Images as Numerical Data

April 26, 2018

1 Images as Grids of Pixels

1.0.1 Import resources

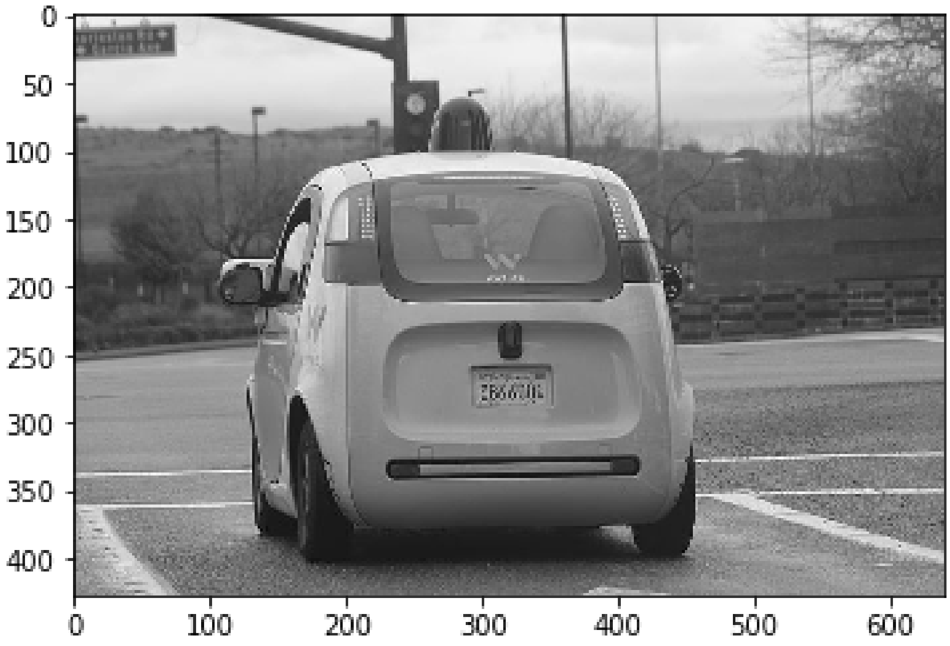
```
In [1]: import numpy as np
        import matplotlib.image as mpimg  # for reading in images

        import matplotlib.pyplot as plt
        import cv2  # computer vision library

        %matplotlib inline
```

1.0.2 Read in and display the image

```
In [2]: # Read in the image
        image = mpimg.imread('images/waymo_car.jpg')

        # Print out the image dimensions
        print('Image dimensions:', image.shape)

        # Change from color to grayscale
        gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

        plt.imshow(gray_image, cmap='gray')
```

```
Image dimensions: (427, 640, 3)
```

```
Out[2]: <matplotlib.image.AxesImage at 0x7fc599667438>
```



```
In [3]: # Print specific grayscale pixel values
        # What is the pixel value at x = 400 and y = 300 (on the body of the car)?

        x = 400
        y = 300

        print(gray_image[y,x])
```

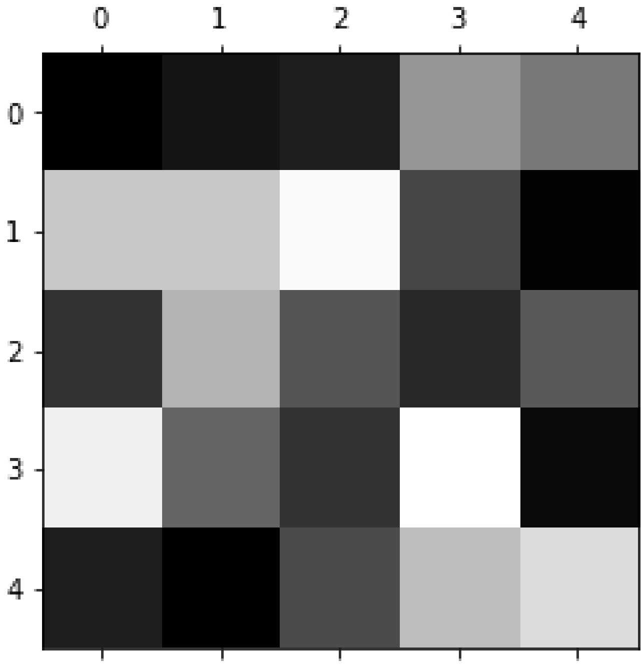
```
159
```

```
In [4]: #Find the maximum and minimum grayscale values in this image

        max_val = np.amax(gray_image)
        min_val = np.amin(gray_image)

        print('Max: ', max_val)
        print('Min: ', min_val)
```

```
Max:  255
Min:  2
```

```
In [5]: # Create a 5x5 image using just grayscale, numerical values
        tiny_image = np.array([[0, 20, 30, 150, 120],
                              [200, 200, 250, 70, 3],
                      [50, 180, 85, 40, 90],
                      [240, 100, 50, 255, 10],
                      [30, 0, 75, 190, 220]])

# To show the pixel grid, use matshow
plt.matshow(tiny_image, cmap='gray')

## TODO: See if you can draw a tiny smiley face or something else!
```

```
Out[5]: <matplotlib.image.AxesImage at 0x7fc5985927b8>
```


0
1
2
3
4
0
1
2
3
4