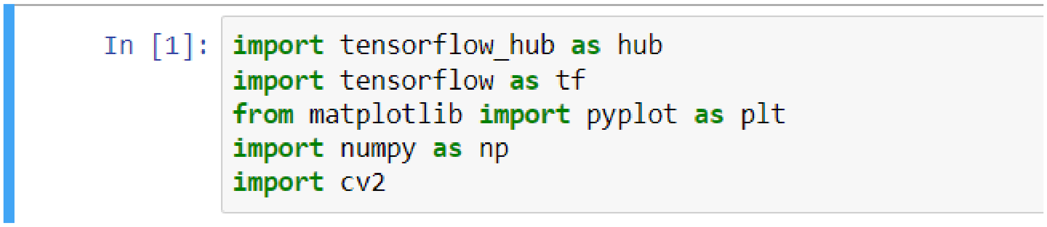
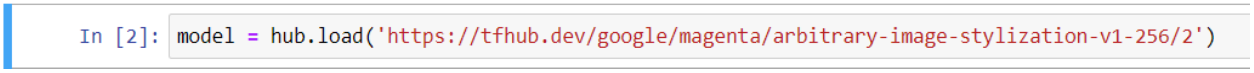
The optimization method of neural style transfer combines two images (content image and style reference image) so that the output image resembles the content image while being "painted" in the manner of the style reference image. In this case, one content image was used to form two stylized images using two different styles from famous painters.



```
In [1]: import tensorflow_hub as hub
        import tensorflow as tf
        from matplotlib import pyplot as plt
        import numpy as np
        import cv2
```

Figure 1: Importing Libraries

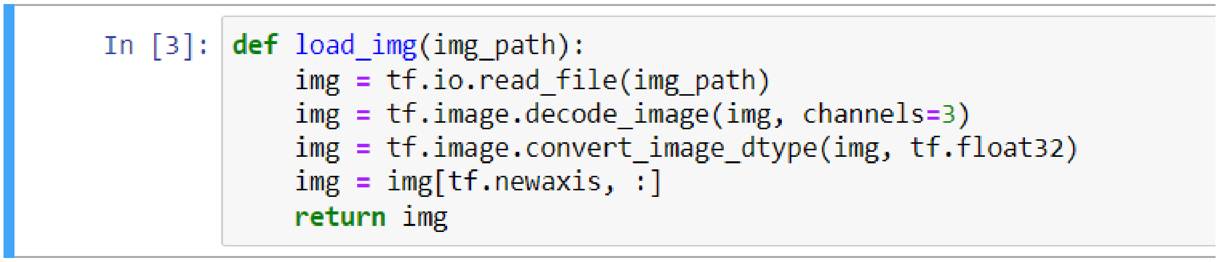
In this line of codes, the libraries shown were used to process the needed input data in order to produce the desired output.



```
In [2]: model = hub.load('https://tfhub.dev/google/magenta/arbitrary-image-stylization-v1-256/2')
```

Figure 2: Importing Model

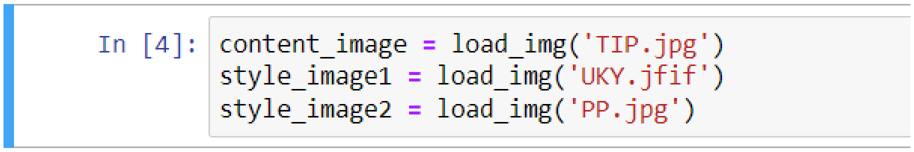
This is a pre-trained neural style transfer model which was imported from TensorFlow Hub. This model will be used as structure model for the codes.



```
In [3]: def load_img(img_path):
            img = tf.io.read_file(img_path)
            img = tf.image.decode_image(img, channels=3)
            img = tf.image.convert_image_dtype(img, tf.float32)
            img = img[tf.newaxis, :]
            return img
```

Figure 3: Preparing the Data

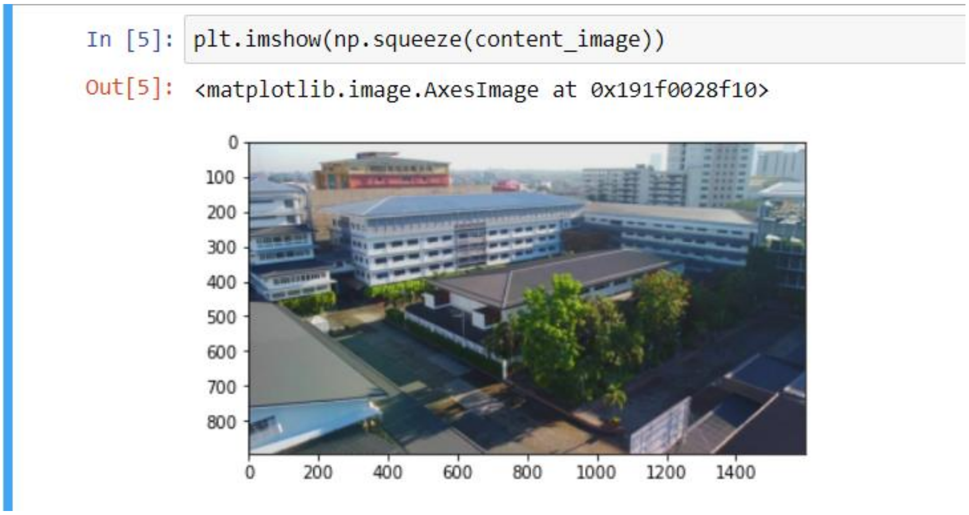
This line of codes shows the preparation of the data. This part shows how the data will be read, decoded, and converted using tensorflow library.



```
In [4]: content_image = load_img('TIP.jpg')
        style_image1 = load_img('UKY.jfif')
        style_image2 = load_img('PP.jpg')
```

Figure 4: Importing the Images

This line of codes shows the storing of three images into a designated variables as an input.



```
In [5]: plt.imshow(np.squeeze(content_image))
Out[5]: <matplotlib.image.AxesImage at 0x191f0028f10>
```

Figure 5: Visualizing the Content Image

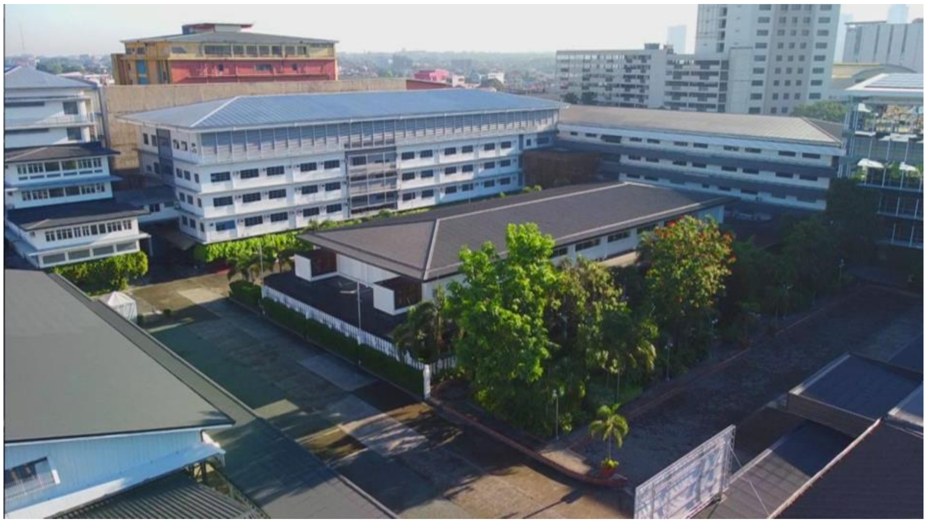
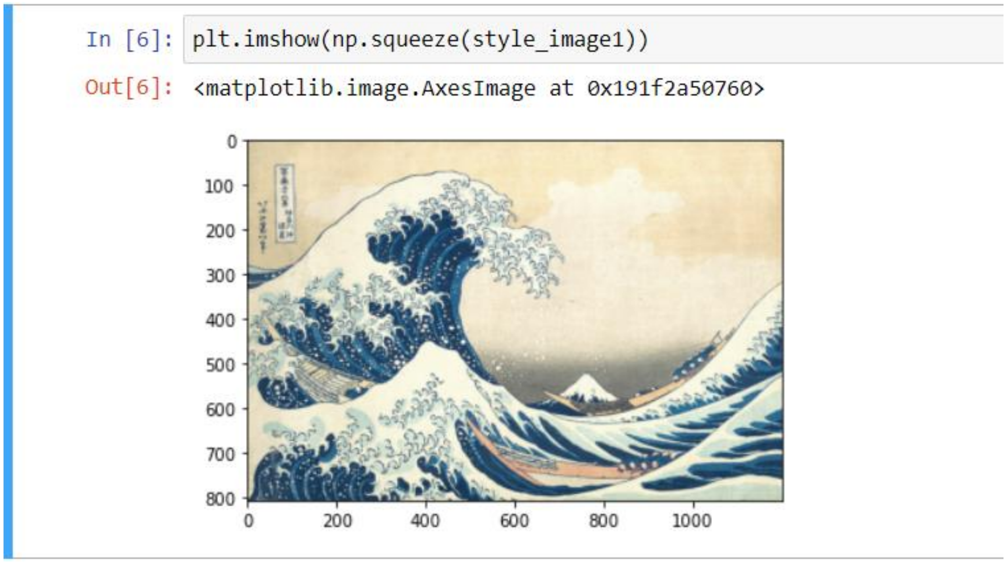


Figure 6: TIP.jpg

This image is a partial aerial view of Technological Institute of the Philippines (TIP) Quezon City. This will be used as the content image for this neural style transfer. This image will be converted into two different styles as an output.



```
In [6]: plt.imshow(np.squeeze(style_image1))

Out[6]: <matplotlib.image.AxesImage at 0x191f2a50760>
```

Figure 7: Visualizing the First Style Image

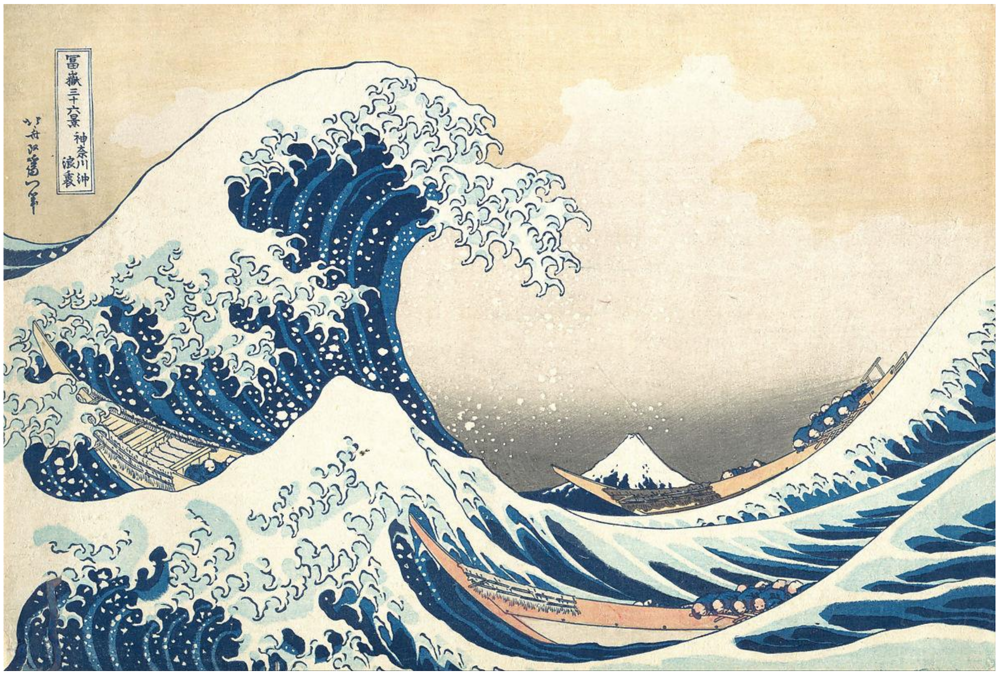
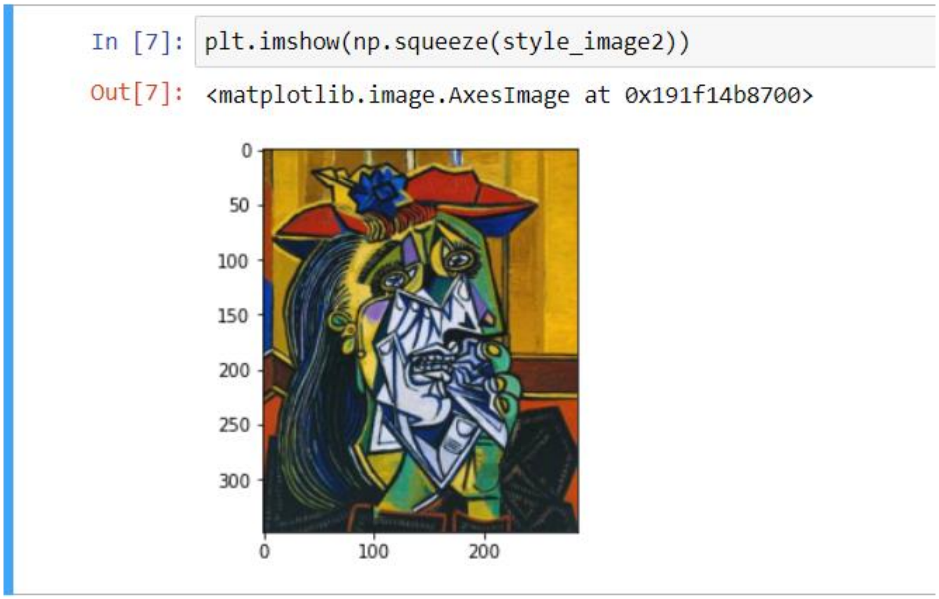


Figure 8: UKY.jfif

This image is an art style called Ukiyo-e. It is a genre of Japanese art which flourished from 17th through 19th centuries. This will be used as the first style reference for the content image and produce the first output image.



```
In [7]: plt.imshow(np.squeeze(style_image2))
Out[7]: <matplotlib.image.AxesImage at 0x191f14b8700>
```

Figure 9: Visualizing the Second Style Image

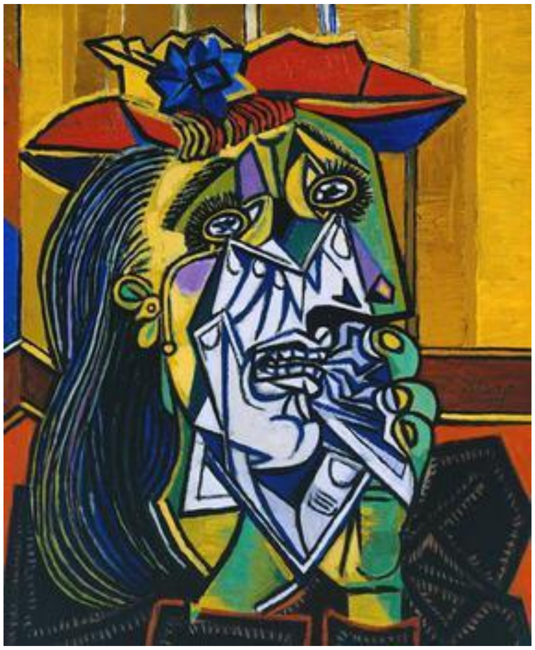
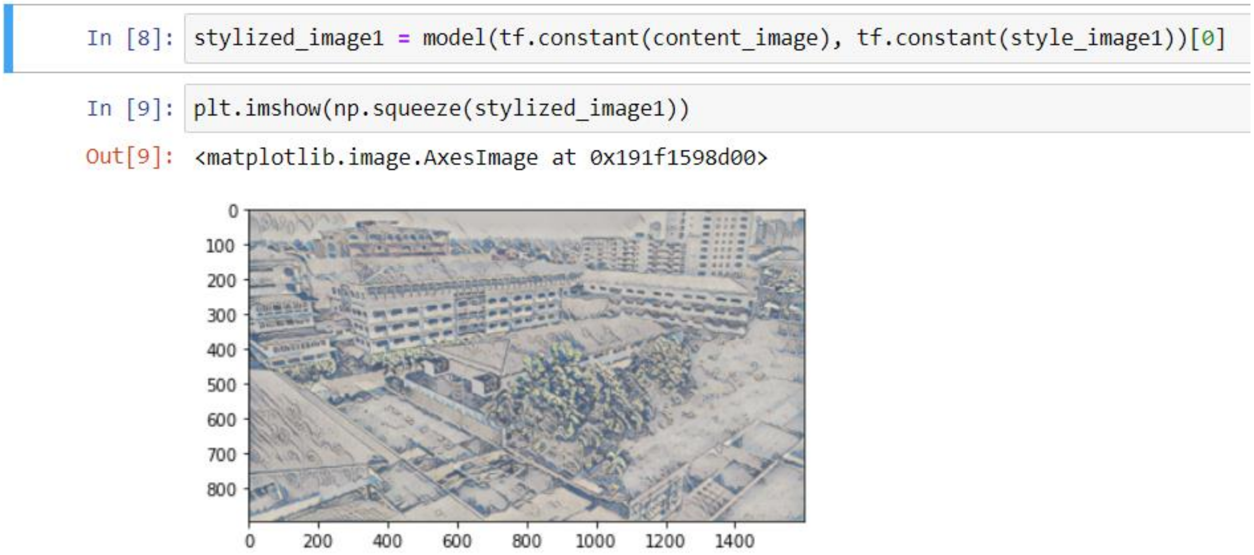


Figure 10: PP.jpg

This image is called “The Weeping Woman”, a masterpiece by Pablo Picasso in 1937. This will be used as the second style reference for the content image and produce the second output image.



```
In [8]: stylized_image1 = model(tf.constant(content_image), tf.constant(style_image1))[0]

In [9]: plt.imshow(np.squeeze(stylized_image1))

Out[9]: <matplotlib.image.AxesImage at 0x191f1598d00>
```

Figure 11: Generating First Output Image

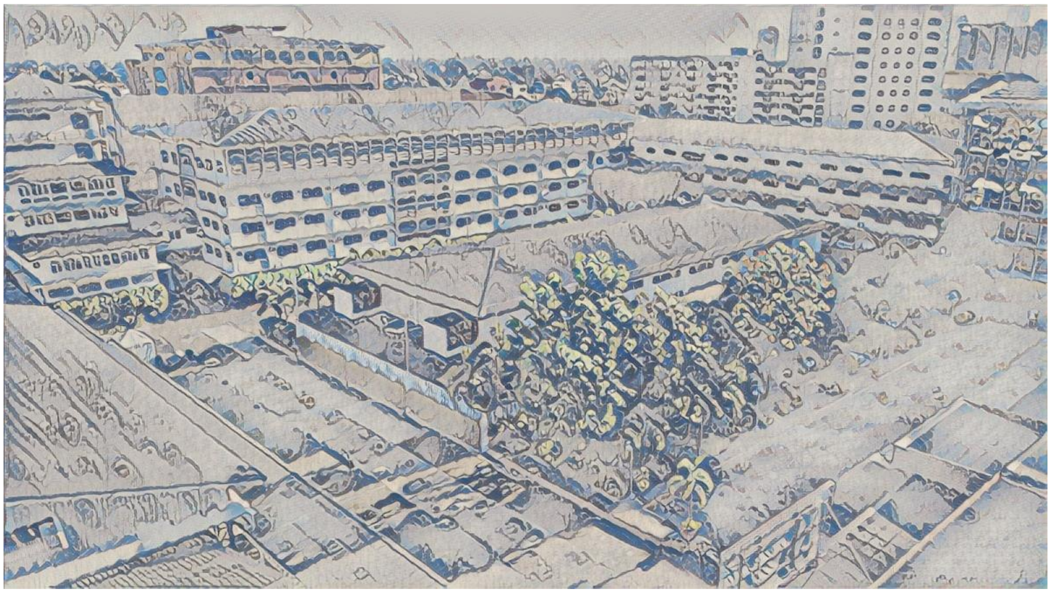
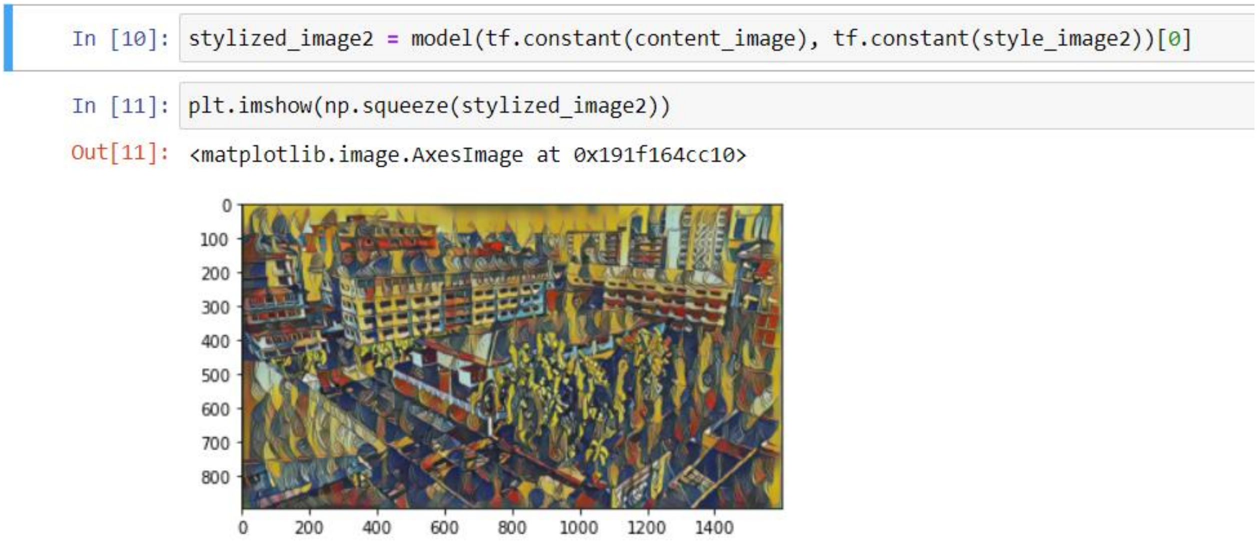


Figure 12: UKY_image

This image was the first generated output image when the aerial view of TIP and Ukiyo-e painting style were merged to form an aerial view of TIP “painted” in Ukiyo-e style.



```
In [10]: stylized_image2 = model(tf.constant(content_image), tf.constant(style_image2))[0]

In [11]: plt.imshow(np.squeeze(stylized_image2))
Out[11]: <matplotlib.image.AxesImage at 0x191f164cc10>
```

Figure 13: Generating Second Output Image

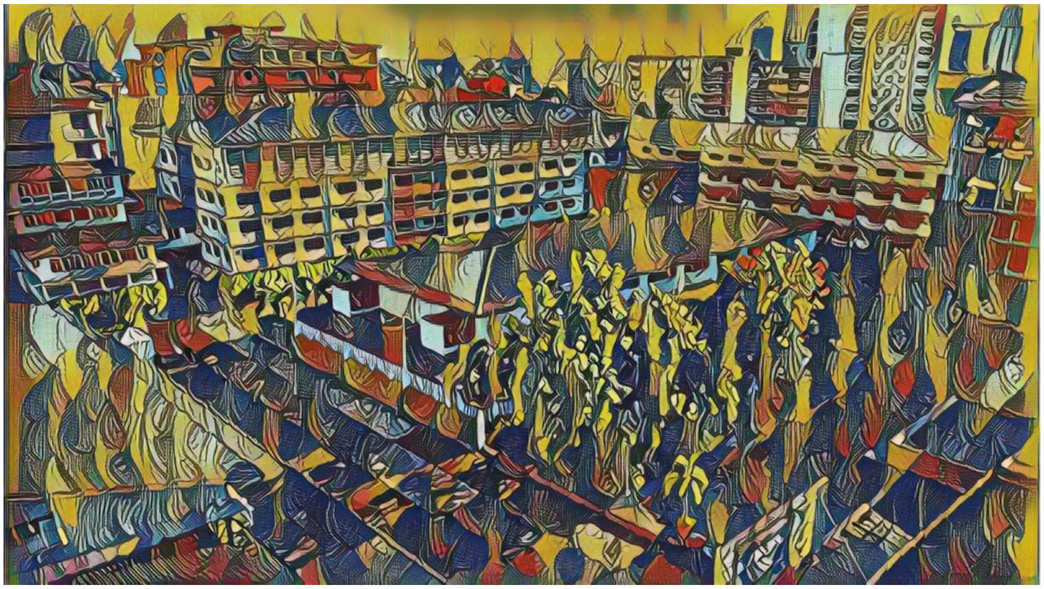
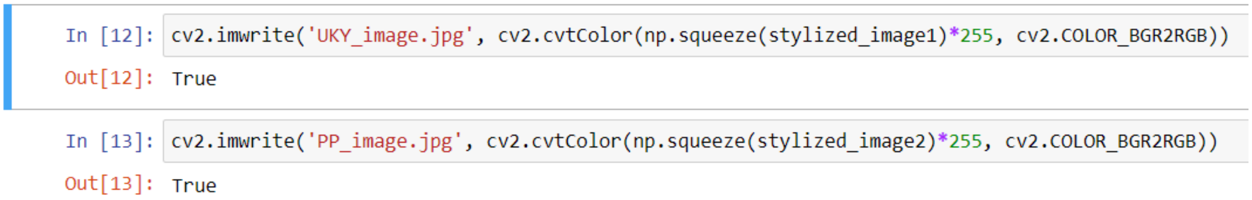


Figure 14: PP_image

This was the second generated output image when the aerial view of TIP and cubism art style were merged to form an aerial view of TIP “painted” by Pablo Picasso.



```
In [12]: cv2.imwrite('UKY_image.jpg', cv2.cvtColor(np.squeeze(stylized_image1)*255, cv2.COLOR_BGR2RGB))
Out[12]: True

In [13]: cv2.imwrite('PP_image.jpg', cv2.cvtColor(np.squeeze(stylized_image2)*255, cv2.COLOR_BGR2RGB))
Out[13]: True
```

Figure 15: Storing Output Images

These lines of codes were used to export the output images and save them as image files.

In this activity, we were tasked to find a style transfer code based on CNN architectures. We chose one from two photos in which we will apply the style transferring code. I chose Photo A, since the natural lighting seems more pleasant. We also chose two from the given list of styles in which we will use as style references for the content image from the photo chosen. I chose Ukiyo-e and Pablo Picasso’s art styles. Photo A, which was the content image, is a partial aerial view of Technological Institute of the Philippines, Quezon City, shown in Figure 6. The first reference style image was a painting from Ukiyo-e style, a genre of Japanese art which flourished from 17th through 19th centuries, shown in Figure 8. The second reference style image was called “The Weeping Woman”, a masterpiece by Pablo Picasso, shown in Figure 10. The first generated output image was shown in Figure 12, while the second is shown in figure 14. Those images were the output images when the content image was merged with different reference style images.