

# A Taste of Image Processing-Live

October 31, 2018

## 1 Processing Images

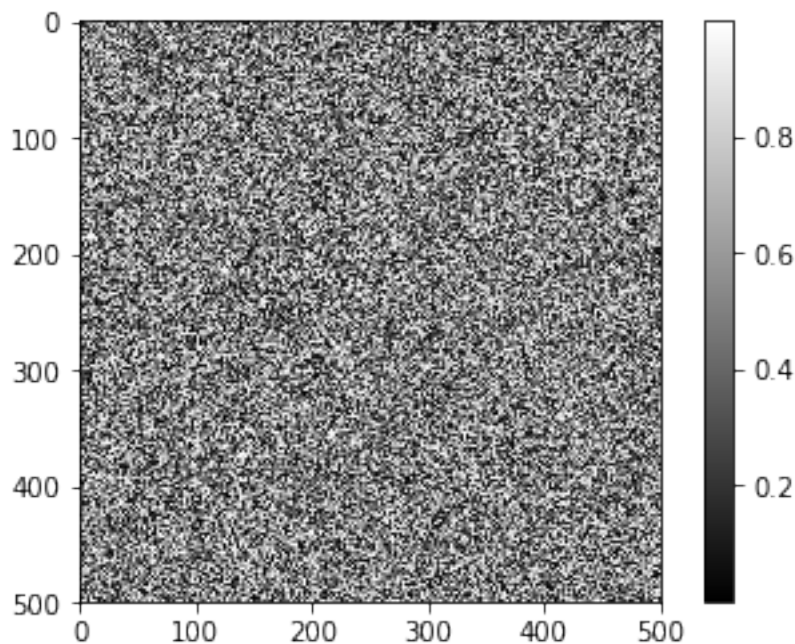
Several libraries you could use

- numpy (really basic stuff)
- pillow (formally PIL this has lots of image formats)
- opencv (Computer vision library wrapping C/C++)
- scikit-image (simpler and less features than opencv ... focused on image processing)

```
In [2]: import skimage
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
```

```
In [3]: random_image = np.random.random((500,500))
plt.imshow(random_image, cmap='gray')
plt.colorbar()
```

```
Out [3]: <matplotlib.colorbar.Colorbar at 0x7f55ed769a20>
```



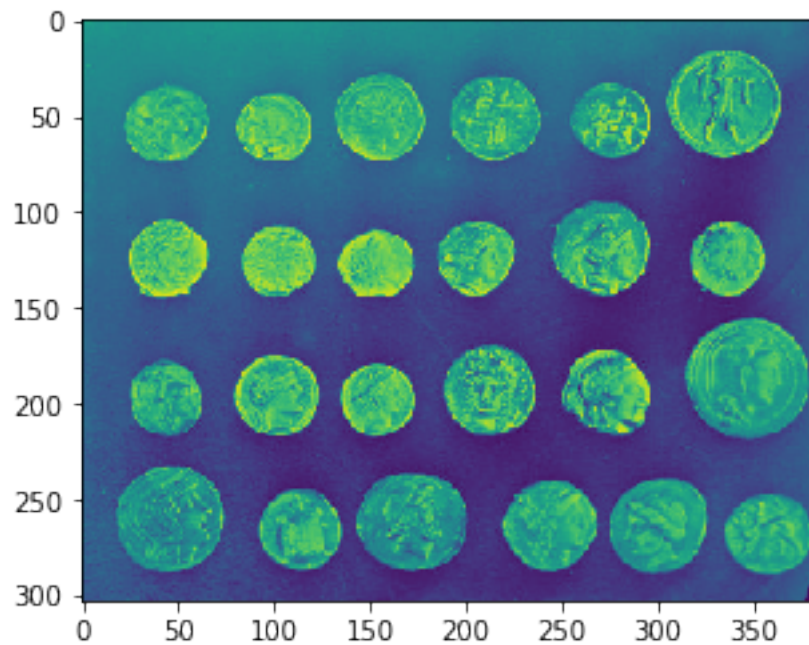
```
In [4]: from skimage import data
        coins = data.coins()

        print("Type: ", type(coins))
        print('dtype', coins.dtype)
        print('shape', coins.shape)
```

```
Type: <class 'numpy.ndarray'>
dtype uint8
shape (303, 384)
```

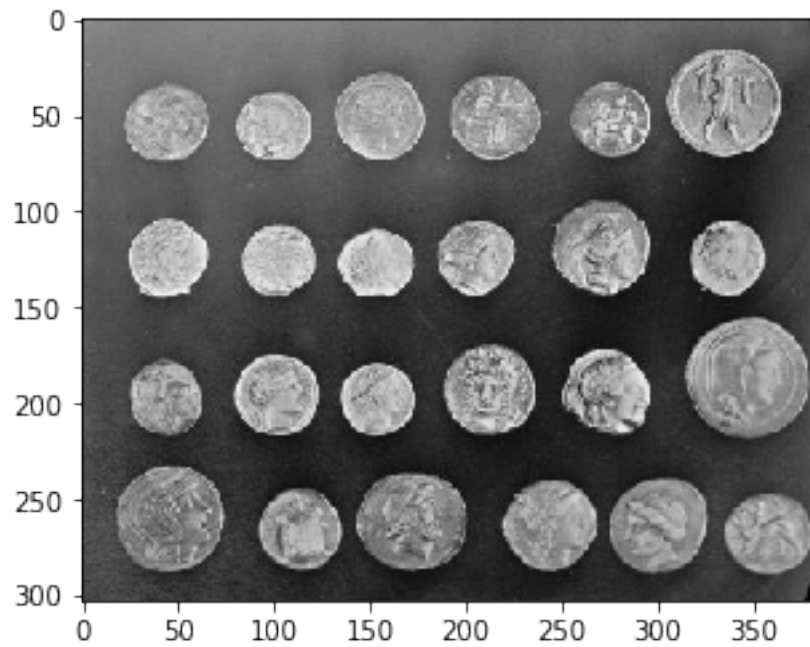
```
In [5]: plt.imshow(coins)
```

```
Out[5]: <matplotlib.image.AxesImage at 0x7f55ed61be80>
```



```
In [6]: plt.imshow(coins, cmap='gray')
```

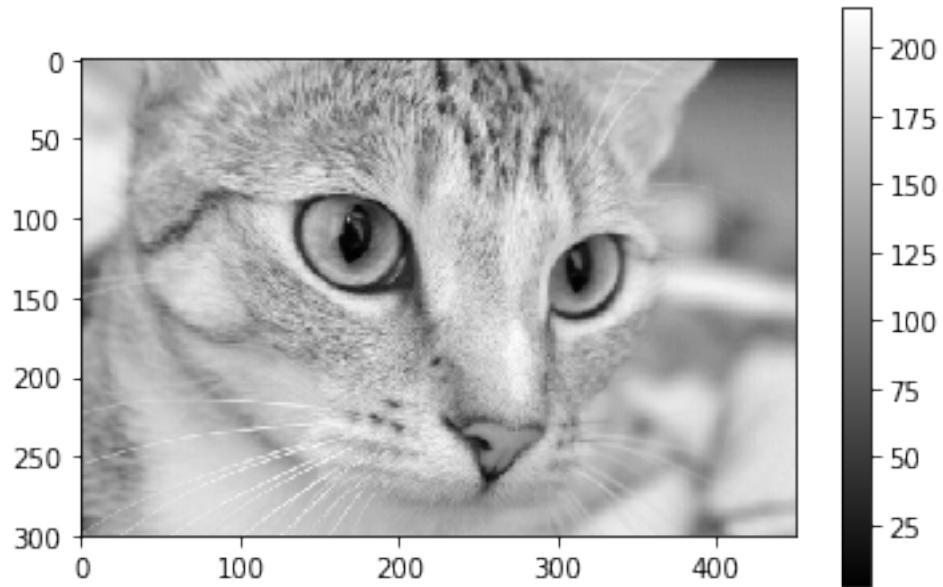
```
Out[6]: <matplotlib.image.AxesImage at 0x7f55ed604588>
```



```
In [8]: cat = data.chelsea()
        print(cat.shape)
        print(cat.max(),cat.min())
        plt.imshow(cat[:,:,:0],cmap='gray')
        plt.colorbar()
```

```
(300, 451, 3)
231 0
```

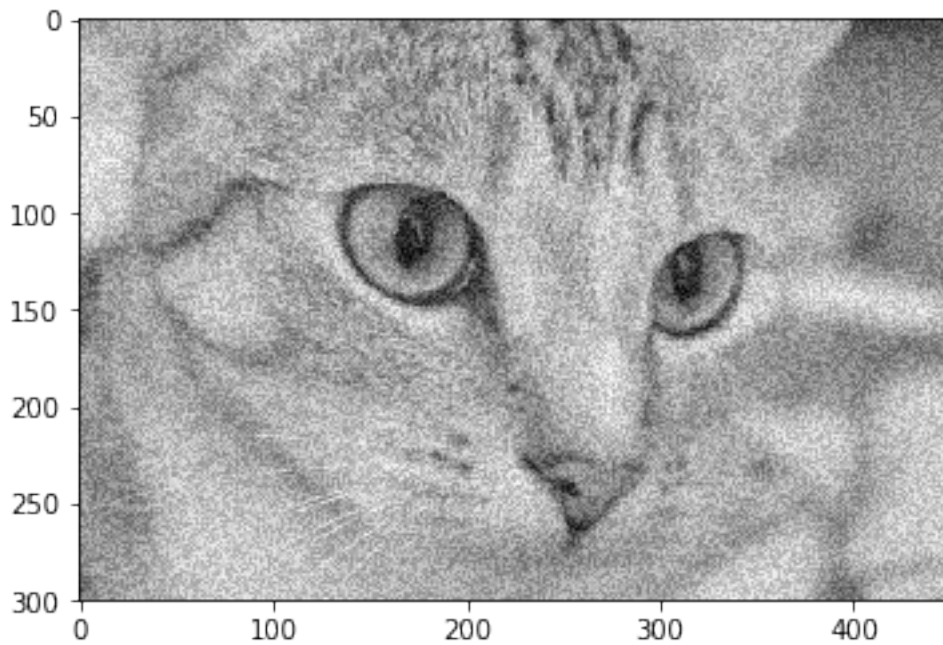
```
Out[8]: <matplotlib.colorbar.Colorbar at 0x7f55ec078588>
```



```
In [12]: red_cat = cat[:, :, 0]/255.0
```

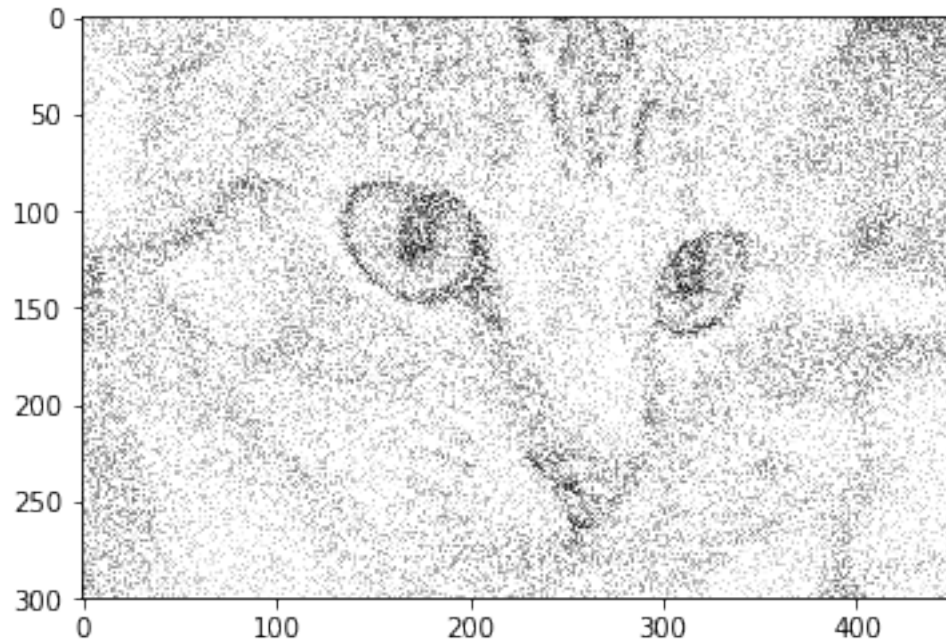
```
In [14]: red_cat_shape = red_cat.shape  
noise = np.random.random(red_cat_shape)*(0.3)  
  
plt.imshow(red_cat+noise, cmap='gray')
```

```
Out[14]: <matplotlib.image.AxesImage at 0x7f55ddecbdd8>
```



```
In [21]: plt.imshow(red_cat+noise*3,cmap='gray',vmin=0,vmax=1)
```

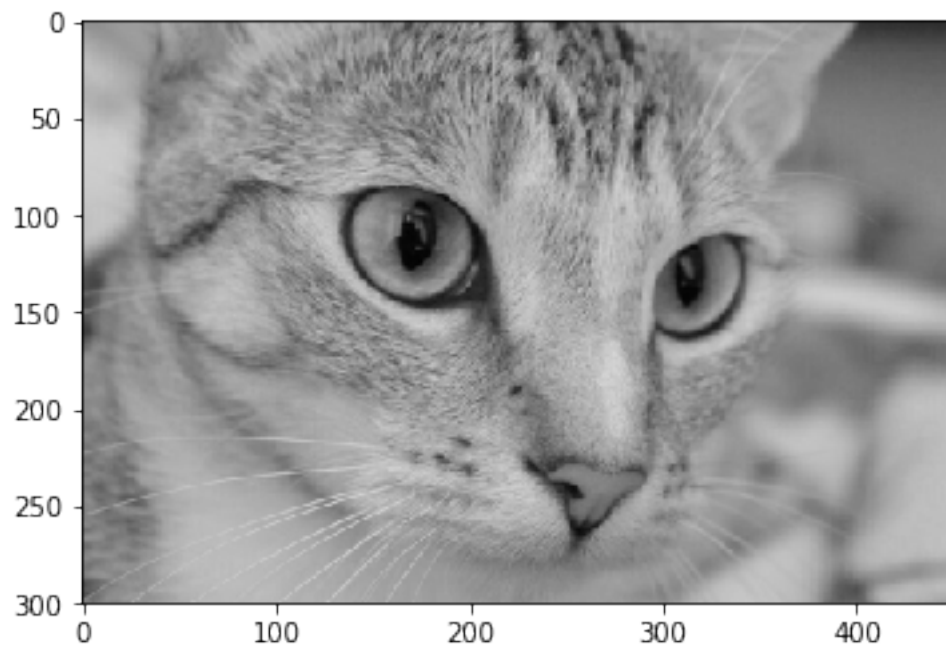
```
Out[21]: <matplotlib.image.AxesImage at 0x7f55ddeb7ef0>
```



```
In [22]: plt.imshow(red_cat,cmap='gray',vmin=0,vmax=1)
```

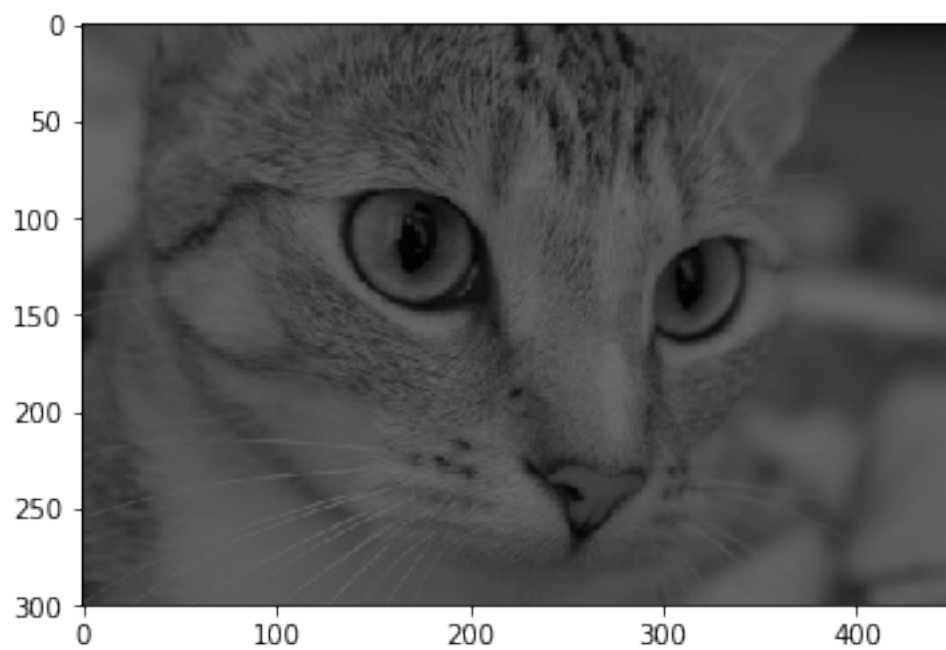
```
Out[22]: <matplotlib.image.AxesImage at 0x7f55dde3e588>
```





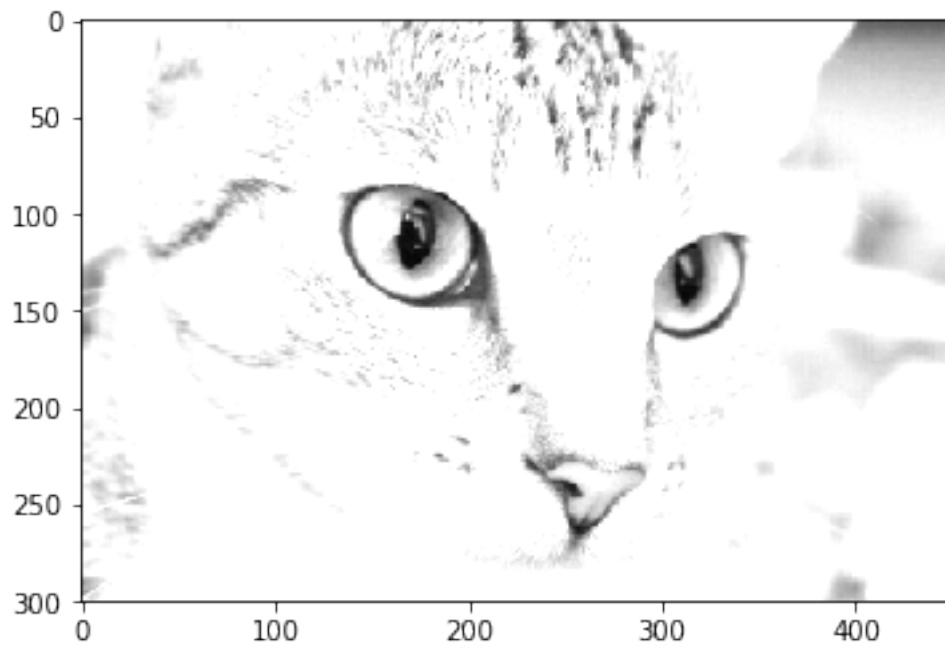
```
In [24]: plt.imshow(red_cat*0.5,cmap='gray',vmin=0,vmax=1)
```

```
Out[24]: <matplotlib.image.AxesImage at 0x7f55ddb89e10>
```



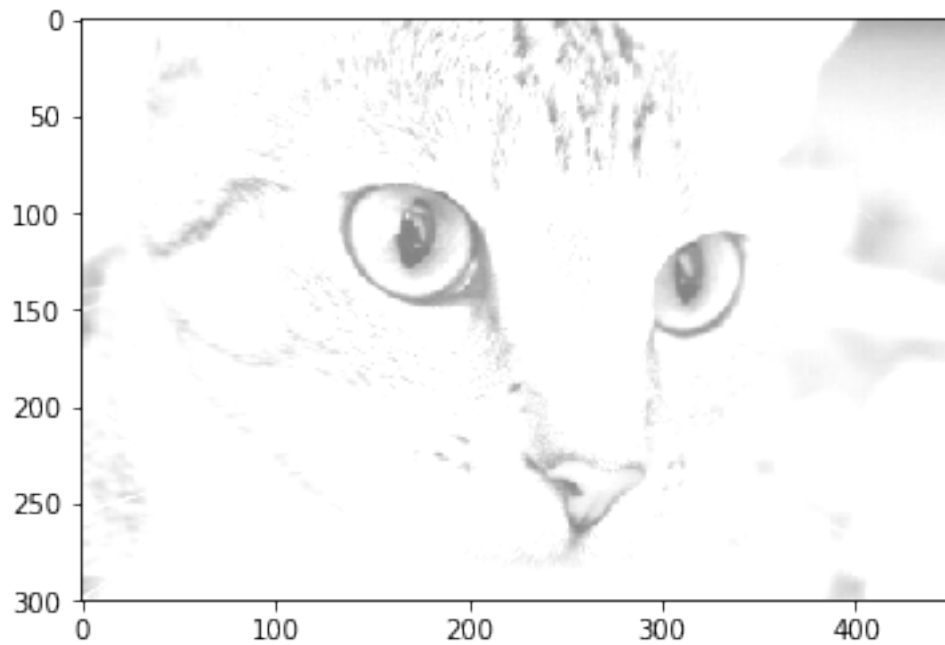
```
In [25]: plt.imshow(red_cat*2,cmap='gray',vmin=0,vmax=1)
```

```
Out[25]: <matplotlib.image.AxesImage at 0x7f55ddb61a58>
```



```
In [26]: plt.imshow(red_cat+0.5,cmap='gray',vmin=0,vmax=1)
```

```
Out[26]: <matplotlib.image.AxesImage at 0x7f55ddaba6a0>
```



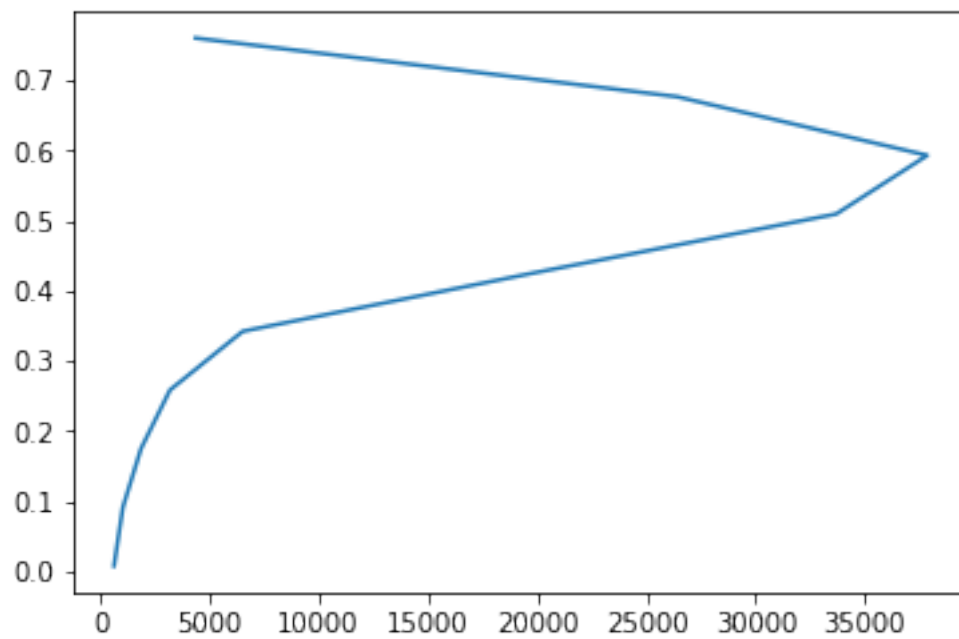
```
In [29]: red_cat.shape
```

```
Out[29]: (300, 451)
```

```
In [30]: vals, bins = np.histogram(red_cat)
```

```
plt.plot(vals,bins[:-1])
```

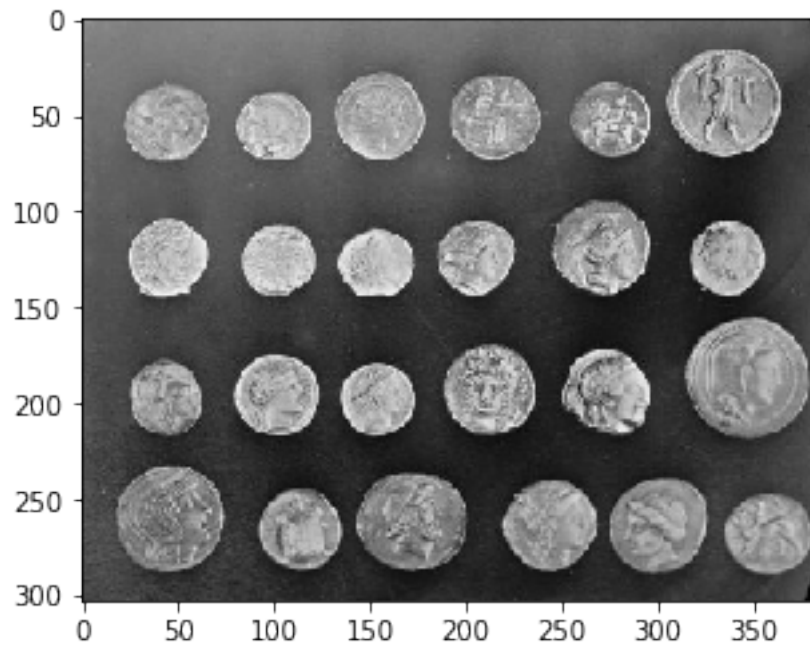
```
Out[30]: [<matplotlib.lines.Line2D at 0x7f55dd653400>]
```



```
In [32]: plt.imshow(coins, cmap='gray')
```

```
Out[32]: <matplotlib.image.AxesImage at 0x7f55dcee9898>
```

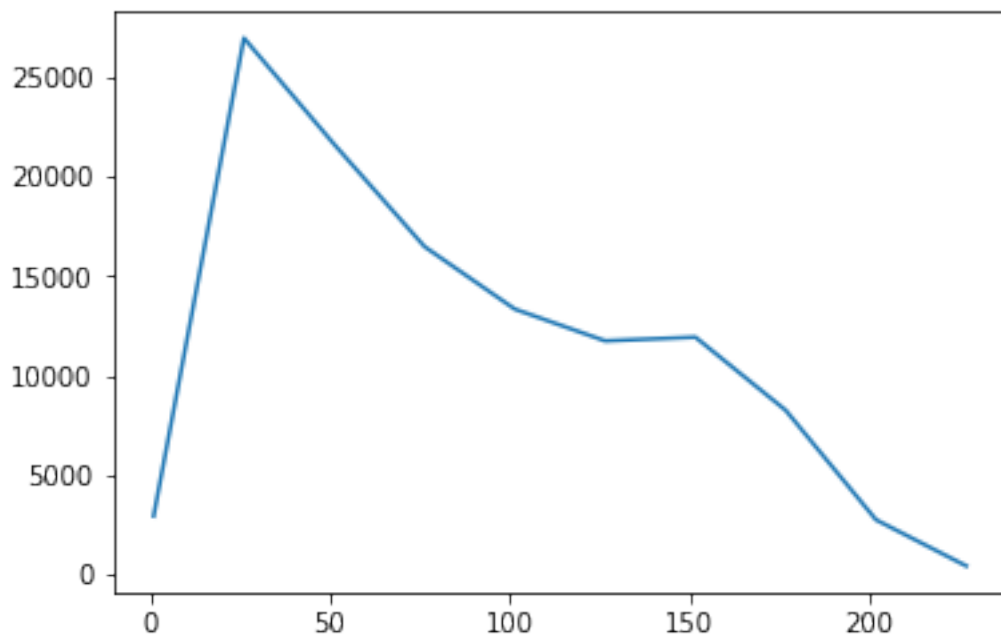




```
In [33]: vals, bins = np.histogram(coins)
```

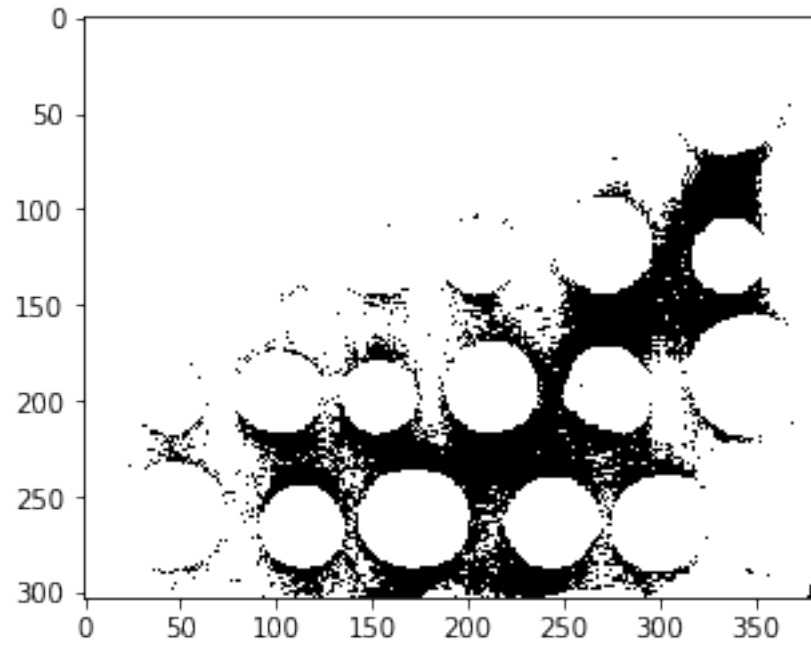
```
plt.plot(bins[:-1],vals)
```

```
Out[33]: [<matplotlib.lines.Line2D at 0x7f55dc7b78d0>]
```



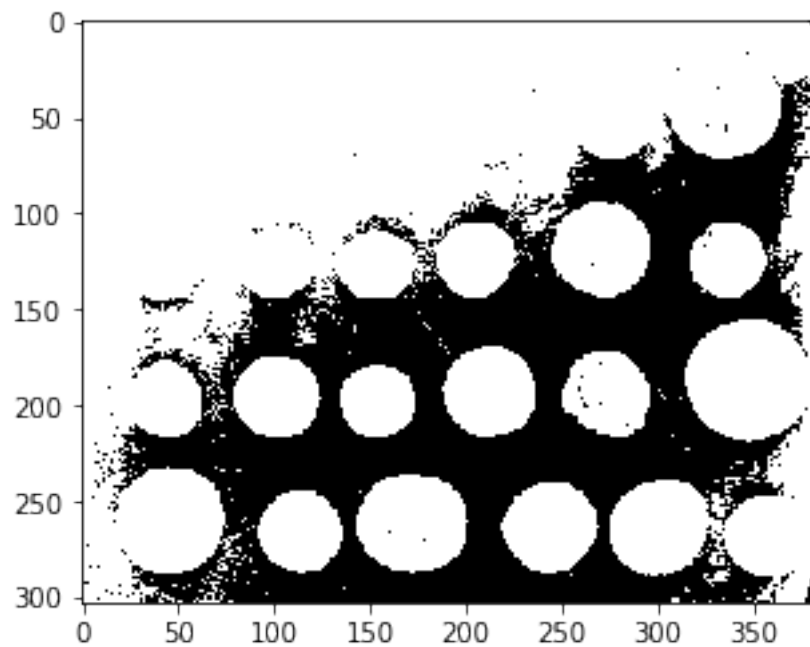
```
In [34]: plt.imshow(coins>40, cmap='gray')
```

```
Out[34]: <matplotlib.image.AxesImage at 0x7f55dc1704a8>
```



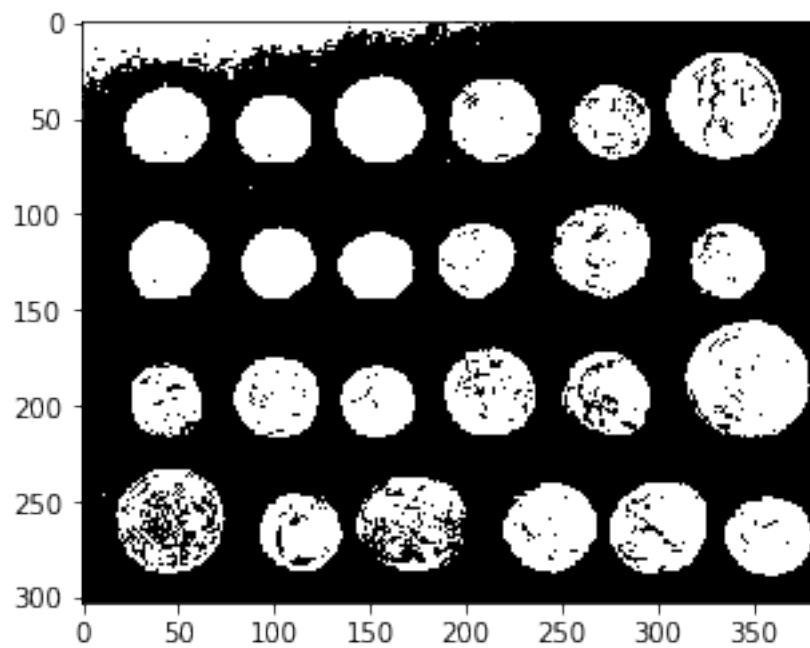
```
In [35]: plt.imshow(coins>60, cmap='gray')
```

```
Out[35]: <matplotlib.image.AxesImage at 0x7f55c79c8ef0>
```



```
In [36]: plt.imshow(coins>120, cmap='gray')
```

```
Out[36]: <matplotlib.image.AxesImage at 0x7f55c72e4278>
```



```
In [37]: threshold60 = coins>120
```

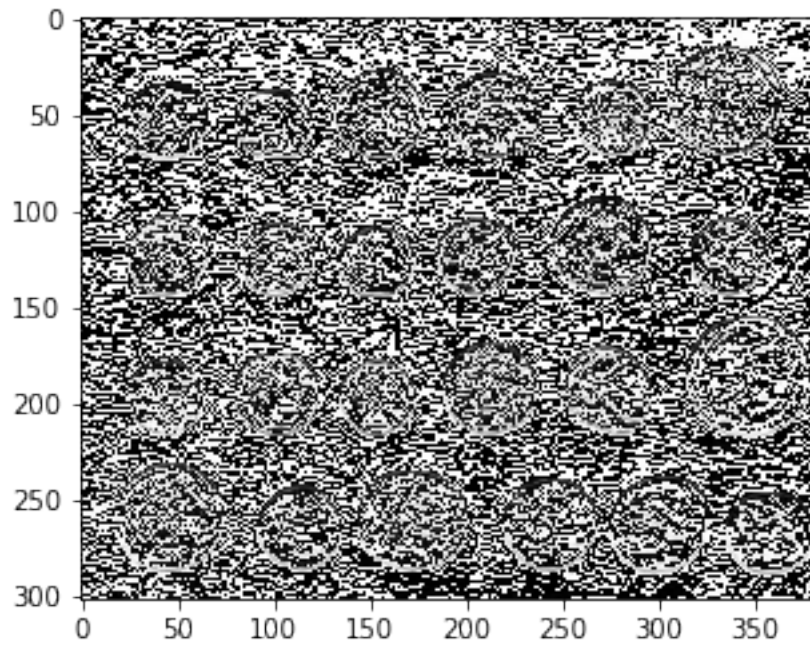
```
In [38]: coins.shape
```

```
Out[38]: (303, 384)
```

```
In [39]: xcoins = coins[1:,:]-coins[:-1,:]
```

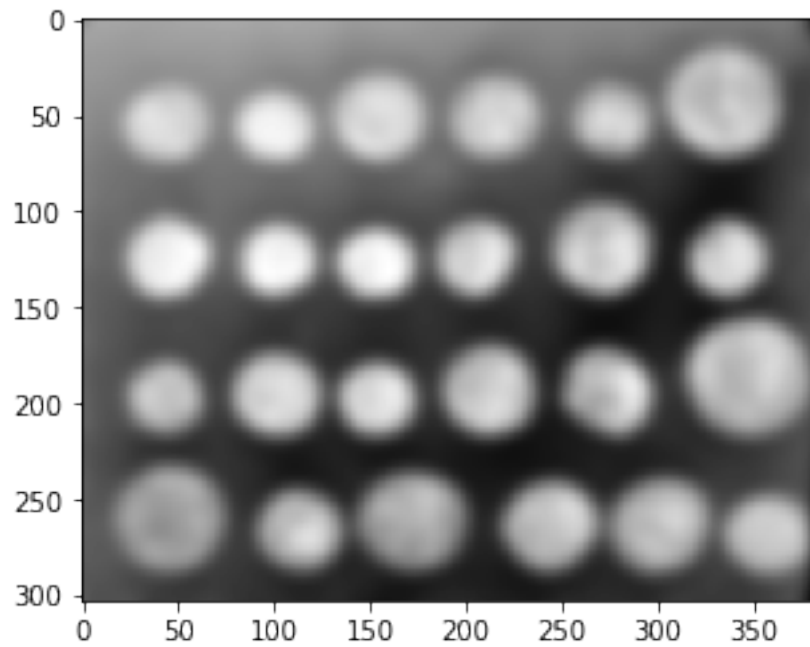
```
In [41]: plt.imshow(xcoins,cmap='gray')
```

```
Out[41]: <matplotlib.image.AxesImage at 0x7f55c6cc8198>
```



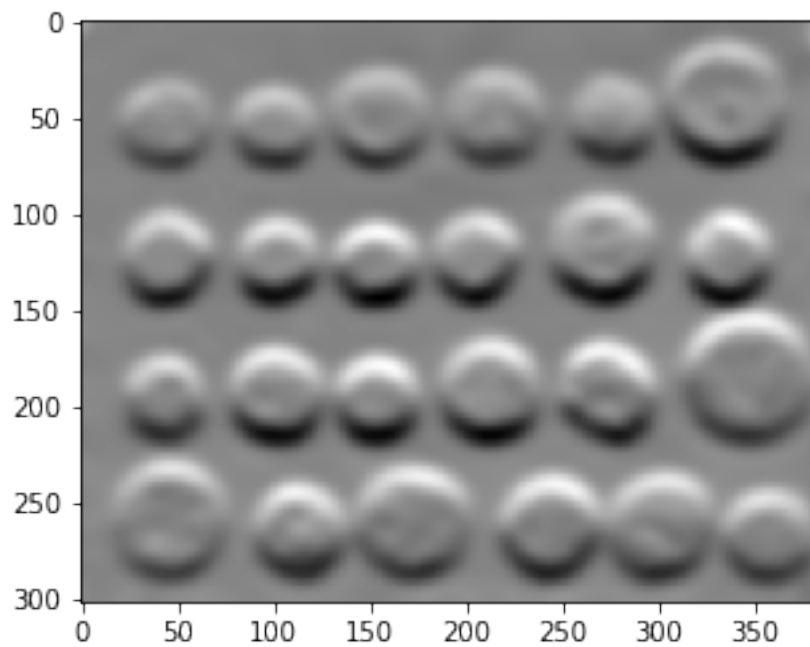
```
In [44]: plt.imshow(skimage.filters.gaussian(coins,sigma=5),cmap='gray')
```

```
Out[44]: <matplotlib.image.AxesImage at 0x7f55c6c0ff28>
```



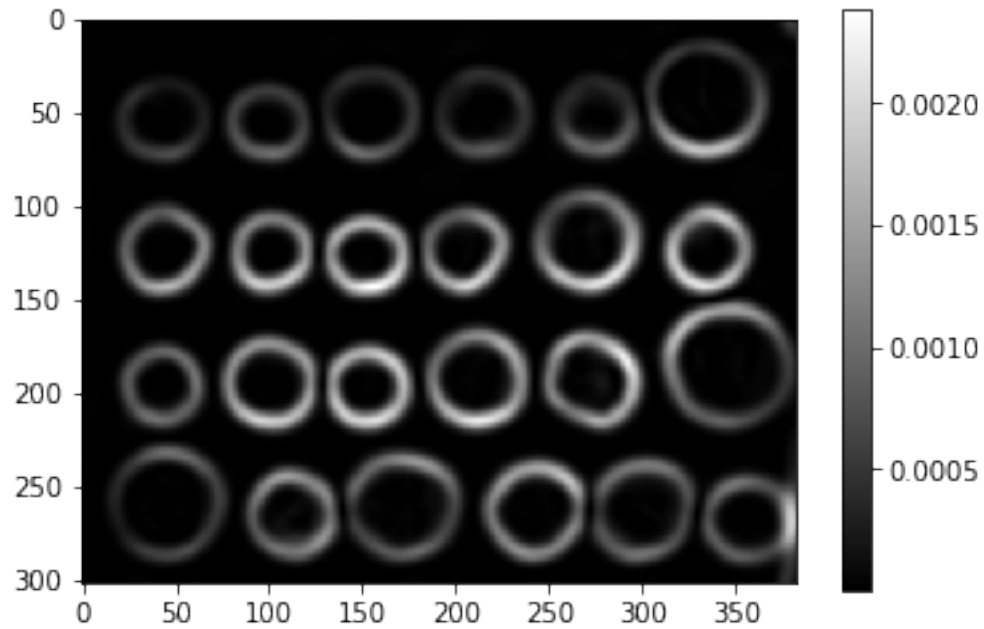
```
In [46]: bcoins = skimage.filters.gaussian(coins,sigma=5)
         xcoins = bcoins[1:,:]-bcoins[:-1,:]
         plt.imshow(xcoins,cmap='gray')
```

```
Out[46]: <matplotlib.image.AxesImage at 0x7f55c6bb0208>
```



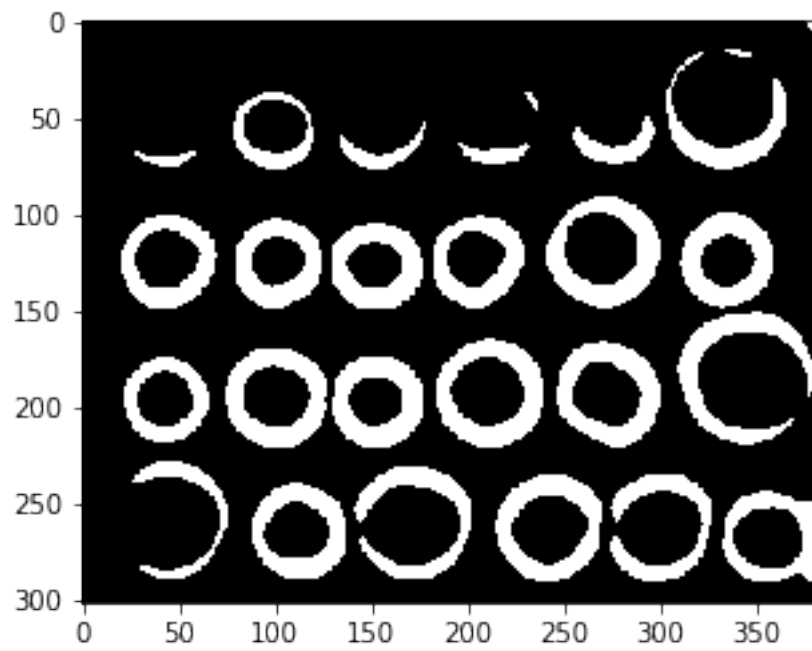
```
In [54]: xbcoins = (bcoins[1:,1:]-bcoins[:-1,1:])**2 + (bcoins[1:,1:]-bcoins[1:,-1])**2  
plt.imshow(xbcoins,cmap='gray')  
plt.colorbar()
```

```
Out[54]: <matplotlib.colorbar.Colorbar at 0x7f55c6925748>
```



```
In [55]: plt.imshow(xbcoins>0.0005,cmap='gray')
```

```
Out[55]: <matplotlib.image.AxesImage at 0x7f55c68970f0>
```



In [ ]: