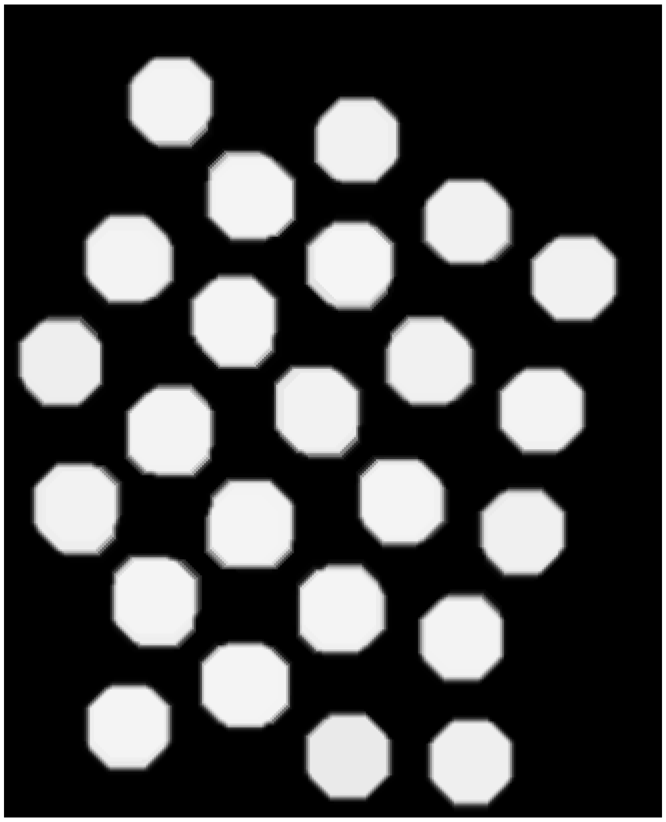
Qamar Syed
ENPM 673
3/16/2022

Midterm Report

Problem 1

Part 1:



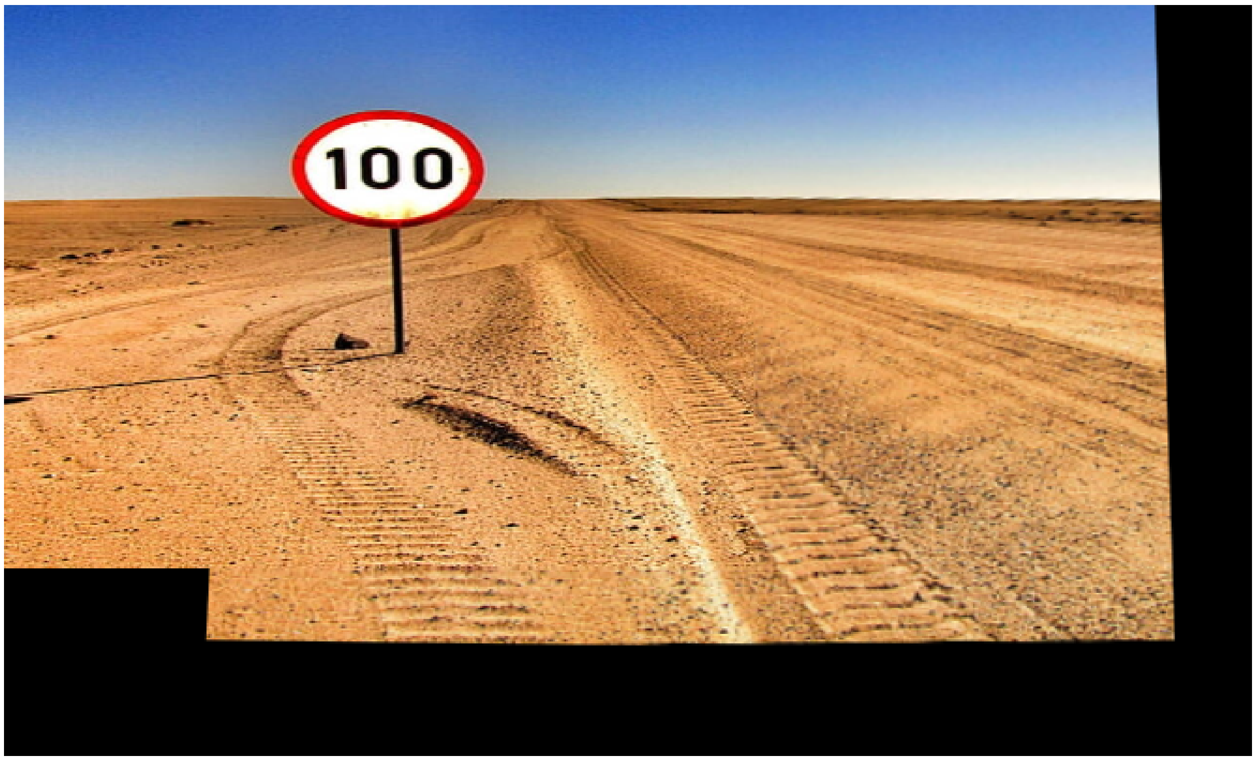
Part 2:
Project returned 24 coins

Pipeline:

1. Initialize 5x5 Disk Kernel
2. Use cv2.erode() and erode multiple times with kernel
3. Use cv2.dilate() with multiple iterations to return to circular shape
4. Use cv2.connectedComponents to find connected components and subtract 1 for the black background to get number of coins
    a. 4-connected or 8-connected didn't seem to affect the results here

Qamar Syed
ENPM 673
3/16/2022

Problem 2:



Pipeline:

1. Find features using cv2.BRISK detection
    a. Wanted to use SIFT but got error saying it was patented, ORB wasn't working all too well either
2. Use cv2.BFMatcher to find all matches on the features detected
3. Sort matches by smallest distance to find the best n number of matches
    a. Program used 25 for n
4. Use cv2.findHomography and cv2.warpPerspective to get the homography from B to A and then apply it onto B
5. Index image array to fill in image A and stitch images together

Qamar Syed
ENPM 673
3/16/2022

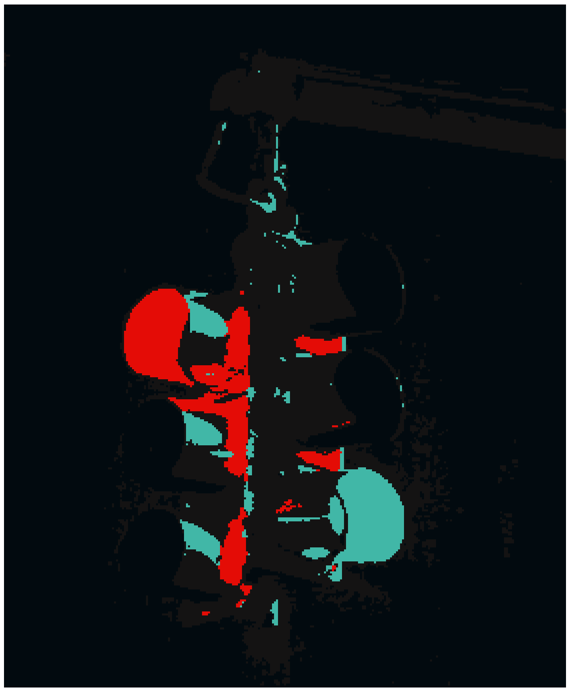
Problem 3:

1. 6 sets of points (pair of image and real world coordinates) are needed
2. Pipeline:
   a. cv2.Canny to reduce image to edges of the grid pattern
   b. Find lines using cv2.findContours or cv2.HoughLines
   c. Find intersections in lines (can use logical operators for contours)
   d. Associate each intersection with a world coordinate from the origin intersection
   e. Pair world coordinates with image pixel coordinates
   f. Use np.linalg.svd to solve for P and Given rotations to find K
3. Steps in math to find K:
   a. Calculate P matrix using x = PX
      i. Fill in world coords for X and homogeneous pixel coordinates for x and use SVD to solve for Ap = 0
   b. Reshape p to 3x4 P matrix and then take the left 3x3 to use as the M matrix for M = KR
   c. Solve for K using Given Rotations in which K = MRxRyRz
4. K = **[[8.55588372e-02, 8.95567396e-05, -4.18473264e-02]**
   [-6.95499291e-19, -8.52106611e-02, -3.21231811e-02]
   [-2.05432702e-21, -1.70856294e-22, -5.26435732e-05]]

   The 0 values are there, a very small number was calculated for them (e <= -19)

Qamar Syed
ENPM 673
3/16/2022

Problem 4:



Clustered image can be variable (image above isn't always returned, often the red blends into the yellow or the blue into the black)

Pipeline:

1. Defined function to calculate Euclidean distance between colors
2. Defined class Cluster to hold current mean color and previous mean color, initiate with random color, and have function to add colors to current mean
3. Initiated list of cluster classes up to K number and an empty list for 'finished' clusters
4. Looped through each x and y data point as long as there was a Cluster object in the clusters list
5. Closest cluster was found for each data point and the data was added to that cluster's mean
6. After going through every data point, clusters mean values were compared to previous mean values, if they did not change after going through all the data, they were popped into the finished list
7. If they did change, the loop continued and data points were continuously evaluated until the clusters were stagnant
8. Each pixel on the original image was mapped to the closest end cluster to return final image