Pic_vision

In [1]:
```
#pip install opencv-python==3.4.2
#pip install opencv-contrib-python==3.3.1

#pip3 install opencv-python
#pip install opencv-contrib-python
```

In [2]:
```
import cv2
cv2.__version__
```

Out[2]: '4.2.0'

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [4]:
```
# Import the image
img = cv2.imread('burano.jpg')
plt.imshow(img)
```

Out[4]: <matplotlib.image.AxesImage at 0x1de46da6128>

In [5]:
```
# Convert the image(BGR) into RGB
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img_rgb)
```

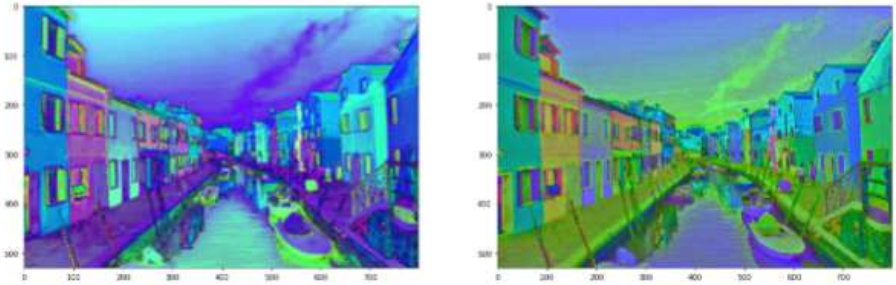
Out[5]: <matplotlib.image.AxesImage at 0x1de46e21940>

In [6]:
```
# Convert the image into gray scale
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(img_gray, cmap = 'gray')
```

Out[6]: <matplotlib.image.AxesImage at 0x1de46e68588>

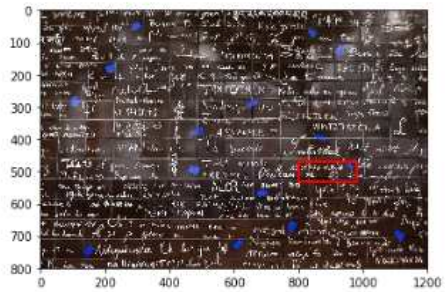
In [7]:
```
fig, axs = plt.subplots(nrows = 1, ncols = 3, figsize = (20, 20))
for i in range(0, 3):
    ax = axs[i]
    ax.imshow(img_rgb[:, :, i], cmap = 'gray')
plt.show()
```

In [8]:
```
# Transform the image into HSV and HLS models
img_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
img_hls = cv2.cvtColor(img, cv2.COLOR_BGR2HLS)
# Plot the converted images
fig, (ax1, ax2) = plt.subplots(nrows = 1, ncols = 2, figsize = (20, 20))
ax1.imshow(img_hsv)
ax2.imshow(img_hls)
plt.show()
```



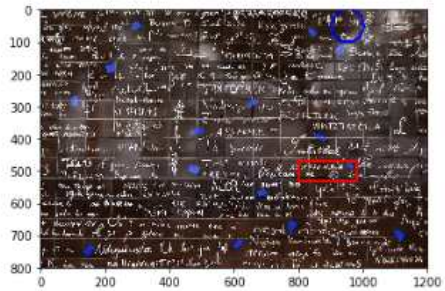
In [9]:
```
# Import the image
img2 = cv2.imread('thewallofiove.jpg')
plt.imshow(img2)
# Copy the image
img_copy = img2.copy()
# Draw a rectangle
cv2.rectangle(img_copy, pt1 = (800, 470), pt2 = (980, 530),
              color = (255, 0, 0), thickness = 5)
plt.imshow(img_copy)
```

Out[9]:
```
<matplotlib.image.AxesImage at 0x1de4a9ddf98>
```



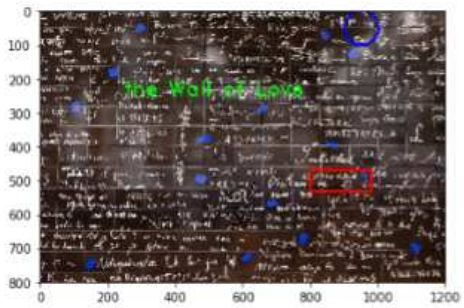
In [10]:
```
# Draw a circle
cv2.circle(img_copy, center = (950, 50), radius = 50,
           color = (0, 0, 255), thickness = 5)
plt.imshow(img_copy)
```

Out[10]:
```
<matplotlib.image.AxesImage at 0x1de4abe2438>
```



In [11]:
```
# Add text
cv2.putText(img_copy, text = "the Wall of Love",
            org = (250, 250),
            fontFace = cv2.FONT_HERSHEY_DUPLEX,
            fontScale = 2,
            color = (0, 255, 0),
            thickness = 2,
            lineType = cv2.LINE_AA)
plt.imshow(img_copy)
```

Out[11]:
```
<matplotlib.image.AxesImage at 0x1de4acleda0>
```



In [12]:
```
# Step 1. Define callback function
def draw_circle(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
            cv2.circle(img, center = (x, y), radius = 5,
                       color = (87, 184, 237), thickness = -1)
    elif event == cv2.EVENT_RBUTTONDOWN:
            cv2.circle(img, center = (x, y), radius = 10,
                       color = (87, 184, 237), thickness = 1)
```

In [13]:
```
# Step 2. Call the window
img = cv2.imread('map.jpg')
cv2.namedWindow(winname = 'my_drawing')
cv2.setMouseCallback('my_drawing', draw_circle)
```

In [14]:
```
# Step 3. Execution
while True:
    cv2.imshow('my_drawing',img)
    if cv2.waitKey(10) & 0xFF == 27:
        break
cv2.destroyAllWindows()
```

In [15]:
```python
# Step 1. Define callback function
drawing = False
ix = -1
iy = -1

def draw_rectangle(event, x, y, flags, params):

    global ix, iy, drawing

    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        ix, iy = x, y

    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing == True:
            cv2.rectangle(img, pt1 = (ix, iy), pt2 = (x, y),
                          color = (87, 184, 237), thickness = -1)

    elif event == cv2.EVENT_LBUTTONUP:
        drawing = False
        cv2.rectangle(img, pt1 = (ix, iy), pt2 = (x, y),
                      color = (87, 184, 237), thickness = -1)


# Step 2. Call the window
img = cv2.imread('map.jpg')

cv2.namedWindow(winname = 'my_drawing')
cv2.setMouseCallback('my_drawing', draw_rectangle)


# Step 3. Execution
while True:
    cv2.imshow('my_drawing', img)
    if cv2.waitKey(10) & 0xFF == 27:
        break

cv2.destroyAllWindows()
```