

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

## QuMaDe: Quick Foreground Mask and Monocular Depth Data Generation



Anonymous CVPR 2021 submission

Paper ID 68

### Abstract

Segmentation of the desired object along with depth estimation is useful in various applications like robotics and autonomous navigation. Any deep learning workflow to estimate monocular depth and segment the desired foreground object in a scene require significant training data. The data generation process usually involve expensive hardware like RGB-D sensors, Laser Scanners or significant manual involvement. Moreover, for every specific foreground object, the data collection process need to repeat. This paper presents a novel way to utilize only a small number of readily available png images with transparency for the foreground object, and representative background images from the internet and combine them to generate a large dataset for deep learning, utilizing current state of the art monocular depth estimation techniques. To illustrate the effectiveness of the data generation approach, this paper presents a baseline model for depth and foreground mask estimation for detecting cattle on road using the generated data from the proposed approach. The baseline model exhibits strong generalization to real scenarios. The generated dataset is available for public use.

### 1. Introduction

Depth estimation and segmentation of desired objects in the scene are often used together in many vision tasks [1], like autonomous navigation of agents, augmented reality,

self driving cars and other robotics applications. In all these applications, identification of desired objects precisely in the scene and its depth estimate from the camera are crucial for safe and effective navigation. Modern RGB-D sensors like OAK-D<sup>1</sup> are capable of simultaneously running advanced neural networks while providing depth from two stereo cameras and color information from a single 4K camera in the center. Deep learning based techniques using convolution neural nets have effective solution in both depth estimation and semantic segmentation. In general for high accuracy outcomes, a deep learning network is dependent on large training dataset availability. To gather such data itself incurs high cost and time. For specific applications requiring several foreground objects against variety of backgrounds become even more challenging in terms of simulating those scenarios. Synthetic datasets using Virtual Reality have been proposed to that end [2].

Recent research indicates effective use of readily available images on the internet to curate training data [3]. This paper introduces *QuMaDe*, a way to curate custom dataset containing hundreds of thousands to millions of images by multiplexing desired foreground objects over representative background scenes, while also generating corresponding depth and foreground mask images. This significantly reduces the cost and time overheads. The authors also experiment by creating baseline models for several application contexts and show that the generated data successfully gen-

<sup>1</sup>OpenCV AI Kit: <https://opencv.org/introducing-oak-spatial-ai-powered-by-opencv/>

108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161

eralizes to detect relevant objects in real scenes. Multiplexing, combined with random cropping, scaling and translation, makes the data generation fast and effective. With only 100 pairs of background and foreground images, the authors generate 0.4 million triplets of foreground over background, monocular depth and foreground mask images by effectively leveraging existing SOTA models for depth estimation.

The main contributions of this paper are the following:

1. A novel effort to mix and match foreground and background images reducing the need for complex scene generation for data curation.
2. Curate large dataset to effectively train models to detect depth and mask for specific foreground objects over any target background, from a limited input of ready available internet images.
3. Combine image, depth map and foreground mask in a single dataset using current SOTA models for depth estimation.
4. To release curated dataset and the trained models making them publicly available. Researchers can use this single dataset to do segmentation, train models to predict depth, or to predict both depth and mask.

The title figure represents an example from the generated dataset to help detect cattle on road, a common happening on the Indian roads that leads to several accidents each year involving loss of life and property. The generated dataset, *MODES*<sup>2</sup> and the trained model are publicly available

## 2. Related Work

A depth image is an image channel in which each pixel relates to a distance between the image plane and the corresponding object in the RGB image. Monocular depth gives information about depth and distance and Monocular Depth Estimation is the task of estimating scene depth using a single image[4]. Image Segmentation is the process of partitioning an image into multiple segments and it can be used for locating objects and boundaries [5]. RGBD image is a combination of a RGB image and its corresponding depth image[6].

Depth information is integral to many problems in robotics including mapping, localization and obstacle avoidance for terrestrial and aerial vehicles, autonomous navigation, and in computer vision, including augmented and virtual reality[7]. RGBD datasets usually collected using depth sensors, monocular cameras and LiDAR scanners are expensive and data collection is a time consuming job. The well known datasets for monocular 3D object detection

are Context-Aware MixEd ReAlity (CAMERA), Objectron, Kitty3D, Cityscape3D, Synthia, etc. and these datasets have limitations like indoor only images, small number of training examples and sparse sampling. Some of the frequently used RGBD datasets are the Kitti dataset [8], the Synthia dataset [2], Make3D dataset [9] and NYU dataset [10]

The dataset Kitti [8] is collected using a vehicle equipped with a sparse Velodyne VLP-64 LiDAR scanner and RGB cameras, and features street scenes in and around the German city of Karlsruhe. The Primary application of this dataset involves perception tasks in the context of self-driving. Synthia [2] is a street scene dataset with depth maps of synthetic data, requiring domain adaptation to apply to real world settings. Cityscapes [11] provides a dataset of street scenes, albeit with more diversity than KITTI. Sintel [12] is another synthetic dataset which mainly comprises of outdoors scenes.

Megadepth [3] is a large-scale dataset of outdoor images collected from internet, with depth maps reconstructed using structure-from-motion techniques, but this dataset lacks in ground truth depth and scale. The RedWeb [13] dataset provide depth maps generated from stereo images which are freely available in large-scale data platforms such as Flickr. The datasets MegaDepth and RedWeb can be easily computed with the existing MVS methods.

Make3D [9] provides RGB and depth information for outdoor scenes. The NYUV2 dataset [10] is widely used for monocular depth estimation in indoor environments. The data was collected with a Kinect RGBD camera, which provides sparse and noisy depth returns. These returns are generally in-painted and smoothed before they are used for monocular depth estimation tasks. As a result, while the dataset includes sufficient samples to train modern machine learning pipelines, the “ground-truth” depth does not necessarily correspond to true scene depth.

Most of the existing datasets consists of indoor images, or outdoor images of city streets. For every specific application, like detecting animals roaming on roads for self driving or assisted driving cars, or people inside a room for autonomous room cleaners etc. researchers need to curate specific dataset to train relevant deep learning models. This paper proposes a technique to come up with a custom dataset by using existing accurate depth predictor models, like High Quality Monocular Depth Estimation via Transfer Learning(nyu.h5) [14] making the task of curating dataset extremely simple and cost effective.

## 3. Method

The curated dataset must have following objectives:

1. It should always include the foreground object.
2. It should drive deep learning models that generalize well.

<sup>2</sup>Curated Dataset for Cattle on Road: *link hidden to maintain anonymity*



Figure 1. Scene and foreground object images

**Algorithm 1:** Generate Dataset([bgimages], [fgimages], k, b)

---

```

input : m Background Image paths, 2n Foreground Image paths, multiplexing factor k, batch size b must be
         multiple of k
output: Yield  $2kmn$  fg-bg, mask and depth images in batches of size b
for offline use it creates 3 folders with fg_bg, mask and depth each having  $2kmn$  images;
for  $bg \leftarrow 1$  to  $m$  do
    for  $fg \leftarrow 1$  to  $2n$  do
        for  $i \leftarrow 1$  to  $k$  do
             $croppedbg \leftarrow$  take maximal random crop of  $448 \times 448$  from  $bg$  without affecting the aspect ratio;
            randomly pick a center point  $(x, y)$  in range  $[0, 447]$ ;
            randomly pick a scale in range  $[0.3, 0.6]$  (ratio of area  $fg$  covers  $bg$ );
            create  $fg - bg$  image by resizing the  $fg$  to scale and place it on top of  $croppedbg$  centered at  $x, y$ 
            calculated;
            calculate binary mask from current placement of  $fg$  by thresholding the transparency channel. save
             $fg - bg$  image and mask add  $fg - bg$  image to a batch;
        if  $b$  new  $fg - bg$  images generated then
            run depth model on batch and save corresponding depth images.
    
```

---

- 248 3. It should provide accurate dense depth maps in line  
249 with SOTA models.
- 250 4. It should provide accurate foreground mask.
- 251 5. It should be able to generate the data online during  
252 training phase dynamically.

### 3.1. Data Acquisition

257 The first step to curate data is to determine a target application scenario and thus determine the foreground object(s) and the representative background context. At the same time the dataset must have sufficient variability to include majority of the types and views that the trained deep network model may see when deployed.

263 We propose to download or take RGB image of  $n$  foreground object(s) and  $m$  background images (we used  $n = m = 100$ ) balancing the types and views. For example, for MODES dataset, we chose several cow, bull and calf types, individual or in group, sitting, standing or walking, and from various angles. Similarly for background, we chose backgrounds of streets, storefronts, main roads, highways,

markets, railway tracks, landscapes, garbage piles etc. PNG images with transparency are readily available on the internet for almost any desired foreground object. Such images will easily allow to generate foreground mask from non-transparent pixels. If not, tools like GIMP [15], combined with deep learning foreground extractors<sup>3</sup> can help generate the required PNG foreground images. Figure 1 shows few of the sample scene and foreground images used for the creation of this dataset.

### 3.2. Multiplexing and Depth Generation

This step is to place each foreground object several times on to the background images generating an fg-bg image and the foreground mask corresponding to the foreground placement and scale. Depth is computed from the fg-bg image via the model proposed by Ibraheem Alhashim et al. in their paper titled “High Quality Monocular Depth Estimation via

<sup>3</sup><https://www.remove.bg/> uses a combination of Image based techniques and DNN to separate foreground from background



Figure 2. Three image sets resulted from the algorithm used. (top) A scene image on which a foreground object is positioned at random location with random scale, (middle) respective mask for the foreground, and (bottom) calculated depth by using a model.

Transfer Learning” [14]<sup>4</sup>. This model takes  $448 \times 448$  size images as input, hence we resize all background images to this size while maintaining their aspect ratio.

The data generation process is completely online and produces one batch of images for training a deep model. By randomly repeating one foreground object  $k$  times at varied locations and scales for each background image and repeating another  $k$  times with horizontally flipped version of the same foreground, one can generate  $2kmn$  fg-bg images. In addition a random crop from background image instead of a fixed initial crop from the source image can add to more variability in the input. For  $k = 20$  and  $n = m = 100$  this becomes 400,000 fg-bg images. Algorithm 1 describes the data generation process

## 4. Experimental Analysis

In this section, we provide a baseline for monocular depth estimation and foreground segmentation on the generated *MODES* dataset. Convolutional Neural Networks(CNN) are progressive in exploring structural features and spatial image formation. To come up with baseline, we started training simple CNN, Resnet, and Unet++. The state-of-the-art models for image segmentation are variants of U-Net and fully convolutional networks (FCN)[16]. Long skip connections are used to skip features from the contracting path to the expanding path in order to recover spatial information lost during downsampling [17]. Short skip connections can be used to build deep FCNs.

### 4.1. Model

By using both long and short skip connections we proposed a light weight model following U-Net architecture with two decoder networks meant for foreground mask prediction and depth prediction. The architecture of the model is shown in Figure 3. The total number of parameters of this model are 5,525,568 including both the decoders.

<sup>4</sup>source code for depth estimation model: <https://github.com/ialhashim/DenseDepth/blob/master/DenseDepth.ipynb>

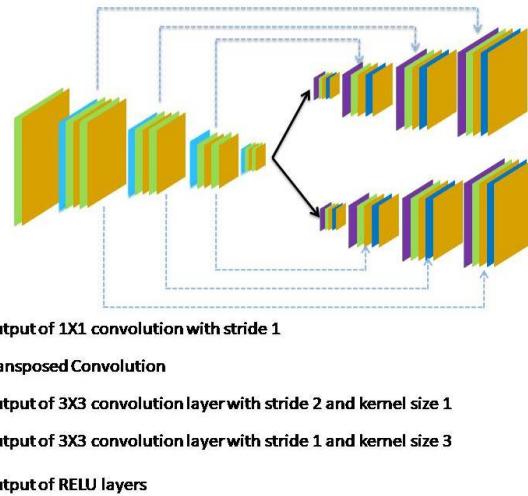


Figure 3. Network Architecture

The encoder part of the network is comprised of four downsampling units. Every downsampling unit compresses the input scene image with the help of a series of convolutional operations. In our implementation, the source image of size  $128 \times 128$ , changed into  $64 \times 64 \rightarrow 32 \times 32 \rightarrow 16 \times 16 \rightarrow 8 \times 8$ . This model has DepthDecoder and MaskDecoder and each of them is comprised of four up-sampling units. The compressed source image is expanded with the help of Atrous and Transposed convolution operations. The encoder outcome  $8 \times 8$  is expanded into  $16 \times 16 \rightarrow 32 \times 32 \rightarrow 64 \times 64 \rightarrow 128 \times 128$ . As shown in model architecture Figure 3 the outcome of encoder downsampling units were added to the outcomes of decoder up-sampling units.

We have trained this model on the entire *MODES* dataset from scratch with train-test-split of 70 – 30%. During training, the network is trained with the batch size of 64 for 10 epochs using SGD optimizer [18]. Every epoch took one hour of time on GPU because of the huge training data. We have used OneCycleLR scheduler [19] with a maximum



Figure 4. Foreground mask and depth inference on validation data

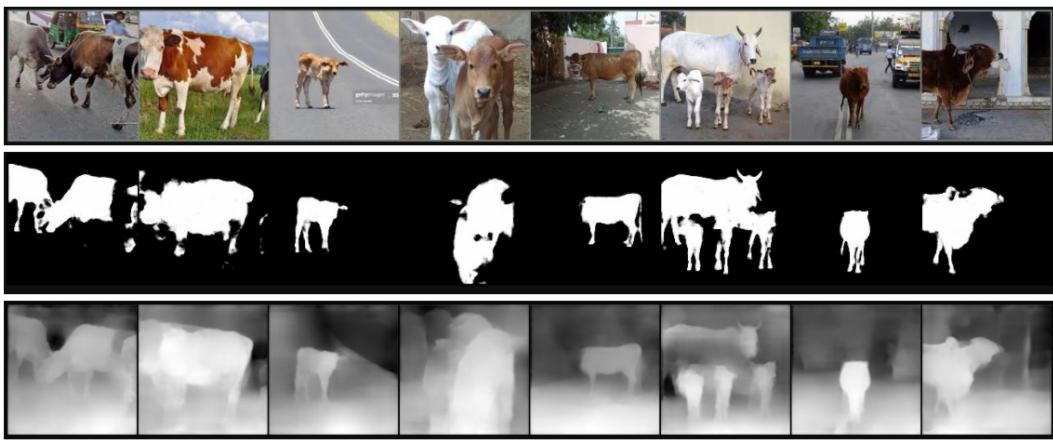


Figure 5. Foreground mask and depth inference on real life unseen data.

Learning rate of 0.1. This made the initial learning rate as 0.0099. The Deep Convolutional Neural Networks encoder is fed with an image ( $128 \times 128$ ) and the first decoder outputs a mask image and the second decoder outputs a depth image. To reduce overfitting [20], and achieve generalization this work employed the augmentation techniques of Random Rotation, Random Grayscale, Color Jitter, random horizontal flips and random channel swaps.

## 4.2. Loss function

Deciding a universal loss function is not possible for complex objectives like foreground segmentation and depth prediction. Based on the survey done by Shruti Jadon [21] we have picked L1 loss and SSIM(Structural Similarity Index) loss [22]. Their work also suggested to use a penalty term, which helps the network to focus towards hard-to-segment boundary regions. The Loss is calculated with the help of L1 and SSIM at both the decoders and employed regularization for weight penalty.

For training our network with two decoders, we defined the same loss function  $L$  for depth and mask prediction, be-

tween  $y$  and  $\hat{y}$  as the weighted sum of two loss function values.

$$L(y, \hat{y}) = \lambda L_{term1}(y, \hat{y}) + (1 - \lambda)L_{term2}(y, \hat{y}) \quad (1)$$

The first loss term  $L_{term1}(y, \hat{y})$  is the point-wise L1 loss defined on the predictions of Mask Decoder and Depth Decoder units of the network.

$$L_{term1}(y, \hat{y}) = \frac{1}{n} \sum_{x=1}^n |y_i - \hat{y}_i| \quad (2)$$

The second loss term  $L_{term2}(y, \hat{y})$  uses a commonly used metric for image reconstruction task i.e., SSIM. Many recent depth prediction CNNs employed this metric. The loss term is redefined as shown in equation as SSIM has an upper bound of one.

$$L_{term1}(y, \hat{y}) = \frac{1 - SSIM(y, \hat{y})}{2} \quad (3)$$

540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558

input image segmentation map segmentation overlay

road sidewalk building fence pole traffic light traffic sign vegetation terrain sky person rider car truck motorcycle bicycle

594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612

Images with wrong scale

Adaptive scaling did fix but still there were issues

594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612

Figure 6. Using Semantic Segmentation for adaptive placement and scaling. (Top) semantic segmentation. (Middle) Example of poor scaling and placement in proposed approach. (Bottom) Attempted correct placement and scaling

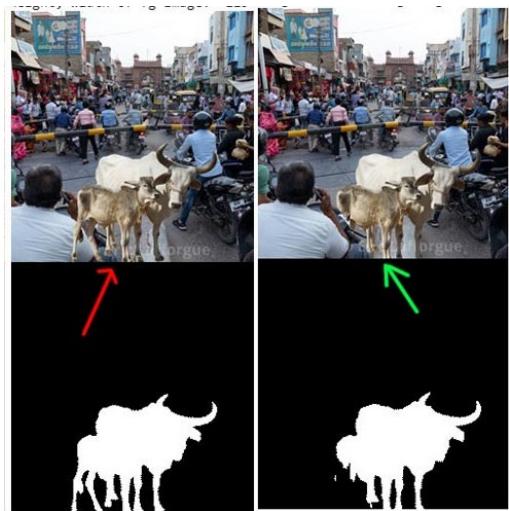


Figure 7. Restoring occlusion from semantic segmentation

Different weight parameters  $\lambda$  were tried and we have ended with a value  $\lambda = 0.84$ . The final loss function is as follows.

$$L(y, \hat{y}) = 0.84 * L_{term1}(y, \hat{y}) + 0.16 * L_{term2}(y, \hat{y}) \quad (4)$$

### 4.3. Results

The model is trained on the entire dataset and obtained significant accuracy and minimal loss. The outcome of the model on validation dataset is shown in Figure 4, and on the

unseen data is shown in Figure 5. The unseen data fed to the model is a real picture and not one curated as in *QuMaDe* where foreground is placed on background. From the obtained depths and foreground masks, it is very clear that the model generalized well. Few exceptions like the spots on cow and two calves not having very good detection indicate the non-prediction of such examples in the training set. However, it should be straightforward to introduce few more examples to make the application more robust.

## 5. Conclusion and Future Work

We presented a novel approach to generate a robust large dataset for depth and foreground mask estimation, where the dataset creation cost is kept low by intelligently multiplexing few foreground objects and high quality scene images collected from the Internet. We demonstrated the use of generated data to predict depth and mask for cattle on road. The light-weight baseline model presented, generalizes well to real life unseen data. We also demonstrate successful use of State-of-the-art models like DenseDepth to generate depth images for the curated foreground-background combined images. In addition we release the 400K multiplexed images, depth, and masks in *MODES* data for public use.

Due to random scaling and placement the fg-bg images are often not so realistic. Figure 6 middle show sample images with wrong placement and scale. Regardless of this, the trained model is generalizing well. However, we experimented with detection of ground and sky re-

648 regions from semantic segmentation by following DeepLab  
649 architecture[23]. That combined with the depth information  
650 of the background gives necessary cues as where to place  
651 the foreground image and at what scale. Figure 6 shows  
652 the outcomes. We further experimented with adding occlu-  
653 sions from semantic segmentation information where non  
654 ground/sky pixels that belong to regions starting below the  
655 top margin of the foreground location, are placed above the  
656 foreground. Figure 7 illustrates this. The effect of such data  
657 on training is yet to be explored. At the same time, the gen-  
658 erated images by such informed scaling and placement are  
659 also not free from artifacts. The orientation and unknown  
660 size of foreground object pose a challenge, leading to use  
661 of several experimental constants in the transformation pro-  
662 cess. We would like to formalize that and analyze its out-  
663 come on training as future work.  
664

## 665 6. Acknowledgements

666 *Removed to maintain anonymity.*

## 667 6. References

- 670 [1] Zhe Lin, Scott D Cohen, Peng Wang, SHEN Xiaohui, and  
671 Brian L Price. Joint depth estimation and semantic seg-  
672 mentation from a single image, July 10 2018. US Patent  
673 10,019,657. 1
- 674 [2] German Ros, Laura Sellart, Joanna Materzynska, David  
675 Vazquez, and Antonio M Lopez. The synthia dataset: A large  
676 collection of synthetic images for semantic segmentation of  
677 urban scenes. In *Proceedings of the IEEE conference on*  
678 *computer vision and pattern recognition*, pages 3234–3243,  
679 2016. 1, 2
- 680 [3] Zhengqi Li and Noah Snavely. Megadepth: Learning single-  
681 view depth prediction from internet photos. In *Proceedings*  
682 *of the IEEE Conference on Computer Vision and Pattern*  
683 *Recognition*, pages 2041–2050, 2018. 1, 2
- 684 [4] Abdullah Abuolaim and Michael S Brown. Defocus deblur-  
685 ring using dual-pixel data. In *European Conference on Com-  
686 puter Vision*, pages 111–126. Springer, 2020. 2
- 687 [5] Catalin Amza. A review on neural network-based image seg-  
688 mentation techniques. *De Montfort University, Mechanical*  
689 *and Manufacturing Engg., The Gateway Leicester, LE1 9BH,*  
690 *United Kingdom*, pages 1–23, 2012. 2
- 691 [6] Yinda Zhang and Thomas Funkhouser. Deep depth comple-  
692 tion of a single rgb-d image. In *Proceedings of the IEEE*  
693 *Conference on Computer Vision and Pattern Recognition*,  
694 pages 175–185, 2018. 2
- 695 [7] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler.  
696 Pose estimation for augmented reality: a hands-on survey.  
697 *IEEE transactions on visualization and computer graphics*,  
698 22(12):2633–2651, 2015. 2
- 699 [8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel  
700 Urtasun. Vision meets robotics: The kitti dataset. *The Inter-  
701 national Journal of Robotics Research*, 32(11):1231–1237,  
2013. 2

- [9] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d:  
Depth perception from a single still image. In *AAAI*, volume 3, pages 1571–1576, 2008. 2
- [10] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob  
Fergus. Indoor segmentation and support inference from  
rgbd images. In *European conference on computer vision*,  
pages 746–760. Springer, 2012. 2
- [11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo  
Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe  
Franke, Stefan Roth, and Bernt Schiele. The cityscapes  
dataset for semantic urban scene understanding. In *Pro-  
ceedings of the IEEE conference on computer vision and pattern*  
*recognition*, pages 3213–3223, 2016. 2
- [12] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer,  
Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A  
large dataset to train convolutional networks for disparity,  
optical flow, and scene flow estimation. In *Proceedings of*  
*the IEEE conference on computer vision and pattern recog-  
nition*, pages 4040–4048, 2016. 2
- [13] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao,  
Ruibo Li, and Zhenbo Luo. Monocular relative depth percep-  
tion with web stereo data supervision. In *Proceedings of the*  
*IEEE Conference on Computer Vision and Pattern Recog-  
nition*, pages 311–320, 2018. 2
- [14] Ibraheem Alhashim and Peter Wonka. High quality monocu-  
lar depth estimation via transfer learning. *arXiv preprint*  
*arXiv:1812.11941*, 2018. 2, 4
- [15] Ian M Howat, A Negrete, and Benjamin E Smith. The green-  
land ice mapping project (gimp) land classification and sur-  
face elevation data sets. *The Cryosphere*, 8(4):1509–1518,  
2014. 3
- [16] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand,  
Samuel Kadoury, and Chris Pal. The importance of skip con-  
nections in biomedical image segmentation. In *Deep learn-  
ing and data labeling for medical applications*, pages 179–  
187. Springer, 2016. 4
- [17] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima  
Tajbakhsh, and Jianming Liang. Unet++: Redesigning skip  
connections to exploit multiscale features in image segmen-  
tation. *IEEE transactions on medical imaging*, 39(6):1856–  
1867, 2019. 4
- [18] Léon Bottou. Large-scale machine learning with stochastic  
gradient descent. In *Proceedings of COMPSTAT'2010*, pages  
177–186. Springer, 2010. 4
- [19] Leslie N Smith. A disciplined approach to neural network  
hyper-parameters: Part 1—learning rate, batch size, momen-  
tum, and weight decay. *arXiv preprint arXiv:1803.09820*,  
2018. 4
- [20] Luis Perez and Jason Wang. The effectiveness of data aug-  
mentation in image classification using deep learning. *arXiv*  
*preprint arXiv:1712.04621*, 2017. 5
- [21] Shruti Jadon. A survey of loss functions for semantic seg-  
mentation. In *2020 IEEE Conference on Computational*  
*Intelligence in Bioinformatics and Computational Biology*  
*(CIBCB)*, pages 1–7. IEEE, 2020. 5

756	[22] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss	810
757	functions for neural networks for image processing. <i>arXiv preprint arXiv:1511.08861</i> , 2015. 5	811
758		812
759	[23] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian	813
760	Schroff, and Hartwig Adam. Encoder-decoder with atrous	814
761	separable convolution for semantic image segmentation. In	815
762	<i>ECCV</i> , 2018. 7	816
763		817
764		818
765		819
766		820
767		821
768		822
769		823
770		824
771		825
772		826
773		827
774		828
775		829
776		830
777		831
778		832
779		833
780		834
781		835
782		836
783		837
784		838
785		839
786		840
787		841
788		842
789		843
790		844
791		845
792		846
793		847
794		848
795		849
796		850
797		851
798		852
799		853
800		854
801		855
802		856
803		857
804		858
805		859
806		860
807		861
808		862
809		863