Assignment 1.3
Image Pre-processing

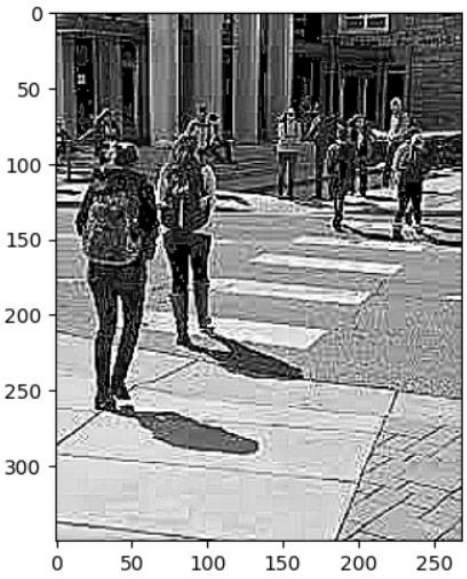
1. **Image 01:**

The Input: image01

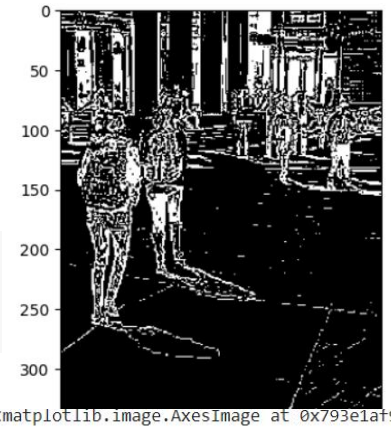
Segmentation

(place the code snippet and the corresponding output here)

```
img = cv2.imread("/content/image01.png", 0)

sharpening_filter = np.array([[-1,-1,-1],
                              [-1, 9 , -1],
                              [-1,-1,-1]])

#kernel_size = np.ones((5, 5), float) / 25

image = cv2.filter2D(img, -1, sharpening_filter)

#image = cv2.filter2D(img, -1, kernel_size)

#image = cv2.equalizeHist(image)
plt.imshow(image, cmap='gray')
scale = 0.6 #microns/pixel
```



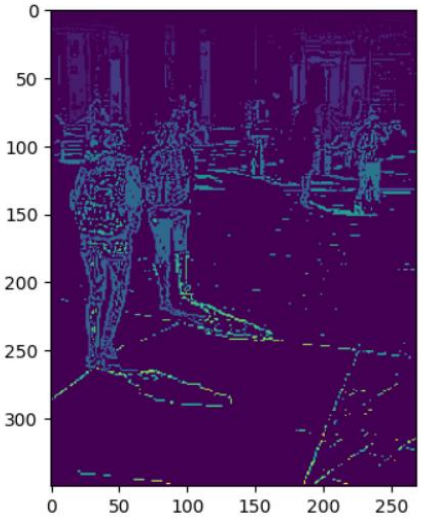
```
thresholded_img = image < 19 #for pedestrian picture

thresholded_img = image < threshold
plt.imshow(thresholded_img, cmap='gray')
```



```
from skimage.segmentation import clear_border
edge_touching_removed = clear_border(thresholded_img)
plt.imshow(edge_touching_removed, cmap='gray')
```



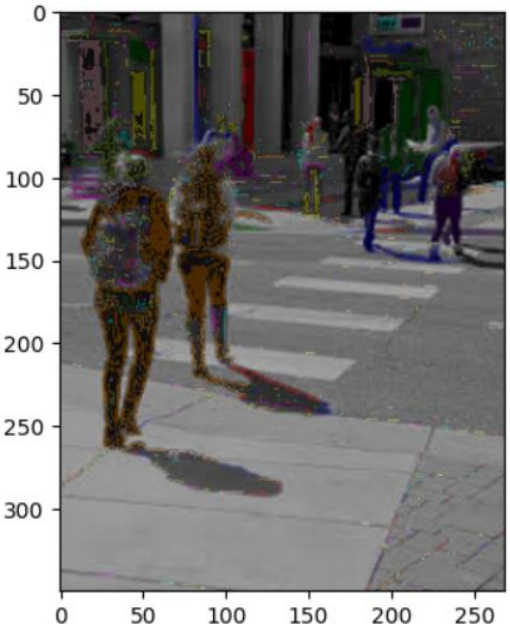
```
label_image = measure.label(edge_touching_removed, connectivity=image.ndim)

plt.imshow(label_image)
```



The presentation:

(place the code snippet and/or the corresponding output here)

```
[154] image_label_overlay = label2rgb(label_image, image=img)
      plt.imshow(image_label_overlay)
```

Description:

(place the code snippet and/or the corresponding output here)

```
[218] props = measure.regionprops_table(label_image, image,
                                        properties=['label',
                                                    'area', 'equivalent_diameter',
                                                    'mean_intensity', 'solidity'])
```

```
[219] import pandas as pd
      df = pd.DataFrame(props)
      print(df.head())
```

```
   label  area  equivalent_diameter  mean_intensity  solidity
0      1     6             2.763953        5.666667  1.000000
1      2   197            15.837556        1.162437  0.358834
2      3     1             1.128379        0.000000  1.000000
3      4     4             2.256758        4.750000  1.000000
4      5     1             1.128379        0.000000  1.000000
```

```
      df = df[df['area'] > 50]
      print(df.head())
```

```
    label  area  equivalent_diameter  mean_intensity  solidity
1       2   197            15.837556        1.162437  0.358834
54     55   117            12.205287        0.769231  0.639344
58     59   573            27.010484        0.675393  0.547801
59     60    80            10.092530        2.162500  0.509554
61     62    91            10.764051        3.131868  0.446078
```

```
[160] df['area_sq_microns'] = df['area'] * (scale**2)
      df['equivalent_diameter_microns'] = df['equivalent_diameter'] * (scale)
      print(df.head())
```

```
    label  area  equivalent_diameter  mean_intensity  solidity  \
1       2   197            15.837556        1.162437  0.358834
54     55   117            12.205287        0.769231  0.639344
58     59   573            27.010484        0.675393  0.547801
59     60    80            10.092530        2.162500  0.509554
61     62    91            10.764051        3.131868  0.446078

    area_sq_microns  equivalent_diameter_microns
1             70.92                     9.502534
54            42.12                     7.323172
58           206.28                    16.206291
59            28.80                     6.055518
61            32.76                     6.458431
```

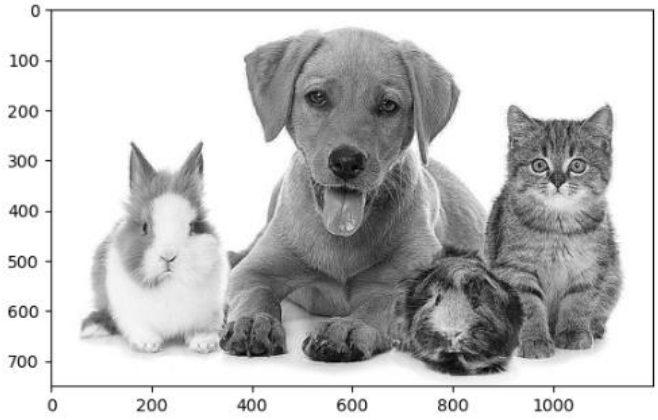
2. **Image 2:**

The Input: image2

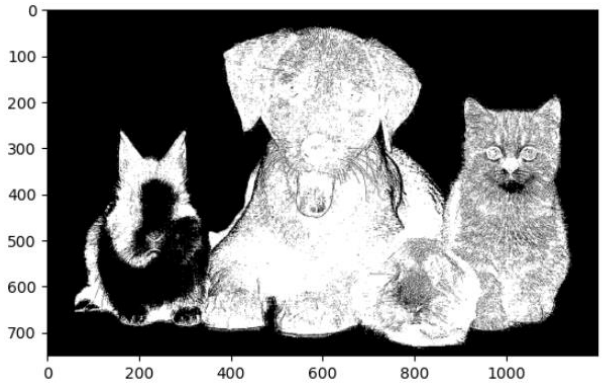
Segmentation

(place the code snippet and the corresponding output here)

```
img = cv2.imread("/content/image2.png", 0)

sharpening_filter = np.array([[-1,-1,-1],
                              [-1, 9 , -1],
                              [-1,-1,-1]])

#kernel_size = np.ones((5, 5), float) / 25

image = cv2.filter2D(img, -1, sharpening_filter)

#image = cv2.filter2D(img, -1, kernel_size)

#image = cv2.equalizeHist(image)
plt.imshow(image, cmap='gray')
scale = 0.6 #microns/pixel
```

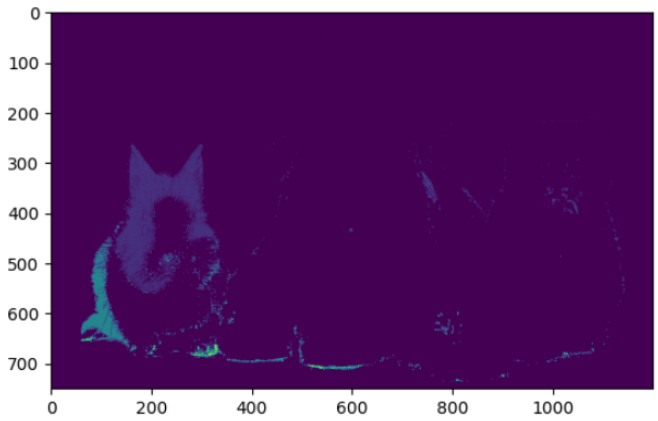



```
thresholded_img = image < 200

#thresholded_img = image < threshold
plt.imshow(thresholded_img, cmap='gray')
```




```
from skimage.segmentation import clear_border
edge_touching_removed = clear_border(thresholded_img)
plt.imshow(edge_touching_removed, cmap='gray')
```

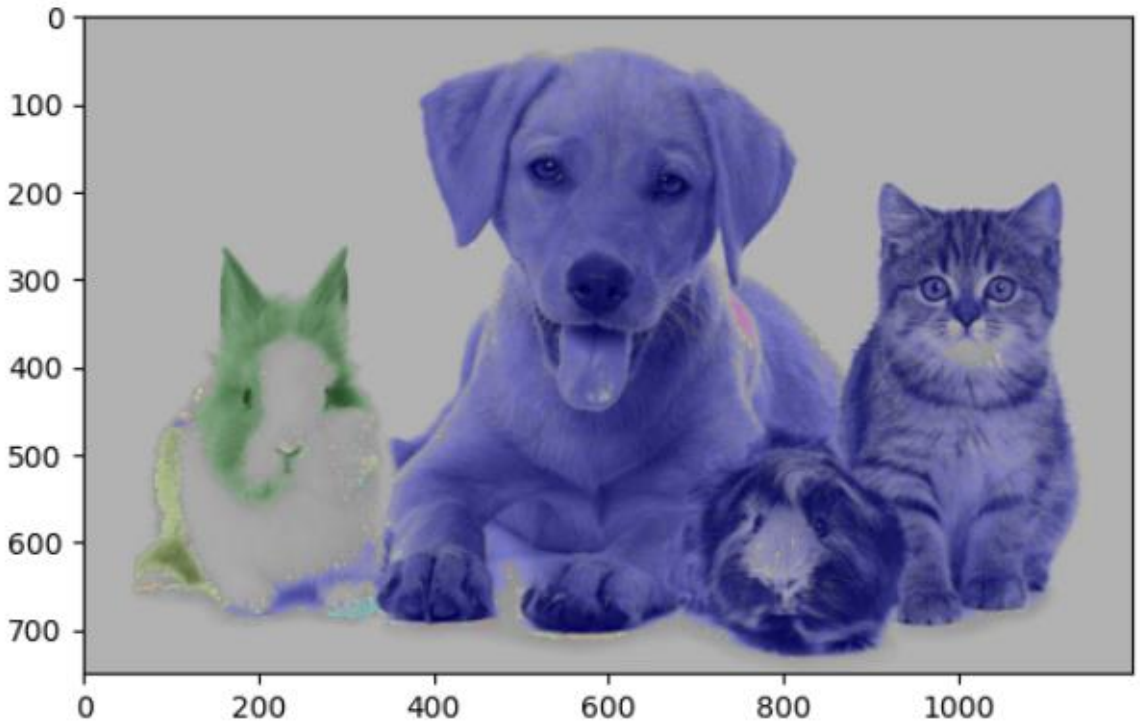
```
[182] label_image = measure.label(edge_touching_removed, connectivity=image.ndim)

      plt.imshow(label_image)
```



The presentation:

(place the code snippet and/or the corresponding output here)

```
[183] image_label_overlay = label2rgb(label_image, image=img)
      plt.imshow(image_label_overlay)
```



Description:

(place the code snippet and/or the corresponding output here)

```
[253] props = measure.regionprops_table(label_image, image,
                                        properties=['label',
                                                    'area', 'equivalent_diameter',
                                                    'mean_intensity', 'solidity'])
```

```
import pandas as pd
df = pd.DataFrame(props)
print(df.head())
```

```
   label    area  equivalent_diameter  mean_intensity  solidity
0      1       1             1.128379      198.000000  1.000000
1      2  330293           648.492181      100.458832  0.626074
2      3       3             1.954410      191.000000  1.000000
3      4       2             1.595769      195.500000  1.000000
4      5      16             4.513517      183.937500  0.592593
```

```
[255] df = df[df['area'] > 50]
print(df.head())
```

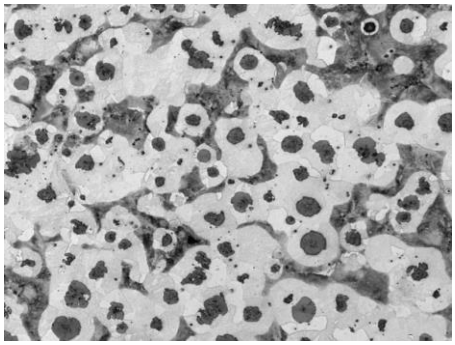
```
     label    area  equivalent_diameter  mean_intensity  solidity
1        2  330293           648.492181      100.458832  0.626074
164    165   21484           165.391289      142.648529  0.456931
253    254     393            22.369245      162.791349  0.545076
339    340      53             8.214724      172.622642  0.452991
541    542      85            10.403142      169.141176  0.566667
```

```
df['area_sq_microns'] = df['area'] * (scale**2)
df['equivalent_diameter_microns'] = df['equivalent_diameter'] * (scale)
print(df.head())
```

```
     label    area  equivalent_diameter  mean_intensity  solidity  \
1        2  330293           648.492181      100.458832  0.626074
164    165   21484           165.391289      142.648529  0.456931
253    254     393            22.369245      162.791349  0.545076
339    340      53             8.214724      172.622642  0.452991
541    542      85            10.403142      169.141176  0.566667

     area_sq_microns  equivalent_diameter_microns
1          118905.48                   389.095309
164          7734.24                    99.234773
253           141.48                    13.421547
339            19.08                     4.928835
541            30.60                     6.241885
```

3. **Image 3:**

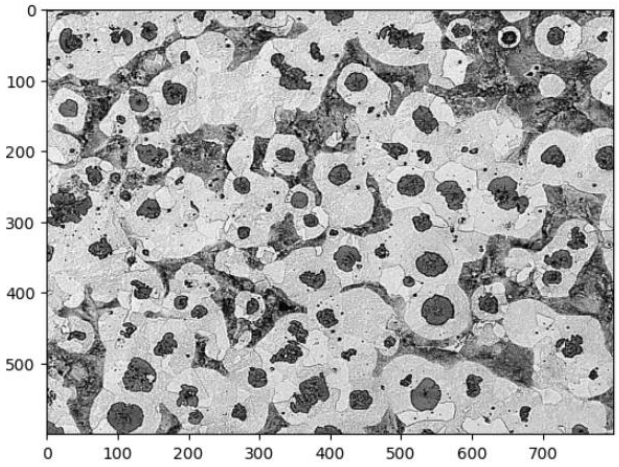
The Input: image3



Segmentation

(place the code snippet and the corresponding output here)

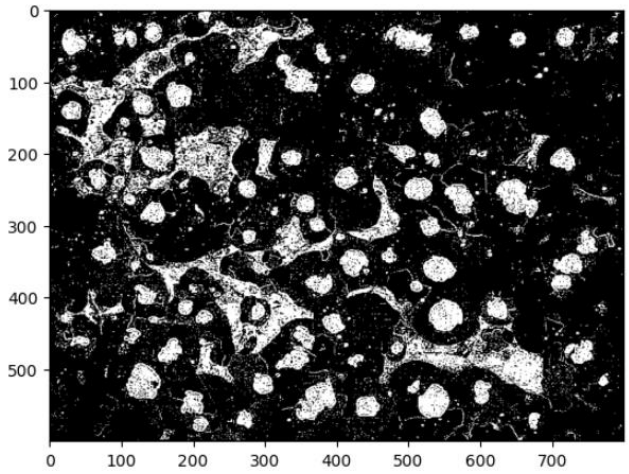
```
img = cv2.imread("/content/image3.jpg", 0)

sharpening_filter = np.array([[-1,-1,-1],
                              [-1, 9 , -1],
                              [-1,-1,-1]])

#kernel_size = np.ones((5, 5), float) / 25

image = cv2.filter2D(img, -1, sharpening_filter)

#image = cv2.filter2D(img, -1, kernel_size)

#image = cv2.equalizeHist(image)
plt.imshow(image, cmap='gray')
scale = 0.6 #microns/pixel
```



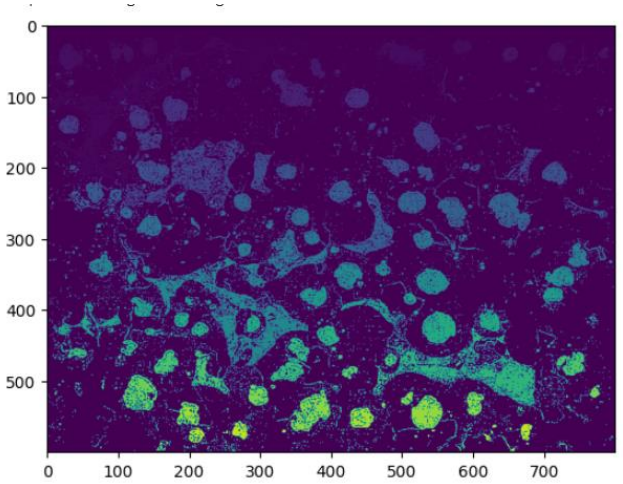

```
#thresholded_img = image < 200

    thresholded_img = image < threshold
    plt.imshow(thresholded_img, cmap='gray')
```

```
from skimage.segmentation import clear_border
edge_touching_removed = clear_border(thresholded_img)
plt.imshow(edge_touching_removed, cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x793e1ae96a70>
```



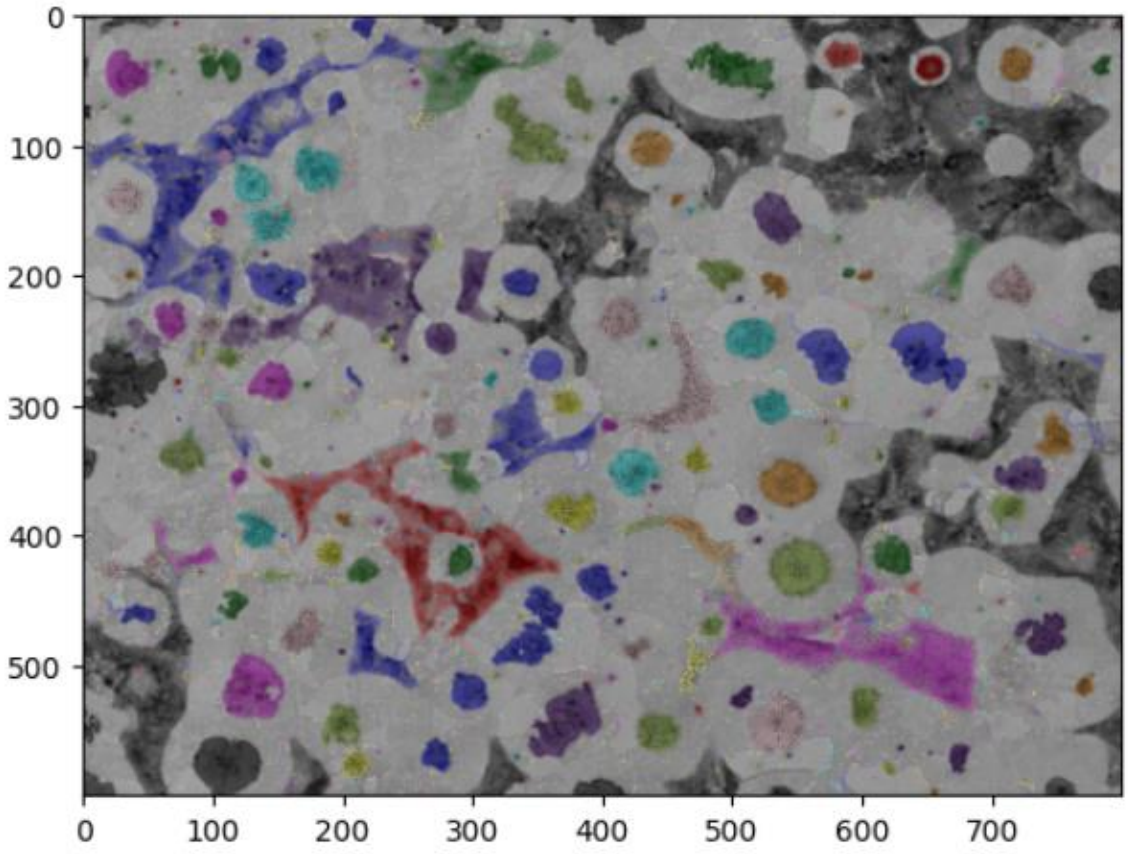
```
[195] label_image = measure.label(edge_touching_removed, connectivity=image.ndim)

      plt.imshow(label_image)
```



The presentation:

(place the code snippet and/or the corresponding output here)

```
image_label_overlay = label2rgb(label_image, image=img)
plt.imshow(image_label_overlay)
```



Description:

(place the code snippet and/or the corresponding output here)

```
[199] props = measure.regionprops_table(label_image, image,
                                        properties=['label',
                                                    'area', 'equivalent_diameter',
                                                    'mean_intensity', 'solidity'])
```

```
[200] import pandas as pd
      df = pd.DataFrame(props)
      print(df.head())
```

```
   label  area  equivalent_diameter  mean_intensity  solidity
0      1     1             1.128379      123.000000  1.000000
1      2   204            16.116478       41.691176  0.816000
2      3     8             3.191538      100.125000  0.888889
3      4     2             1.595769      128.500000  1.000000
4      5     1             1.128379      132.000000  1.000000
```

```
df = df[df['area'] > 50]
print(df.head())
```

```
     label  area  equivalent_diameter  mean_intensity  solidity
1        2   204            16.116478       41.691176  0.816000
54      55  1996            50.412163       74.792084  0.451379
121    122  6781            92.918445       70.967409  0.233602
141    142   456            24.095585       53.335526  0.798599
170    171   384            22.111626       53.786458  0.728653
```

```
[202] df['area_sq_microns'] = df['area'] * (scale**2)
      df['equivalent_diameter_microns'] = df['equivalent_diameter'] * (scale)
      print(df.head())
```

```
     label  area  equivalent_diameter  mean_intensity  solidity  \
1        2   204            16.116478       41.691176  0.816000
54      55  1996            50.412163       74.792084  0.451379
121    122  6781            92.918445       70.967409  0.233602
141    142   456            24.095585       53.335526  0.798599
170    171   384            22.111626       53.786458  0.728653

     area_sq_microns  equivalent_diameter_microns
1              73.44                     9.669887
54            718.56                    30.247298
121          2441.16                    55.751067
141           164.16                    14.457351
170           138.24                    13.266975
```