

In [2]:

```
1 from keras.preprocessing.image import load_img, img_to_array
2 # from tensorflow.keras.preprocessing.image import load_img, img_to_array
3
4 # This is the path to the image you want to transform.
5 #target_image_path = '/home/ubuntu/data/portrait.png'
6 # This is the path to the style image.
7 #style_reference_image_path = '/home/ubuntu/data/popova.jpg'
8
9 #jf
10 target_image_path = '/home/jefferson/dataset/photos/GoldenGate.jpg'
11 #target_image_path = '/home/jefferson/dataset/photos/Jefferson.jpg'
12 #target_image_path = '/home/jefferson/dataset/photos/Sherry.jpg'
13
14 #style_reference_image_path = '/home/jefferson/dataset/photos/Picasso.jpg'
15 style_reference_image_path = '/home/jefferson/dataset/photos/VanGaugh.jpg'
16
17 # Dimensions of the generated picture.
18 width, height = load_img(target_image_path).size
19 img_height = 400
20 #jf img_height = 100
21 img_width = int(width * img_height / height)
22
23 #jf
24 import matplotlib.pyplot as plt
25
26 # Content image
27 plt.imshow(load_img(target_image_path, target_size=(img_height, img_width)))
28 plt.figure()
29
30 # Style image
31 plt.imshow(load_img(style_reference_image_path, target_size=(img_height, img_width)))
32 plt.figure()
33
34 plt.show()
```

<Figure size 640x480 with 1 Axes>

<Figure size 640x480 with 1 Axes>

<Figure size 640x480 with 0 Axes>

We will need some auxiliary functions for loading, pre-processing and post-processing the images that will go in and out of the VGG19 convnet: