Activity 12 - Feature Extraction

de Castro, Crizzia Mielle | 2015-08076

Images of Flowers

butterfly peas

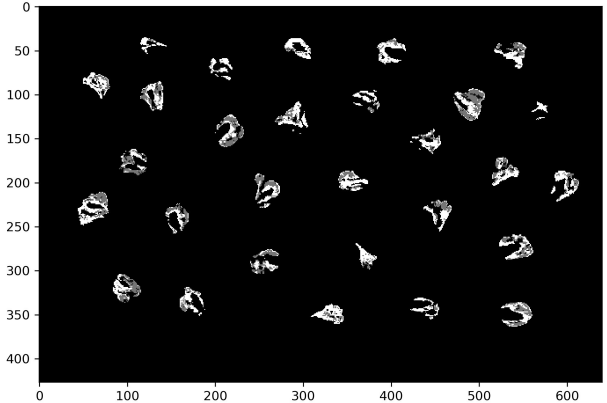
red roses

yellow chrysanthemums

I decided to extract the features of three flowers: butterfly peas, red roses, and yellow chrysanthemums. For this case, I only considered the flower heads. I didn't include the stem or leaves.

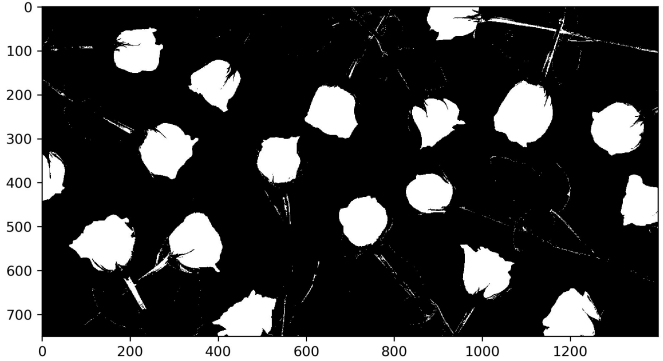
Nonparametric Segmentation

butterfly peas



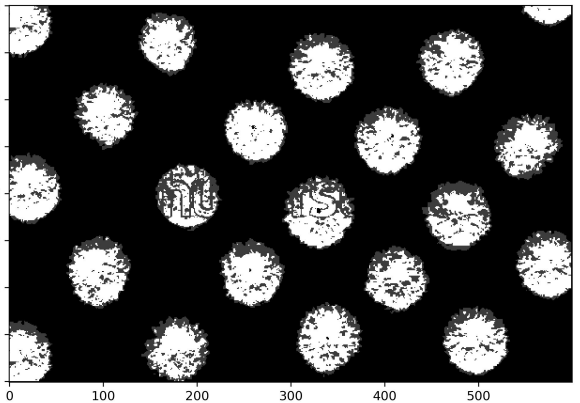
bins = 10

red roses



bins = 2

yellow chrysanthemums

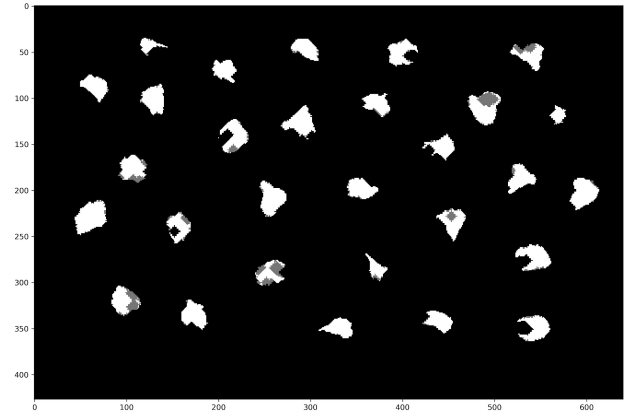


bins = 8

Similar to the previous activities, I applied nonparametric segmentation on the three images to extract the flowers from the background. The patches and number of bins I used for each flower are shown above.

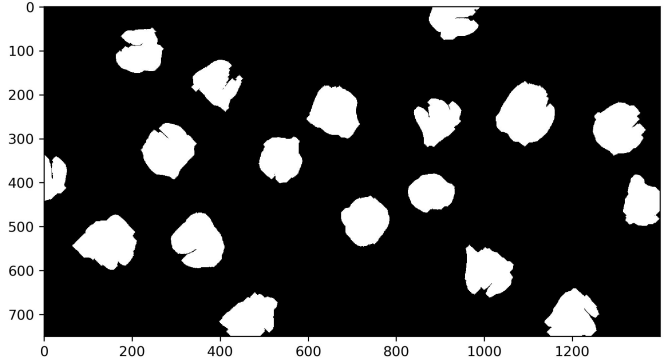
Morphological Operations

butterfly peas



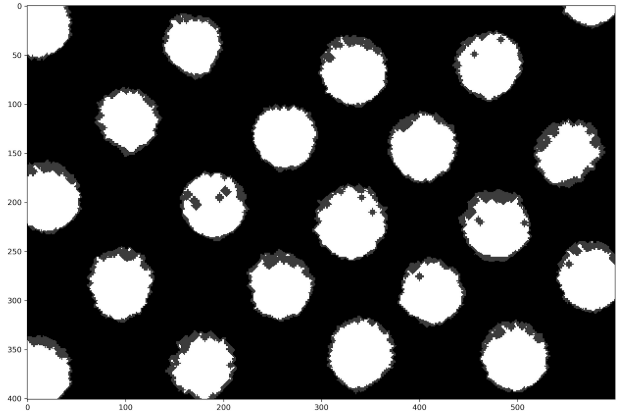
iterations = 3

red roses



iterations = 4

yellow chrysanthemums

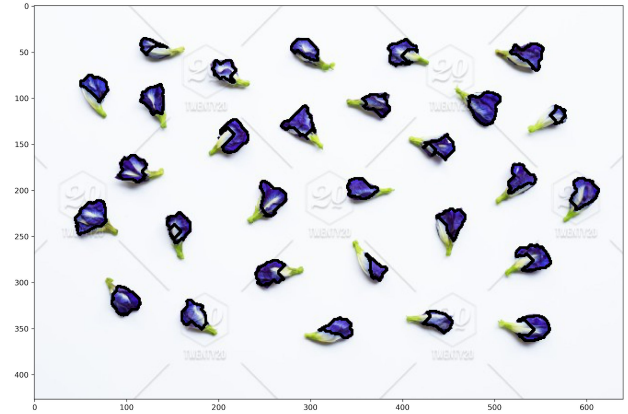


iterations = 2

I used morphological operations on the segmented images. I applied *closing* on the butterfly peas and yellow chrysanthemums to get rid of the holes in the segmented flower heads. I applied *opening* on the red roses to get rid of remnant segmented stems and leaves.

Finding Contours of Cleaned Images

butterfly peas



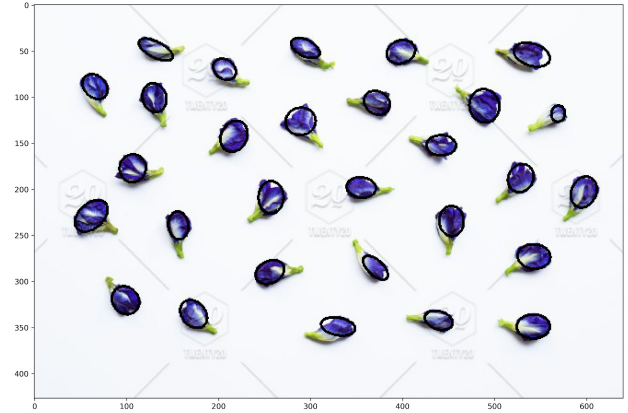
red roses

yellow chrysanthemums

I used *OpenCV's* function *findContours* and *drawContours*[1] to obtain the contours of the flowers from their cleaned binary images.

Fitting an Ellipse to Contours

butterfly peas



threshold = 35

red roses

threshold = 20

yellow chrysanthemums

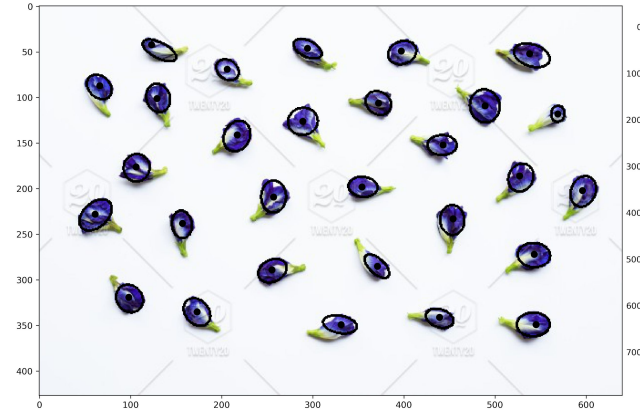
threshold = 30

I used *OpenCV's* function *fitEllipse*[2] to fit an ellipse to the contours I obtained. I had to set a *threshold value for number of points that determine a contour*. This is to prevent fitting ellipses at unnecessary or noisy areas. This function can also output the length of the major and minor axes of the ellipse. I used the following formula to calculate the eccentricity of each ellipse[2]:

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$ where b = minor axis, a = major axis

Calculating Centroid from Contours

butterfly peas

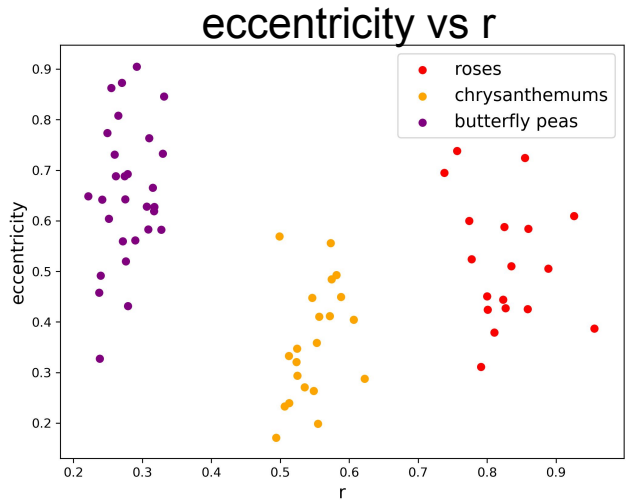


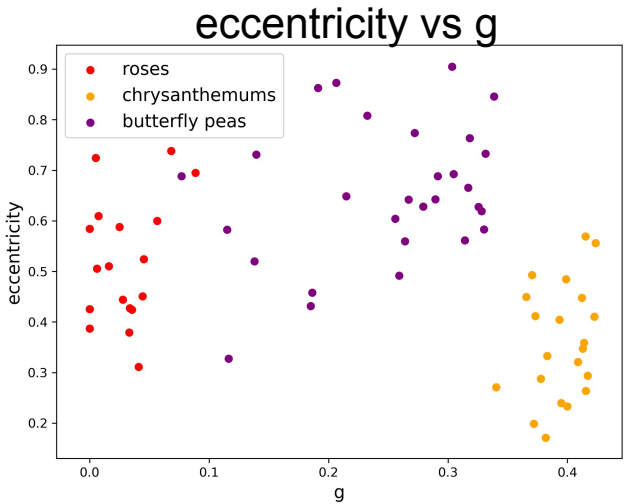
red roses

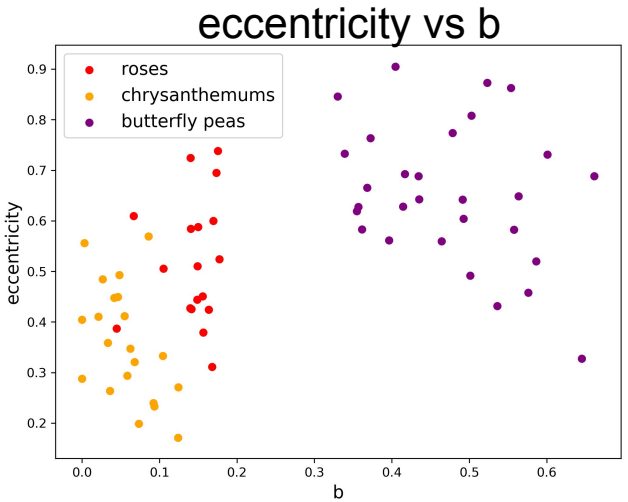
yellow chrysanthemums

I used *OpenCV's* function *moment* to determine the location of the centroid of each contour. I followed the code presented in [2] for calculating the centroid given a contour. I tabulated the NCC rgb values of these centroids in a *csv* file. I set these rgb values, together with the ellipse eccentricities, as the features of each flower.

Plotting Feature Points in 2D Space



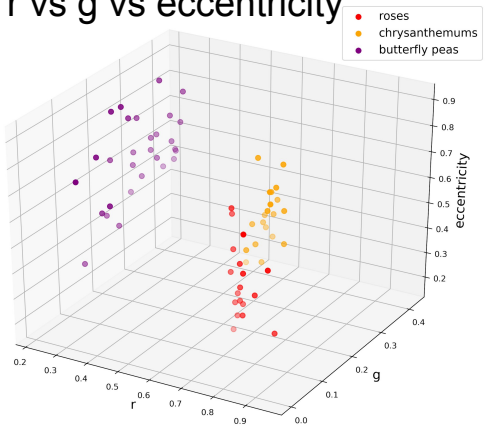




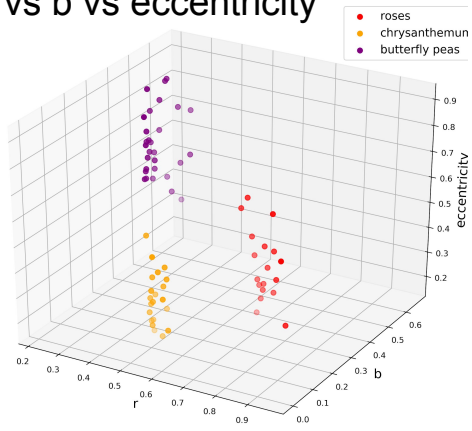
There is a clear **class clustering in the first plot**. All roses have high r values, since they're all red. Butterfly peas have low r values, since they're violet to blue. Chrysanthemums are yellow, a combination of red and green. That's why they're r and g values are close to each other. Butterfly peas generally have high eccentricities because they're more elongated than the other two. Chrysanthemums are almost circular, thus the low eccentricities. Rose eccentricities are somewhere between the two.

Plotting Feature Points in 3D Space

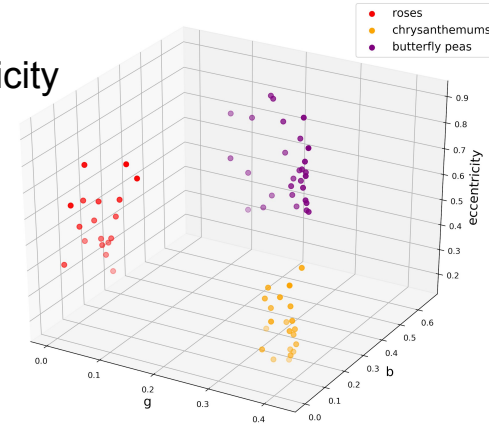
r vs g vs eccentricity



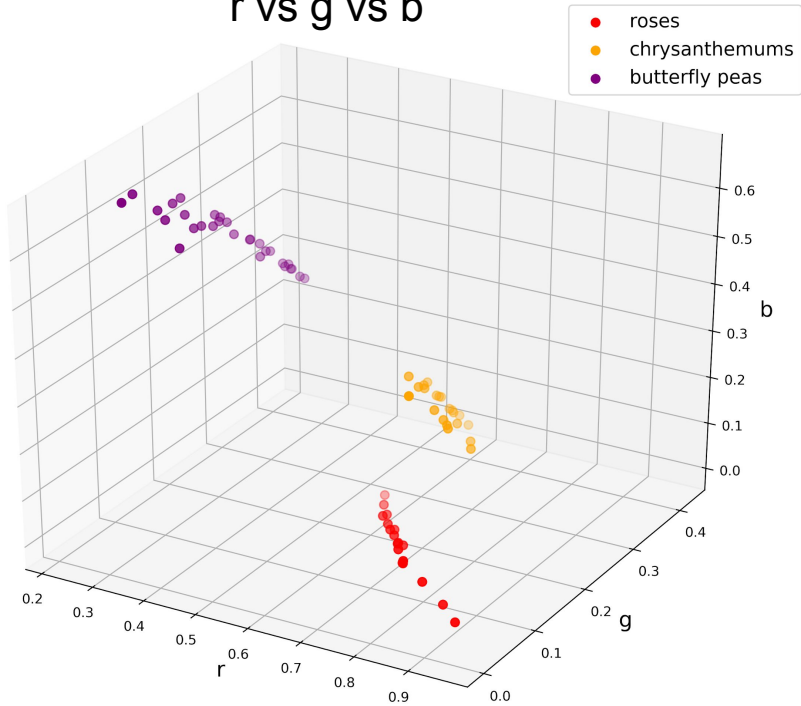
r vs b vs eccentricity



g vs b vs eccentricity



r vs g vs b



The class clustering is clearer when the feature points are plotted in 3D space.

References

1. https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html
2. http://opencvpython.blogspot.com/2012/04/contour-features.html