Super Resolution

Network Architecture

I build 3-layered network for Super Resolution tasks(2x), 3x3 conv with ReLU as activation. (see **SuperResolutionNetwork** at `lib/network.py`)

Detail hyper parameter setup as below:

- Use Adam optimizer with `lr=.001` and `lr_decay=.0`.
- Use **Mean Squared Error** as loss
- Use PSNR as evaluation metric
- Initialize kernel(*random uniform*) and bias(*zeros*) in convolution layers.

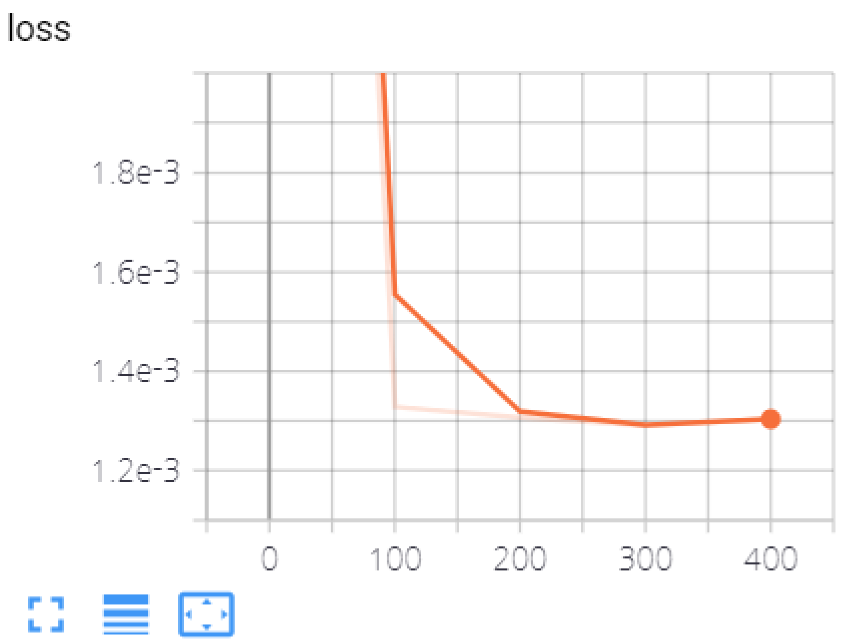
Dataset

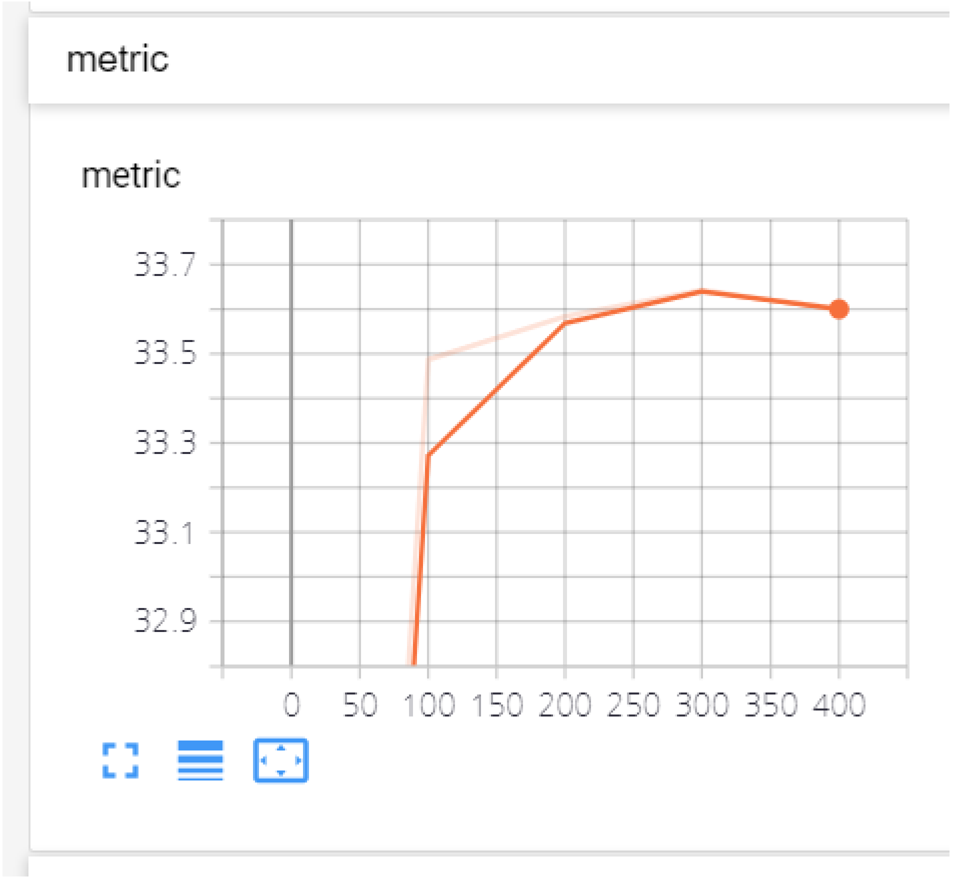
Use all `91` and `291` dataset. Create target image, random crop from image after random scale down and create source image using scale down and up by half.

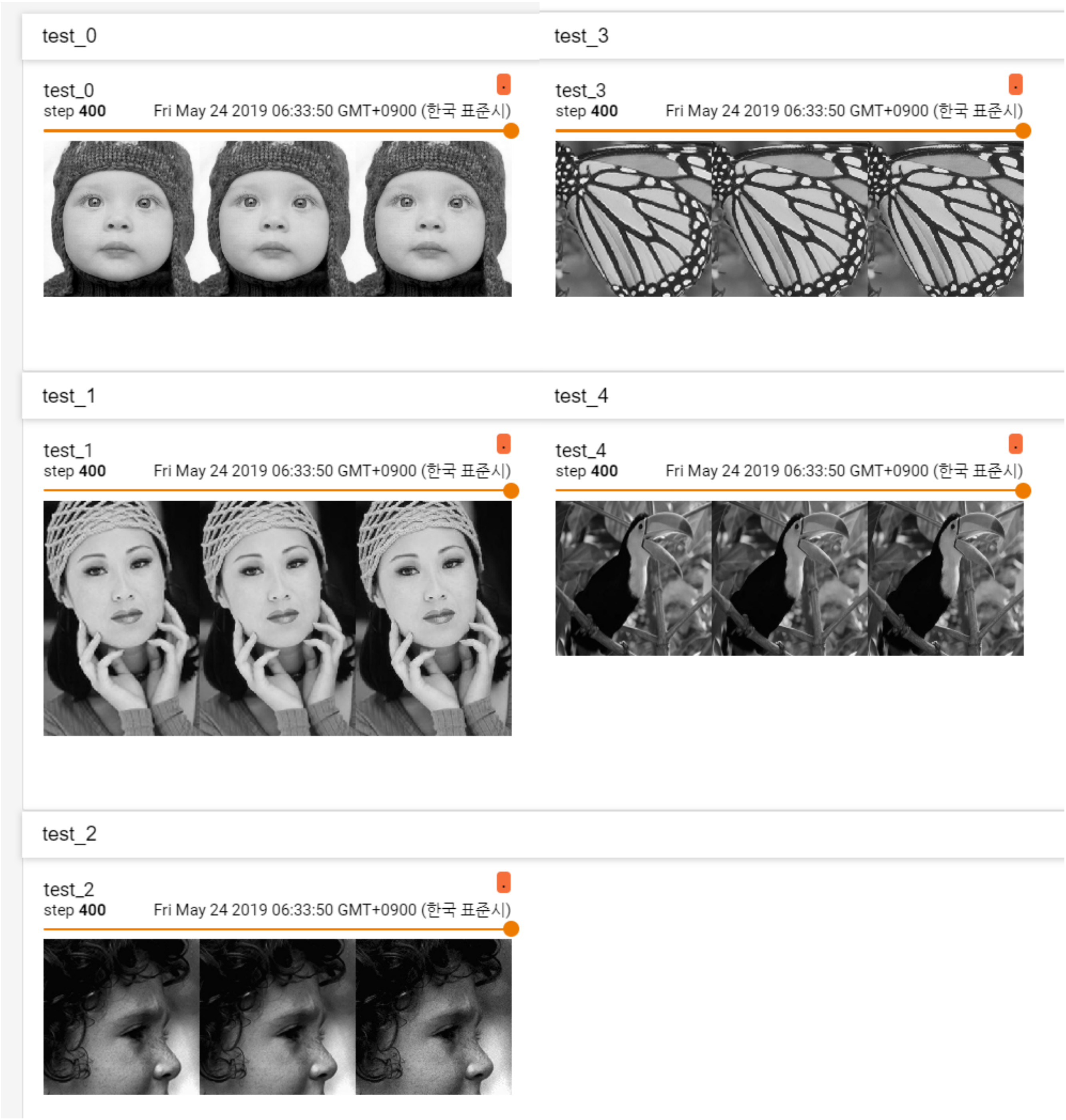
Evaluation

I use PSNR as evaluation metric, (see **SuperResolutionNetwork.metric** at `lib/network.py`) using TensorFlow implementation, `tf.image.psnr`.

Implement **Custom Callback** (see **CustomCallback** at `utils/callbacks.py`) for logging loss, accuracy and sample images. For each interval, write inferenced image summary for one train set and all test set.


loss
1.8e-3
1.6e-3
1.4e-3
1.2e-3
0
100
200
300
400


metric
metric
33.7
33.5
33.3
33.1
32.9
0
50
100
150
200
250
300
350
400



test_0

test_0
step **400** Fri May 24 2019 06:33:50 GMT+0900 (한국 표준시)

test_1

test_1
step **400** Fri May 24 2019 06:33:50 GMT+0900 (한국 표준시)

test_2

test_2
step **400** Fri May 24 2019 06:33:50 GMT+0900 (한국 표준시)

test_3

test_3
step **400** Fri May 24 2019 06:33:50 GMT+0900 (한국 표준시)

test_4

test_4
step **400** Fri May 24 2019 06:33:50 GMT+0900 (한국 표준시)