

User Manual

This is the user manual of my SRGAN code, to run this program, you should follow the next steps. If you want to train this GAN network, I suggest you run it on a machine with extremely advanced GPU hardware. It takes my 3 days training the current pre-trained model. However, I still provide a pre-trained model with you for your testing.

1 Prerequisite

To run this program, firstly, you should install all those software and dependencies on your machine.

- Python 2.7/3.5/3.6

I suggest you install anaconda instead of python. Anaconda is a powerful software that will help you install a large number of useful python packages. Therefore, you don't need to install your python packages one by one.

- Tensorflow(suggest the version above 1.2)

You can install your Tensorflow for CPU or GPU, detailed steps are on

<https://www.tensorflow.org/install/>

- Matlab

2 To pre-processing your OCT and CT image

You can find matlab code in /lib folder, modify dcm2png.m as you want so you can iteratively transfer dcm file to png in your directory. The detailed explanation of code is in 10.1 in the dissertation.

3 To test your images

Firstly, if you don't train your own model for SRGAN, you should download our pre-trained model from

<https://drive.google.com/file/d/1i664FE1s0WVTpd5lKUZMFum-nrBnvwGI/view?usp=sharing>

According to my dissertation, you can use different perceptual loss function. You can choose either MSE content loss or VGG loss. If you want to use VGG loss, please download VGG19 model in advance. You can download from this URL:

http://download.tensorflow.org/models/vgg_19_2016_08_28.tar.gz

Then open run_SRGAN.sh, you can modify the parameter of your testing process in terminal, use "vi run_SRGAN.sh"

Parameter name	Optional arguments	Explanations
CUDA_VISIBLE_DEVICE	-	Your gpu device number
output_dir	./result	Your HR image output directory
summary_dir	./result/log	Your HR image result log
mode	train/test/run	The mode of your script, you can choose either train or test or run

ls_training	False/True	If you set to true, it means training, otherwise testing/running
task	SRGAN	The task of your program, the parameter is used for future work(add more task in code)
Input_dir_LR	./data/myImages	The directory of the input resolution input data
num_resblock	16(any number, preferably multiples of four)	How many residual blocks are there in the generator
perceptual_mode	MSE or VGG54 or VGG22	The type of feature used in perceptual loss as described in dissertation
pre_trained_model	True/False	If set True, the weight will be loaded but the global_step will still be 0. That means you start a new training process again. If set False, you are going to continue the training from the checkpoint you choose. The global_step will be initialized from the checkpoint.
checkpoint	./SRGAN_pre-trained/model-200000	If provided, the weight will be restored from the provided checkpoint

Finally, after you modify your script, you can run the script. On mac and Ubuntu, you should use “sh run_SRGAN.sh”

4 To train your model

You can train your model yourself. The method of training the model is similar to run the program but costs more.

There are more parameter to tune in train_SRGAN.sh, first you need some training set. The training set should include high-resolution image and low-resolution image. I strongly recommend you to use ImageNet database or RAISE database. You should transfer them into low-resolution image as well.

Similarly, the parameter in “train_SRGAN.sh” is as follows:

Parameter name	Optional arguments	Explanations
CUDA_VISIBLE_DEVICE	-	Your gpu device number
output_dir	./experiment_SRGAN_VGG54/	It means the output directory of your model checkpoint
summary_dir	./experiment_SRGAN_VGG54/log/	It means the output directory of your pre-trained mode
batch_size	16	Batch size of the input batch, each training takes

		batchsize number of samples in the training set
flip	False/True	Whether random flip data augmentation is applied
random_crop	True/False	Whether perform the random crop
crop_size	24	The size of trimming the training set image, the image after crop will be 24*24
input_dir_LR	./data/RAISE_LR/	The directory of the low resolution input data
input_dir_HR	./data/RAISE_HR/	The directory of the high resolution input data
name_queue_capacity	4096	The capacity of the filename queue (suggest large to ensure enough random shuffle)
image_queue_capacity	4096	The capacity of the image queue (suggest large to ensure enough random shuffle)
ratio	0.001	The ratio between content loss and adversarial loss
learning_rate	0.0001	Learning rate
decay_step	100000	The steps needed to decay the learning rate
decay_rate	0.1	The decay rate of each decay step
beta	0.9	The beta parameter for the Adaptive Moment Estimation(Adam) Optimizer(we don't use SGD)
vgg_scaling	0.0061	The scaling factor for the perceptual loss if using vgg perceptual loss
max_iteration	200000	The number of global steps

Finally, just use “sh Train_SRGAN.sh” to train your model. If you stopped at some point, you can continue use checkpoint.