Ashley Chen
DSCI549 - HW4

1. Image processing

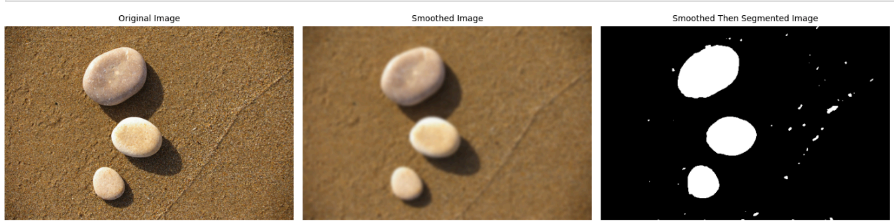
```
img=loadIMG()
displaySmooth(img)
```

```
Please input the image filename: rocks.jpg
<Figure size 640x480 with 0 Axes>
```


```
[9]:
displaySegment(img)
```

```
<Figure size 640x480 with 0 Axes>
```

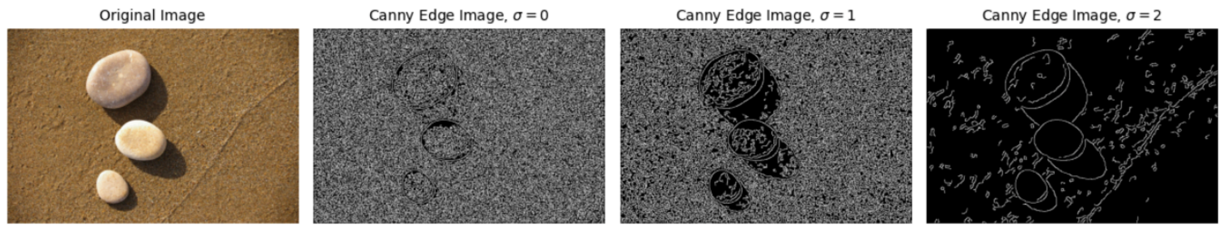

```
plotSmoothThenSegment(img)
```



a. Without smoothing, the segmented image shows a fair amount of noise in the background. The sand's subtle intensity variations cause many small regions or speckles to appear as foreground. With smoothing, the background in the smoothed image is more uniform, so thresholding is less likely to misclassify noise pixels. The resulting segmented mask is cleaner, with the rocks clearly defined and very few extraneous white spots in the background. They differ because of noise reduction and sharper object boundaries. Smoothing reduces local intensity fluctuations in the sand, so small bright/dark flecks do not get segmented as separate regions. And although excessive smoothing can blur edges, a moderate amount helps unify the background intensities without losing the main boundaries of the rocks.

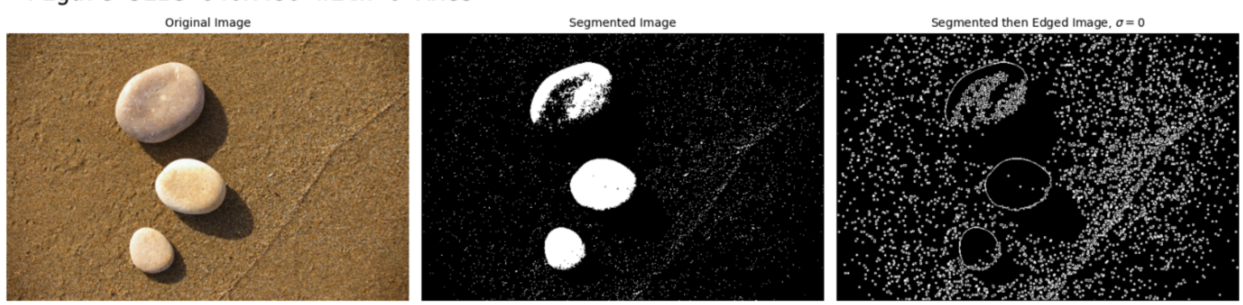
```
CannyEdge(img, sigma)
```

```
<Figure size 640x480 with 0 Axes>
```

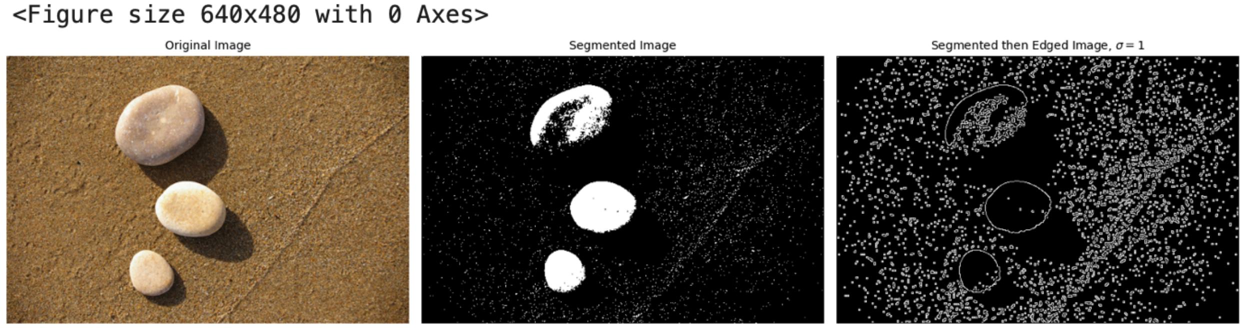



```
sigma0, sigma1, sigma2=0,1,2
displaySegmentThenCanny(img,sigma0)
displaySegmentThenCanny(img,sigma1)
displaySegmentThenCanny(img,sigma2)
```

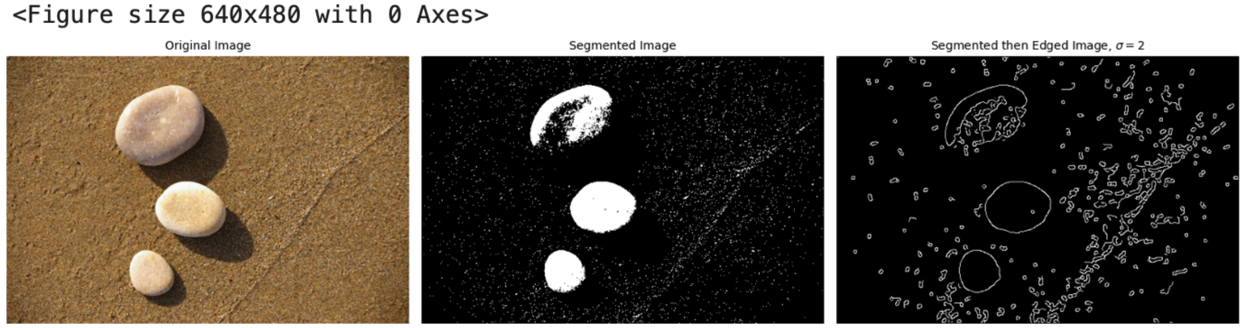
```
<Figure size 640x480 with 0 Axes>
```




```
<Figure size 640x480 with 0 Axes>
```




```
<Figure size 640x480 with 0 Axes>
```

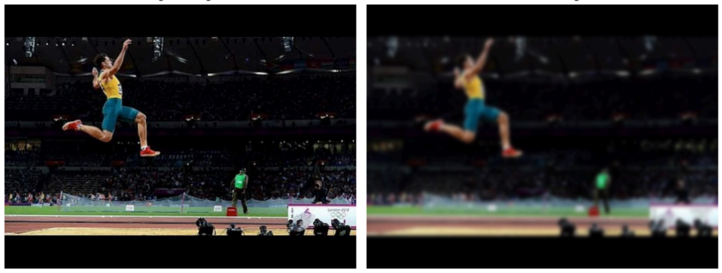


b. Looking at Canny on the original, unsegmented image, a dense field of edges representing the sand's texture can be detected along with the outlines of the rocks. Because the sand has a lot of microscopic, grainy detail, Canny picks up many tiny edges. Looking at Canny on the segmented image, the segmented image is largely black background with white rocks. So applying Canny here yields edges primarily along the rock boundaries. The background is uniform black, so almost no extra edges are detected in the background. They differ because on the full-color or grayscale original, there are countless small gradients in the sand. Canny's gradient detection sees them all, producing a busy edge map. Once segmented, the image contains just two intensities (foreground vs. background), so the only strong gradient is at the rock boundary.
c. When $\sigma = 0$, there is no smoothing before edge detection, so there are very noisy edges. Every tiny grain in the sand triggers edges, and the rock boundaries are hidden among many stray contours. When $\sigma = 1$, moderate smoothing reduces some noise. The rock outlines become clearer, though the background still has some detectable edges where the sand's texture remains. When $\sigma = 2$, heavier smoothing averages out more of the background texture. Fewer edges remain, and the primary rock boundaries are most visible. Fine details are sacrificed in exchange for reduced noise.

```
img=loadIMG()
displaySmooth(img)
```

```
Please input the image filename: longjump.jpg
<Figure size 640x480 with 0 Axes>
```

Original Image Smoothed Image

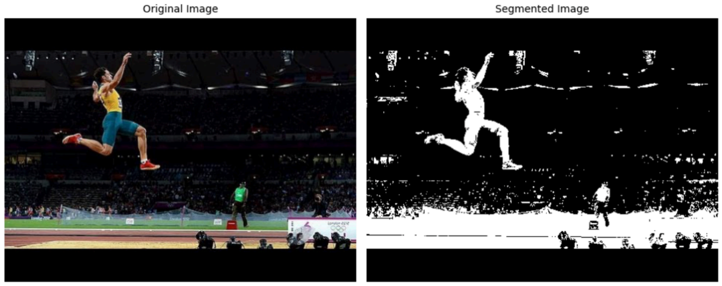


[23]:

```
displaySegment(img)
```

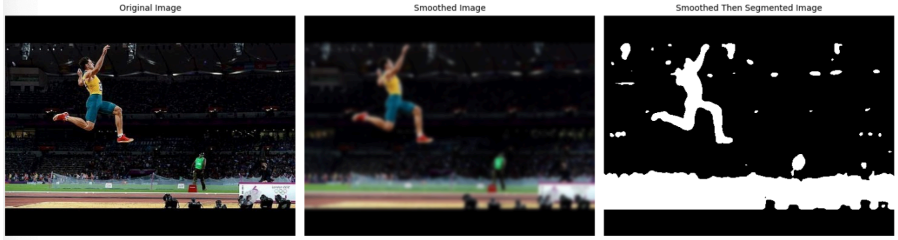
```
<Figure size 640x480 with 0 Axes>
```

Original Image Segmented Image



```
plotSmoothThenSegment(img)
```

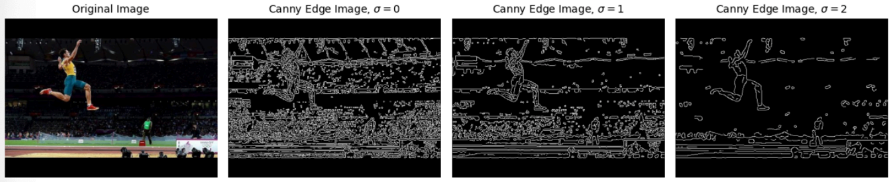
Original Image Smoothed Image Smoothed Then Segmented Image



```
CannyEdge(img, sigma)
```

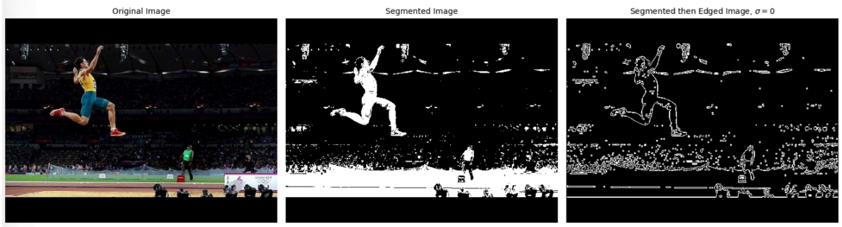
```
<Figure size 640x480 with 0 Axes>
```

Original Image Canny Edge Image, $\sigma = 0$ Canny Edge Image, $\sigma = 1$ Canny Edge Image, $\sigma = 2$

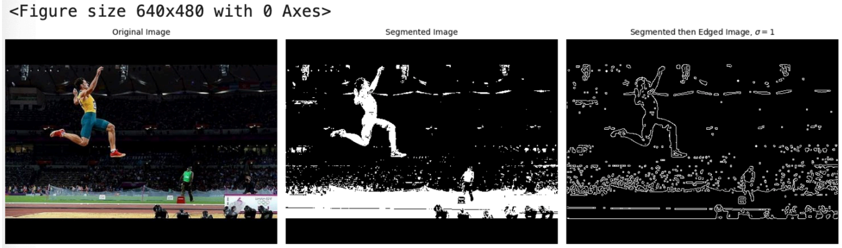


```
sigma0, sigma1, sigma2=0,1,2
displaySegmentThenCanny(img,sigma0)
displaySegmentThenCanny(img,sigma1)
displaySegmentThenCanny(img,sigma2)
```

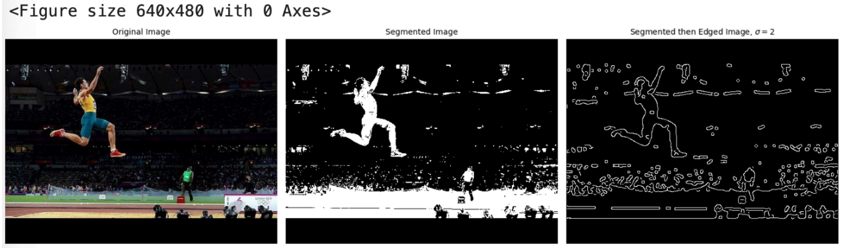
```
<Figure size 640x480 with 0 Axes>
```




```
<Figure size 640x480 with 0 Axes>
```




```
<Figure size 640x480 with 0 Axes>
```



d. longjump.jpg

i. *Segmentation With and Without Smoothing:* On the long jump image, smoothing helps blend out the stadium crowd and lights in the background a bit, so the athlete is easier to isolate. However, because the background has large areas of varying intensity (people, seats, track, etc), even with smoothing, some of those background details can still appear in the segmented result. Compared to the rocks image, which had a fairly uniform background of sand, the long jump shot has more going on like the crowd, scoreboard, and ground lines. Therefore, the segmentation picks up more patchy regions if it’s not smoothed enough or if the thresholding is set in a certain way.

ii. *Canny Edge Detection With and Without Segmentation:* As with the rocks, Canny on the original (no segmentation) sees edges around everything: the athlete, stadium, seats, signage, people in the stands, etc. Because there’s so much high-contrast detail, the edge image can look quite busy. For Canny on the segmented image, once you have a mostly black-and-white (foreground vs. background) image, the edges mainly outline the athlete. Some edges can still be seen where the background was mis-segmented, but it’s nowhere near as cluttered as applying Canny directly on the original.

iii. *Effect of Sigma in Canny:* Just like the rocks image, a higher sigma (more smoothing) in Canny means fewer edges in the background, making the athlete’s contour more prominent. However, too much smoothing can lose smaller details, like arms, fingers, or shoe edges. In a complex image like long jump, $\sigma = 0$ produces a very dense edge map, while $\sigma = 2$ weeds out more background edges, so the jumper’s figure and major stadium lines stand out.

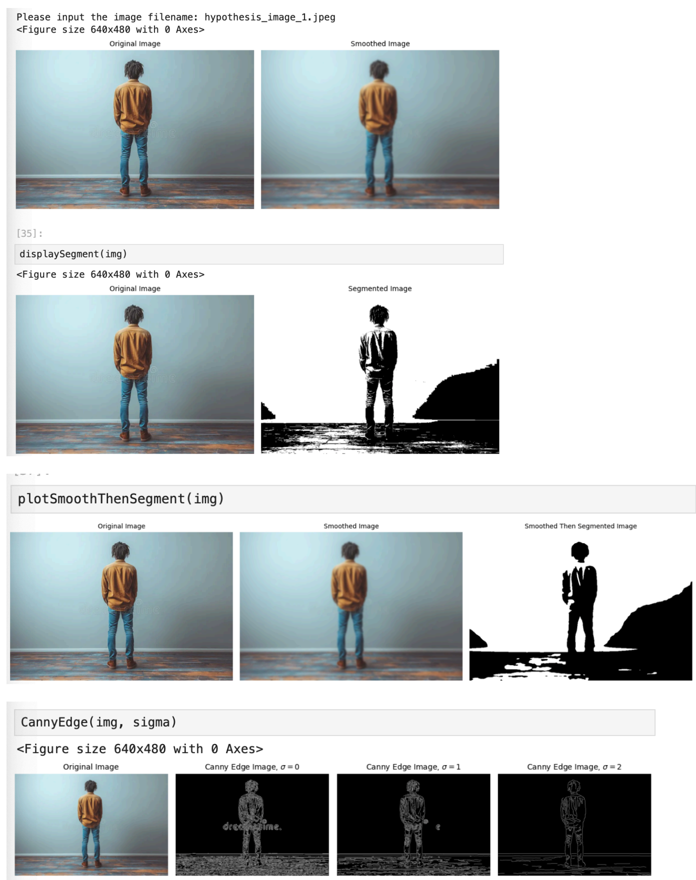
e. Some hypotheses include: “If the background is relatively uniform or has low texture, then a simple threshold plus smoothing will segment objects more cleanly”. Low-detail backgrounds don’t confuse the segmentation algorithm, so fewer random patches are seen. Another hypothesis is that “When the background is highly detailed or has many objects, you’ll need more sophisticated thresholding or more smoothing to avoid tons of noise in the segmentation”. High-texture or varied backgrounds secret many small intensity changes that might be misinterpreted as foreground. Finally, another hypothesis is “increasing the Gaussian sigma in Canny edge detection will be more beneficial for busy images than for simple ones because it will filter out a lot of irrelevant detail”. For a complicated scene, heavy smoothing helps isolate only the strongest contours. But in a simpler scene, like the rocks on sand, you may not need as much because you already have fewer false edges.



f. Each image above should apply to their respective hypotheses. For image 1, from a nearly uniform background, segmentation should work almost perfectly with minimal smoothing, because there’s no complex background to throw it off. In image 2, we see a busy street. Without heavy smoothing or advanced thresholding, you’ll likely get a chaotic segmentation. Using bigger blur or an adaptive threshold might help. In image 3, a partially textured background creates a not super busy, but not super uniform background, giving you results somewhere in between. Moderate smoothing might be enough to isolate the main subject, but the pattern could cause partial missegmentations.

g. Image 1’s results support hypothesis 1 because the uniform wall lets the segmentation work with fewer artifacts and little to no advanced thresholding is needed. Image 2’s results support hypothesis 2, showing that a busy background yields a noisy segmentation. Even with smoothing, many details remain, so more advanced techniques might be needed to get a cleaner separation of foreground vs. background. Image 3 also confirms hypothesis 3. In a more textured setting, bigger sigma values help filter out extraneous background edges. But you also see the natural trade-off that if you smooth too aggressively, you lose fine details. (See images below).

   i. Image 1:



```
sigma0, sigma1, sigma2=0,1,2
displaySegmentThenCanny(img,sigma0)
displaySegmentThenCanny(img,sigma1)
displaySegmentThenCanny(img,sigma2)
```

```
<Figure size 640x480 with 0 Axes>
```

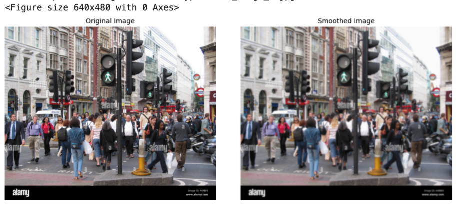

```
<Figure size 640x480 with 0 Axes>
```


```
<Figure size 640x480 with 0 Axes>
```

ii. Image 2:

```
img=loadIMG()
displaySmooth(img)
```

```
Please input the image filename: hypothesis_image_2.jpg
<Figure size 640x480 with 0 Axes>
```



[45]:

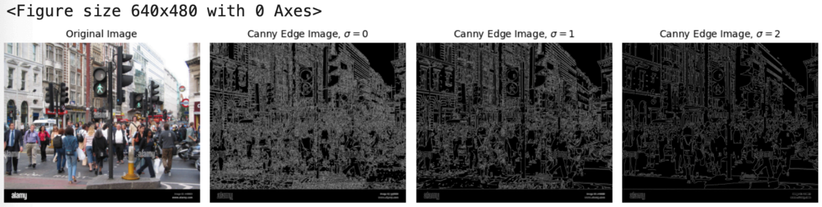
```
displaySegment(img)
```

```
<Figure size 640x480 with 0 Axes>
```


```
plotSmoothThenSegment(img)
```


```
CannyEdge(img, sigma)
```

```
<Figure size 640x480 with 0 Axes>
```



```
sigma0, sigma1, sigma2=0,1,2
displaySegmentThenCanny(img,sigma0)
displaySegmentThenCanny(img,sigma1)
displaySegmentThenCanny(img,sigma2)
```

```
<Figure size 640x480 with 0 Axes>
```


```
<Figure size 640x480 with 0 Axes>
```


```
<Figure size 640x480 with 0 Axes>
```

iii. Image 3:

```
img=loadIMG()
displaySmooth(img)
```

```
Please input the image filename: hypothesis_image_3.jpeg
<Figure size 640x480 with 0 Axes>
```

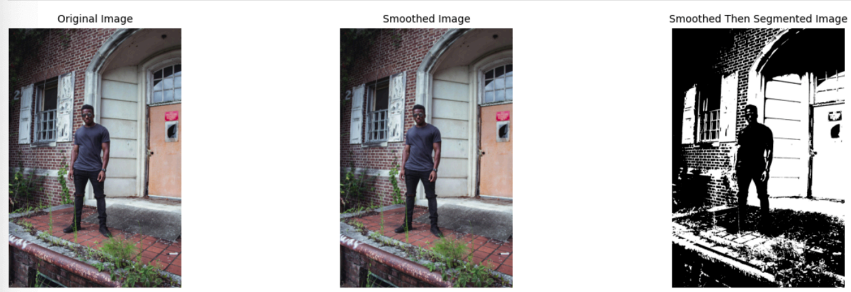
[3]:

```
displaySegment(img)
```

```
<Figure size 640x480 with 0 Axes>
```

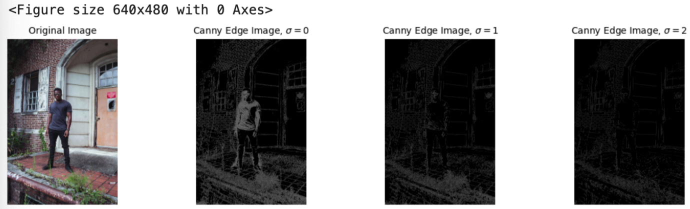
[5]:

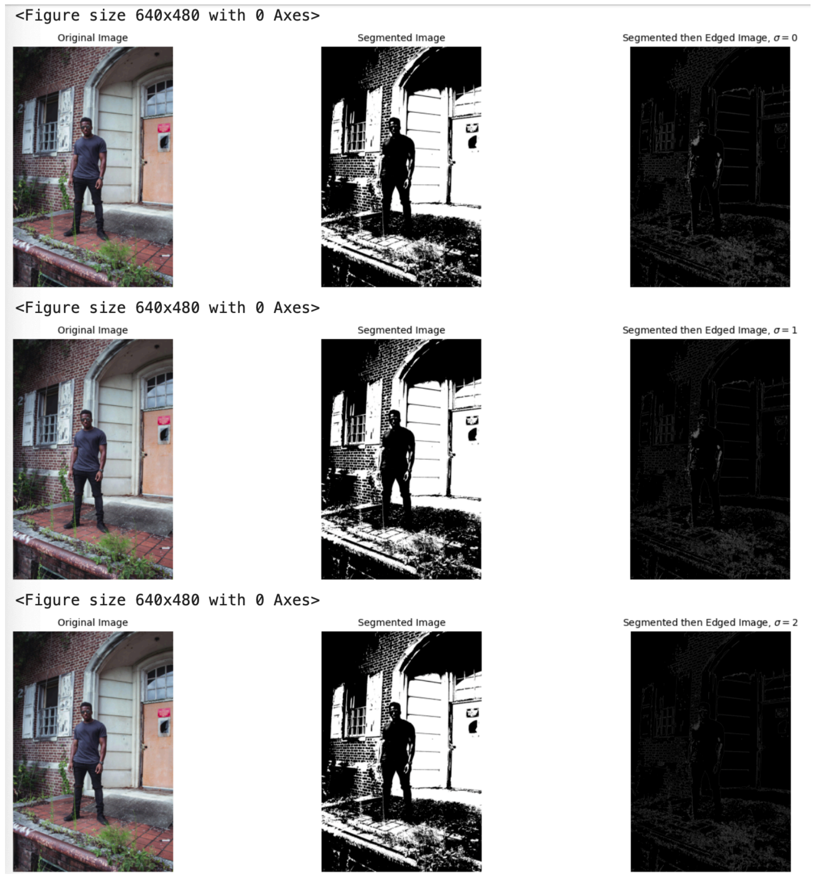
```
plotSmoothThenSegment(img)
```



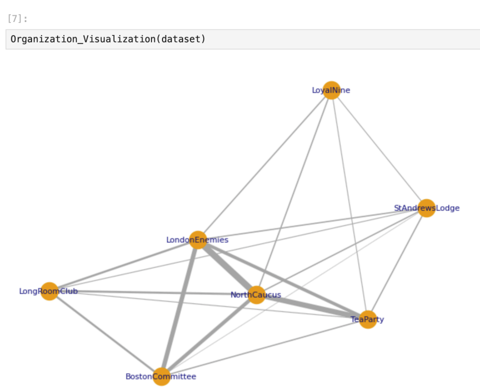

```
CannyEdge(img, sigma)
```

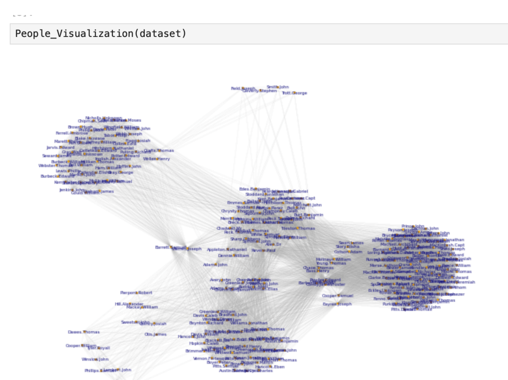
```
<Figure size 640x480 with 0 Axes>
```





2. Network visualization

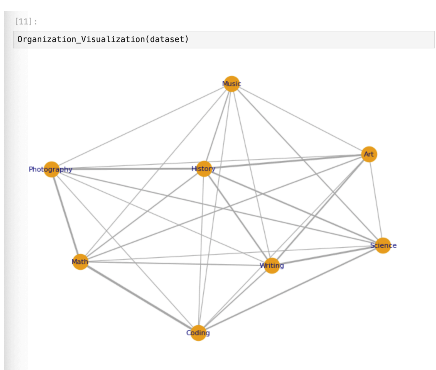


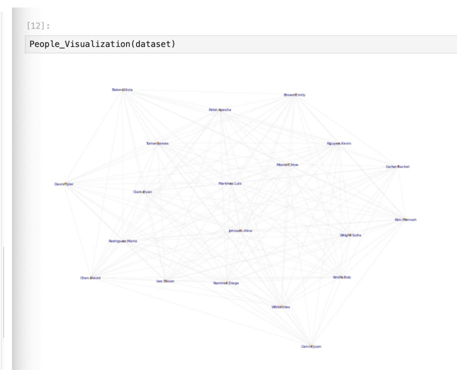


a.

b. Based on the people visualization, three individuals that stand out in different ways are John Smith, Joseph Barrett, and Samuel Dexter. John Smith is highly connected and seems like the “life of the party” in this network, making him a central figure. In network terms, he has high degree centrality because he’s linked to so many others. Plus, he probably acts as a bridge between different groups, giving him high betweenness centrality as well. He was likely an influential leader or coordinator during the Revolution. Joseph Barrett, while still pretty social, is more like the head of a smaller friend group rather than a whole community. He has a decent number of connections, but not as many as John Smith. His network is more tightly knit, meaning his friends also tend to be friends with each other. This gives him moderate centrality and clustering. He's probably an important player within his own little bubble, but not as influential on the larger scale. Samuel Dexter, on the other hand, is hanging out on the edge of the network with few connections. He's a bit of a loner compared to the social butterflies. In network terms, he has low degree centrality and doesn't really connect different groups so his betweenness centrality is also low. This suggests he wasn't a super involved or influential figure.

c. When looking at the organization network, I noticed that the Loyal Nine and the Long Room Club don't seem to have any members in common. It's like they're separate friend groups that never really overlap. In network theory, this is called being "disconnected components", which basically means there's no link between them whatsoever. They probably operated pretty independently, without sharing members or resources, almost like two different clubs at school that never collaborate on anything.





d. Yes, the networks were generally what I expected. I expected popular interests like music and art to be well-connected since they tend to attract more students. Subjects like math and science were moderately connected but not as central as more creative or social hobbies. Some subjects formed tight-knit clusters while others appeared more spread out due to overlapping members.

3. Time Series - Data Exploration
    a. Between 2010 and 2019, the number of collisions in LA County are pretty consistent with around 50,000 to 60,000 collisions per year. The numbers peaked around 2017 and 2018, with a sudden drop in 2020 because of the COVID-19 pandemic.

b. They must often happen on the streets, followed by parking lots.
c. The 99 age group has a big spike, which can be explained by unknown ages and bad data entry. The spikes at 25, 30, 35, 40, 45, and so on may be due to rounding errors. People tend to round their age to a nice, even number when they're in a hurry or don't want to be precise.
d. The time-of-day plot shows that collisions peak around 3:00 to 7:00 PM, with the highest point being around 5:00 to 6:00 PM. This is rush hour when everyone's headed home from work or school. With so many tired, stressed, and frustrated drivers on trafficked roads, accidents are more likely to occur. My hypothesis is that collisions are most frequent during rush hour because of heavy traffic volume, making it harder to maneuver, driver fatigue and stress after a long day, and more distractions such as being on the phone and being in a hurry. Other data that would help confirm this hypothesis would be traffic volume data during different times of the day to see how it lines up with the collisions rates, weather data to see if rain or fog make it worse, road condition reports to check if potholes or poor road designs are a factor, and driver impairment data, like DUI stats or reports of distracted driving.