In [24]:

```python
import sys
import keras
import cv2
import numpy
import matplotlib
import skimage
import os
```

In [27]:

```python
print(os.getcwd())
print('Keras: {}'.format(keras.__version__))
print('OpenCV: {}'.format(cv2.__version__))
print('NumPy: {}'.format(numpy.__version__))
print('Matplotlib: {}'.format(matplotlib.__version__))
print('Skimage: {}'.format(skimage.__version__))
```

```
/Users/mokshkant/Desktop/GAN
Keras: 2.3.0
OpenCV: 4.1.2
NumPy: 1.17.2
Matplotlib: 3.1.1
Skimage: 0.16.2
```

In [1]:

```python
# import the necessary packages
from keras.models import Sequential
from keras.layers import Conv2D
from keras.optimizers import Adam
from skimage.measure import compare_ssim as ssim
from matplotlib import pyplot as plt
import cv2
import numpy as np
import math
import os

# python magic function, displays pyplot figures in the notebook
%matplotlib inline
```

```
Using TensorFlow backend.
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorflow/python/framewor
```

```
k/dtypes.py:516: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a fut
ure version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorflow/python/framewor
k/dtypes.py:517: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a fut
ure version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorflow/python/framewor
k/dtypes.py:518: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a fut
ure version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorflow/python/framewor
k/dtypes.py:519: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a fut
ure version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)]
)
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorflow/python/framewor
k/dtypes.py:520: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a fut
ure version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorflow/python/framewor
k/dtypes.py:525: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a fut
ure version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)]
)
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorboard/compat/tensorf
low_stub/dtypes.py:541: FutureWarning: Passing (type
, 1) or '1type' as a synonym of type is deprecated;
in a future version of numpy, it will be understood
as (type, (1,)) / '(1,)type'.
```

```
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorboard/compat/tensorf
low_stub/dtypes.py:542: FutureWarning: Passing (type
, 1) or '1type' as a synonym of type is deprecated;
in a future version of numpy, it will be understood
as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorboard/compat/tensorf
low_stub/dtypes.py:543: FutureWarning: Passing (type
, 1) or '1type' as a synonym of type is deprecated;
in a future version of numpy, it will be understood
as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorboard/compat/tensorf
low_stub/dtypes.py:544: FutureWarning: Passing (type
, 1) or '1type' as a synonym of type is deprecated;
in a future version of numpy, it will be understood
as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)]
)
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorboard/compat/tensorf
low_stub/dtypes.py:545: FutureWarning: Passing (type
, 1) or '1type' as a synonym of type is deprecated;
in a future version of numpy, it will be understood
as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/Library/Frameworks/Python.framework/Versions/3.7/li
b/python3.7/site-packages/tensorboard/compat/tensorf
low_stub/dtypes.py:550: FutureWarning: Passing (type
, 1) or '1type' as a synonym of type is deprecated;
in a future version of numpy, it will be understood
as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)]
)
```

In [21]:

```
# define a function for peak signal-to-noise ratio (PSNR)
def psnr(target, ref):

    # assume RGB image
    target_data = target.astype(float)
    ref_data = ref.astype(float)

    diff = ref_data - target_data
    diff = diff.flatten('C')

    rmse = math.sqrt(np.mean(diff ** 2.))

    return 20 * math.log10(255. / rmse)

# define function for mean squared error (MSE)
def mse(target, ref):
    # the MSE between the two images is the sum of the squared di
    err = np.sum((target.astype('float') - ref.astype('float')) *
    err /= float(target.shape[0] * target.shape[1])

    return err

# define function that combines all three image quality metrics
def compare_images(target, ref):
    scores = []
    scores.append(psnr(target, ref))
    scores.append(mse(target, ref))
    scores.append(ssim(target, ref, multichannel =True))

    return scores
```

In [40]:

```
def prepare_images(path, factor):

    # loop through the files in the directory
    for file in os.listdir(path):

        # open the file
        img = cv2.imread(path + '/' + file)
        if img is None:
            continue

        # find old and new image dimensions
        h, w, _ = img.shape
        new_height =int( h / factor)
        new_width = int(w / factor)

        # resize the image - down
        img = cv2.resize(img, (new_width, new_height), interpolat

        # resize the image - up
        img = cv2.resize(img, (w, h), interpolation = cv2.INTER_L

        # save the image
        print('Saving {}'.format(file))
        cv2.imwrite('SRCNN_Data/low_res/{}'.format(file), img)
```

In [41]:

```
prepare_images('SRCNN_Data/Images', 2)
```

```
Saving 1.bmp
Saving 2.bmp
Saving 3.bmp
Saving 4.bmp
```

In [ ]: