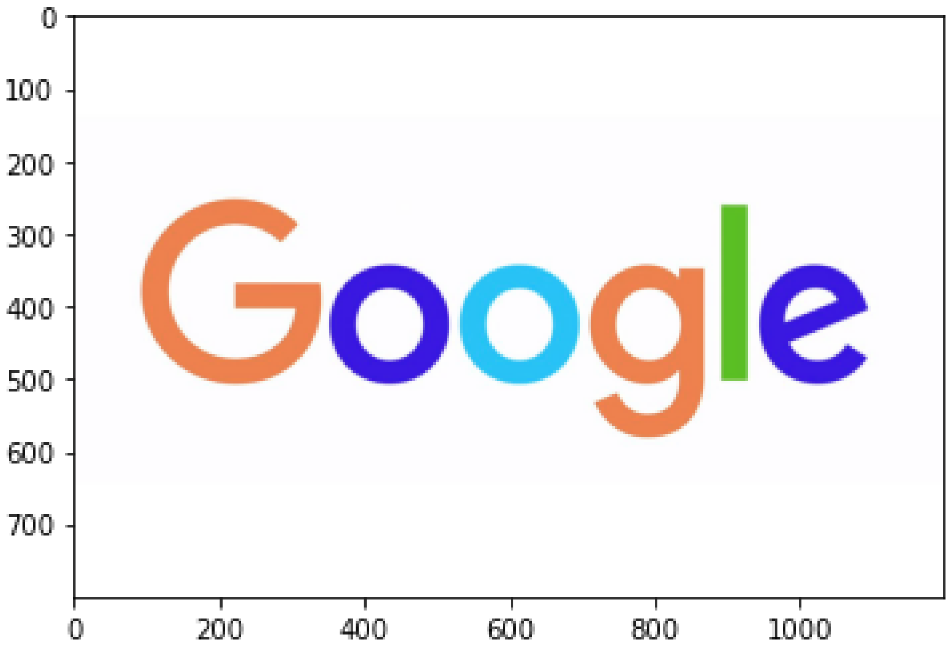
chap13-analyzingImageData

July 12, 2022

[5]:

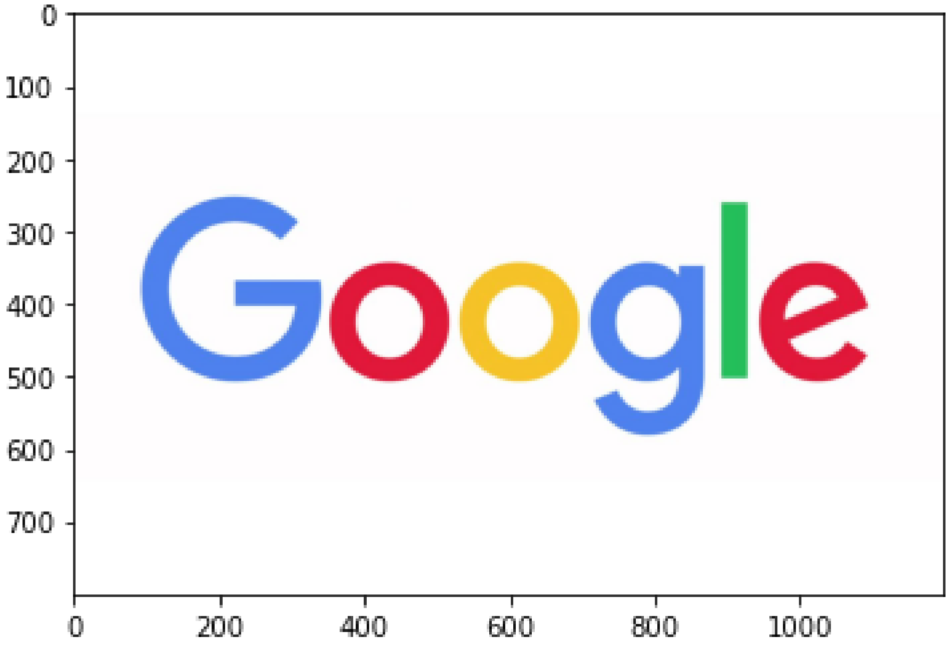
```python
## analyzing image data

# import cv2 latest version of OpenCV library
import cv2

# import numeric python (NumPy) library
import numpy as np

# import matplotlib for showing the image
import matplotlib.pyplot as plt

# magic function to render the figure in a notebook
%matplotlib inline

# read image using imread() function
image = cv2.imread('google.jpg')

# let's check image data type
print('Image Type: ', type(image))

# let's check dimension of image
print('Image Dimension: ', image.shape)

# let's show the image
plt.imshow(image)
plt.show()
```

```
Image Type:  <class 'numpy.ndarray'>
Image Dimension:  (800, 1200, 3)
```



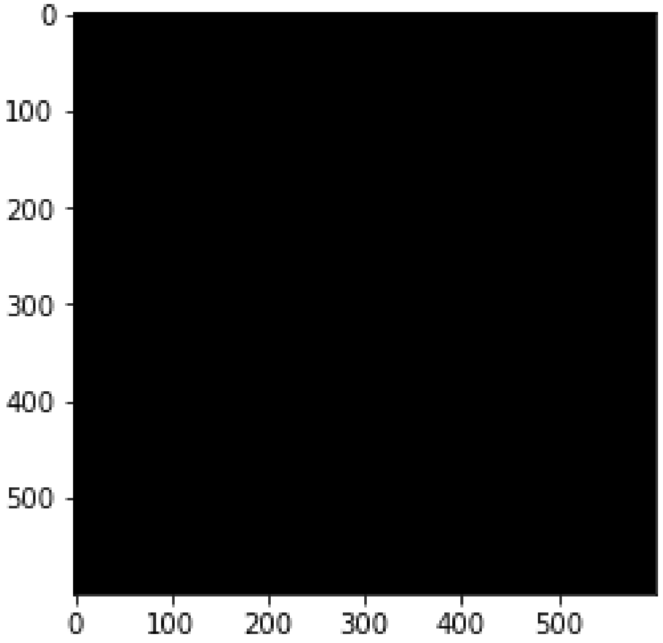
[6]:

```
# convert image color space BGR to RGB
rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# display the image
plt.imshow(rgb_image)
plt.show()
```



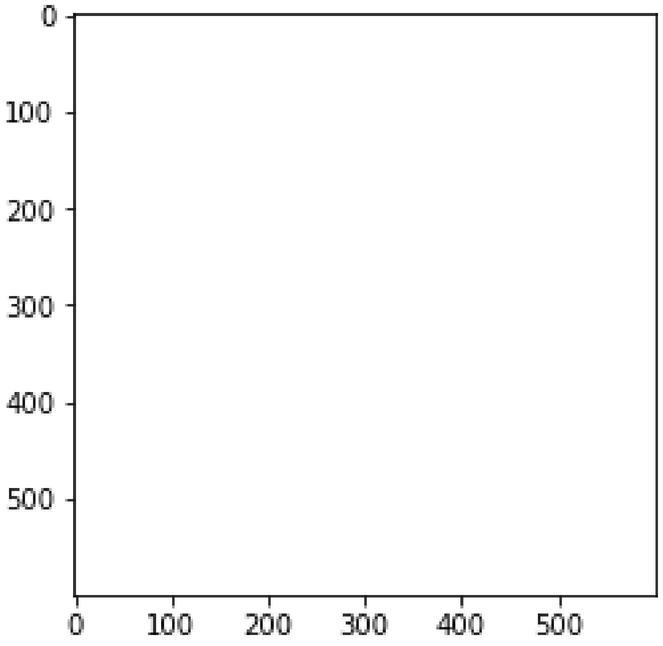
```
[14]: # let's create a black image
      image_shape = (600, 600, 3)
      black_image = np.zeros(shape=image_shape, dtype=np.int16)

      # show the image
      plt.imshow(black_image)
```

```
[14]: <matplotlib.image.AxesImage at 0x2e1fe72e460>
```



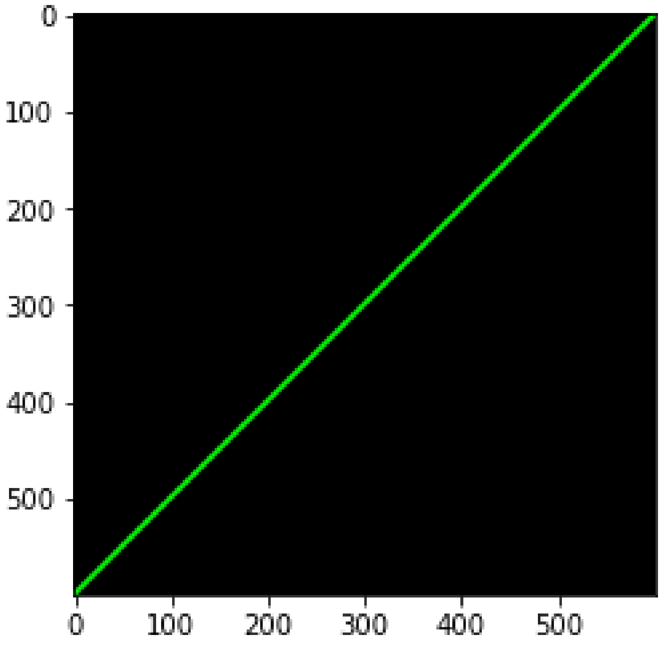
```
[15]: # create a white image
      white_image = np.zeros(shape=image_shape, dtype=np.int16)

      # set every pixel of the image to 255
      white_image.fill(255)

      # show the image
      plt.imshow(white_image)
```

```
[15]: <matplotlib.image.AxesImage at 0x2e1fe75c760>
```



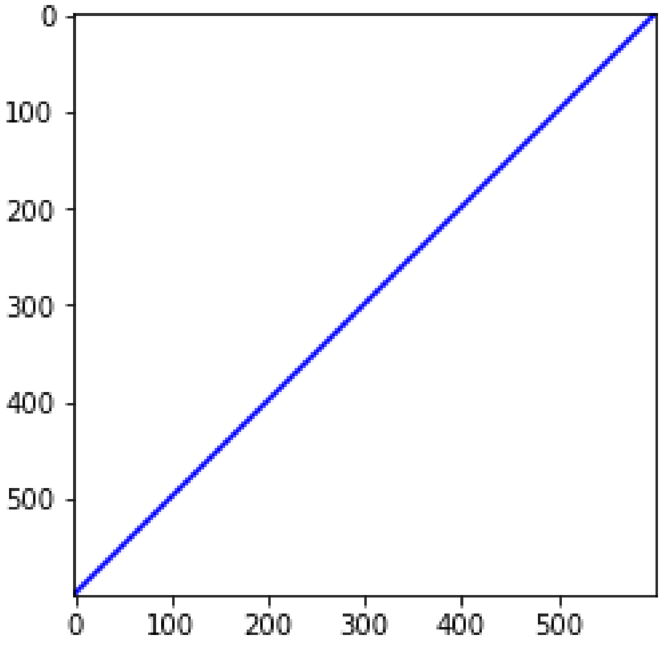
[16]:
```
# draw a line on black image
line = cv2.line(black_image, (599,0), (0,599), (0,255,0), 4)

# show image
plt.imshow(line)
```

[16]:
```
<matplotlib.image.AxesImage at 0x2e180120e20>
```



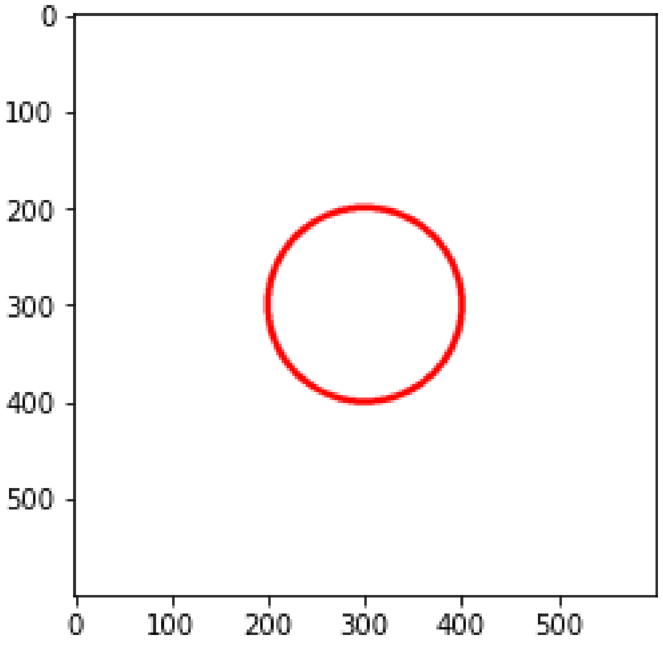
[17]:
```
# let's draw a blue line on white image
line = cv2.line(white_image, (599,0), (0,599), (0,0,255), 4)

# show the image
plt.imshow(line)
```

[17]:
```
<matplotlib.image.AxesImage at 0x2e180186730>
```



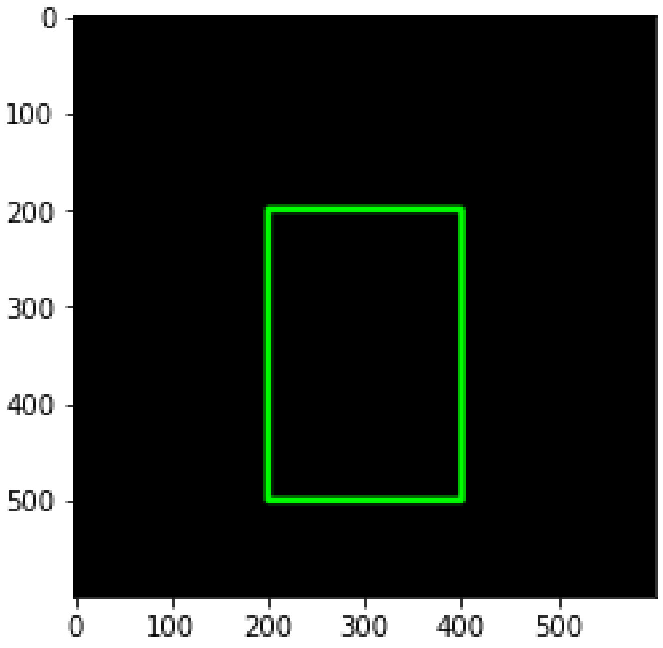
```
[18]: # let's create a white image
      img_shape = (600, 600, 3)
      white_image = np.zeros(shape=image_shape, dtype=np.int16)

      # set every pixel of the image to 255
      white_image.fill(255)

      # draw a red circle on white image
      circle = cv2.circle(white_image, (300, 300), 100, (255, 0, 0), 6)

      # show the image
      plt.imshow(circle)
```

```
[18]: <matplotlib.image.AxesImage at 0x2e180311a30>
```



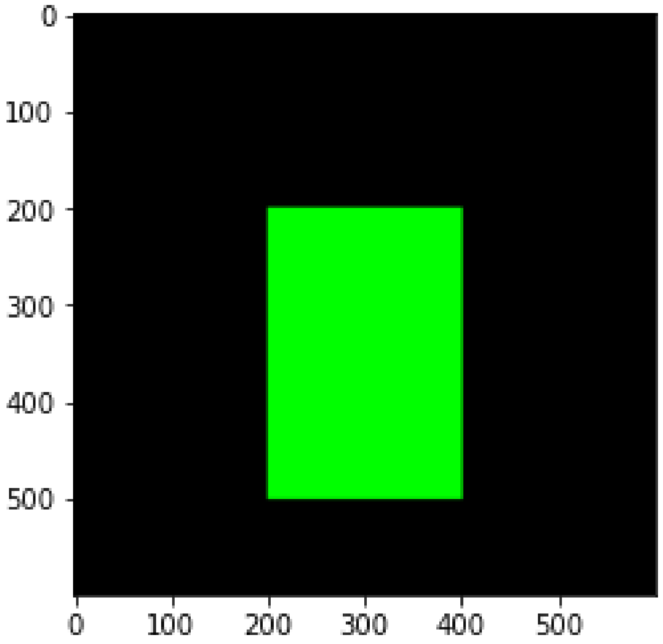
[19]:
```
# let's create a black image
img_shape = (600, 600, 3)
black_image = np.zeros(shape=image_shape, dtype=np.int16)

# draw a green rectangle on black image
rectangle = cv2.rectangle(black_image, (200, 200), (400, 500), (0, 255, 0), 5)

# show the image
plt.imshow(rectangle)
```

[19]:
```
<matplotlib.image.AxesImage at 0x2e1803749d0>
```



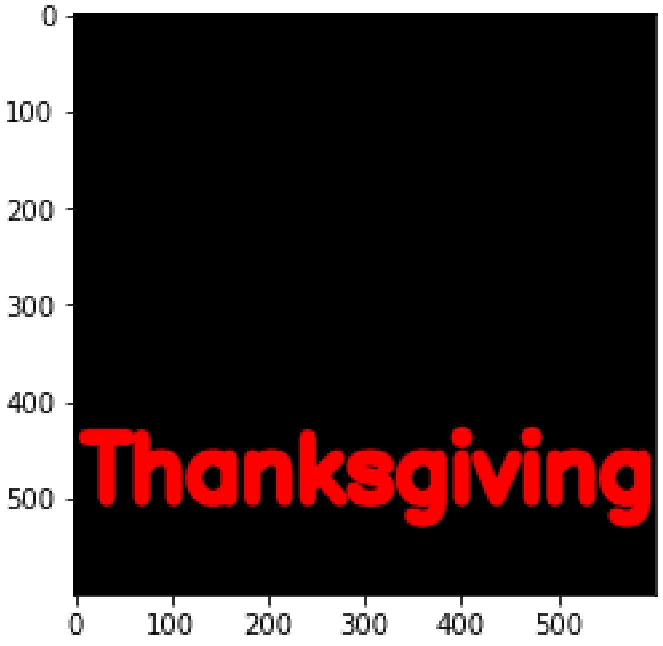
```
[20]: # let's create a black image
      img_shape = (600, 600, 3)
      black_image = np.zeros(shape=image_shape, dtype=np.int16)

      # draw a green filled rectangle on black image
      rectangle = cv2.rectangle(black_image, (200, 200), (400, 500), (0, 255, 0), -1)

      # show the image
      plt.imshow(rectangle)
```

```
[20]: <matplotlib.image.AxesImage at 0x2e1806d59d0>
```



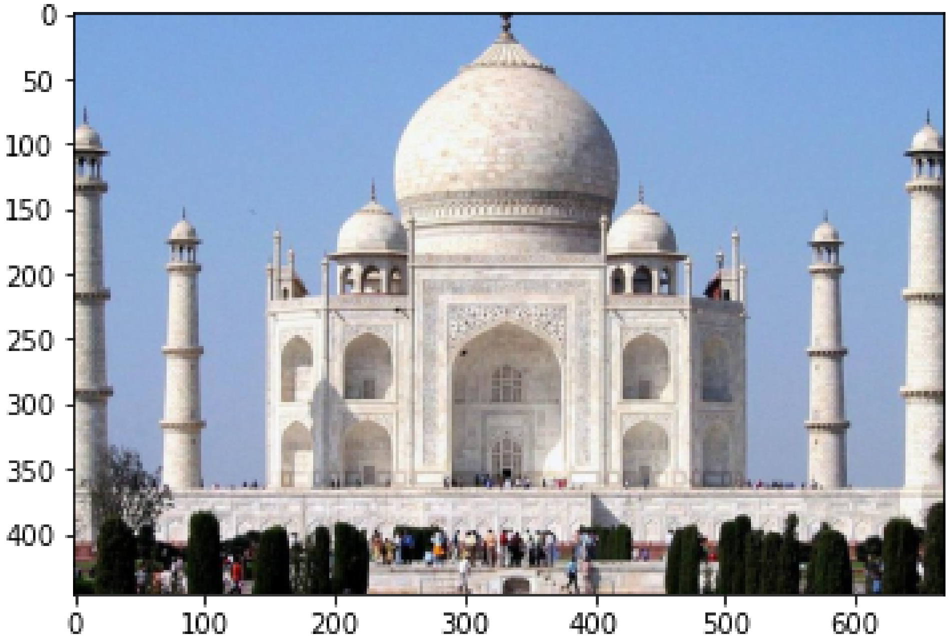
```
[23]: ## writing on images

      # let's create a black image
      black_image = np.zeros(shape=image_shape, dtype=np.int16)

      # write on black image
      text = cv2.putText(black_image, 'Thanksgiving', (10, 500),
                         cv2.FONT_HERSHEY_SIMPLEX, 3, (255, 0, 0), cv2.LINE_AA)
      # display the image
      plt.imshow(text)
```

```
[23]: <matplotlib.image.AxesImage at 0x2e180b2fee0>
```



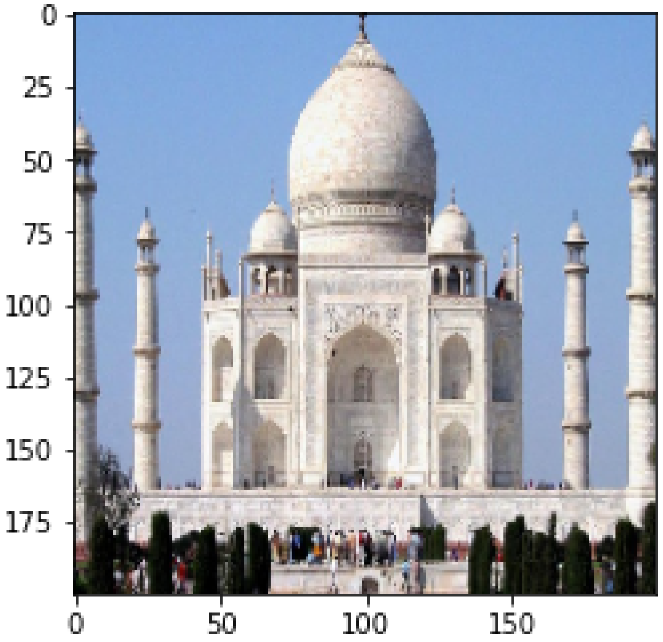
[24]:
```
# resizing images

# import cv2 module
import cv2

# import matplotlib for showing the image
import matplotlib.pyplot as plt

# magic function to render the figure in a notebook
%matplotlib inline

# read image
image = cv2.imread('tajmahal.jpg')

# convert image color space BGR to RGB
rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# display the image
plt.imshow(rgb_image)
```

[24]:
```
<matplotlib.image.AxesImage at 0x2e180b867c0>
```



[25]:
```
# resize the image
image_resized = cv2.resize(rgb_image, (200, 200))
interpolation = cv2.INTER_NEAREST

# display the image
plt.imshow(image_resized)
```

[25]:
```
<matplotlib.image.AxesImage at 0x2e180b4a070>
```



```
[ ]: # flipping images, changing the brightness, blurring an image - book
     # face detection - book
```