VRDL HW3: Instance segmentation:

Name:陳政彥
Student ID:310552006

Link of my code:

Github repo link:
https://github.com/egghead2630/VRDL_HW3

Model:
https://drive.google.com/file/d/1Mc9ts3-ChO_9GNzbLhPwVQpAO7_K7_p7/view?usp=sharing

README:
https://github.com/egghead2630/VRDL_HW3/blob/main/README.md

Reference:

1.https://github.com/open-mmlab/mmdetection

Brief introduction

My model is trained for the HW3 instance segmentation problem, I use mainly mmdetection to accomplish the HW3.

Basic structure:
I split the training data into train-part and validation-part with proportion: **9:1**. To be precise, I use 2 pictures as validation-part. The model which performs best on validation-part would be chosen.

Technics applied:
1. Some preprocess and augmentation on the train-part
2. Proper hyperparameters setting(Tuning)
3. Model selection

In the HW2, I apply above methods to train my model better, I will explain them in detail in the following part: **Methodology**

Methodology:

This part, I simply introduce how I apply the above methods in the HW2.

About Data Preprocess and Augmentation:

I simply use the default augmentation and preprocess setting of mmdetection, since the default settings are good enough for me to gain a satisfying predictive model

About Hyperparameters tuning:

Image size:

I first apply the (500*500) image size to deal with Out Of Memory issue, however, I find it not suitable to do so because all of the pictures in data set are of great size, that is (1000 * 1000).

Therefore, I latter choose to use the original size (1000 * 1000). To achieve this without OOM occuring, I change some settings in config files and backbone as suggested by mmdetection creators, and finally get a good model that outperforms the baseline model.

learning rate:

I tried to set the learning rate to some other values instead of the default 0.02, however, I find there is no great difference in the models' final performance. Therefore, for simplicity, I apply the default learning rate as my learning rate.

I will list some results below in: **experimental results** part to prove the idea.

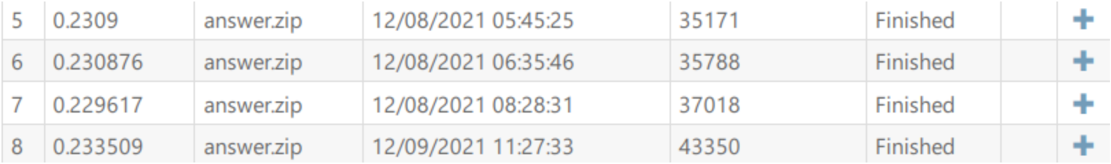
About Model Selection:

It has been a hard time searching for models in the model zoo of mmdetection. Actually, there are hundreds of models to pick from, and it's not clear to me which model is suitable for my task.

Therefore, I decided to start following TA's suggestion: using mask-rcnn. Based on this, I find a tutorial in mmdetection github. In the tutorial, a mask-rcnn using resnet50 as backbone is used as the model to perform an instance segmentation task, exactly the same task in this homework! So I chose the same model in the tutorial as my first beginning model, and surprisingly, I found this tutorial model: mask_rcnn_r50_caffe_fpn_mstrain-poly_1x_coco.py is good enough if pretrained, so that I selected this as my final model.

Experimental Results:

In this part, I will show some results, all trained from the same network: mask_rcnn_r50_caffe_fpn_mstrain-poly_1x_coco.py



5	0.2309	answer.zip	12/08/2021 05:45:25	35171	Finished		+
6	0.230876	answer.zip	12/08/2021 06:35:46	35788	Finished		+
7	0.229617	answer.zip	12/08/2021 08:28:31	37018	Finished		+
8	0.233509	answer.zip	12/09/2021 11:27:33	43350	Finished		+

Above are some results with different learning rates, I 've tried 0.02(default),0.01, and 0.005, and it seems there's only a slight difference between the mAP, that is, learning rate influenced little if it is in a suitable range.

Summary:

I will summarize **what I have learned in this homework.**

What I have learned in the homework:

Basically I learn some some important aspects including:

1.How to take advantage of open-source tools:

Before writing this homework, I had little experience using open-source tools, and now I realize that making good use of these tools helps save a lot of time and effort.

2.How to pick proper hyperparameters:

It is important to pick proper hyperparameters, in this homework, I learned that picking wrong hyperparameters can lead to terrible accuracy of our model. Therefore, we should be careful when picking hyperparameters.

3.How to choose a proper model:

In this homework, I learned that when I need to search for a good model from a great number of models, it would be a good start following other’s advice first.

Above is my report for HW3, thanks for the patience reading it all down here!

Have a Nice day!

Sincerely,
Chen.