors (Eigen et al., 2014) and (Garg et al., 2016). Spatial pyramidal pooling is used for reducing the spatial resolution of feature maps.

Deep networks based on VGG and ResNet as feature extractors have been able to beat the previous techniques (Garg et al., 2016) and (Eigen et al., 2014). A multi scale network was used the low spatial resolution depth map to high spatial resolution (Eigen et al., 2014). This helped reduce the recurring pooling operation which helped decrease the spatial resolution of feature maps. The network was divided into 2 parts: coarse network which predicts depth of the scene at a global level and fine network which uses local information to refine the depth.

Multi layer deconvolutional network has been used which uses high resolution feature maps (Laina et al., 2016). Residual upsampling modules were used along with a Resnet based feature extractor. (Jiao et al., 2018) proposed a multi task convolutional neural network which uses lateral sharing approach between the individual networks. Also a new loss function was used to tackle the imbalanced depth distribution. (Fu et al., 2018) used a multi scale approach by discretizing the weights thus better taking uncertainty at various depths into account. VGG and Resnet based feature extractors were benchmarked along with atrous-spatial-pyramid-pooling approach to enhance the receptive field of the network.

A skip connection based approach was used to fuse low spatial resolution depth map at deeper layers to high spatial resolution depth maps at previous layers (Xie et al., 2016). To reduce the computational burden multi scale network has been used for extracting the features (Liu et al., 2015a) and skip connections (Xie et al., 2016). Also work has been done recently using unsupervised learning or semi supervised learning (Garg et al., 2016). Reconstruction losses are used for estimating the disparity map by using information from both the left and right view.

Our main contributions can be summarized as:

• We propose a novel end to end trainable network for monocular depth estimation.

• We present the network architecture, training details, loss functions and ablation studies.

• Our network outperforms previous state of the art networks on Make3D Range Image Data, NYU Depth Dataset V2 and Kitti dataset.

3 Method

3.1 Dataset

The following datasets have been used for training and testing our network:

1. **Make3D Range Image Data** - This dataset was one of the first proposed to infer the depth map from a single image. It has the range data corresponding to each image. Examples from the dataset include outdoor scenes, indoor scenes and synthetic objects (Saxena et al., 2008).

2. **NYU Depth Dataset V2** - This dataset is made up of video sequences from a variety of indoor scenes which have been recorded using both RGB and depth cameras. It has 1449 densely labeled pairs of aligned RGB and depth images. The objects present in the dataset have been individually labelled with a class id (Silberman et al., 2012). The official split consists of 249 training and 215 testing scenes. The images are of resolution is 480×640.

3. **Kitti dataset** - This large dataset has over 93 thousand depth maps with corresponding raw Lidar scans and RGB images. This has been the benchmark dataset for depth estimation using a single image for autonomous driving (Geiger et al., 2013). For benchmarking, Eigen split was done by (Eigen et al., 2014). The training set consists of approximately 22 600 frames from a total of 28 different scenes and the validation set contains of 888 frames. The test set contains 697 frames from 28 different scenes. The images are of resolution 376×1242.

3.2 Data Augmentation

Data Augmentation is the process in which the dataset size is manually increased by performing operations on the individual samples of the dataset. This leads to better generalization ability thus avoiding overfitting of the network. Data Augmentation has been used successfully for depth estimation (Alhashim and Wonka, 2018) and (Li et al., 2018).

The training data was increased using data augmentation:

- **Scale**: Colour images are scaled by a random number $s \in [1, 1.5]$.
- **Rotation**: The colour and depth images are both rotated with a random degree $r \in [-5, 5]$.
- **Colour Jitter**: The brightness, contrast, and saturation of color images are each scaled by $k \in [0.6, 1.4]$.
- **Colour Normalization**: RGB images are normalized through mean subtraction and division by standard deviation.
- **Flips**: Colour and depth images are both horizontally flipped with a 50% chance.

Also nearest neighbor interpolation was used.

3.3 Network Architecture

The task is to learn a direct mapping from a colour image to the corresponding depth map. Our network fuses multi scale depth features which is important for depth estimation. Our network removed all the fully connected layers which adds a lot of computational overhead. Although fully connected layers are important in inferring long range contextual information but still it is not required. Instead we use dilated convolutions which enlarges the receptive field without increasing the number of parameters involved.

The network takes as input an image and uses a pre trained ResNet backbone for feature extraction. Convolutions are used at multiple scales with combinations of 1×1 convolution, 3×3 convolution, 5×5 convolution and 7×7 convolution. Instance-wise concat operation is performed to merge the feature maps. This multi scale block is repeated for 4 times. The receptive field of our network increases considerably due to this operation and is able to capture global contextual information in addition to the local information.

The fused features is propagated to another multi scale block. This block is made up of plain convolutional layer and dilated convolutions with dilation rates of 2 and 4 respectively. This block is also repeated for 4 times and instance-wise concat operation is used for merging the feature maps. The network architecture used in this work is presented in Figure 1:

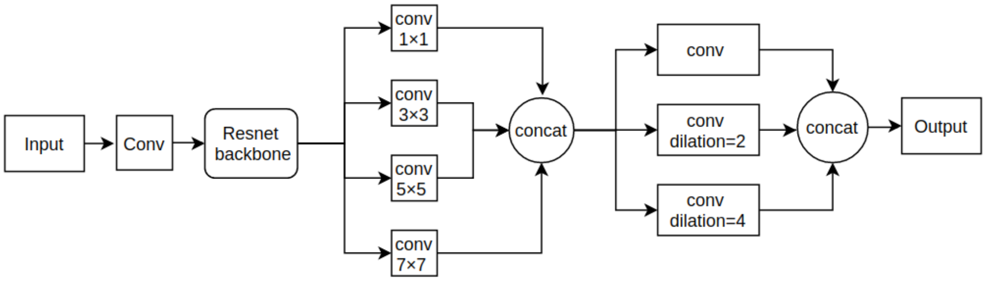

Figure 1: Network architecture used in this work

3.4 Multi Scale Fusion

The high level neurons have a larger receptive field in convolutional neural network. Although low level neurons has a smaller receptive field, it contains more detailed information. Hence for better results, we combined feature maps at different scales. We concatenated the high level and intermediate level feature maps using a concat operator. Skip connections also helps the multi scale fusion operation by creating an additional pathway for flow of information.

3.5 Loss Functions

The standard loss function for training depth estimation network is a regression loss which is the difference between the ground-truth depth map y and the prediction of the network $\hat{y}$ (Eigen et al., 2014). Loss functions are very important for avoiding training instability as well as achieving better results. A lot of loss functions have been proposed in literature for depth estimation (Fu et al., 2018) and (Laina et al., 2016). We design our loss function by minimizing the reconstruction depth and penalizing the high frequency details. Our loss function is made up of 3 terms: depth term, Structural Similarity Index Measure (SSIM term) and a multinomial logistic loss term. The depth term is a L1 loss defined on the depth values as shown in Equation 1:

$$L_{depth}(y, \hat{y}) = \frac{1}{n} \sum_{p}^{n} |y_p - \hat{y}_p| \tag{1}$$

SSIM metric is frequently used for measuring the image quality and similarity between two images. (Godard et al., 2017) first used this while training depth estimation network. The upper bound of SSIM metric is 1, hence the loss term can be defined as in Equation 2:

$$L_{SSIM}(y, \hat{y}) = \frac{1 - SSIM(y, \hat{y})}{2} \tag{2}$$

We cast depth estimation task as a kind of image classification one. Multinomial logistic loss term is defined as in Equation 3:

$$L(\theta) = -\left[\sum_{i=1}^{N} \sum_{k=1}^{K} \left\{ y^{(i)} \right\} \log \frac{\exp\left(\theta^{(k)T} y^{(i)}\right)}{\sum_{i=1}^{K} \exp\left(\theta^{(i)T} y^{(i)}\right)} \right] \tag{3}$$

where N is the number of training samples, $exp(\theta(k)Tx(i))$ is the probability of label k of sample i, and k is the ground truth label.

The three terms can be combined together to yield the complete loss function which is used to train the network as in Equation 4:

$$L(y, \hat{y}) = \alpha L_{\text{depth}}(y, \hat{y}) + \beta L_{SSIM}(y, \hat{y}) + \gamma L(\theta) \tag{4}$$

Where α, β and γ are constants.

3.6 Evaluation Metrics

For evaluating depth predicting networks, the error metrics used by (Eigen et al., 2014) are commonly used. Let y_i denotes the prediction value of pixel, $y_i^\star$ the ground truth value of pixel i and T denotes the total number of pixels which are valid. The error metrics are defined in the form of Root Mean Square Error (RMSE) and RMSE(log) as defined in Equation 5 and Equation 6 respectively:

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{i} \|y_i - y_i^*\|^2} \tag{5}$$

$$\text{RMSE(log)} = \sqrt{\frac{1}{T} \sum_{i} \|\log(y_i) - \log(y_i^*)\|^2} \tag{6}$$

The SILog error metric was defined by (Eigen et al., 2014) to measure the relationship between points in the scene irrespective of the absolute global scale which is shown in Equation 8. The value of d_i can be computed using Equation 7:

$$d_i = \log\left(y_i\right) - \log\left(y_i^*\right) \tag{7}$$

$$\text{SILog} = \frac{1}{T}\sum_i d_i^2 - \frac{1}{T^2}\left(\sum_i d_i\right)^2 \tag{8}$$

The Averaged Relative Error (ARE) and Squared Relative Error (SRE) metrics is defined in Equation 9 and Equation 10 respectively:

$$\text{ARE} = \sqrt{\frac{1}{T}\sum_i \frac{\left|y_i - y_i^*\right|}{y_i^*}} \tag{9}$$

$$\text{SRE} = \sqrt{\frac{1}{T}\sum_i \frac{\left\|y_i - y_i^*\right\|^2}{y_i^*}} \tag{10}$$

Accuracy with a threshold metric, divides the error ratios into intervals determined by the threshold value λ. The accuracy is defined as the number of pixels with a error ratio less than the threshold divided by the total number of pixels present. This error metric is shown in Equation 11:

$$\frac{1}{T}\sum_i \left(\max\left(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}\right) = \delta < \text{thr}\right), \text{thr} = \left[\lambda, \lambda^2, \lambda^3\right] \tag{11}$$

The value of λ is taken as 1.25.

For quantitative evaluation, error metrics Mean relative error and Mean $\log_{10}$ error is defined in Equation 12 and Equation 13 respectively:

$$\text{Rel } = \frac{1}{|T|}\sum_{d\in T} |\hat{d} - d|/d \tag{12}$$

$$\log_{10} = \frac{1}{T\,|}\sum_{d\in T}\left|\log_{10}\hat{d} - \log_{10} d\right| \tag{13}$$

Where d represents the ground truth depth, $\hat{d}$ represents the estimated depth, and T denotes the set of all points in the images.

3.7 Implementation Details

State of the art ResNet backbone was used as feature extractor which is trained on the Imagenet dataset. In all the experiments, ADAM optimizer was used with a learning rate value of 0.0001, parameter values momentum as 0.9, weight decay value of 0.0004 and batch size is set to 8. The network was trained using Stochastic Gradient Decent (SGD) for 500K iterations for NYU Depth v2 dataset, 100K iterations for Make3D dataset and 300K iterations for Kitti dataset.

4 Results

The comparison of our network with previous state of the art methods on NYU Depth v2 dataset is shown in Table 1:

The model predictions compared along with ground truth depth map on NYU v2 dataset is shown in Figure 2:

Table 1: Performance on NYU Depth v2 dataset. 2nd, 3rd and 4th column: higher is better; 5th, 6th and 7th column: lower is better.

Method	δ_1	δ_2	δ_3	rel	log_{10}	rms
(Saxena et al., 2008)	0.447	0.745	0.897	0.349	-	1.214
(Karsch et al., 2014)	-	-	-	0.35	0.131	1.2
(Liu et al., 2010)	-	-	-	0.335	0.127	1.06
(Li et al., 2018)	0.621	0.886	0.968	0.232	0.094	0.821
(Wang et al., 2015)	0.605	0.890	0.970	0.220	-	0.824
(Roy and Todorovic, 2016)	-	-	-	0.187	-	0.744
(Liu et al., 2010)	0.650	0.906	0.976	0.213	0.087	0.759
(Eigen et al., 2014)	0.769	0.950	0.988	0.158	-	0.641
(Laina et al., 2016)	0.629	0.889	0.971	0.194	0.083	0.790
(Xu et al., 2017)	0.811	0.954	0.987	0.121	0.052	0.586
(Fu et al., 2018)	0.828	0.965	0.992	0.115	0.051	0.509
Ours	0.823	0.962	0.994	0.101	0.054	0.456

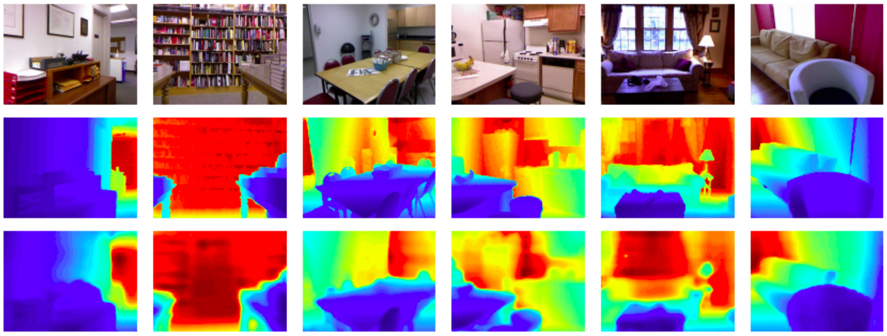

Figure 2: Qualitative comparison of the estimated depth map on the NYU v2 dataset. Color indicates depth (red is far, blue is close). First row: RGB image, second row: Ground Truth depth map, third row: Results of our proposed method

The comparison of our network with previous state of the art methods on Kitti dataset is shown in Table 2:

Table 2: Performance on KITTI dataset. All the methods are evaluated on the test split by (Eigen et al., 2014). 3rd, 4th and 5th column: higher is better; 6th, 7th, 8th and 9th column: lower is better.

Method	cap	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Sq Rel	RMSE	$RMSE_{\log}$
(Saxena et al., 2008)	0 - 80 m	0.601	0.820	0.926	0.280	3.012	8.734	0.361
(Eigen et al., 2014)	0 - 80 m	0.692	0.899	0.967	0.190	1.515	7.156	0.270
(Liu et al., 2010)	0 - 80 m	0.647	0.882	0.961	0.217	1.841	6.986	0.289
(Godard et al., 2017)	0 - 80 m	0.861	0.949	0.976	0.114	0.898	4.935	0.206
(Kuznietsov et al., 2017)	0 - 80 m	0.862	0.960	0.986	0.113	0.741	4.621	0.189
(Fu et al., 2018)	0 - 80 m	0.915	0.980	0.993	0.081	0.376	3.056	0.132
(Fu et al., 2018)	0 - 80 m	0.932	0.984	0.994	0.072	0.307	2.727	0.120
(Garg et al., 2016)	0 - 50 m	0.740	0.904	0.962	0.169	1.080	5.104	0.273
(Godard et al., 2017)	0 - 50 m	0.873	0.954	0.979	0.108	0.657	3.729	0.194
(Kuznietsov et al., 2017)	0 - 50 m	0.875	0.964	0.988	0.108	0.595	3.518	0.179
(Fu et al., 2018)	0 - 50 m	0.920	0.982	0.994	0.079	0.324	2.517	0.128
(Fu et al., 2018)	0 - 50 m	0.936	0.985	0.995	0.071	0.268	2.271	0.116
Ours	0 - 50 m	0.945	0.987	0.997	0.066	0.268	2.042	0.110

The model predictions compared along with ground truth depth map on test image number 1 on Kitti dataset is shown in Figure 3:

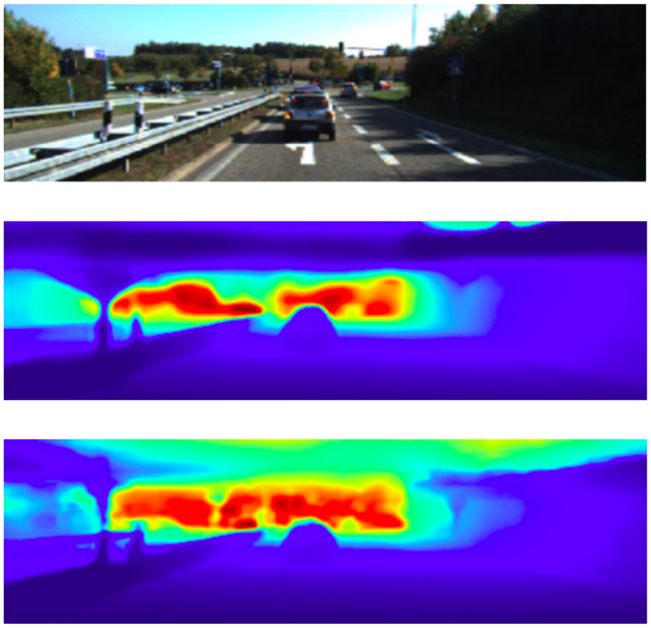

Figure 3: The output predictions of our network on test image number 1. First row: input image, second row: ground truth depth map, third row: model prediction depth map. Color indicates depth (red is far, blue is close).

The model predictions compared along with ground truth depth map on test image number 5 on Kitti dataset is shown in Figure 4:

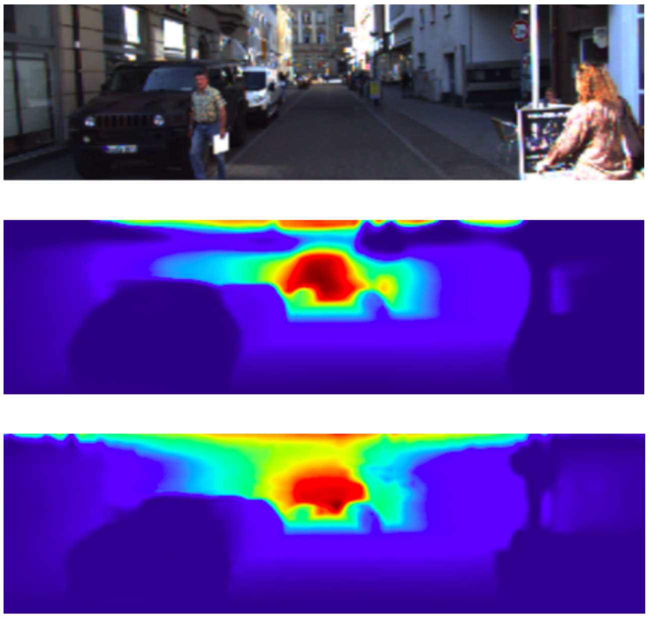

Figure 4: The output predictions of our network on test image number 5. First row: input image, second row: ground truth depth map, third row: model prediction depth map. Color indicates depth (red is far, blue is close). Our network fails to detect person in front of the car as well as the person in the bottom left corner

The comparison of our network with previous state of the art methods on Make3D dataset is shown in Table 3:

Table 3: Performance on Make3D dataset. 2nd, 3rd and 4th column represents C1 error; 5th, 6th and 7th column represents C2 error. Both lower C1 and C2 error is better.

Method	rel	$\log_{10}$	rms	rel	$\log_{10}$	rms
(Saxena et al., 2008)	-	-	-	0.370	0.187	-
(Liu et al., 2010)	-	-	-	0.379	0.148	-
(Karsch et al., 2014)	0.355	0.127	9.20	0.361	0.148	15.10
(Liu et al., 2014)	0.335	0.137	9.49	0.338	0.134	12.60
(Liu et al., 2015a)	0.278	0.092	7.12	0.279	0.102	10.27
(Liu et al., 2015b)	0.287	0.109	7.36	0.287	0.122	14.09
(Roy and Todorovic, 2016)	-	-	-	0.260	0.119	12.40
(Laina et al., 2016)	0.176	0.072	4.46	-	-	-
(Xie et al., 2016)	1.000	2.527	19.11	-	-	-
(Godard et al., 2017)	0.443	0.156	11.513	-	-	-
(Kuznietsov et al., 2017)	0.421	0.190	8.24	-	-	-
(Xu et al., 2018)	0.184	0.065	4.38	0.198	-	8.56
(Fu et al., 2018)	0.236	0.082	7.02	0.238	0.087	10.01
(Fu et al., 2018)	0.157	0.062	3.97	0.162	0.067	7.32
Ours	0.139	0.060	2.64	0.144	0.059	6.36

4.1 Ablation Studies

We perform ablation studies to analyze the performance of our network. The comparative performance using dilation and concat layers is shown in Table 4:

Table 4: Ablation Study of our CNN architecture design on Kitti dataset. 2nd, 3rd and 4th column: higher is better; 5th, 6th and 7th column: lower is better.

Method	$\delta < 1.25$ (%)	$\delta < 1.25^2$ (%)	$\delta < 1.25^3$ (%)	Rel	$\log_{10}$	rms
no dilation no cocat	76.06	94.29	97.56	0.156	0.056	0.536
no dilation yes concat	79.24	96.2	97.80	0.145	0.056	0.520
yes dilation no concat	81.52	95.43	98.63	0.132	0.060	0.533
yes dilation yes concat	83.10	95.3	98.70	0.134	0.051	0.515

5 Conclusions

In this paper, we proposed a novel network architecture for monocular depth estimation using multi scale feature fusion. We present the network architecture, training details, loss functions and the evaluation metrics used. We used Make 3D dataset, NYU Depth V2 dataset and Kitti dataset for training and testing our network. Our network not only beats the previous state of the art methods on monocular depth estimation but also has lesser parameters thus making it feasible in a real time setting.

Acknowledgments

We would like to thank Nvidia for providing the GPUs for this work.

References

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

V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017a.

L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017b.

L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

X. Cheng, P. Wang, and R. Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018.

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.

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.

R. Garg, V. K. Bg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

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.

C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3838, 2019.

I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

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.

J. Jiao, Y. Cao, Y. Song, and R. Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European conference on computer vision (ECCV)*, pages 53–69, 2018.

K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014.

Y. Kuznietsov, J. Stuckler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6647–6655, 2017.

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.

B. Li, Y. Dai, and M. He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition*, 83:328–339, 2018.

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

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, 2015a.

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, 2015b.

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.

R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018.

F. Mal and S. Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

A. Rajagopalan, S. Chaudhuri, and U. Mudenagudi. Depth estimation and image restoration using defocused stereo pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11): 1521–1525, 2004.

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.

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, 2008.

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017.

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.

J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016.

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.

D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2018.

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2614–2622, 2015.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.