CS172 Computer Vision I: Project3 Depth Estimation

Ziyun Zeng
2019533086
zengzy1@shanghaitech.edu.cn

Abstract

In this homework, an single image depth estimation task is performed. Junjie Hu's paper named Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries is reproduced in this homework.

1. Introduction

As a fundamental problem in computer vision, depth estimation shows the geometric relations within a scene, which leads to improvement in some recognition tasks including autonomous driving, robotics, 3D reconstruction, etc. Many depth estimation techniques are based on stereo vision, but there are many scenarios that require monocular image depth estimation. However, estimating depth from a single monocular image is an ill-posed problem due to a lot of difficulties in techniques. In this homework, Junjie Hu's paper is reproduced, and the experiment is carried out on NYU Depth V2 dataset.

2. Estimation algorithm

2.1. Network

In this homework, I will choose ResNet18 as the encoder to feature extraction. Here is the structure block of ResNet18.

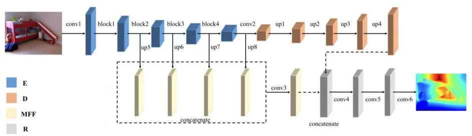

Figure 1. the structure block of ResNet

The table below show what each layer of my ResNet18 do.

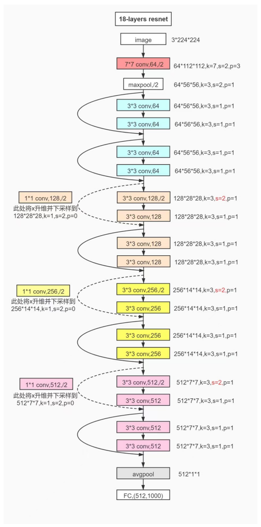

Figure 2. the structure block of ResNet18

Layer	Output Size	Input channels	Output channels
conv1	112*112	3	64
block1	56*56	64	64
block2	28*28	64	128
block3	14*14	128	256
block4	7*7	256	512
conv2	7*7	512	512
up1	14*14	512	256
up2	28*28	256	128
up3	56*56	128	64
up4	112*112	64	32
up5	112*112	64	8
up6	112*112	128	8
up7	112*112	256	8
up8	112*112	512	8
conv3	112*112	32	32
conv4	112*112	64	64
conv5	112*112	64	64
conv6	112*112	64	1

2.2. Loss function

In this homework, I employ the sum of the difference between the depth estimate d_i and its ground truth g_i for a loss:

$$l_1 = \frac{1}{n}\sum_{i=1}^{n} e_i$$

2.3. Accuracy Measures for Depth Estimation

Denoting the total number of (valid) pixels used in all evaluated images by T , we use the following accuracy measures that are commonly employed in the previous studies:

- Mean relative error (REL):$\frac{1}{T}\sum_{i=1}^{T}\frac{\|d_i-g_i\|_1}{g_i}$
- Mean log_{10} error (log_{10}):$\frac{1}{T}\sum_{i=1}^{T}\|log_{10}d_i - log_{10}g_i\|_1$

2.4. Code

2.4.1 Code Repo

- ResNet18.py It contains the whole part of dataloader,resnet18,train and test.
- ./nyuv2/train a dataset for training
- ./nyuv2/test a dataset for testing
- res i the trained model which can be used to test immediately

2.4.2 Classes and Functions in ResNet18

- class DepthDataset It gets the image from dataset iteratively
- class BasicBlock A basic operation for a block
- class ResNet The whole network of resnet
- getTrainingData Get the data for training
- getTestingData Get the data for testing
- resnet18 Gives the layer numbers to use ResNet
- train Use resnet18 to train model and save the model
- test Load a model and test the accuracy of test pictures

2.4.3 How to run

- If you want to train a new model and get the accuracy of the test, you can run ResNet18.py with train() and test() in the main function
- If you want to test the accuracy, you can run ResNet18.py with only test() in the main function

2.5. Result of testing

Due to the use of an optimizer, if we run the train dataset for several times, it may have a better train model. We call the iter times epoch. For each epoch, I run 5 times to get the 5 REL and Log10, then get the average,and all the data are shown below.

epoch	REL	Log10
2	0.4185	0.5081
5	0.4140	0.2460
6	0.3947	0.1933
7	0.3830	0.1669
8	0.3798	0.1575
9	0.3755	0.1543
10	0.3721	0.1532

From the table, we can find that REL and Log10 are always decrease with the increasing of epoch, which is the same as my prediction that more calculation with an adam optimizer can help me get a better model.

3. Visualization

3.1. Code

- ./nyuv2/test xxx.png: a depth picture get from estimation
- visualization.py : change depth picture into a 3D point cloud
  Run visualization.py directly and then you will a popupwindow will show the 3D point cloud of the depth picture.

3.2. Result of visualization

Here is the result of visualization.

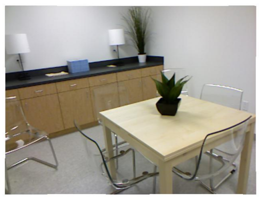
Figure 3. Origin Picture

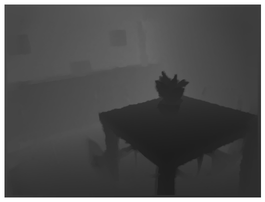
Figure 4. Depth Picture

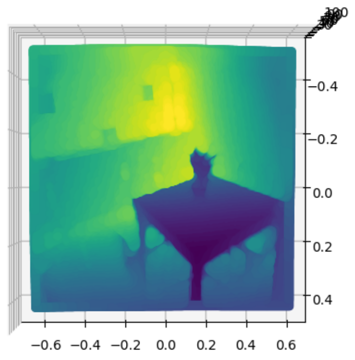

Figure 5. 3D point cloud

4. Problems

4.1. Pixels affect the output depth most

According to my opinion, pixels on the edge between two objects may affect the output depth most, because the pixels on the edges have a large gradient change, so if we miss the edge then we can't distinguish an object from another one and we can't get the related depth of two objects.

For demonstrating the correctness of my proposal, I decide to design a certification program. We can use a sliding window to cover a 9*12 pixels area, and let the area be white, then get the depth picture of these pictures iteratively. After that, calculate the difference between these pictures with the origin depth picture, that find the picture with the biggest difference with the origin picture, and then check where the 9*12 area contain edge pixels.

4.2. Low resolution pictures

Due to the limit of time, I decide to resize the origin color pictures from 480*640 to 224*224 and resize the depth pictures from 480*640 to 112*112. Downsampling of picture may cause less evidence can be used, so it may cause the REL and Log10 be larger than using 480*640 color pictures and depth pictures.