Setting up the environment

```
!git clone -b master https://github.com/erickcfarias/ml-deployment
```

```
    fatal: destination path 'ml-deployment' already exists and is not an empty directory.
```

```
import os
os.chdir('ml-deployment/capstone_gan_circle/')
```

```
!apt-get update && apt-get install python3-venv && python3 -m venv .env && source .env/bin/activate
```

```
    Hit:1 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
    Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
    Hit:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
    Hit:4 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
    Hit:5 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease
    Get:6 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
    Hit:7 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
    Get:8 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
    Ign:9 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64  InRelease
    Ign:10 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64  InRelease
    Hit:11 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64  Release
    Hit:12 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64  Release
    Fetched 163 kB in 2s (92.9 kB/s)
    Reading package lists... Done
    Reading package lists... Done
    Building dependency tree
    Reading state information... Done
    python3-venv is already the newest version (3.6.7-1~18.04).
    0 upgraded, 0 newly installed, 0 to remove and 48 not upgraded.
```

```
!python3 -m pip install -r source/requirements.txt --quiet
```

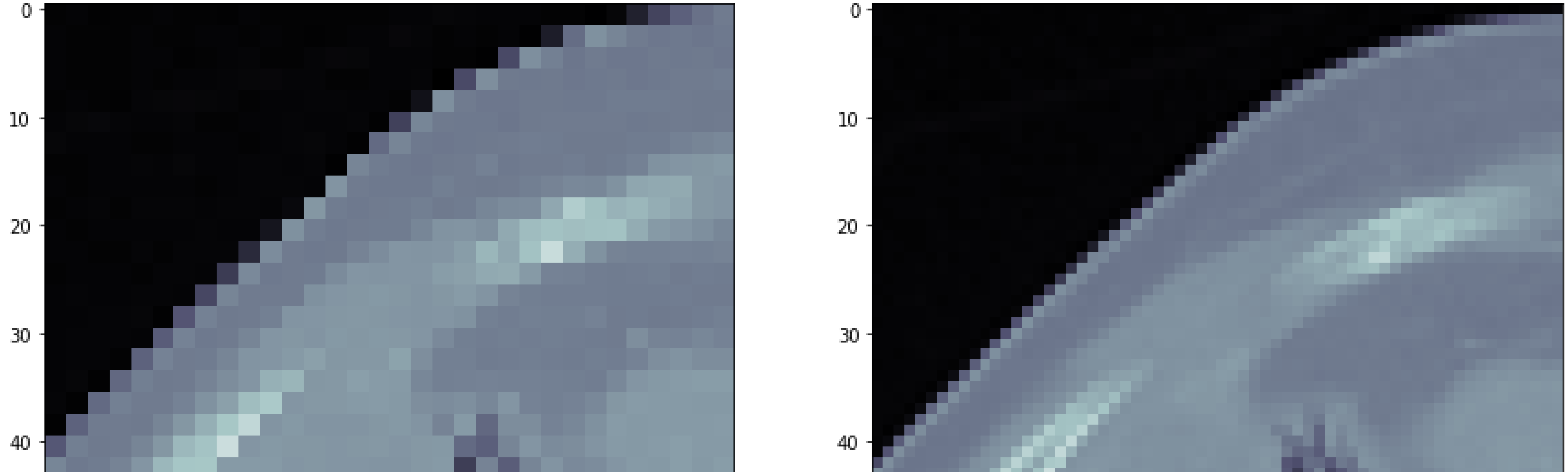
1. Downloading the data

```
from source.utils.preprocessor import DeepLesionPreprocessor
config = {
    'input_path': 'raw_data/',
    'output_path': 'preprocessed_data/',
    'data_urls': ['https://nihcc.box.com/shared/static/l52tpmmkgjlfa065ow8czhivhu5vx27n.zip'],
    'download': True,
    'delete_raw': True,
    'train': True,
    'test': True,
    'multi_size_input': True,
    'crop_size': 128,
    'input_size': 64
}
preprocessor = DeepLesionPreprocessor(config)
preprocessor.run()
```

2. Visualizing the problem

```
from glob import glob
import matplotlib.pyplot as plt
import random
import numpy as np
import tensorflow as tf
import SimpleITK as sitk

files = glob("preprocessed_data/train/*")
file = random.choice(files)

fig, axs = plt.subplots(1, 2, figsize=(15, 10))
img = sitk.ReadImage(file)
img = sitk.GetArrayFromImage(img)
img = img * 1. + (-32768)
hu_0 = -10
hu_1 = 400
img = 2. * ((img - (hu_0)) / (hu_1 - (hu_0))) - 1
img = 2. * ((img - (hu_0)) / (hu_1 - (hu_0))) - 1.
img = tf.expand_dims(img, 2)

hr_patch = tf.cast(tf.image.resize(
                img,
                (64, 64),
                method=tf.image.ResizeMethod.BICUBIC,
                preserve_aspect_ratio=True
            ), tf.float32)

lr_patch = tf.cast(tf.image.resize(
                hr_patch,
                (32, 32),
                method=tf.image.ResizeMethod.NEAREST_NEIGHBOR,
                preserve_aspect_ratio=True
            ), tf.float32)

lr_patch = tf.cast(tf.image.resize(
                lr_patch,
                (64, 64),
                method=tf.image.ResizeMethod.NEAREST_NEIGHBOR,
                preserve_aspect_ratio=True
            ), tf.float32)


lr_patch = np.squeeze(lr_patch)
axs[0].imshow(lr_patch, cmap=plt.cm.bone)

hr_patch = np.squeeze(hr_patch)
axs[1].imshow(hr_patch, cmap=plt.cm.bone)
```

```
<matplotlib.image.AxesImage at 0x7f23505c3588>
```



3. Building the network - GAN CIRCLE

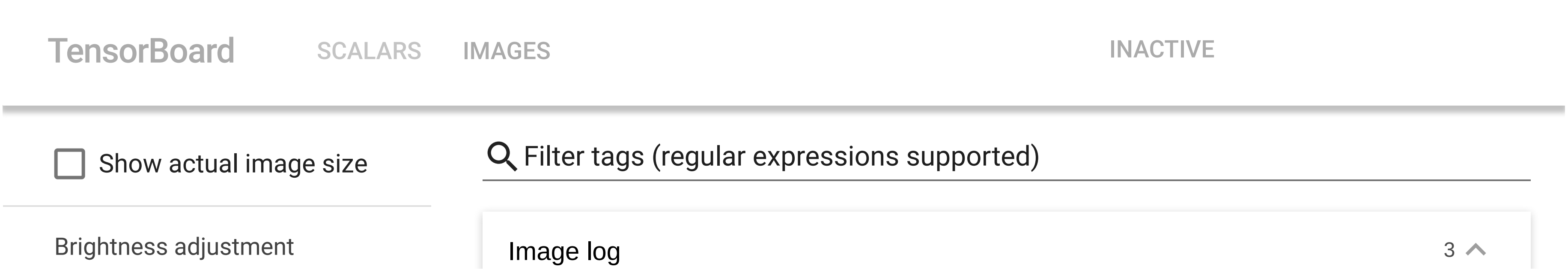
```
!pygmentize 'source/model/circle_gan.py'
```

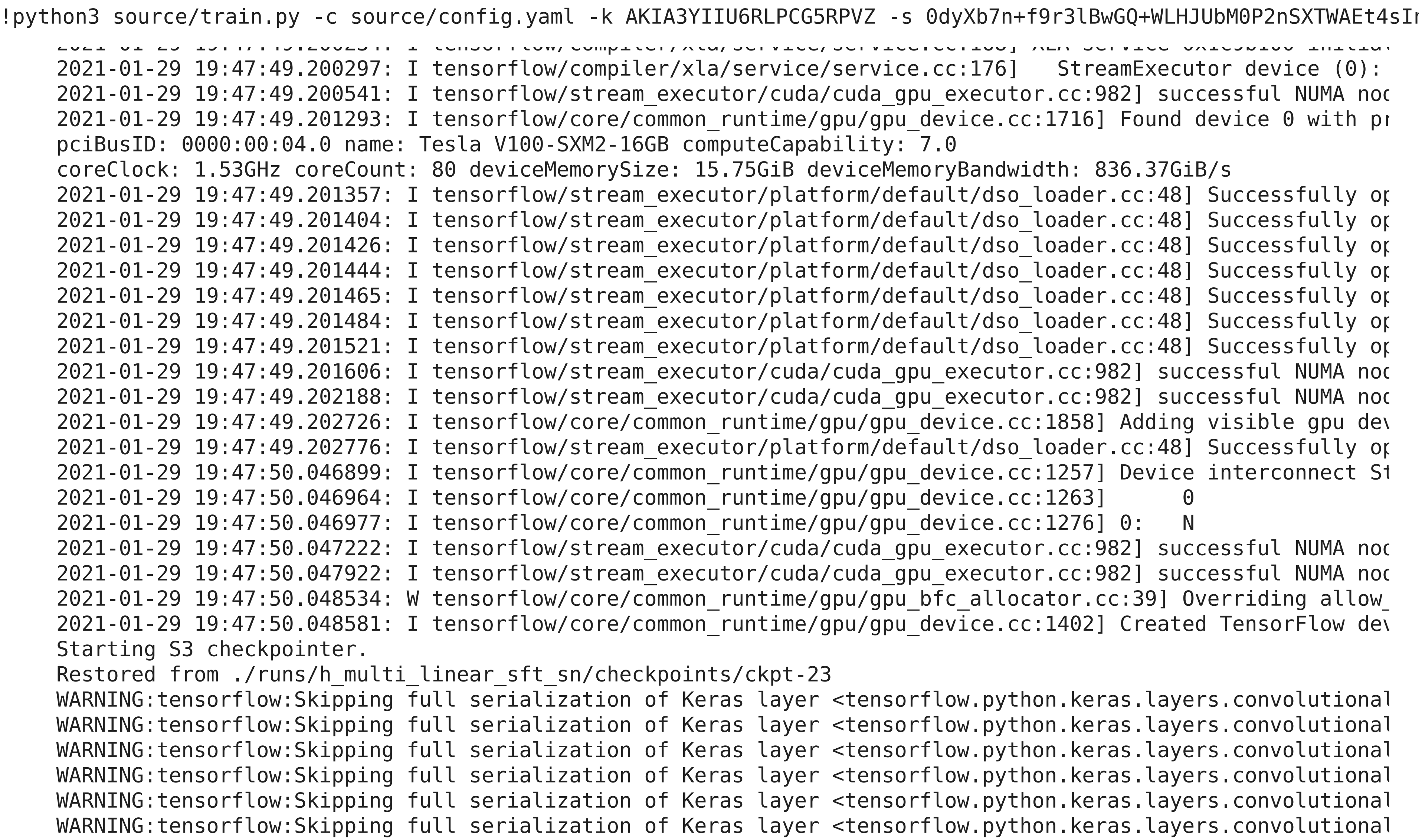
```
from model.models import discriminator, generator, sft_generator, sa_generator
from utils.cloud import S3Manager
from utils.image_tools import calculate_image_similarity
import tensorflow as tf
import numpy as np
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import Progbar
import subprocess
import os
import pickle as pkl


class ganCIRCLE:

    def __init__(self,
                 config,
                 s3_key=None,
                 s3_secret=None
                 ):
```

```
learning_rate=config['learning_rate']
ttur_rate=float(config['ttur_rate'])
lambda_cycle=config['cycle_loss_weight']
lambda_id=config['identity_loss_weight']
lambda_val=config['validation_loss_weight']
lambda_jst=config['joint_loss_weight']
checkpoint_every=config['checkpoint_every']
version_name=config['version_name']
input_type=config['input_type']
hu_scale_min=config['hu_scale_min']
hu_scale_max=config['hu_scale_max']
conditioning=config['conditioning']
spectral_normalization=config['spectral_normalization']
gen_output_activation=config['generator_output_activation']

self.lambda_0 = lambda_val
self.lambda_1 = lambda_cycle
self.lambda_2 = lambda_id
self.lambda_3 = lambda_jst
self.checkpoint_every = checkpoint_every
self.version_name = version_name
self.input_type = input_type
self.s3_key = s3_key
self.s3_secret = s3_secret
self.batch_size = config['batch_size']

# Create Folders for checkpointing and images saving
self._create_folders()

# Build Models

# Build the critics
self.d_hr = discriminator(img_shape=(None, None, 1), spectr_norm=spectral_normalization)
self.d_lr = discriminator(img_shape=(None, None, 1), spectr_norm=spectral_normalization)

# Build the generators
if conditioning == 'sft':
    self.g_hr_lr = sft_generator(
        img_shape=(None, None, 1), hu_min=hu_scale_min, hu_max=hu_scale_max,
        spectr_norm=spectral_normalization, gen_out=gen_output_activation
```

4. Training the network

```
%load_ext tensorboard
%tensorboard --logdir runs/h_multi_linear_sft_sn/tensorboard_log
```





```
!python3 source/train.py -c source/config.yaml -k AKIA3YIIU6RLPCG5RPVZ -s 0dyXb7n+f9r3lBwGQ+WLHJUbM0P2nSXTWAEt4sIr
    2021-01-29 19:47:49.200234: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x1e9b100 initia
    2021-01-29 19:47:49.200297: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0):
    2021-01-29 19:47:49.200541: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA noc
    2021-01-29 19:47:49.201293: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1716] Found device 0 with pr
    pciBusID: 0000:00:04.0 name: Tesla V100-SXM2-16GB computeCapability: 7.0
    coreClock: 1.53GHz coreCount: 80 deviceMemorySize: 15.75GiB deviceMemoryBandwidth: 836.37GiB/s
    2021-01-29 19:47:49.201357: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
    2021-01-29 19:47:49.201404: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
    2021-01-29 19:47:49.201426: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
    2021-01-29 19:47:49.201444: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
    2021-01-29 19:47:49.201465: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
    2021-01-29 19:47:49.201484: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
    2021-01-29 19:47:49.201521: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
    2021-01-29 19:47:49.201606: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA noc
    2021-01-29 19:47:49.202188: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA noc
    2021-01-29 19:47:49.202726: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1858] Adding visible gpu dev
    2021-01-29 19:47:49.202776: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
    2021-01-29 19:47:50.046899: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1257] Device interconnect St
    2021-01-29 19:47:50.046964: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1263]      0
    2021-01-29 19:47:50.046977: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1276] 0:   N
    2021-01-29 19:47:50.047222: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA noc
    2021-01-29 19:47:50.047922: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA noc
    2021-01-29 19:47:50.048534: W tensorflow/core/common_runtime/gpu/gpu_bfc_allocator.cc:39] Overriding allow_
    2021-01-29 19:47:50.048581: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1402] Created TensorFlow dev
    Starting S3 checkpointer.
    Restored from ./runs/h_multi_linear_sft_sn/checkpoints/ckpt-23
    WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
    WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
    WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
    WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
    WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
    WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
```

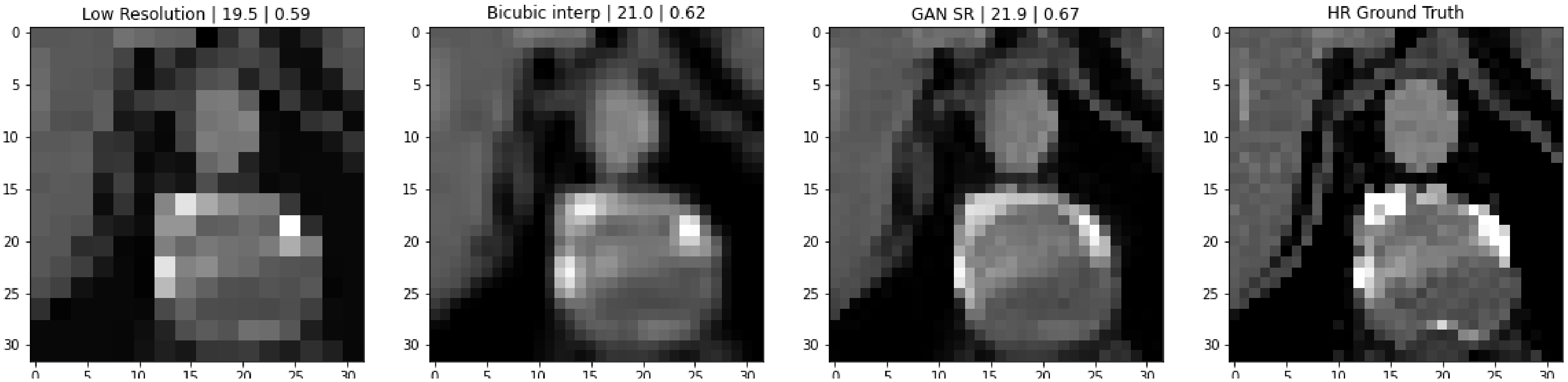
```
WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
WARNING:tensorflow:Skipping full serialization of Keras layer <tensorflow.python.keras.layers.convolutional
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/training/tracking/tracking
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
2021-01-29 19:48:07.183034: W tensorflow/python/util/util.cc:348] Sets are not currently considered sequenc
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/training/tracking/tracking
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
Each epoch will have 441 iterations

 epoch 0/5
2021-01-29 19:48:37.845411: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
2021-01-29 19:48:38.780550: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully op
 441/2205 [=>........] - ETA: 22:31
 epoch 1/5

 882/2205 [===>......] - ETA: 13:51
 epoch 2/5
1323/2205 [=====>....] - ETA: 8:27
 epoch 3/5
1764/2205 [=======>..] - ETA: 4:01
 epoch 4/5
2205/2205 [==========] - 1174s 533ms/step
```

5. Results and conclusion

```
import tensorflow_addons as tfa
from source.model.layers import condition, sft

sr_gen = tf.keras.models.load_model('sr_generator_model')
```

```
WARNING:tensorflow:No training configuration found in save file, so the model was *not* compiled. Compile it
```

```
from skimage.metrics import structural_similarity as ssim
from skimage.metrics import structural_similarity as ssim
from skimage.metrics import peak_signal_noise_ratio as psnr

files = glob("preprocessed_data/test/*")

while True:
  img = random.choice(files)
  img = sitk.ReadImage(img)
  img = sitk.GetArrayFromImage(img)
  img = np.squeeze(img)
  img = img * 1. + (-32768.)
  img = (img - (-100)) / (500 - (-100))
  img[img > 1.] = 1.
  img[img < 0.] = 0.
  img = tf.expand_dims(img, 2)
  break

hr = img
lr = tf.image.resize(
    hr,
    (int(img.shape[0]/2), int(img.shape[1]/2)),
    method=tf.image.ResizeMethod.BICUBIC,
    preserve_aspect_ratio=True
)
cubic = tf.image.resize(
    lr,
    (int(img.shape[0]), int(img.shape[1])),
    method=tf.image.ResizeMethod.BICUBIC,
    preserve_aspect_ratio=True
)

lr = tf.image.resize(
    lr,
    (int(img.shape[0]), int(img.shape[1])),
    method=tf.image.ResizeMethod.NEAREST_NEIGHBOR,
    preserve_aspect_ratio=True
)

hr = np.squeeze(hr.numpy())
lr = np.squeeze(lr.numpy())
lr = np.squeeze(lr.numpy())

cubic = np.squeeze(cubic.numpy())

sr = sr_gen(tf.expand_dims(cubic, 0)).numpy()[0]
sr = np.squeeze(sr)

sr[sr < 0.] = 0.
sr[sr > 1.] = 1.
cubic[cubic < 0.] = 0.
cubic[cubic > 1.] = 1.

L = hr.max() - hr.min()

fig, axs = plt.subplots(1, 4, figsize=(20, 5))
p = np.round(psnr(hr, lr, data_range=L),1)
s = np.round(ssim(hr, lr, gaussian_weights=True, sigma=1.5, win_size=11, data_range=L, multichannel=True),2)
axs[0].imshow(lr, cmap=plt.cm.gray)
axs[0].set_title(f"Low Resolution | {p} | {s}")

axs[1].imshow(cubic, cmap=plt.cm.gray)
p = np.round(psnr(hr, cubic, data_range=L),1)
s = np.round(ssim(hr, cubic, gaussian_weights=True, sigma=1.5, win_size=11, data_range=L, multichannel=True),2)
axs[1].set_title(f"Bicubic interp | {p} | {s}")

axs[2].imshow(sr, cmap=plt.cm.gray)
p = np.round(psnr(hr, sr, data_range=L),1)
s = np.round(ssim(hr, sr, gaussian_weights=True, sigma=1.5, win_sigze=11, data_range=L, multichannel=True),2)
axs[2].set_title(f"GAN SR | {p} | {s}")

axs[3].imshow(hr, cmap=plt.cm.gray)
axs[3].set_title(f"HR Ground Truth")

fig.show()
```



```
from collections import defaultdict

files = glob("preprocessed_data/test/*")

sr_images = defaultdict(list)
for n in range(50):
  img = random.choice(files)
  img = sitk.ReadImage(img)
  img = sitk.GetArrayFromImage(img)
  img = np.squeeze(img)
  img = img * 1. + (-32768.)
  img = (img - (-100)) / (500 - (-100))
  img[img > 1.] = 1.
  img[img < 0.] = 0.
  img = tf.expand_dims(img, 2)

  hr = img
  lr = tf.image.resize(
      hr,
      (int(img.shape[0]/2), int(img.shape[1]/2)),
      method=tf.image.ResizeMethod.BICUBIC,
      preserve_aspect_ratio=True
  )
  cubic = tf.image.resize(
      lr,
      (int(img.shape[0]), int(img.shape[1])),
      method=tf.image.ResizeMethod.BICUBIC,
      method=tf.image.ResizeMethod.BICUBIC,
      preserve_aspect_ratio=True
  )

  lr = tf.image.resize(
      lr,
      (int(img.shape[0]), int(img.shape[1])),
      method=tf.image.ResizeMethod.NEAREST_NEIGHBOR,
      preserve_aspect_ratio=True
  )

  hr = np.squeeze(hr.numpy())
  lr = np.squeeze(lr.numpy())
  cubic = np.squeeze(cubic.numpy())
  sr = sr_gen(tf.expand_dims(cubic, 0)).numpy()[0]
  sr = np.squeeze(sr)
  sr[sr < 0.] = 0.
  sr[sr > 1.] = 1.
  cubic[cubic < 0.] = 0.
  cubic[cubic > 1.] = 1.

  sr_images['hr'].append(hr)
  sr_images['lr'].append(lr)
  sr_images['cubic'].append(cubic)
  sr_images['sr'].append(sr)
```

```
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:5 out of the last 83 calls to <function recreate_function.<locals>.restored_function_body
WARNING:tensorflow:6 out of the last 84 calls to <function recreate_function.<locals>.restored_function_body
```

```
    WARNING:tensorflow:7 out of the last 85 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:8 out of the last 86 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:9 out of the last 87 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:10 out of the last 88 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 89 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:5 out of the last 18 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:6 out of the last 19 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:7 out of the last 20 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:8 out of the last 21 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:9 out of the last 22 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:10 out of the last 23 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 24 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:5 out of the last 434 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:6 out of the last 435 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:7 out of the last 436 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:8 out of the last 437 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:9 out of the last 438 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:10 out of the last 439 calls to <function recreate_function.<locals>.restored_function_bod
    WARNING:tensorflow:11 out of the last 440 calls to <function recreate_function.<locals>.restored_function_bod
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
    WARNING:tensorflow:11 out of the last 11 calls to <function recreate_function.<locals>.restored_function_body
```

```python
import pandas as pd

metrics = defaultdict(list)

peak_cubic = []
peak_sr = []
sim_cubic = []
sim_sr = []
ambe_cubic = []
ambe_sr = []
for lr, hr, cubic, sr in zip(sr_images['lr'], sr_images['hr'], sr_images['cubic'], sr_images['sr']):
    L = hr.max() - hr.min()

    peak_cubic.append(
        psnr(hr, lr, data_range=L)
    )
    peak_sr.append(
        psnr(hr, sr, data_range=L)
    )

    sim_cubic.append(
        ssim(hr, lr, gaussian_weights=True, sigma=1.5, win_size=11, data_range=L, multichannel=True)
    )
    sim_sr.append(
        ssim(hr, sr, gaussian_weights=True, sigma=1.5, win_size=11, data_range=L, multichannel=True)
    )

    ambe_cubic.append(
        np.abs(np.mean(hr)-np.mean(lr_patch)) / L
    )
    ambe_sr.append(
        np.abs(np.mean(hr)-np.mean(sr)) / L
    )

metrics['psnr_cubic_mean'].append(np.nanmean(peak_cubic))
metrics['psnr_cubic_sd'].append(np.nanstd(peak_cubic))
metrics['psnr_sr_mean'].append(np.nanmean(peak_sr))
metrics['psnr_sr_sd'].append(np.nanstd(peak_sr))
metrics['ssim_cubic_mean'].append(np.nanmean(sim_cubic))
metrics['ssim_cubic_sd'].append(np.nanstd(sim_cubic))
metrics['ssim_sr_mean'].append(np.nanmean(sim_sr))
metrics['ssim_sr_sd'].append(np.nanstd(sim_sr))
metrics['ambe_cubic_mean'].append(np.nanmean(ambe_cubic))
metrics['ambe_cubic_sd'].append(np.nanstd(ambe_cubic))
metrics['ambe_sr_mean'].append(np.nanmean(ambe_sr))
metrics['ambe_sr_sd'].append(np.nanstd(ambe_sr))


df = pd.DataFrame(metrics)
display(df)
display(df)
```

	psnr_cubic_mean	psnr_cubic_sd	psnr_sr_mean	psnr_sr_sd	ssim_cubic_mean	ssim_cubic_sd	ssim_sr_mean	ssi
0	20.608964	3.898599	21.977609	5.099824	0.665291	0.158117	0.673964	