Name: Angelo D. Alvarez **Course:** COE 005 – Machine Learning and Prediction

Section: ECE41S11 **Assignment:** GAN Application

Select a sample application of GAN in this link: [GAN Applications]

5 pages, font size of 12 pt, single space, simulations with discussions.

- I import all the libraries needed for the program and create a path to download the dataset from other website.

```
In [1]: # Import all the necessary libraries needed in this program

        import tensorflow as tf
        import os
        import pathlib
        import time
        import datetime
        from matplotlib import pyplot as plt
        from IPython import display
```

```
In [3]: #Downloading the dataset from the website and labeling with a new name

        dataset_name = "City_Scapes"
        _URL = f'http://efrosgans.eecs.berkeley.edu/pix2pix/datasets/{dataset_name}.tar.gz'

        path_to_zip = tf.keras.utils.get_file(
            fname=f"{dataset_name}.tar.gz",
            origin=_URL,
            extract=True)

        path_to_zip  = pathlib.Path(path_to_zip)

        PATH = path_to_zip.parent/dataset_name

        Downloading data from http://efrosgans.eecs.berkeley.edu/pix2pix/datasets/cityscapes.tar.gz
        103441232/103441232 [==============================] - 93s 1us/step
```

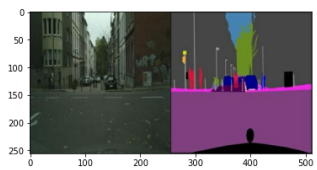
- I extract the windows path of the dataset and defining the size of the images in the dataset

```
In [4]: #Listing the path of the dataset

        list(PATH.parent.iterdir())

Out[4]: [WindowsPath('C:/Users/asus/.keras/datasets/cityscapes'),
         WindowsPath('C:/Users/asus/.keras/datasets/cityscapes.tar.gz')]
```

```
In [11]: #Defining the size of the images

         sample_image = tf.io.read_file(str(PATH / 'train/10.jpg'))
         sample_image = tf.io.decode_jpeg(sample_image)
         print(sample_image.shape)

         (256, 512, 3)
```

- I display a sample image from the dataset to see what kind of images that the program will process.

```
In [12]: #Displaing the sample image from the dataset

         plt.figure()
         plt.imshow(sample_image)

Out[12]: <matplotlib.image.AxesImage at 0x2462d9fb460>
```



- Displaying the train image.

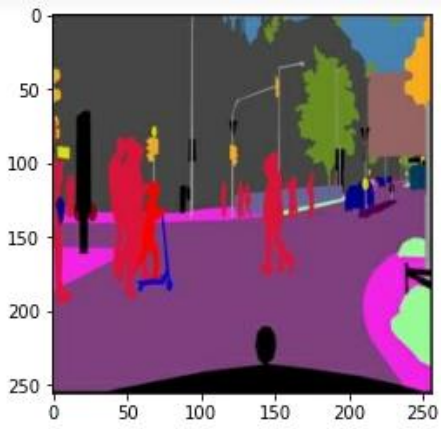
"I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own."

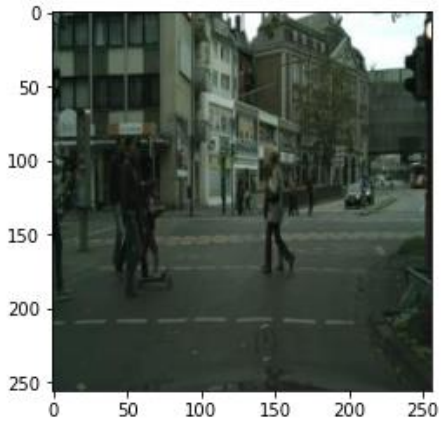
Name: Angelo D. Alvarez **Course:** COE 005 – Machine Learning and Prediction

Section: ECE41S11 **Assignment:** GAN Application

In [15]:
```
#

def load(image_file):

  image = tf.io.read_file(image_file)
  image = tf.io.decode_jpeg(image)

  w = tf.shape(image)[1]
  w = w // 2
  input_image = image[:, w:, :]
  real_image = image[:, :w, :]

  input_image = tf.cast(input_image, tf.float32)
  real_image = tf.cast(real_image, tf.float32)

  return input_image, real_image
```

In [17]:
```
#

inp, re = load(str(PATH / 'train/15.jpg'))

plt.figure()
plt.imshow(inp / 255.0)
plt.figure()
plt.imshow(re / 255.0)
```





- Using the generator to the image.

"I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own."

Name: Angelo D. Alvarez **Course:** COE 005 – Machine Learning and Prediction

Section: ECE41S11 **Assignment:** GAN Application

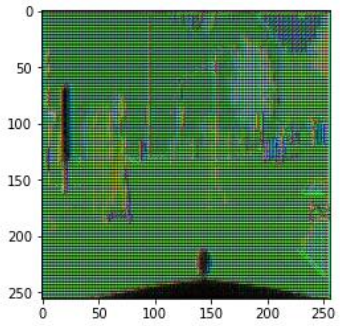
In [34]:
```
generator = Generator()
tf.keras.utils.plot_model(generator, show_shapes=True, dpi=64)
```

```
You must install pydot (`pip install pydot`) and install graphviz (see instructions at https://graphviz.gitlab.io/download/) fo
r plot_model to work.
```

In [35]:
```
gen_output = generator(inp[tf.newaxis, ...], training=False)
plt.imshow(gen_output[0, ...])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

Out[35]: `<matplotlib.image.AxesImage at 0x2462fd82b50>`



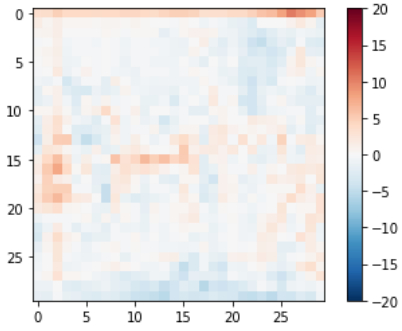
- Using discriminator to the image.

In [40]:
```
discriminator = Discriminator()
tf.keras.utils.plot_model(discriminator, show_shapes=True, dpi=64)
```

```
You must install pydot (`pip install pydot`) and install graphviz (see instructions at https://graphviz.gitlab.io/download/) fo
r plot_model to work.
```

In [41]:
```
disc_out = discriminator([inp[tf.newaxis, ...], gen_output], training=False)
plt.imshow(disc_out[0, ..., -1], vmin=-20, vmax=20, cmap='RdBu_r')
plt.colorbar()
```

Out[41]: `<matplotlib.colorbar.Colorbar at 0x2462fe45a30>`



- Displaying the sample output.

"I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own."

Name: Angelo D. Alvarez **Course:** COE 005 – Machine Learning and Prediction

Section: ECE41S11 **Assignment:** GAN Application

```
In [46]: for example_input, example_target in test_dataset.take(1):
           generate_images(generator, example_input, example_target)
```

- Displaying the final output where the predicted image is generated,

```
In [50]: fit(train_dataset, test_dataset, steps=40000)
```


```
Step: 0k
.............................
```

To conclude this assignment, I say that this program is a failure because as you can see in the predicted image it barely displays the accurate prediction for the image. I don't know what the problem is, but I stop the program before it finishes for the reason that my device is failing drastically so that I stop it to prevent more damages to my device.

"I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own."