SAM Enhanced Depth Estimation

Huang Baixiang
bhuangak@connect.ust.hk

Abstract

SAM model, segment anything model, is recently widely used in different scenes. As learnt from class, the segmentation and depth estimation has a very close relation, So I want to use the SAM to enhance the performance of the depth estimation

1. Introduction

The traditional depth estimation using the machine learning way usually use the skip-connect network as the backbone, we can see its basic structures in Figure 1.

The use of skip-connection network can reserve most of the features in the original images and make a good result. However, only training the original images is not enough. If we can train the segmentation images and the original images together. We could get a better result because of the feature in the segmentation. That is why we want to use the SAM to enhance the depth estimation. [3]

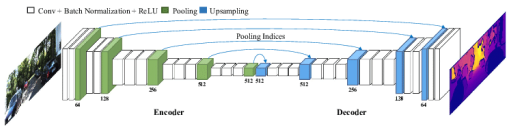

Figure 1. The basic structure of the depth estimation model

2. SAM enhanced depth estimation model

2.1. Describe

The segmentation of the images is not only similar to depth ground truth, but it will also make some contributions to the depth estimation. Here we give out a new model called SAM-enhanced model to help me get a better depth estimation result.

2.2. Data preparation

We are going to implement different datasets for training, validation, and testing. The main datasets we use for training and validation are the Kaggle nyu2 datasets which have already got the original images and the depth ground truth in pair. [2]

for the Kaggle nyu2 datasets, it is divided into different parts based on the scenes. But this make little influence on our training. We simply add them in pairs together when training. The pair information can be found in the .csv files which correspond the images in the dataset.

Another thing is the SAM model. We simply download is from its own web page and also download the related packages for us to use.

2.3. Model building

The basic model we build is a traditional U-skip network structure. This kind of network will reserve the features of the previous layer and is widely used for segmentation and depth estimation. Here, we still tried to use this structure as the basic of the model.

Beside the U-skip network working on the original images, we additionally add an extra path to combine to the output. This path is mainly based on the SAM model which you can see from the Figure 2.

This is the SAM [1] enhanced depth estimation model. I create an extra path for the depth estimation based on the segmented based images. Then I put both images, the original images and segmented images to the convolution layers to extract their features. Here I use the same structure of the convolution layers, U-skip network, with the same weight to train both of them. This is because both images have similar features and colors which would be powerful training with the same model.

After that we add the depth matrix together and calculate their mean to get the final version of the depth estimation. Then we use the ground truth and the output estimation to get the loss and go backward to update the model.

3. Loss Functions

3.1. Describe

The model of the last layer will use a Sigmoid function as an output, which means I should compress the original depth ground truth data to 0 to 1 for training. In this case, we tried to loss functions for training. First one is the L1 loss function and the other one is the MSE function. [4]

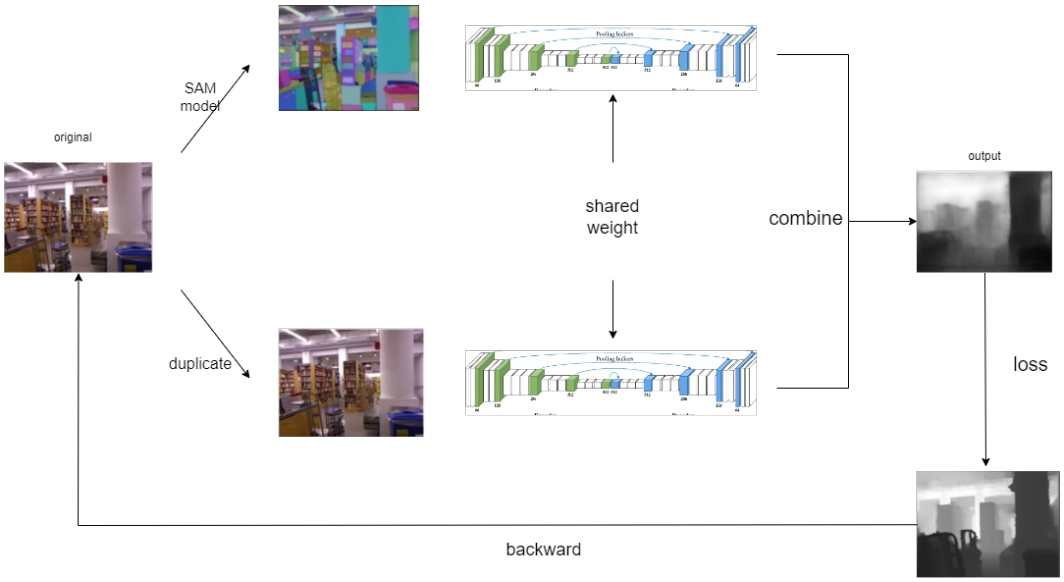

Figure 2. The basic structure of the depth estimation model

Next, I would like to introduce the implementation of the basic functions.

3.2. MSE Loss Function

The MSE loss function is designed to calculate the distance between the predicted depth and the ground truth depth.

The x is the prediction of the depth, while the y is the ground truth depth. This function is mainly used for caculate the overall distance of two different matrix.

$$\mathcal{L}_{MSE}(x, y) = \sum_{i=1}^{N}(x - y)^2$$

This loss function can be simply implemented by using the inner APIs of PyTorch.

3.3. L1 Loss Function

Despite the mse loss function, we also introduce the L1 loss function which takes into the prediction image and the ground truth depth image.

The main difference of the L1 loss and the MSE loss is that L1 loss can tolerate more on the depth estimation because it is one power. In that case, the training progress of L1 is much more smooth.

$$\mathcal{L}_1(x, y) = \sum_{i=1}^{N}|x - y|$$

We can also implement this function through the inner APIs in the PyTorch

4. Result

Here we use mainly use the MSE loss function and you can see the result on Figure 3. Since the MSE loss is power of 2 so the loss may have some fluctuation. But anyway, you can see that the overall training progress is quite good.

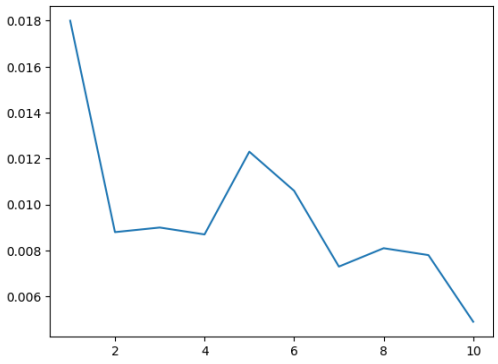

Figure 3. The loss change of the model

The output is shown in Figure 4. You can obviously see that the output of the depth estimation is really like the ground truth one.

You can see from Figure 5, which you can easily see the bad performance if we directly apply a simple U-skip network. We can roughly find the contour of the objects in the images but it is still hard for us to get a good depth estimation.

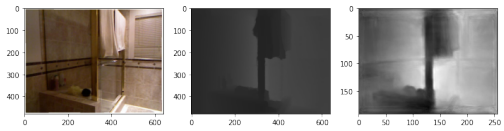

Figure 4. The output of the depth estimation. the first one is the original image, the second one is ground truth, the third one is the prediction

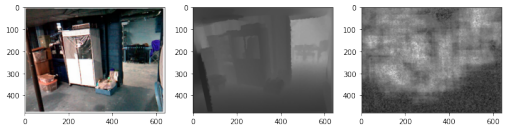

Figure 5. The output of the depth estimation. the first one is the original image, the second one is ground truth, the third one is the prediction

5. Improvement

5.1. Enlarge Dataset

We can find that the Dataset of different folder is often from a single view. So it would be much] better if we can enlarge the dataset and train the model with a specific scenes from different view.

5.2. train the SAM together

The training method we are using now is only focusing on training the U-skip net while using the existing SAM model as a tool. However, I think we can get a better result if we train the SAM model while we are doing the depth estimation. This will strengthen the ability of the SAM model while training and may have a better prediction output.

References

[1] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 1

[2] nyu. nyu dataset on kaggle. 1

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1

[4] Xi Yan. loss choosing. 1