Monocular Depth Estimation using a deep network

Renwen Cui, Zhenye Li

June 21, 2021

1 Introduction

1.1 Background

A dense depth map provides detail 3D geometric relations of objects and their environment within a scene that can be applied to recognition, navigation, image refocusing, and segmentation[1]. Depth estimation from 2D images is an important and basic task in many applications including scene understanding and reconstruction[2]. There are many prior methods for computing the depth map of a scene. Stereo vision matching that simulates human eyes is a popular way to construct 3D depth information by observing the scene from two viewpoints. Using this geometry-based method needs to calibrate the two cameras in advance to obtain the transformation between them[3]. While the above method can efficiently perceive the depth information, binocular vision depends on image pairs. The other method of computing the depth map is based on depth sensors, such as light detection and ranging (LIDAR) that is able to directly capture depth images at high speed, high resolution, and long-range[4]. However, the depth map obtained by LIDAR is sparse rather than dense. Besides, considering the cost and size of this equipment, estimating the dense depth map from a single image attracts more attention.

Much work on monocular depth estimation is focusing on applying deep learning to perform 3D reconstruction from a single RGB image[1]. A variety of deep neural networks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), show their outstanding performance on recovering pixel-level depth maps.

1.2 Project description

For this project, we want to construct a deep neural network to predict the depth map from a single RGB image in an end-to-end way. The input of the network is RGB images, and after applying the network the output of this system is the corresponding estimated depth maps. The two datasets we are planning to use for training the model, NYU Depth v2 and KITTI, contain RGB images and corresponding ground truth of depth maps. Because of the features of our datasets, this supervised monocular depth estimation can be regarded as a regression problem and we can apply machine learning algorithms to our datasets. First, we want to obtain the training datasets including RGB images and ground truth of dense depth maps pairs from the raw datasets. We need to associate depth maps with the closest RGB images and fill invalid depth values via the inpainting method[5]. Then we want to learn and analyze the existing network architectures and strategies, especially the convolutional neural network proposed by I. Alhashim et al.[2]. As shown in Figure 1, I. Alhashim et al. employed an encoder-decoder architecture with skip connections via transfer learning. Finally, we will try to modify the architecture and adjust different loss functions to construct a new model.

1.3 Existing Methods for supervised monocular depth estimation

The problem about depth maps reconstruction from a single RGB image is ill-posed. Many studies aim to improve the resolution and quality of the estimated depth maps with more accurate boundaries using simple model architectures and few parameters. Eigen et al. first solved this problem using the multi-scale framework with two-component slacks, the global coarse-scale network and the local finescale network. The first estimates the global structure, and then a second refines it[6]. Mayer et al. proposed a fully CNN framework to solve this problem[7]. Laina et al. extended ResNet and introduced residual learning to

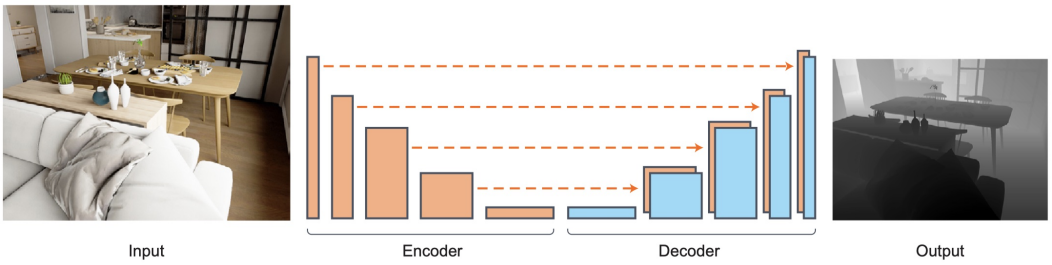

Figure 1: Network architecture[2]

construct a deeper network[8]. I. Alhashim et al. introduced an encoder-decoder network based on the concept of transfer learning[2].

2 Next Steps

- **Jun 17:** Test on our own images with existing models and architectures to see initial results.
- **Jun 25:** Pre-process the two raw datasets of NYU Depth v2 and KITTI to train the model, which includes matching the RGB-depth map pairs and filling invalid depth values of depth maps in these datasets.
- **Jul 4:** Learn and build the existing architecture, encoder-decoder model, and loss function for monocular depth estimation. Besides, evaluate and compare these performances from perspectives of resolution, accuracy, computation, and so on.
- **Jul 25:** Modify and explore the convolutional neural network proposed by I. Alhashim et al.[2], for example, adding super-pixel pooling layers to speed up the convolutional network, using up-sampling blocks or different decoder type to improve the resolution of depth maps. Simultaneously, adjust the loss functions to find the optimal, like creating the vector differential operator to tracing the gradients of the difference between the computed depth and the ground truth in particular directions.
- **Jul 31:** Evaluate the new model architecture from the angles of accuracy and computation.
- **Aug 5:** Clear thoughts, conclude and organize the whole project, and complete the final report as well.

3 Testing with the pre-trained model

We use our own images with the existing encoder-decoder pre-trained model to see initial results.[2] This pre-trained model on NYU Depth v2 dataset focuses on indoor scenes. As shown in Figure 2, RGB images on the left are taken by iPhone 12 in the Medical Science Center of the University of Wisconsin-Madison. After resizing the original pictures with high-resolution to the expected resolution, 640x480, of the pre-trained model, we put them into this model and obtain the estimated depth maps on the right. We put 10 images into the per-trained model, then it takes about 6 seconds to generate corresponding depth maps estimation. Besides, depth maps of objects have clear boundaries and detailed information.

4 Datasets

We plan to train the model on the raw datasets of both NYU Depth v2 and KITTI that are the benchmark and commonest training datasets for the monocular depth estimation.

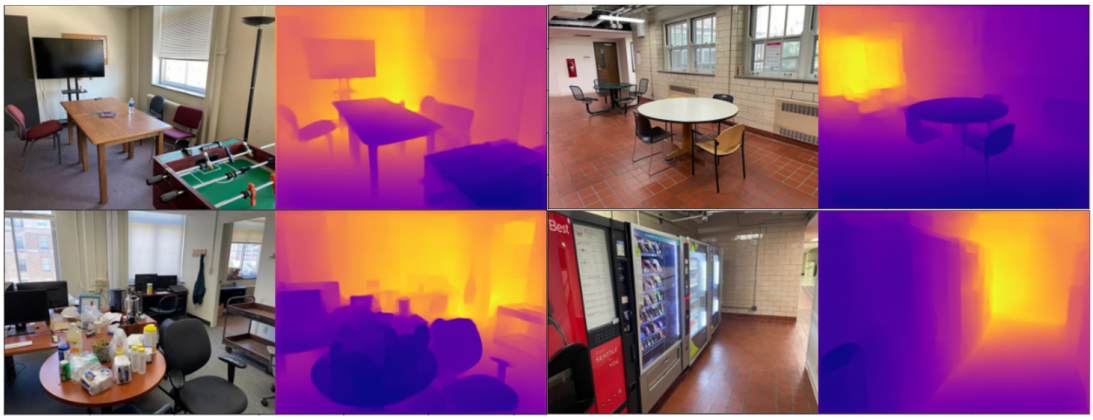

Figure 2: Testing with the pre-trained model

4.1 NYU Depth v2

The NYU Depth dataset v2 is composed of 464 indoor scenes with the resolution of 640x480 as recorded by both the RGB and Depth cameras from the Microsoft Kinect that can collect ground truth of depth directly. The upper bound of the depth maps is 10 meters. The dataset contains 120K training data and 654 testing data. Because sampling rates of the RGB and depth cameras are different, depth maps and RGB images are not one-to-one mapping. Each depth image is matched with its closest RGB image in time as RGB-depth map pair[6]. Besides, depth maps of the raw dataset contain missing or invalid depth values caused by shadows or low albedo surfaces. Using the inpainting method can fill missing depth values[5].

4.2 KITTI

The KITTI is a large dataset that is composed of 56 outdoor scenes including the "city", "residential" categories of the raw data, and so on. Each scene consists of stereo RGB images captured by cameras mounted on a moving vehicle and corresponding sparse 3D laser scans that are sampled at irregularly spaced points by a rotating LIDAR depth sensor[9]. The depth maps have an upper bound of 80 meters. The RGB images are 1224x376 and corresponding depth maps have a very low density with lots of missing depth values. Due to depth maps are sparse, we need to construct the ground truth depths before training the model. Here, we plan to fill missing depth values using the inpainting method[5].

5 Evaluation

Four evaluation indicators of **Abs Rel, RMSE, RMSE log, Accuracies** are often used in evaluating and comparing the performance of different depth estimating networks. These evaluation indicators are defined as:

$$\textbf{average relative error(Abs Rel):}\ \frac{1}{n}\sum_{i\in n}\frac{|d_i-\hat{d}_i|}{\hat{d}_i}$$

$$\textbf{root mean squared error(RMSE):}\ \sqrt{\frac{1}{n}\sum_{i\in n}||d_i-\hat{d}_i||^2}$$

$$\textbf{average error(RMSE log):}\ \sqrt{\frac{1}{n}\sum_{i\in n}||log(d_i)-log(\hat{d}_i)||^2}$$

$$\textbf{threshold accuracy:}\ \%\ \text{of}\ d_i\ \text{s.t}\ \max(\frac{d_i}{\hat{d}_i}\frac{\hat{d}_i}{d_i})=\delta<thr$$

where d_i stands for the predicted depth value of pixel i, $\hat{d}_i$ is the true depth value of pixel i, and n is the total number of pixels for the whole depth image.

References

[1] Zhao, ChaoQiang, et al. "Monocular depth estimation based on deep learning: An overview." Science China Technological Sciences (2020): 1-16.

[2] I. Alhashim and P. Wonka, “High quality monocular depth estimation via transfer learning,” arXiv:1812.11941, 2018.

[3] L. Zou and Y. Li, “A method of stereo vision matching based on opencv,” in 2010 International Conference on Audio, Language and Image Processing. IEEE, 2010, pp. 185–190.

[4] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita, “Lidar scan feature for localization with highly precise 3-d map,” in 2014 IEEE Intelligent Vehicles Symposium Proceedings. IEEE, 2014, pp. 1345–1350.

[5] Levin, Anat, Dani Lischinski, and Yair Weiss. "Colorization using optimization." ACM SIGGRAPH 2004 Papers. 2004. 689-694.

[6] 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, 2014, pp. 2366–2374.

[7] E. Shelhamer, J. T. Barron, and T. Darrell, “Scene intrinsics and depth from a single image,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015, pp. 37–44.

[8] 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). IEEE, 2016, pp. 239–248.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets robotics: The kitti dataset." I. J. Robotics Res., 32:1231–1237, 2013.