

CS 4476/6476 Spring 2020

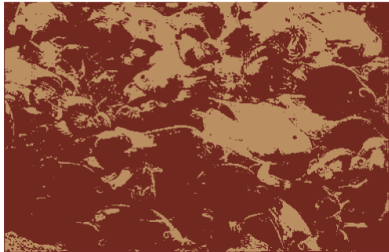
PS4

Sreyans Sipani
ssipani6@gatech.edu
903310164

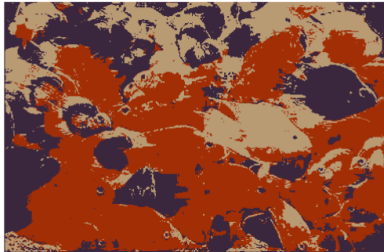
Color Quantization with K-Means

2.1 Color Quantization of RGB images

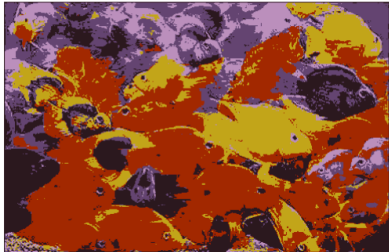
$K = 2$, $\text{err} = e^{-(20.741915091627913)}$



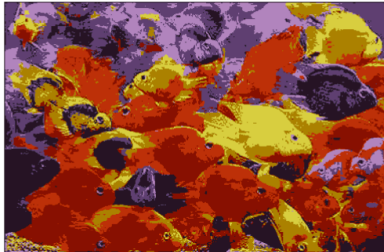
$K = 3$, $\text{err} = e^{-(20.396515020490906)}$



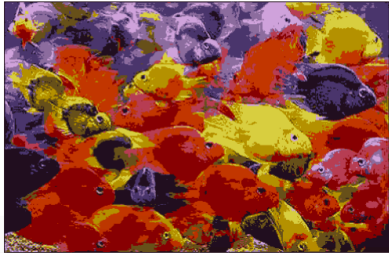
$K = 5$, $\text{err} = e^{-(19.95167762932918)}$



$K = 7$, $\text{err} = e^{-(19.662451398601522)}$



$K = 10$, $\text{err} = e^{-(19.38549622208861)}$



2.2 Color Quantization of HSV images

$K = 2, \text{err} = e^{-(19.813108219240377)}$



$K = 3, \text{err} = e^{-(19.194724734198434)}$



$K = 5, \text{err} = e^{-(18.070853716512776)}$



$K = 7, \text{err} = e^{-(17.51499301850995)}$



$K = 10, \text{err} = e^{-(16.956670095238184)}$



2.3 Quantization error (remember to use the random state initialized in the script).

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

k	RGB	HSV
3	20.396	19.195
5	19.95	18.07
10	19.38	16.956

1.4 Answers (1-2 sentences each)

a) Increasing the number of quantization bins produces richer images as more colors can be used. It also increases the runtime of the algorithm significantly. As one can see, the error reduces as well.

b) The error is lower and reduces faster while the images are clearer and brighter as well. This is because HSV separates the intensity from the color and we only run K-Means on the hue values. Therefore the images look better.

c) Peak Signal to Noise Ratio = $20 * \log(\text{max pixel value}) - 10 * \log(\text{Mean squared error})$ [Monochrome images – can be adapted for color images by averaging MSE over channels]

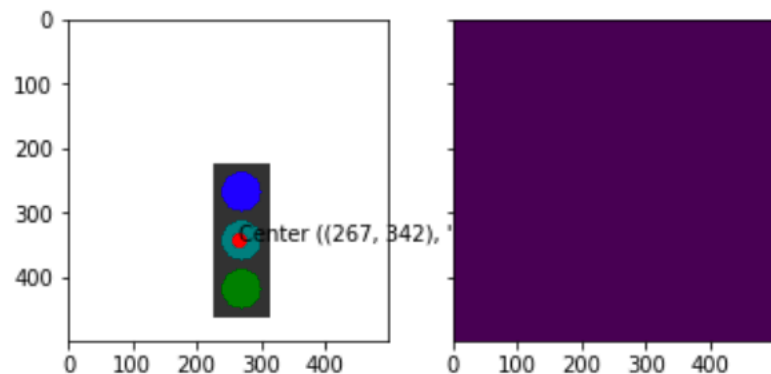
The Structural SIMilarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality. It is implemented in MATLAB and has a rather complicated formula.

Hough Transform

Hough Transform on Generated Images

3.1.1 Traffic lights - Blank

`((267, 342), 'red')`

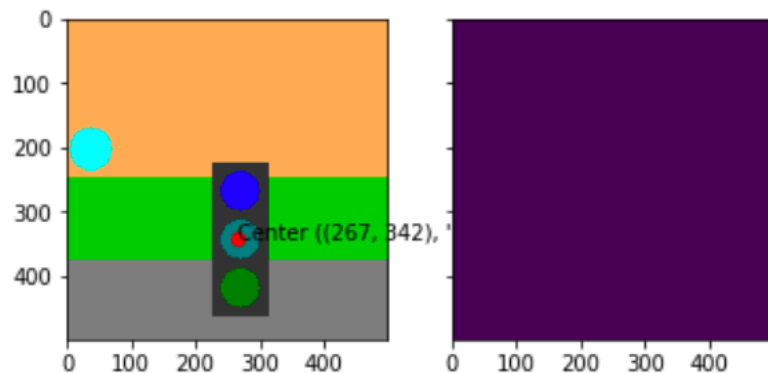


`UnitTest testTrafficLightBlank passed successfully!`

`((267, 342), 'red')`

3.1.2 Traffic lights - Scene

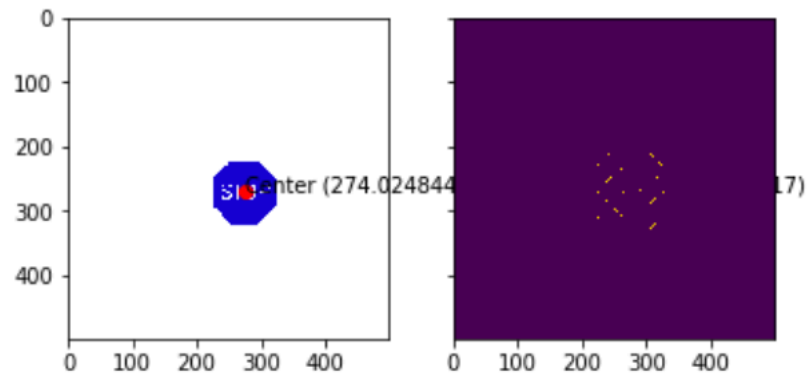
UnitTest testTrafficLightBlank passed successfully!
((267, 342), 'red')



UnitTest testTrafficLightScene passed successfully!

3.2.1 Stop sign detection

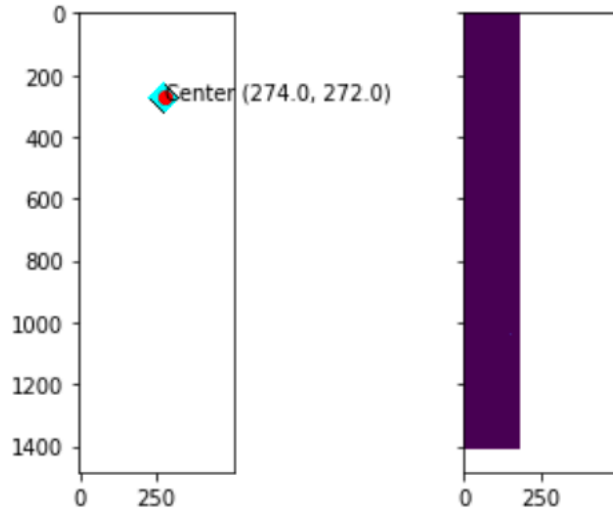
(274.0248447204969, 270.14285714285717)



UnitTest testStopSignBlank passed successfully!

3.2.2 Warning sign detection

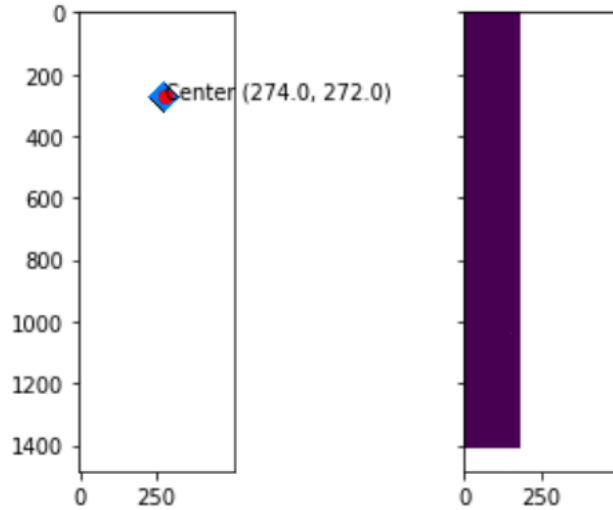
(274.0, 272.0)



UnitTest testWarningSignBlank passed successfully!

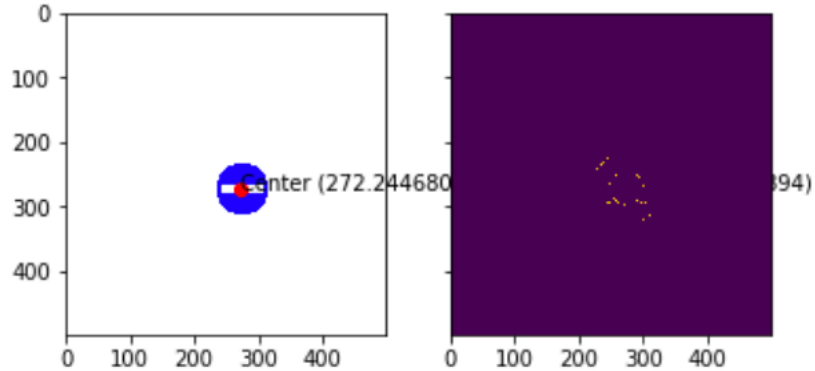
3.2.3 Construction sign detection

(274.0, 272.0)



UnitTest testConstructionSignBlank passed successfully!

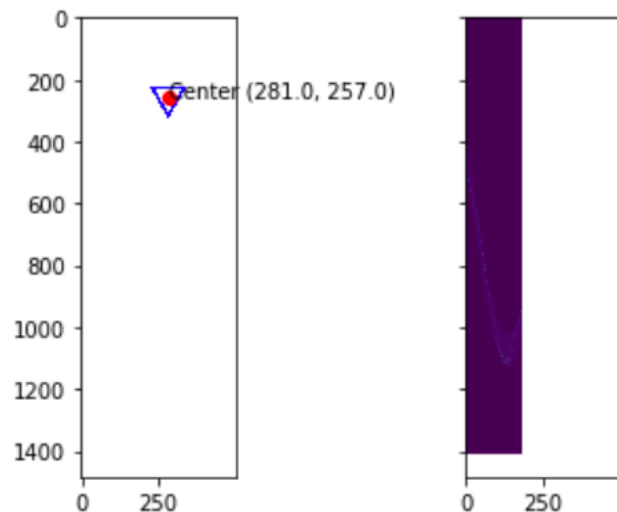
3.2.4 Do not enter sign detection



UnitTest testDoNotEnterSignBlank passed successfully!

3.2.5 Yield sign detection

(281.0, 257.0)



UnitTest testYieldSignBlank passed successfully!

Extra Credit

Hough Transform on Real Images

4.1 Known Radius

<Insert images of detected circles here>

4.1 Known Radius

<Insert images of different thresholds here>

4.2 Unknown Radii

<Insert image here>