186

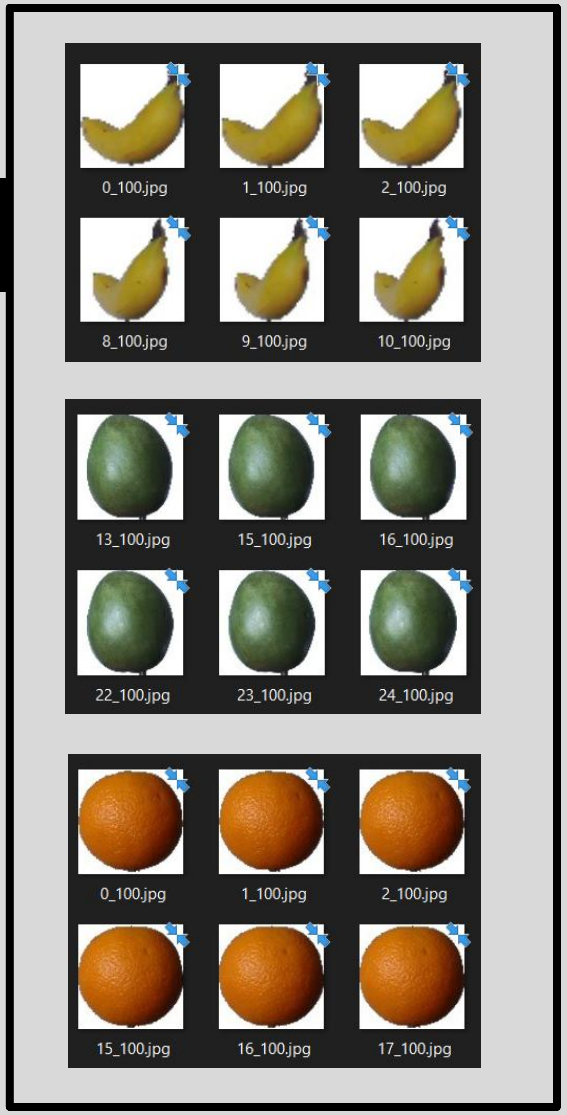
Activity 12: Machine Learning Introduction

Mary Chris Go / 2014 11122

Goal

To be able to extract features of three classes: orange, banana, and mango

Dataset

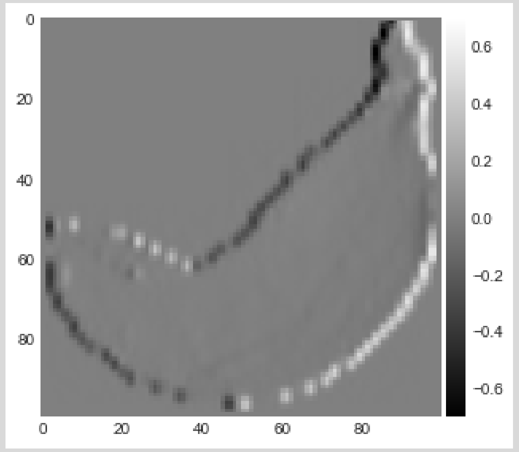
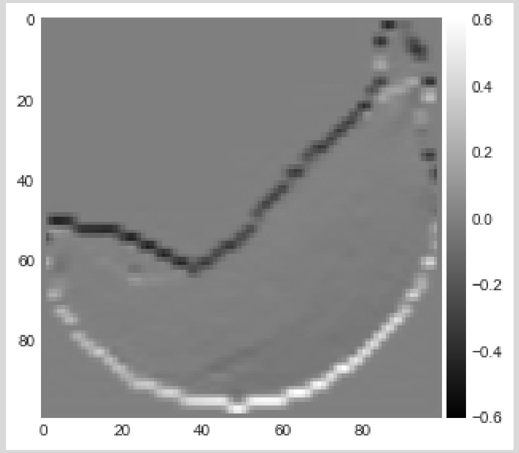


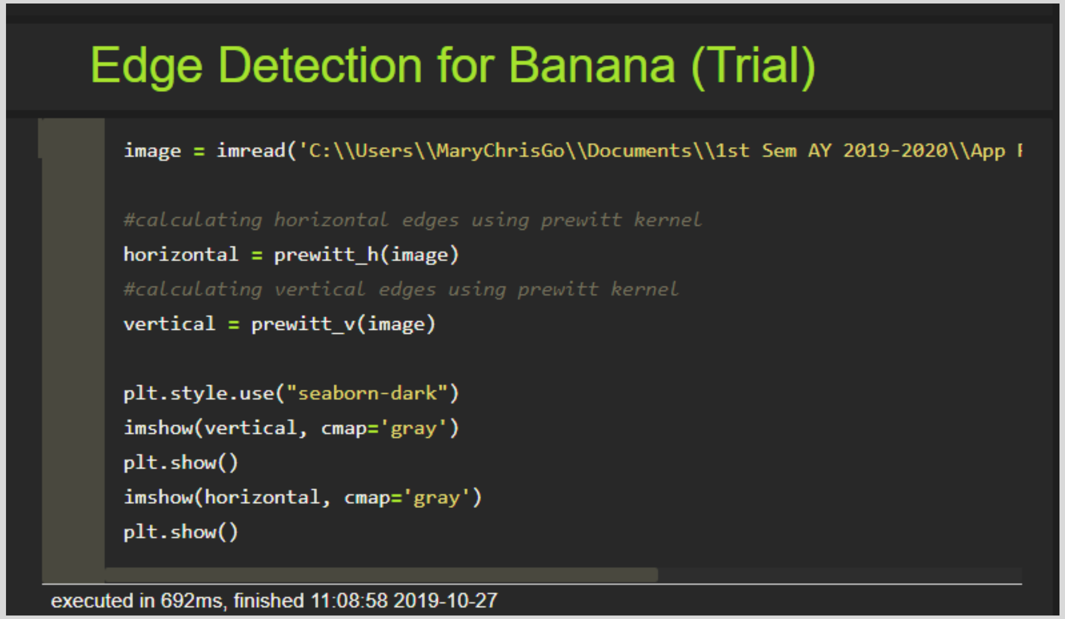
https://www.kaggle.com/moltean/fruits

The dataset was obtained from Kaggle website. The total number of images is 82 213 of 120 fruits and vegetables. But for this activity, only the mango, banana, and orange classes were the chosen. The image size for all is 100 x 100 pixels. Images on the slide are screenshots of the folder.

490 images = banana
490 images = mango
479 images = orange

Trials: Edge Detection and Color Detection

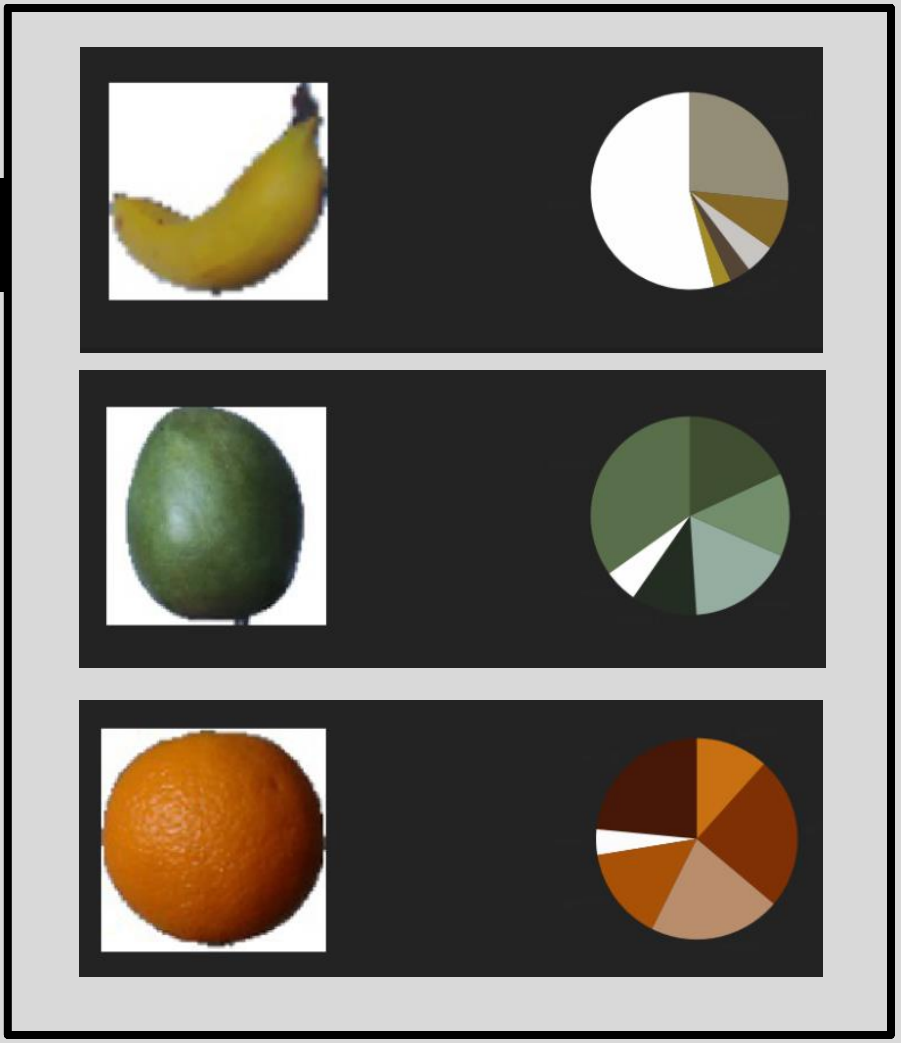

Edge Detection for Banana (Trial)

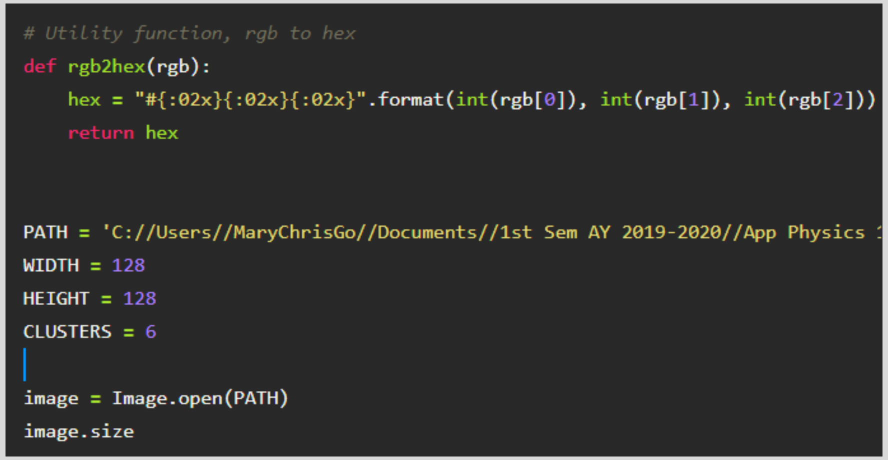
```
image = imread('C:\\Users\\MaryChrisGo\\Documents\\1st Sem AY 2019-2020\\App I

#calculating horizontal edges using prewitt kernel
horizontal = prewitt_h(image)
#calculating vertical edges using prewitt kernel
vertical = prewitt_v(image)

plt.style.use("seaborn-dark")
imshow(vertical, cmap='gray')
plt.show()
imshow(horizontal, cmap='gray')
plt.show()
```

executed in 692ms, finished 11:08:58 2019-10-27

I tried to do the edge detection. In the snippet, only one picture was the subject of this experiment. I was able to do it. But had a hard time extracting the differences between the two classes: mango and orange. Both of them look alike without the color. I thought this method would not be helpful.

Trials: Edge Detection and Color Detection





```
# Utility function, rgb to hex
def rgb2hex(rgb):
    hex = "#{:02x}{:02x}{:02x}".format(int(rgb[0]), int(rgb[1]), int(rgb[2]))
    return hex


PATH = 'C://Users//MaryChrisGo//Documents//1st Sem AY 2019-2020//App Physics 1
WIDTH = 128
HEIGHT = 128
CLUSTERS = 6

image = Image.open(PATH)
image.size
```

What I thought that by converting rgb to hex, I can differentiate the colors. What I encountered in this problem is that every image has a different ratio of colors because of the lighting of fruits. So, it didn't work. But I enjoyed this part as I can extract color palettes from images and transform it in a pie chart.

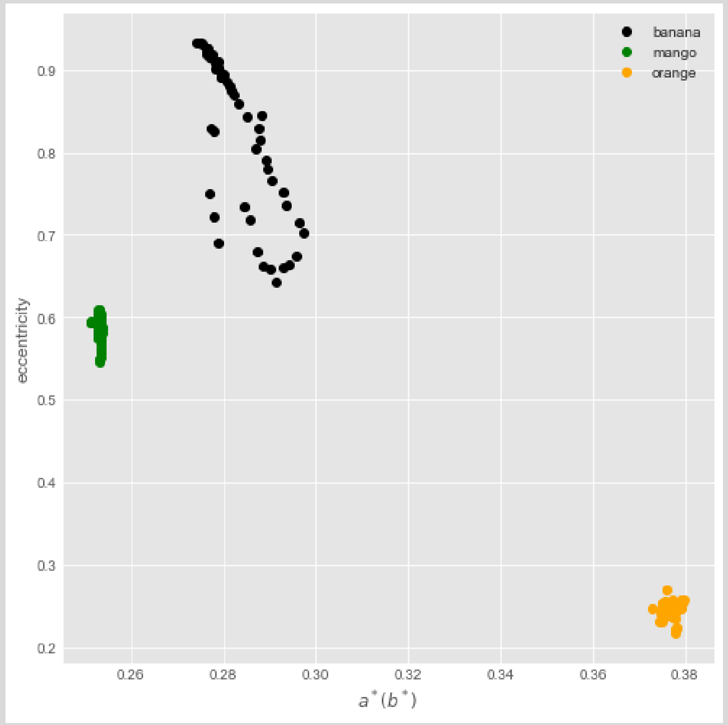
Eccentricity

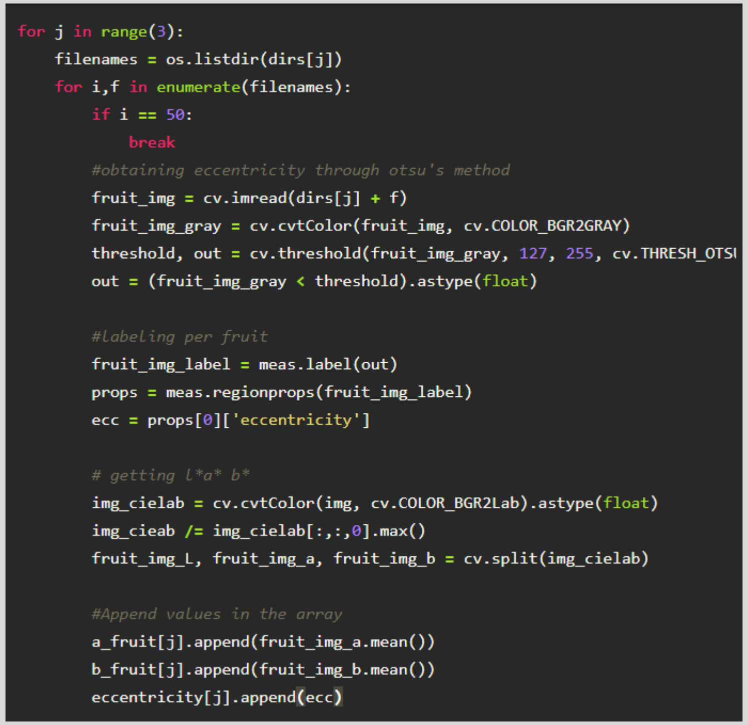
From the past activity, we took the eccentricity of the object through meas.label, meas.regionprops. This part was made easy since we have already done this from the last activity.

A* (b*)

We express the color of the image as three values through CIELAB color space. Now, we obtain the mean of a* and b* through “imgL /= imgL[;,;,0].max()”. We split this by appending it in a* and b*. This method is a good way to differentiate the three classes since it will tell you the color of the object. a* values are the green to red while b* values are the blue to yellow. Since we’re dealing with colored images, we get the a* and b* values only.

Results





```
for j in range(3):
    filenames = os.listdir(dirs[j])
    for i,f in enumerate(filenames):
        if i == 50:
            break
        #obtaining eccentricity through otsu's method
        fruit_img = cv.imread(dirs[j] + f)
        fruit_img_gray = cv.cvtColor(fruit_img, cv.COLOR_BGR2GRAY)
        threshold, out = cv.threshold(fruit_img_gray, 127, 255, cv.THRESH_OTS
        out = (fruit_img_gray < threshold).astype(float)

        #labeling per fruit
        fruit_img_label = meas.label(out)
        props = meas.regionprops(fruit_img_label)
        ecc = props[0]['eccentricity']

        # getting l*a* b*
        img_cielab = cv.cvtColor(img, cv.COLOR_BGR2Lab).astype(float)
        img_cieab /= img_cielab[:,:,0].max()
        fruit_img_L, fruit_img_a, fruit_img_b = cv.split(img_cielab)

        #Append values in the array
        a_fruit[j].append(fruit_img_a.mean())
        b_fruit[j].append(fruit_img_b.mean())
        eccentricity[j].append(ecc)
```

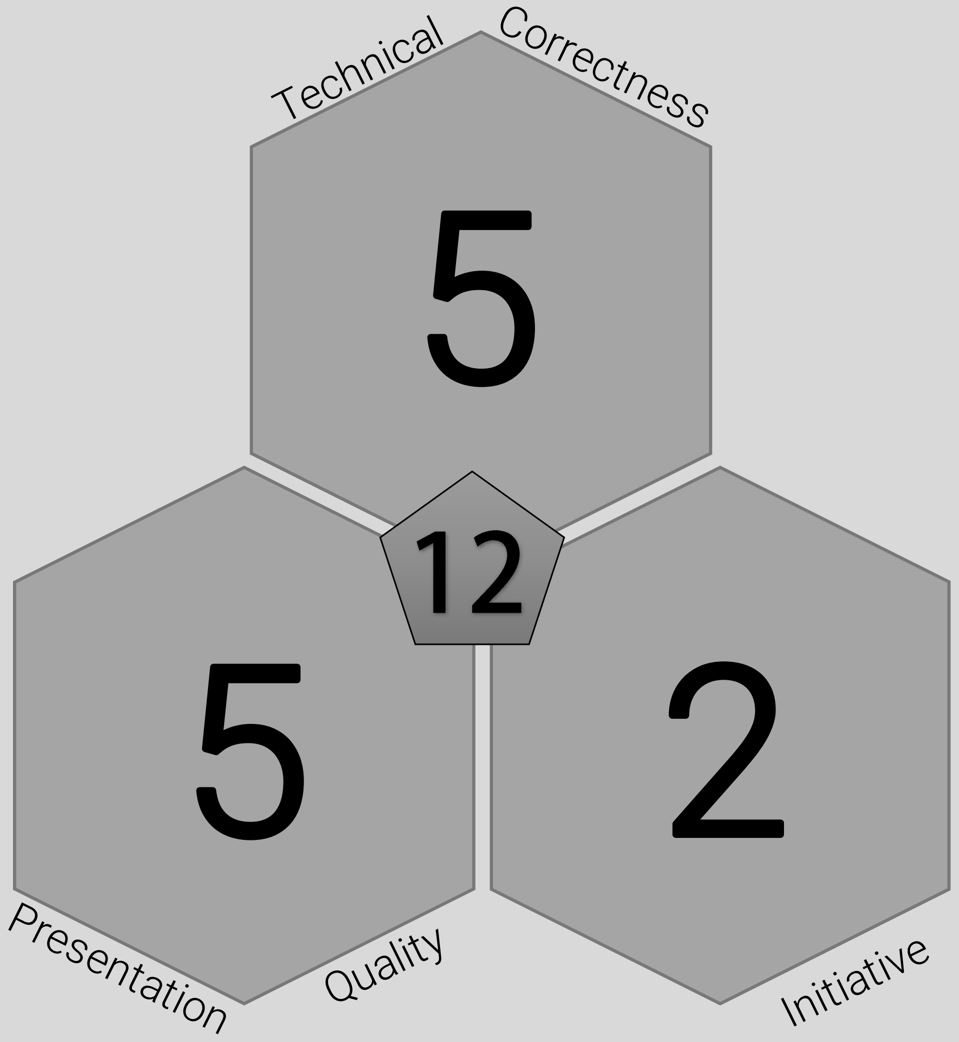
Through multiplying the mean of a* and b*, I was able to get the x-axis. Meanwhile, the y-axis in this part is the eccentricity. The graph makes sense. From the dataset, we saw how the mango and orange look alike. Their eccentricity of each class is almost the same whether turning the fruit in different angles. Meanwhile, for the banana, it is scattered as the eccentricity changes per angle. The snippet of the code is shown. Inside the for loop is the meat of the whole algorithm. It determines the a* and b* values, and the eccentricity through some packages. If you download the whole dataset, you will see that there is a large difference between the banana images. This reason accounts for the scattered black dots.

Summary

The goal was achieved by extracting colors and eccentricity from the images of each classes. Each were differentiated and can be obviously clustered just by looking at the graph. It was an effective method to differentiate them from each other.

It was an integration of what we have learned. We were familiar with extracting the feature of an image. Through ‘for’ loop, we were able to do it for several images. The hard part for this was setting up the ‘for’ loop with two feature extractors. All in all, it was fun doing a simple machine learning activity that was hard coded instead of packages.

Self-Evaluation



References

- Soriano, M., “Machine Learning Introduction”. 2019
- Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.