

In [1]:

```
from matplotlib import pyplot as plt
import numpy as np
import tensorflow as tf
import sys
```

In [2]:

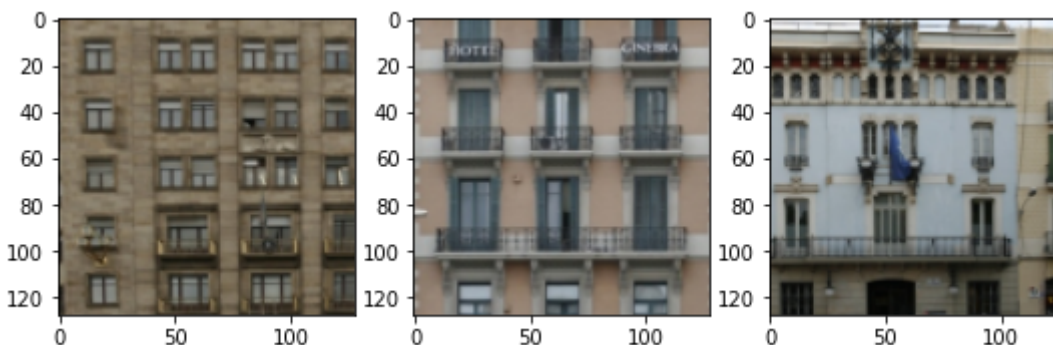
```
from homework6 import *
from framework.dataset import create_dataset, collect_images_from_path
```

In [5]:

```
# Load all images
data_path = "../dataset/train/"
images = collect_images_from_path(data_path)

# Show some examples
fig = plt.figure(figsize=[9,3])
for idx, image in enumerate(images[:3]):
    fig.add_subplot(1,3,idx+1)
    plt.imshow(image)
```

There are 222 images in the dataset.



Task 1 + 6: Try your crop function

In [6]:

```
# Create dataset using your crop function
dataset = create_dataset(
    data_path,
    crop_image, # your crop function
    batch_size=20,
    shuffle=True
)
```

There are 222 images in the dataset.

If your crop function works as expected you should see with the following cell your cropped input and the corresponding groundtruth.

In [7]:

```
# Test your crop image function
for input_image, groundtruth_image in dataset:
    fig = plt.figure(figsize=[5,5])

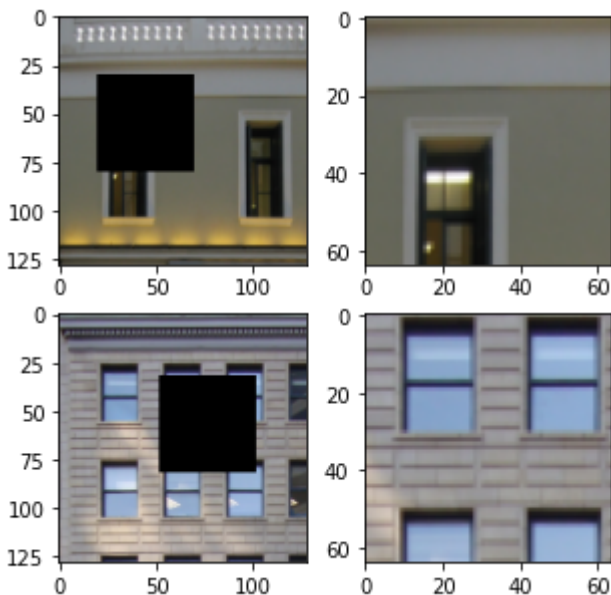
    fig.add_subplot(2, 2, 1)
    plt.imshow(input_image[0,...])

    fig.add_subplot(2, 2, 2)
    plt.imshow(groundtruth_image[0,...])

    fig.add_subplot(2, 2, 3)
    plt.imshow(input_image[1,...])

    fig.add_subplot(2, 2, 4)
    plt.imshow(groundtruth_image[1,...])

    break
```



Task 3: Test your loss function

This is just to test if your loss function runs on tensorflow. This does not cover all the failure case, but if you obtain the same results, it should be good.

In [7]:

```
# Test your loss function (at least for bugs)
for input_image, groundtruth_image in dataset:
    loss_value = reconstruction_loss(groundtruth_image-1, groundtruth_image)
    print(loss_value.numpy())
```

1.1
1.1
1.1
1.1
1.1
1.1
1.1
1.1
1.1
1.1
1.1
1.1

Task 4: Test the reconstruction function

Using the cropped input image, and the corresponding groundtruth, you should be able to obtain the original image

In [6]:

```
# Test your reconstruction function
for input_image, groundtruth_image in dataset:

    # Convert these tensors to numpy
    input_image = input_image.numpy() # size [batch_size, 128, 128, 3]
    groundtruth_image = groundtruth_image.numpy() # size [batch_size, 64, 64, 3]

    fig = plt.figure(figsize=[10,10])
    for i in range(3):
        # Reconstruct some images from the batch, using the groundtruth
        reconstruction = reconstruct_input_image(
            input_image[i,...], groundtruth_image[i,...]
        )

        # Plot the images
        fig.add_subplot(3, 3, 3*i + 1)
        plt.imshow(input_image[i,...])

        fig.add_subplot(3, 3, 3*i + 2)
        plt.imshow(groundtruth_image[i,...])

        fig.add_subplot(3, 3, 3*i + 3)
        plt.imshow(reconstruction)

    break
```

