Feature Extraction

Objectives

- Download images and tabulate their features.
- Decide on at most three visual features by which each of the objects can be differentiated by and measure these features using image processing techniques.
- Plot the feature points in 2D or 3D space and discuss if class clustering is observed.

Summary of my Life

- Romina told me about Scikit module
- It changed my life drastically.
- Goodbye Opencv, Hello Scikit (at least for now)

Step 0: Get Images of Objects

- I used fruits (namely Bananas, Oranges and Mangoes)
- I compiled all the fruits of the same type into a single image collage per type
- Mam Jing said at least 20 images. I tried about 10-15 images first.
- Images have background (white or colored), leaves or shadows. They were eliminated using the feature of MS powerpoint that removes background.
- Some images have watermarks in the fruits. I used online photo editor to retouch those places so that removes watermarks while blending in the fruit color.

Step 0: Get Images of Objects

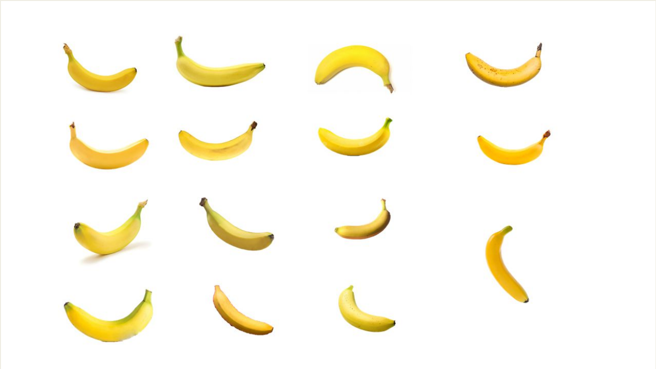


Fig 1: Banana collage

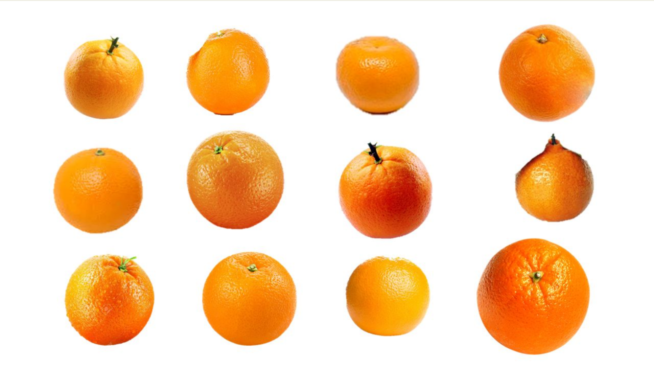


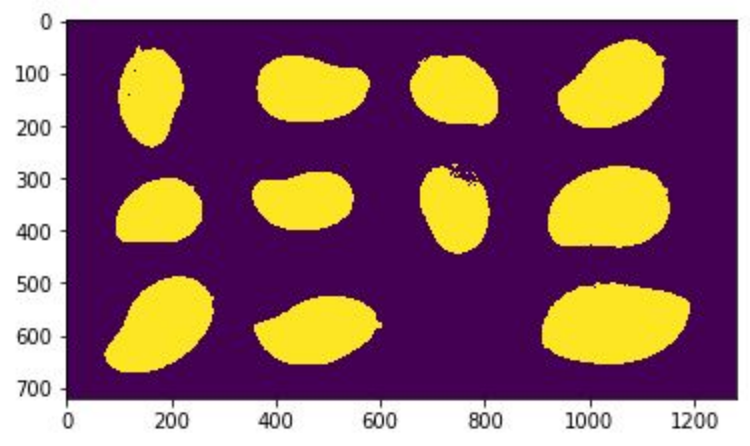
Fig 2: Orange collage

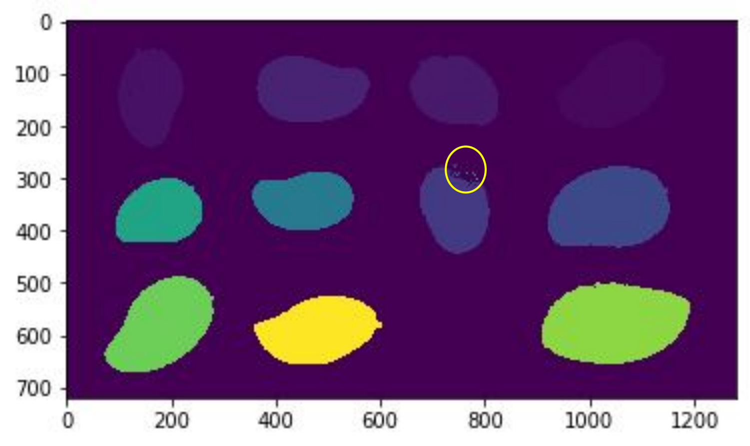
Fig 3: Mango collage

Step 1: Detecting Regions of Interest

- To detect regions, an image collab is set to grayscale.
- Threshold is applied to accept pixel values lower than the threshold value (non-white)
- Using skimage.measure.label(), pixels on unique regions are labeled with a corresponding integer of that region.
- There are some regions (ex. noise due to threshold) that may be included thus I applied a filter that should satisfy a certain area to only include the regions of fruits

Step 1: Detecting Regions of Interest





- The top image shows the pixels lower than the threshold value which where the fruits reside (yellow) compared to the white background (purple).

- There are 11 fruits but the code was able to find and label 41 regions due to noise (ex. encircled in bottom image). So these were filtered out by area and only the 11 regions are left.

Step 2: Featured Used

- I used 4 Features/Classes
- 2 are from the normalized chromaticity values (r and g)
- 1 is the eccentricity of the region
- 1 is the convexity of the region
- These features are very convenient because their range is already normalized [0,1].

Step 2: Features: Chromaticity Values

A colored image can be made up of red, green and blue colors at a certain brightness [0,255]. It can also be interpreted with only 2 values r and g

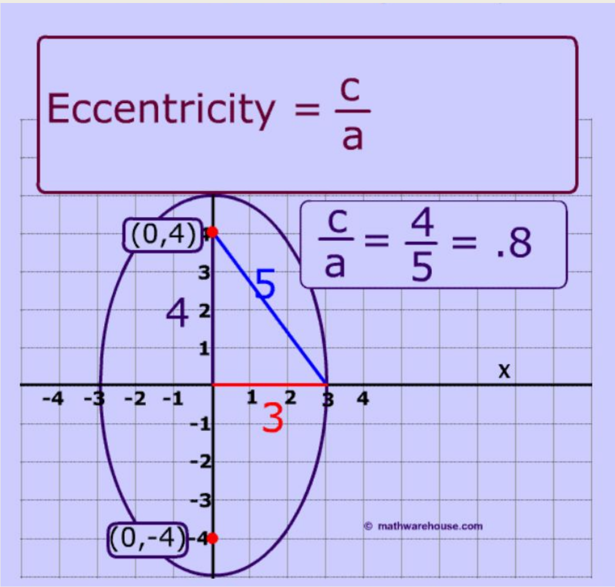
$$r = R/(R+G+B)$$

$$g = G/(R+G+B)$$

where:
R = Red color pixel value
G = Green color pixel value
B = Blue color pixel value
and the range of r,g is [0,1]

Step 2: Features: Eccentricity

Eccentricity is the ratio of the focal distance (distance between focal points) to the center over the major axis length or half. The value is in the interval [0, 1). When it is 0, the ellipse becomes a circle.

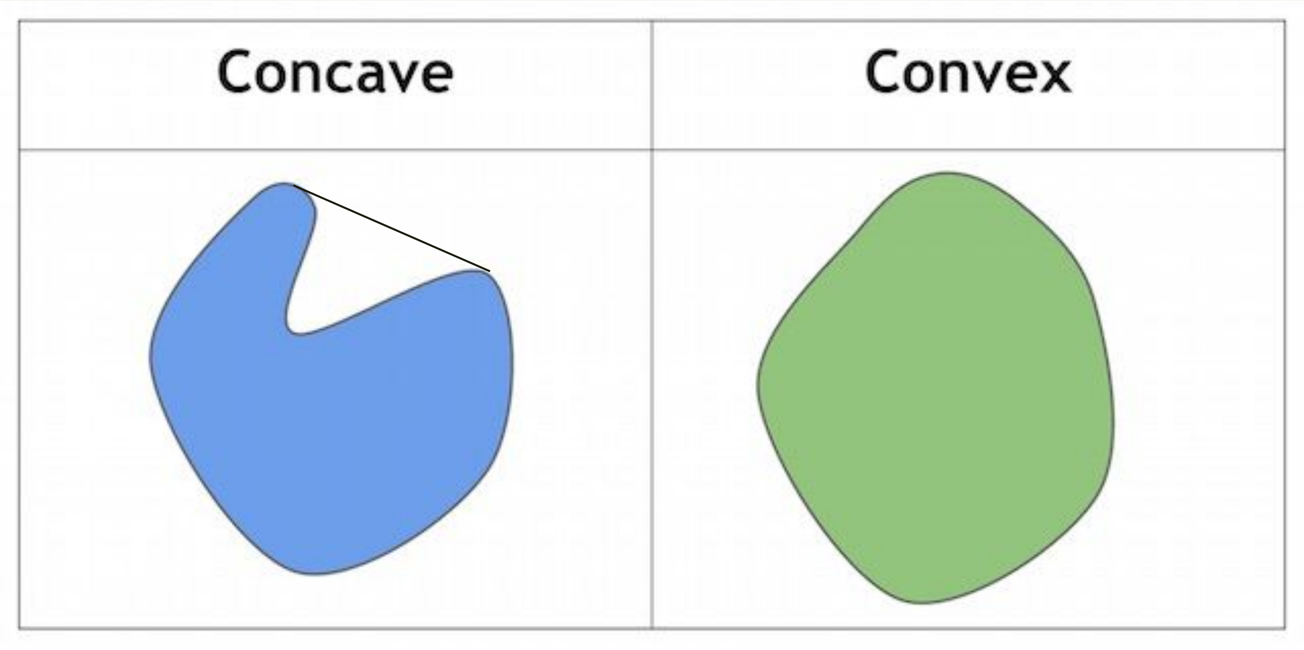


The eccentricity of a circle is zero because both of the focal points are at the center of the circle thus the focal distance is 0 which means the eccentricity is also 0.

https://www.mathwarehouse.com/ellipse/eccentricity-of-ellipse.php

Step 2: Features: Convexity

Convexity is defined as the (Area of the Blob / Area of it's convex hull). A convex hull is the smallest convex polygon than can cover the whole image. Fully convex regions have a value of 1.



The line in the concave image is the edge of a convex hull of that concave image.

https://www.learnopencv.com/blob-detection-using-opencv-python-c/

Step 2: How were they extracted

- Normalized chromaticity values (r and g):

  Images are originally in RGB matrices thus we simply use them
- Eccentricity:

  Using skimage.measure.regionprops(), eccentricity property can be extracted.
- Convexity

  Using skimage.measure.regionprops(), area of region and area of convex hull region can be extracted.

Step 3: Results

- I plotted the features/ classes of the fruits into 2D plots into the ff:

  1.) r-chroma vs g-chroma

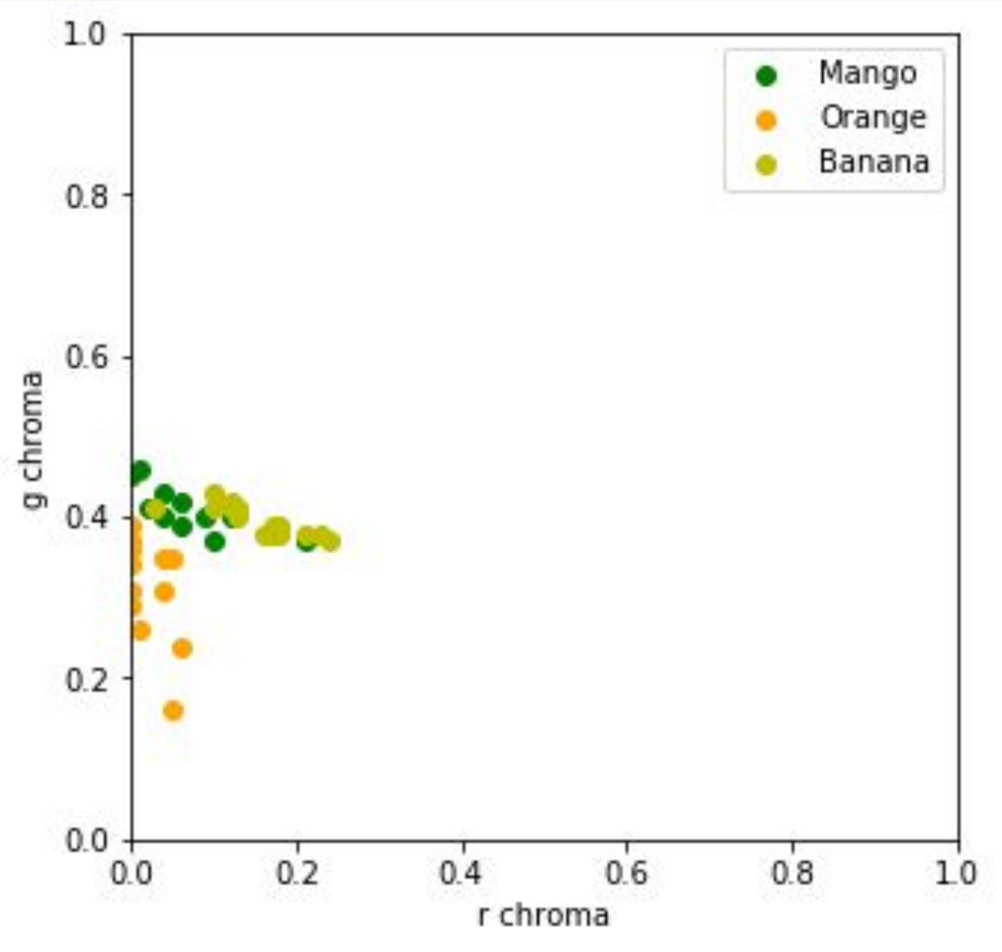
  2.) r-chroma vs eccentricity

  3.) r-chromavs convexity

  4.) eccentricity vs convexity

- I also tried plotting r-value, eccentricity and convexity into a 3D plot.

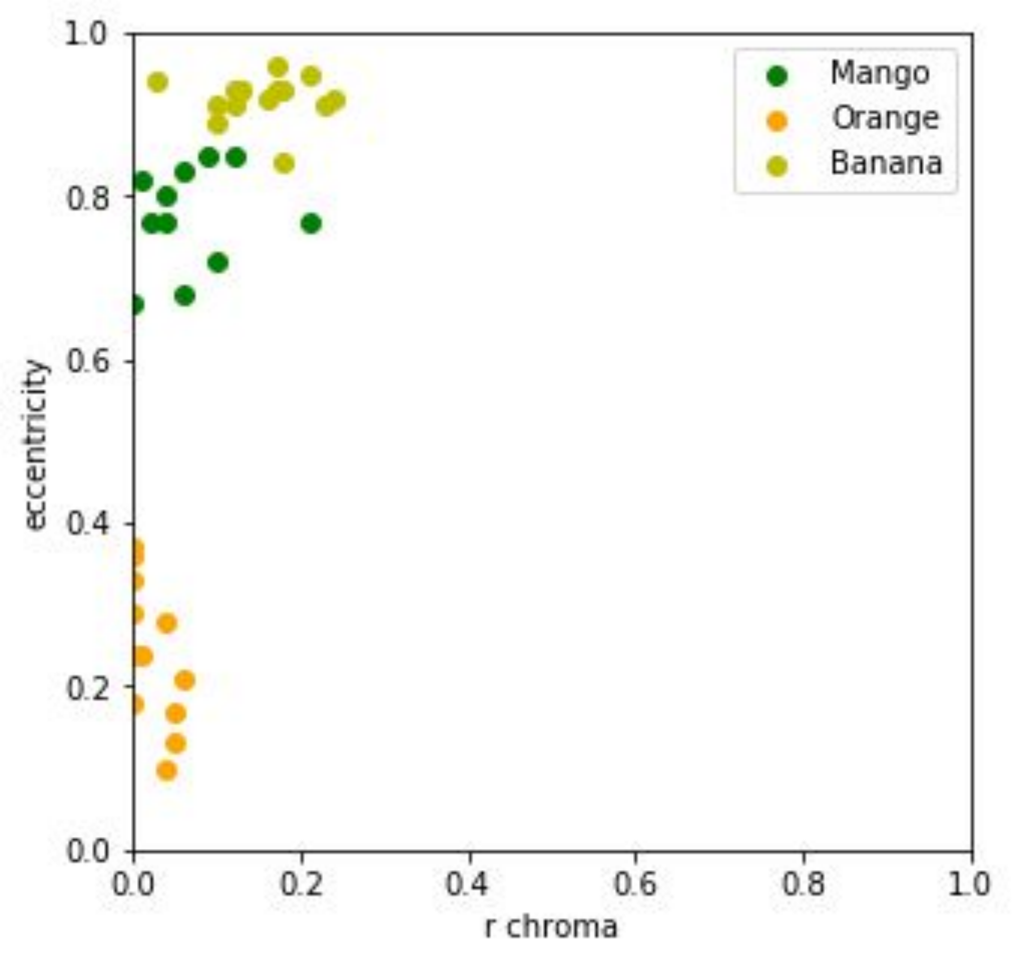
Step 3: r-chroma vs g-chroma



- Class clustering was not observed since bananas and mangos are generally yellow while the color orange is a family of yellow.

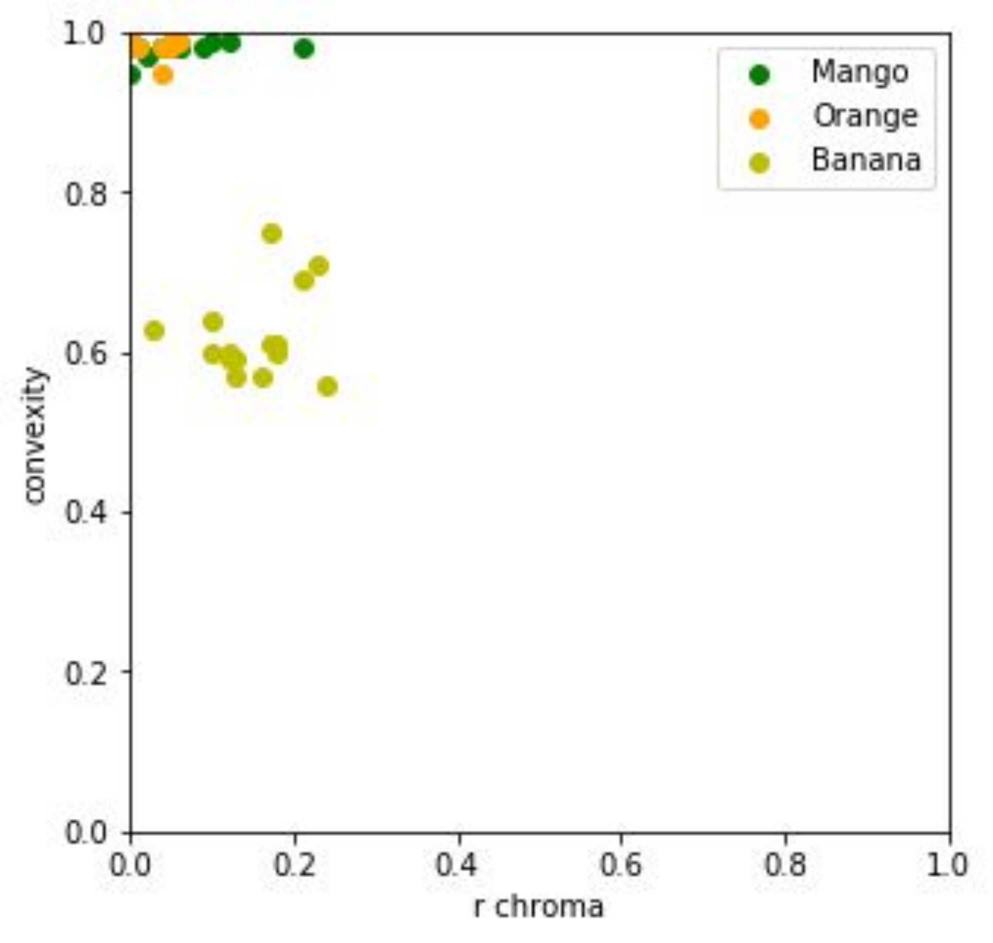
- Mangos and bananas slightly have more green than oranges while bananas slightly have more red than mangos and oranges

Step 3: r-chroma vs eccentricity



- The most distinguishing feature is that bananas have the highest eccentricity since it is the most elliptic of the three.
- Mangos are slightly more circular than bananas while oranges are very circular compared from the two.
- This plot showed more class clustering than the first although r-chroma did little contribution

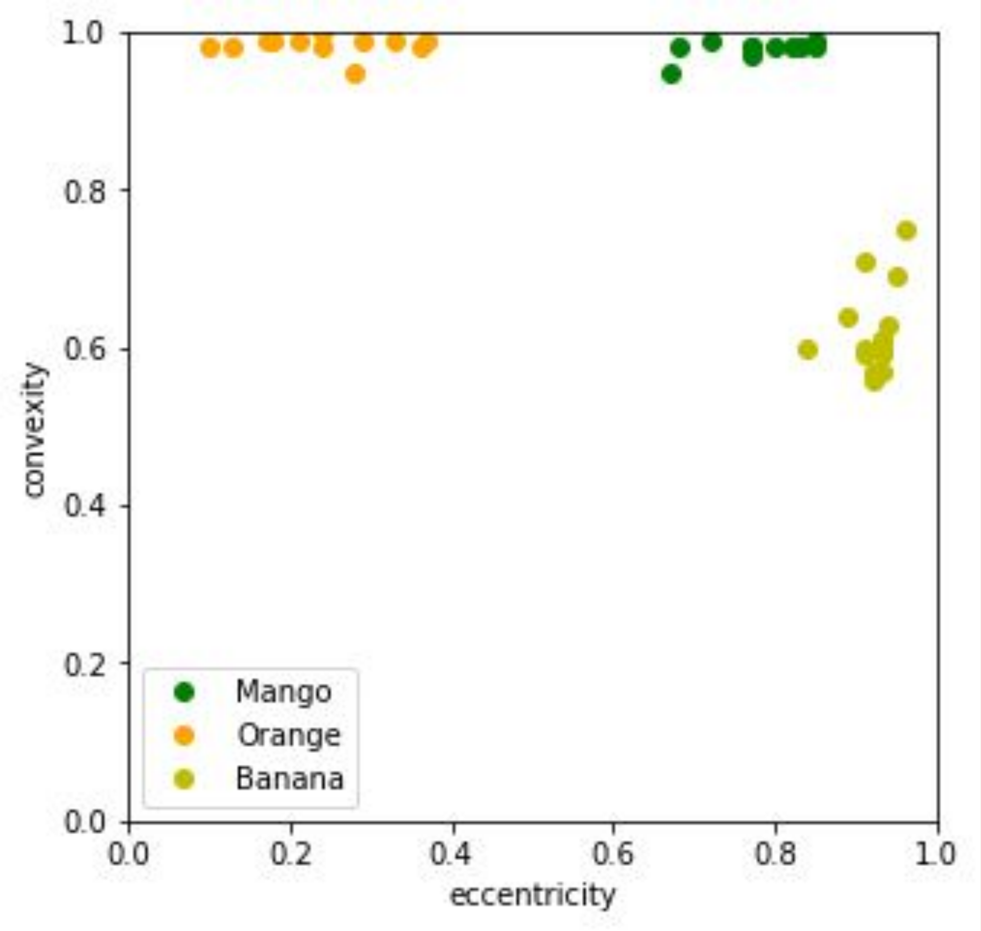
Step 3: r-chroma vs convexity



- The most distinguishing feature is that bananas have the lowest convexity compared to mangoes and oranges since bananas are generally concave.

- Since oranges and mangoes are generally convex, their values are close to 1.

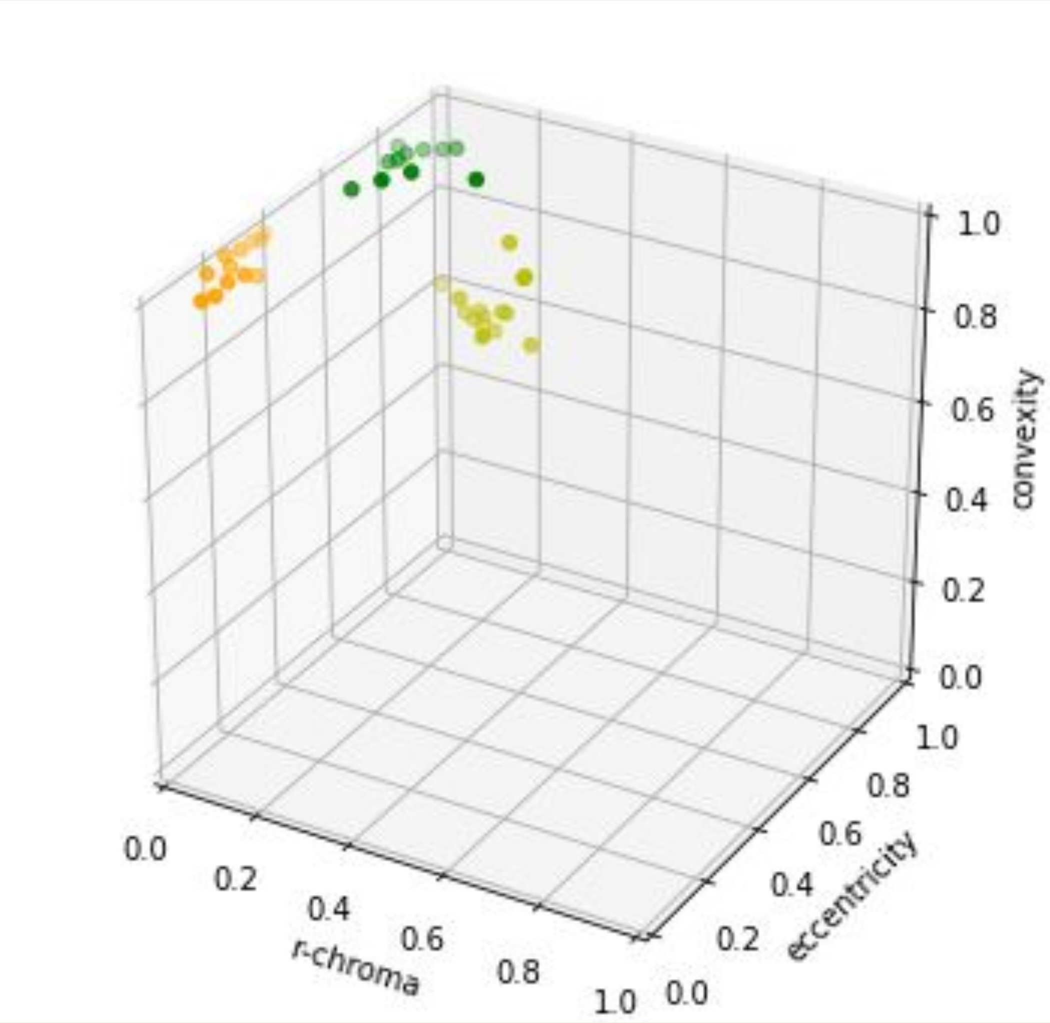
Step 3: eccentricity vs convexity



- Out of the 2D plots, this plot showed the largest separation between clusters and where class clustering is most promising!

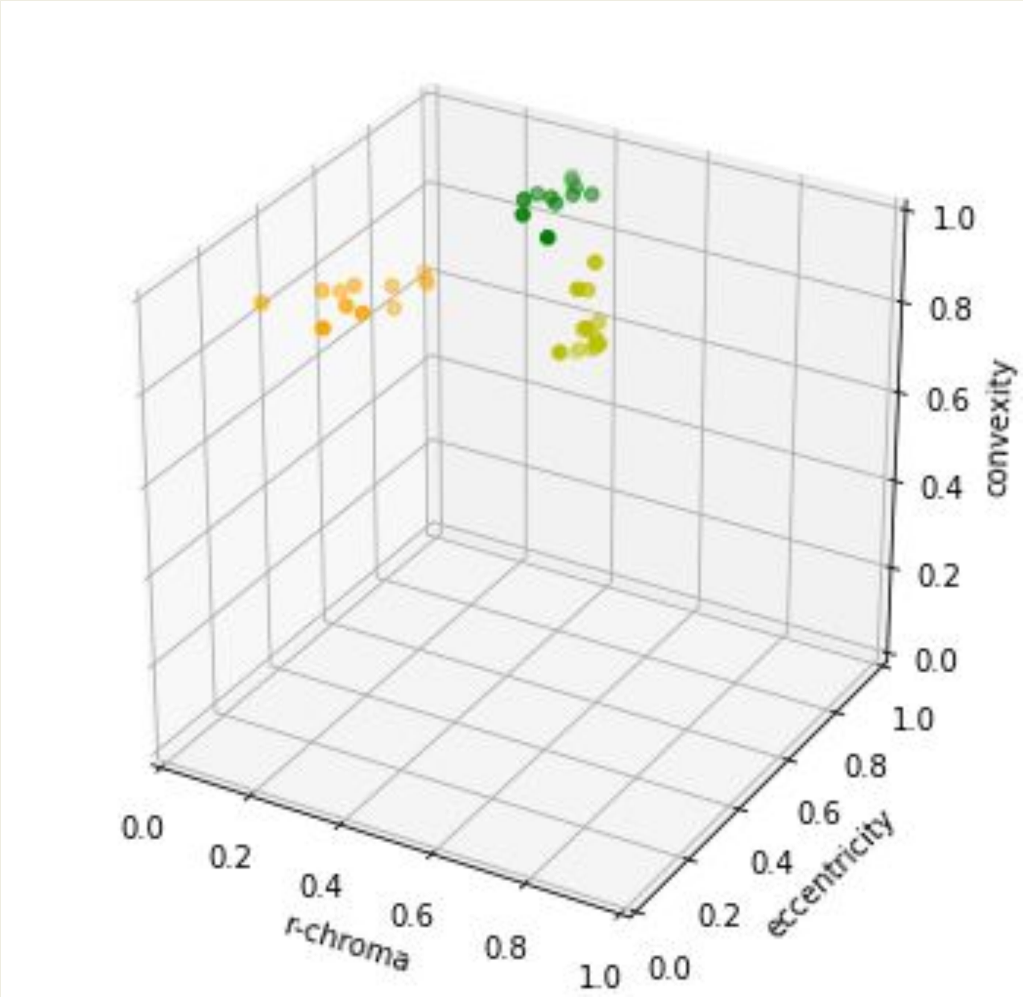
- It showed that bananas have high eccentricity but low convexity, oranges have low eccentricity but high convexity and mangoes have both high eccentricity and convexity.

Step 3: 3D scatter plot (r-chroma)



- In the 3D plot, class clustering becomes more apparent although the clusters seems to be relatively clustered in a region since r-chroma had little contribution to separate the three clusters.

Step 3: 3D scatter plot (g-chroma)



- Same applies to g-chroma.