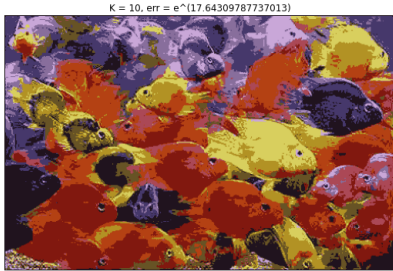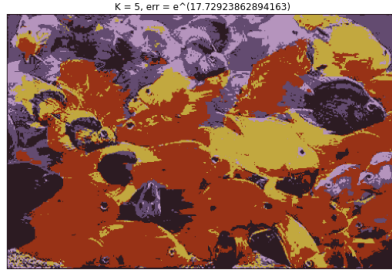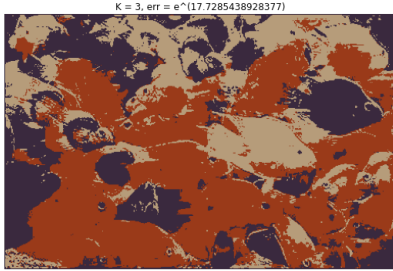# CS 4476 Project 4

Jinghong Peng
jpeng78@gatech.edu
jpeng78
903568613

1.1 Color Quantization of RGB Images

< insert visualizations of the quantized RGB image at k = 3, 5, 10 here as noted proj4.ipynb>

K = 3, err = e^(17.7285438928377)

K = 5, err = e^(17.72923862894163)

K = 10, err = e^(17.64309787737013)

1.2 Color Quantization of HSV Images

< insert visualizations of the quantized HSV image at k = 3, 5, 10 here as noted proj4.ipynb>

K = 3, err = e^(19.19164769405786)

K = 5, err = e^(18.074701228036147)

K = 10, err = e^(16.981677729401962)

1.3 Logarithmic Quantization Error

< Enter values as $\log_e(\text{error})$ >

k	RGB	HSV
3	loge^(17.7285438928377)	loge^(19.19164769405786)
5	loge^(17.72923862894163)	loge^(18.074701228036147)
10	loge^(17.64309787737013)	loge^(16.981677729401962)

1.4 Brief Answers

a) As the number of quantization bins increase, the error gets smaller. More quantization bins means more color can be used in the image, and thus more richly colored image.

b) In HSV color spaces, error reduce faster than RGB space. The image in HSV is clearer than RGB space. The reason is HSV separate color information and color intensity, and the k-means algorithm is run on hue values.

c)
1. structural similarity index measure (SSIM), SSIM is used for measuring the similarity between two images.
2. Compare SIFT features of the two images.

2.1 Circle Detection with Hough Transform

< Briefly explain implementation in concise steps (bullet points / listed steps preferred)>

- Change image to gray image
- Detect the edge of the image
- Iterate through all the edge points
- If the edge point have a high gradient, iterate through all the radians
- Calculate a and b, and increment houghAccumulator of [b, a] by 1
- Get the maximum houghAccumulator and multiply by the threshold
- Filter the houghaccumulator by the new threshold value
- Construct a N x 3 array to store the filtered value

2.2(a) Circle Detection on Synthetic Images

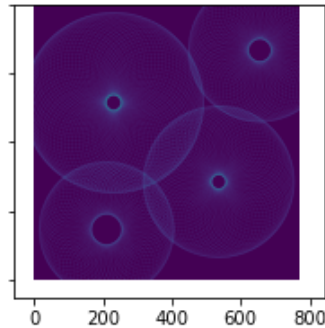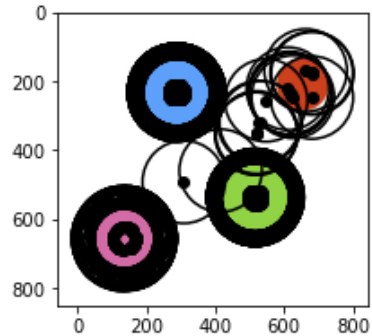< Insert useGradient = True images here>
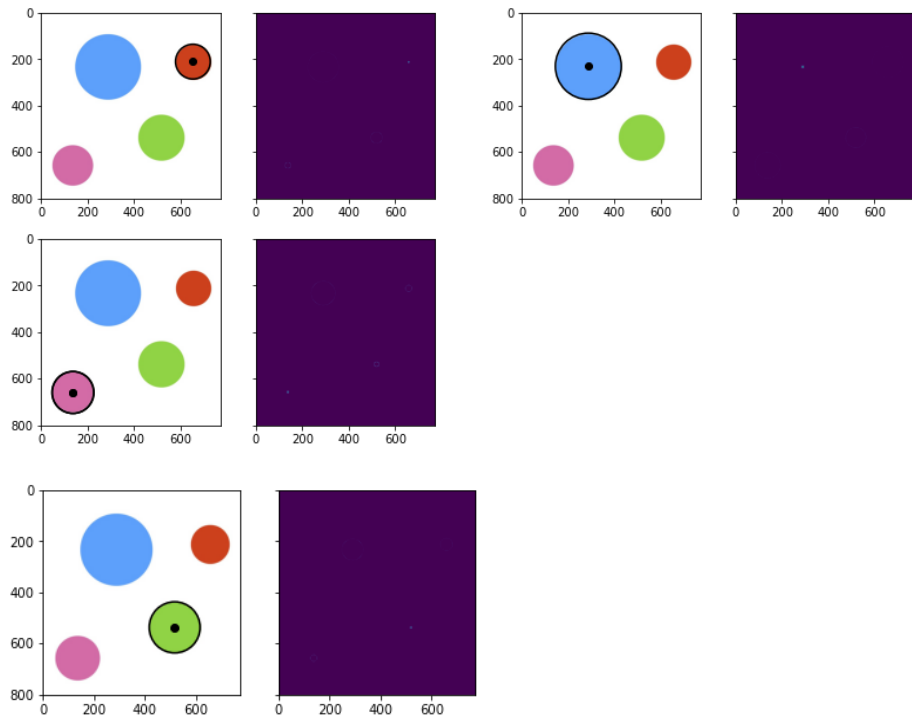
< Insert useGradient = False images here>
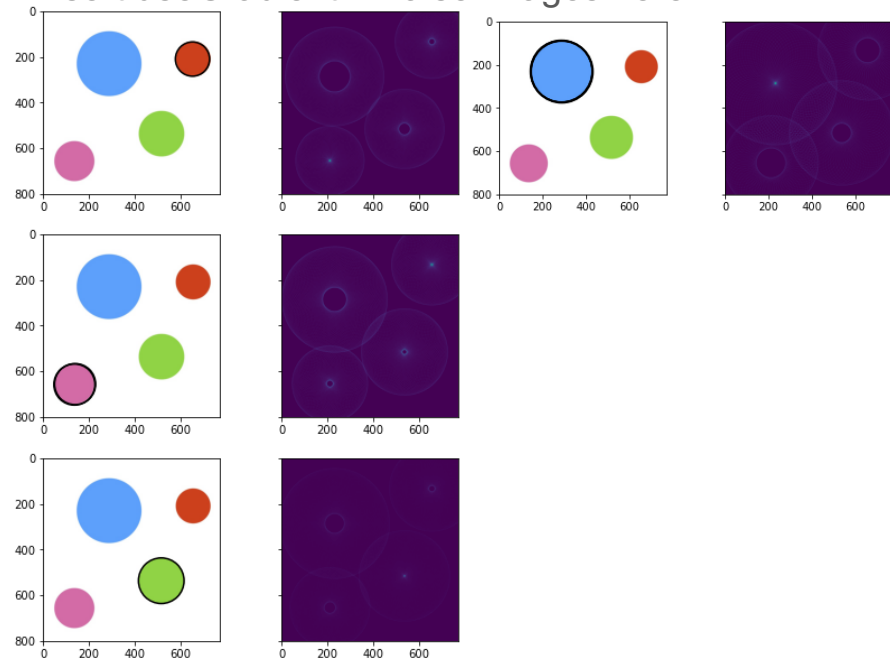
2.2(b) Circle Detection on Synthetic Images

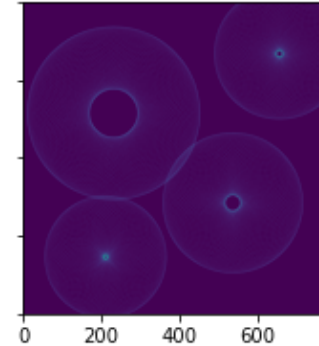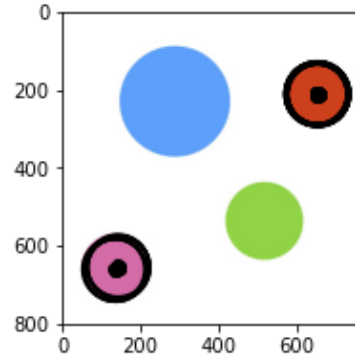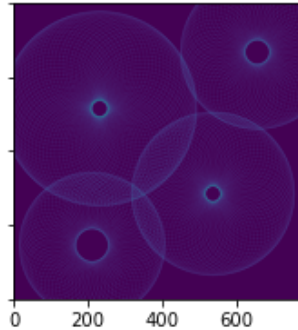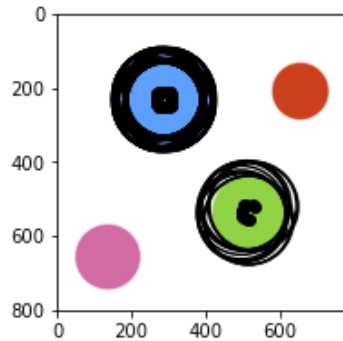< Insert low threshold images and hough accumulator array here>

Threshold = 0.3
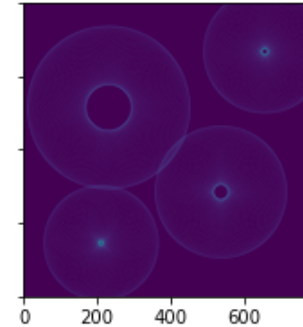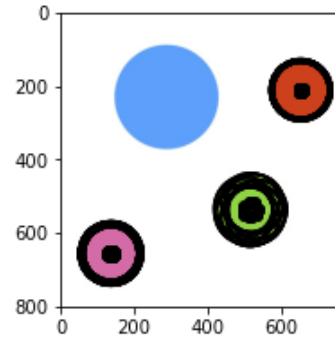
2.2(b) Circle Detection on Synthetic Images

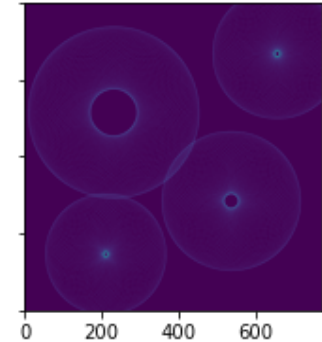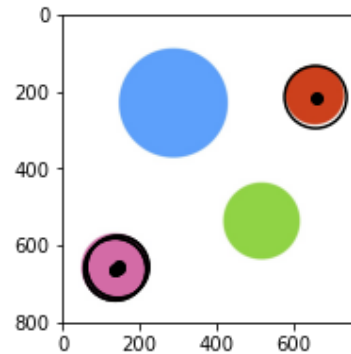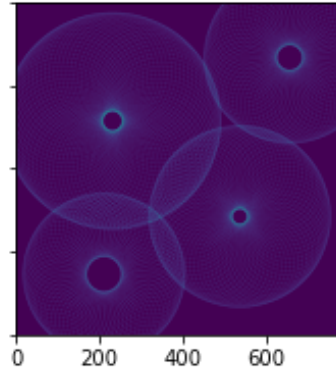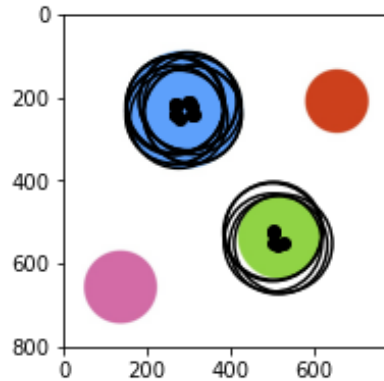< Insert mid-range threshold images and hough accumulator array here>

Threshold = 0.7

2.2(b) Circle Detection on Synthetic Images

< Insert high-range threshold images and hough accumulator array here>

Threshold = 0.9

# 2.2(b) Circle Detection on Synthetic Images

< Explain how results vary with increasing thresholds>

As the threshold increase, the result become more accurate and the number of circles generated is smaller. The circle detection is has better accuracy with high threshold because noises get filtered out.

2.3 Circle Detection on Real Images

< Include image showing detected circles here >
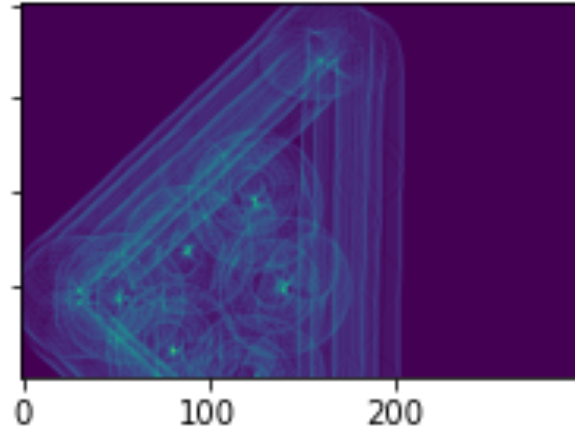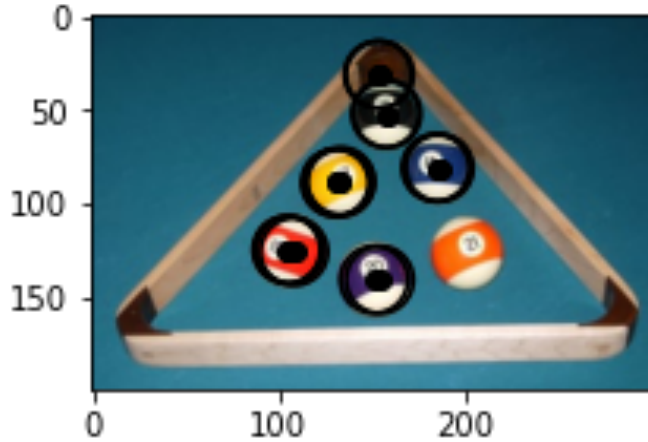
3 Unknown Radii Circle detection

< Include image showing detected circles here >