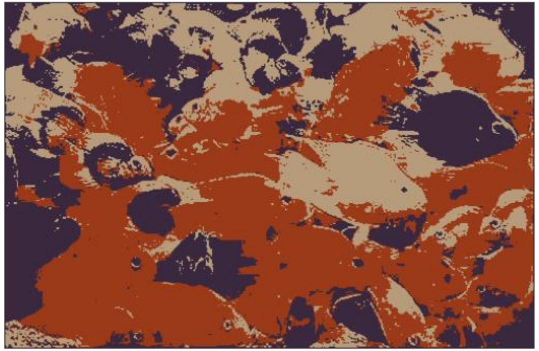
CS 4476 Project 4

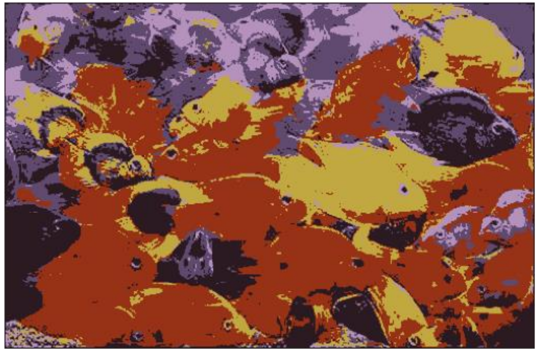
Yunqing Jia
yjia16@gatech.edu
yjia67
903256707

1.1 Color Quantization of RGB Images

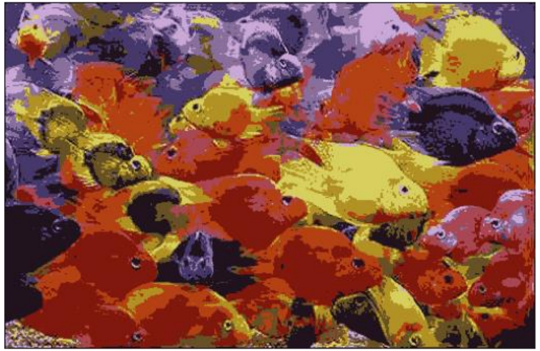
K = 3, err = e^(17.728656492671274)



K = 5, err = e^(17.727821741865014)

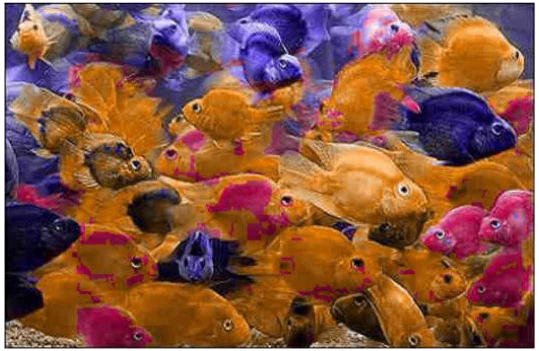


K = 10, err = e^(17.639535401508866)

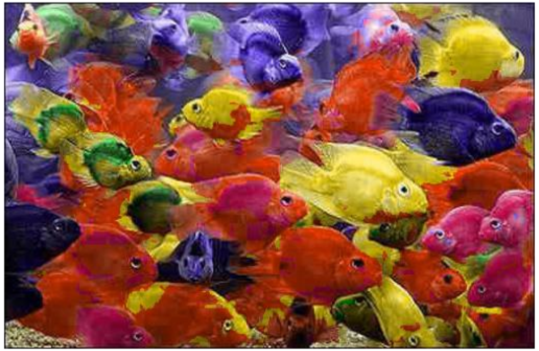


1.2 Color Quantization of HSV Images

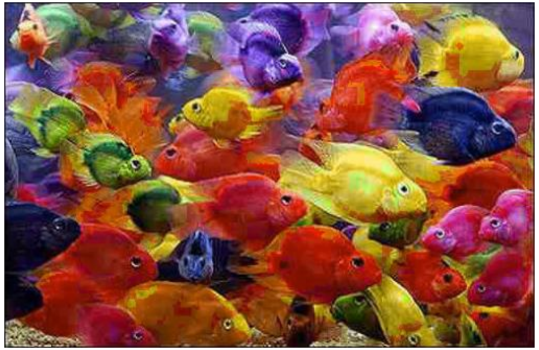
K = 3, err = e^(19.19164769405786)



K = 5, err = e^(18.074701228036147)



K = 10, err = e^(16.981677729401962)



1.3 Logarithmic Quantization Error

k	**RGB**	**HSV**
3	17.729	19.192
5	17.728	18.075
10	17.640	16.982

1.4 Brief Answers

a) Generally, the quantization error decreases with increasing number of bins. The decrease is more evident / drastic when clustering using hue values compared to clustering using RGB pixel values directly.

b) The RGB quantization was highly blurred where fish with similar colors blended together and the details such as texture are not preserved. The HSV quantization, though using the same number of bins (n = [3, 5, 10]), preserves most of the original image because only the hue channel are being reassigned to cluster centers whereas the saturation and the value channels remain the same as the original image.

c) Euclidiean distance or Manhattan distance can also be used as evaluation metrics

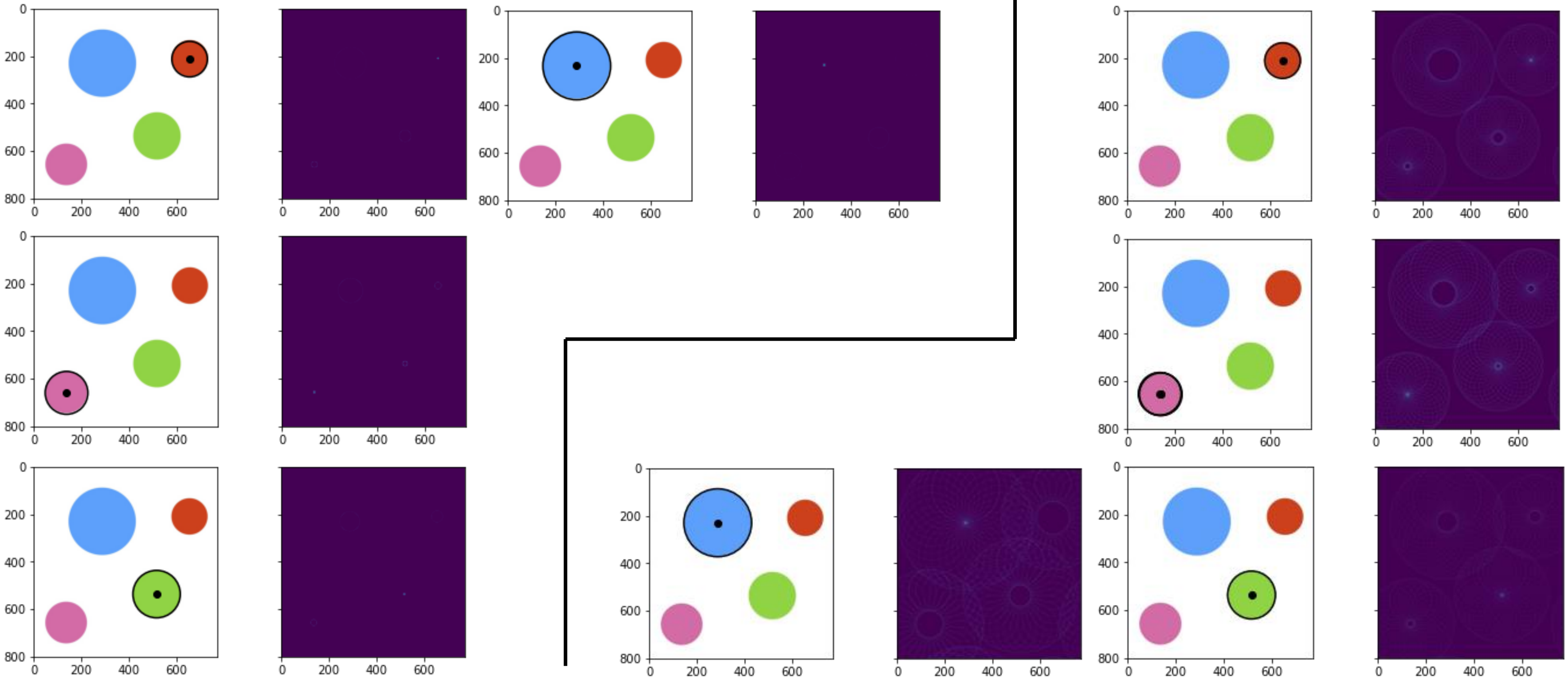
2.1 Circle Detection with Hough Transform

- Convert the image to grayscale and extract potential edge points using feature.canny()
- Initialize a Hough accumulator array that is the same size as the original image
- Discretize the range of theta values to iterate over
- For every potential edge point, iterate over potential theta values, compute the center coordinate (a, b) that coordinate to the x, y, radius, and theta values given
- Check to see if (a, b) falls within the boundary of the array; if so, increment the element in the Hough accumulator array that corresponds to that coordinate by 1
- Points in the accumulator array that have received more votes than some threshold (a percentage of the maximum # of votes received by a singular point) are considered to be circle centers

2.2(a) Circle Detection on Synthetic Images

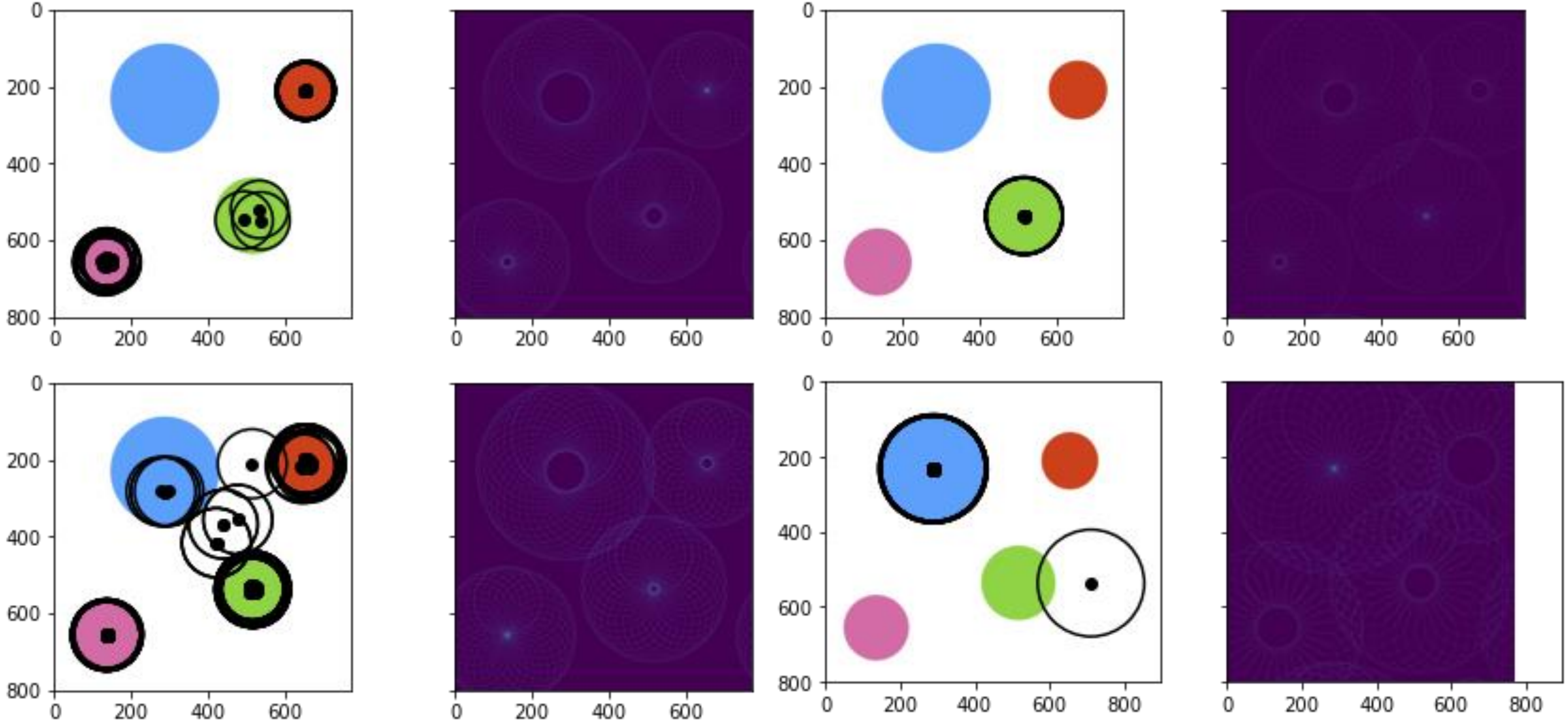
useGradient = True

useGradient = False



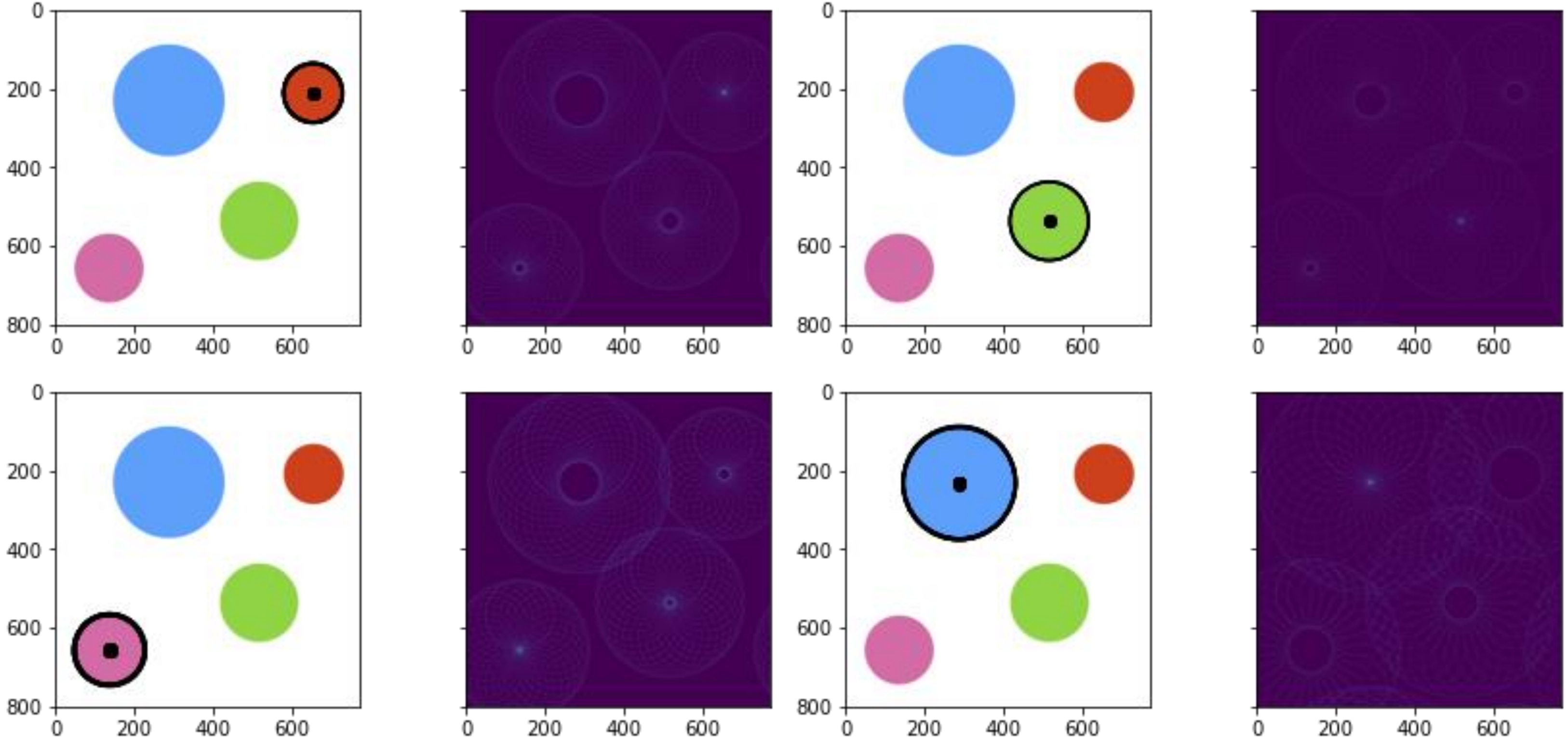
2.2(b) Circle Detection on Synthetic Images

threshold = 0.40



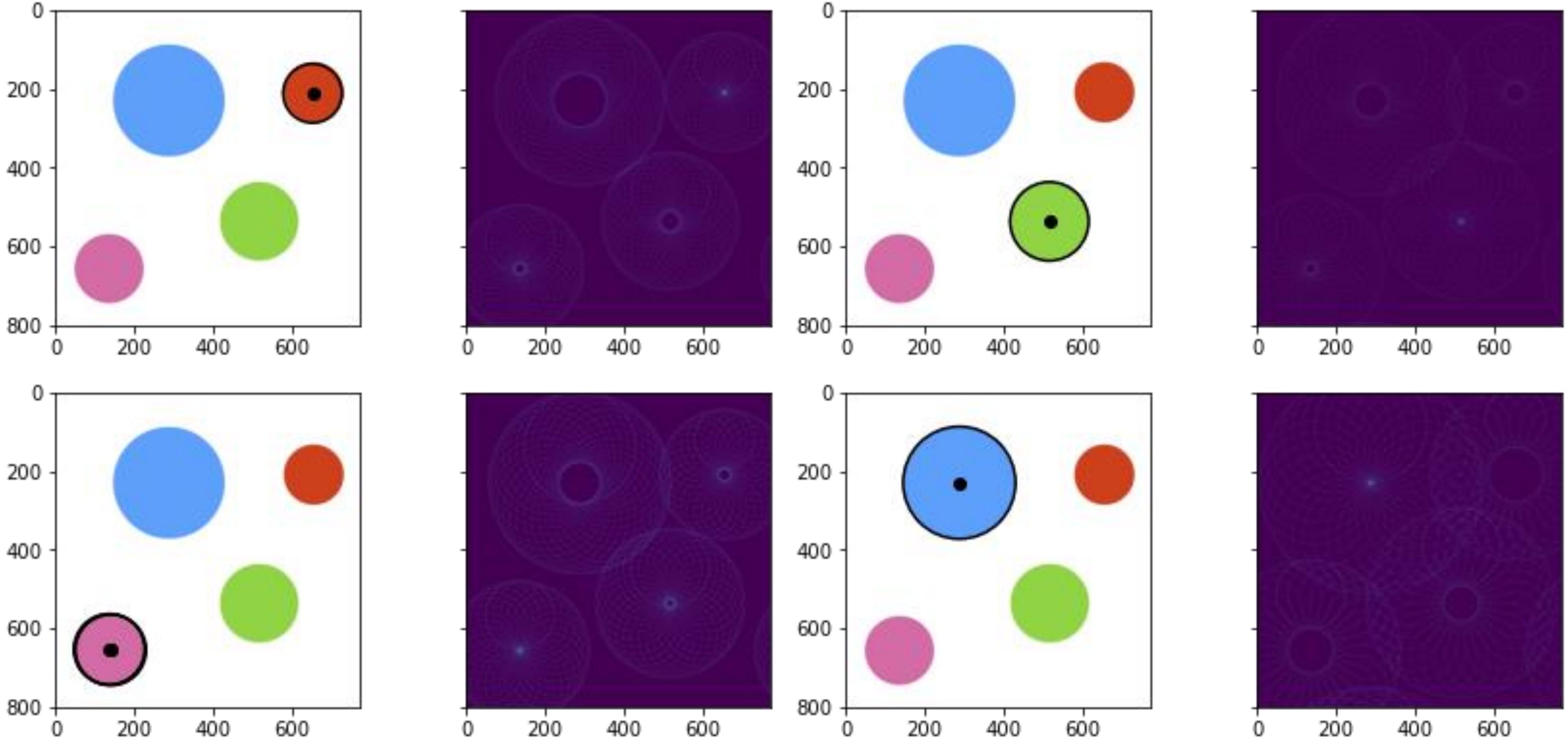
2.2(b) Circle Detection on Synthetic Images

threshold = 0.7



2.2(b) Circle Detection on Synthetic Images

threshold = 0.98

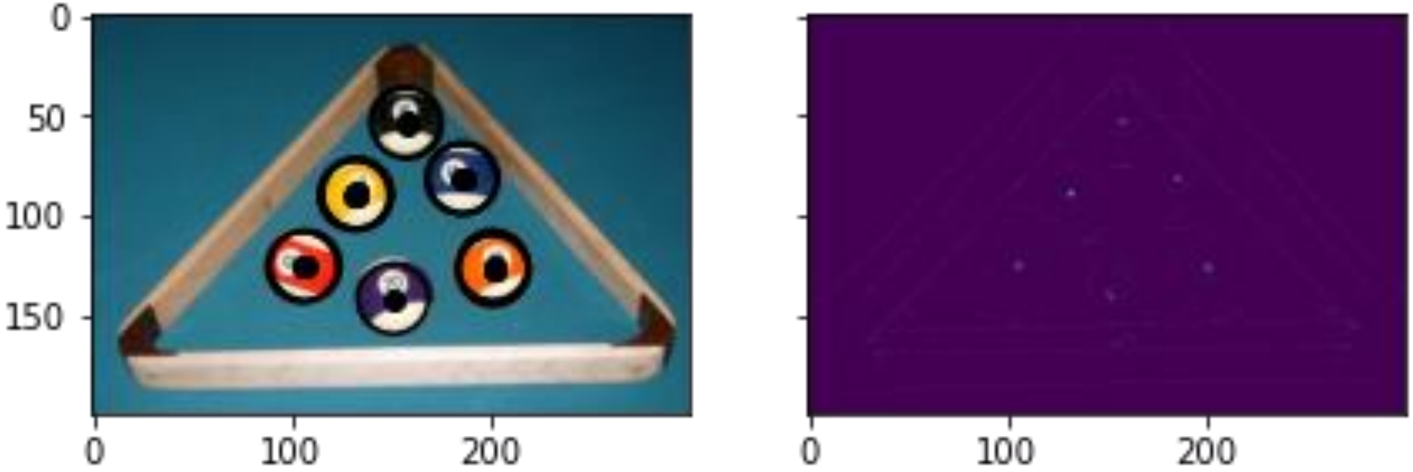


2.2(b) Circle Detection on Synthetic Images

The threshold for determining whether or not a point has received enough votes to be considered a circle center is a hyperparameter we can tune in our Hough transform implementation. The number of centers returned decreases as the threshold increases, thus only returning circle centers that have a high confidence. For a synthetic image like this one, it makes sense to use a high confidence value because the edge is clear and easily detectable. But for a real image, the threshold may need to be lowered if the desired application requires detecting circles that aren’t as obvious.

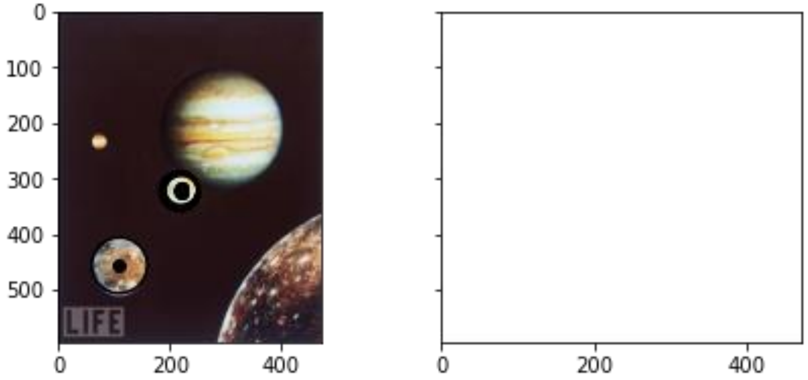
2.3 Circle Detection on Real Images

useGradient = True, radius = 17.6, thresh = 0.3

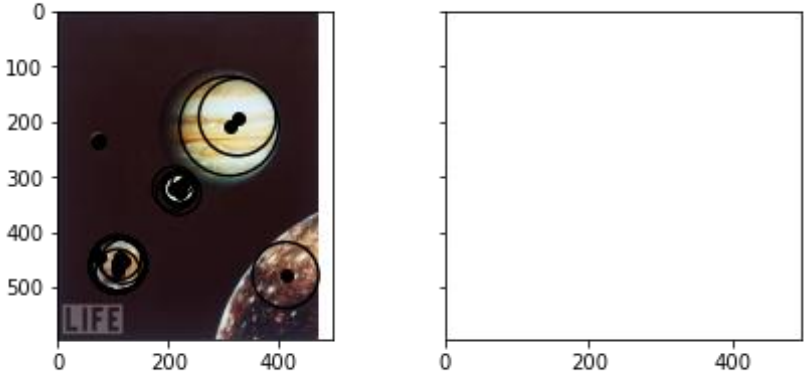


3 Unknown Radii Circle detection

threshold = 0.22



threshold = 0.3


LIFE