Katherine Banis  
CMPSC 497  
Lab#1A (part 1): Round (Circular) Object Detection  
09/12/2024  
  
Objective: Analyze an image of real objects on a table with varying degree of roundness and label objects with a degree of roundness factor.

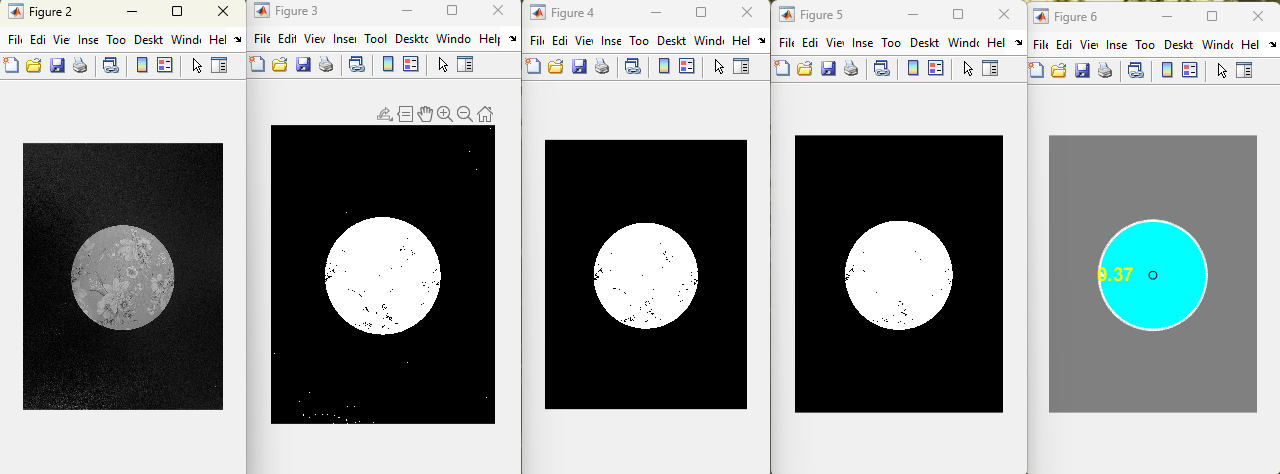
Materials: MATLAB, objects of varying degree of roundness, phone for camera

3 Test Cases (images):

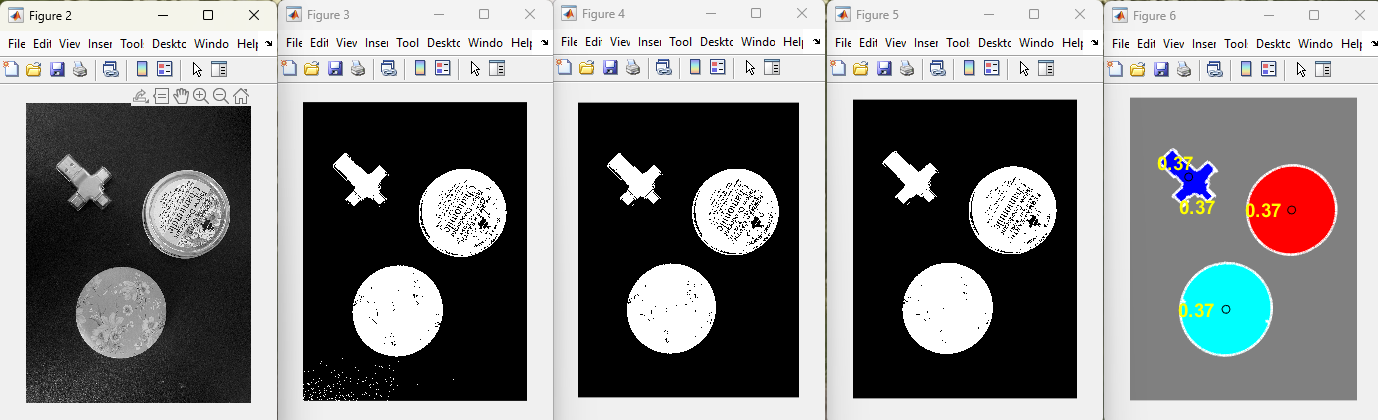
|  |  |  |
| --- | --- | --- |
| Image 1 | Image 2 | Image 3 |

Results:

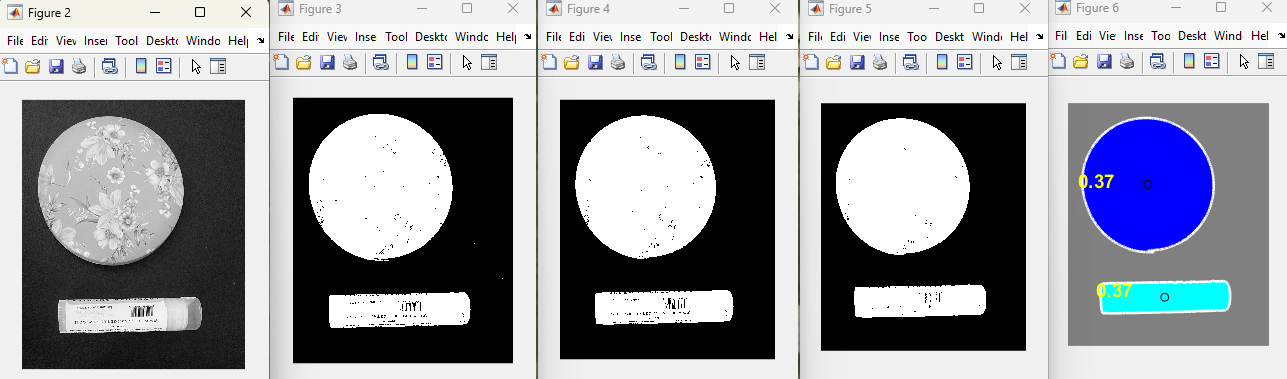
1. Image 1  
   Total number of objects: 1  
   Total number of round objects: 1



1. Image 2  
   Total number of objects: 4  
   Total number of round objects: 3



1. Image 3  
   Total number of objects: 2  
   Total number of round objects: 2



Conclusion:

In two of the three cases, the program was accurate in classifying the number of objects. For the first image, the program correctly recognized the object and correctly classified it as a round object. In the second image, the program incorrectly identified one extra object, potentially confusing the usb drive as two objects. The program also incorrectly classified an extra round object. For the third image, the program correctly recognized two objects, however it also misclassified the cylindrical object as round. My assumption is the program had difficulty differentiating between round and cylinder-shaped objects, especially in 2D images where it may appear round.

I also experienced trouble with background noise. White marks and light shining on my black deskmat were constantly being recognized as round objects. To fix this issue, I kept adjusting the bwareaopen function parameter from a lower threshold to 200.

1. Can this algorithm be used to count the number of M&Ms? Discuss.
   1. Yes, I think this algorithm can be used to count the number of M&Ms, because it can count the number of objects. Some modifications may need to be made such as thresholding value and size filtering.

MATLAB script:

% access original rgb image

rgb = imread('img1.jpg');

imshow(rgb);

% grayscale

gray = rgb2gray(rgb);

threshold = graythresh(gray);

figure, imshow(gray)

% binary image

bw = im2bw(gray, threshold);

figure, imshow(bw)

% remove all objects with < 30 px

bw1 = bwareaopen(bw, 200);

figure, imshow(bw1)

% fill gap

se = strel('disk', 2);

bw2 = imclose(bw1, se);

figure, imshow(bw2);

% fill any holes

bw3 = imfill(bw2, 'holes');

figure, imshow(bw3)

% get px for boundaries of each object

[B,L] = bwboundaries(bw3, 'noholes');

% L = label matrix - try imtool(L) and imshow(L)

% display L and draw each boundary

imshow(label2rgb(L, @jet, [.5 .5 .5]))

hold on % allow graphics to be added to same plot

for k = 1 : length(B) % length(B) = number of objects

boundary = B{k}; % B = cell data type (set)

plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth',2)

End

% find area in px and centroid (x,y) for each object in L

stats = regionprops(L, 'Area', 'Centroid');

% arbitrary value (change as needed)

threshold = 0.1;

% loop over each object (each object has a boundary)

roundObjects = 0;

for k = 1 : length(B)

boundary = B{k};

delta\_sq = diff(boundary).^2;

perimeter = sum(sqrt(sum(delta\_sq, 2)));

area = stats(k).Area;

metric = (4 \* pi \* area) / perimeter^2;

if metric > threshold

centroid = stats(k).Centroid;

plot(centroid(1), centroid(2),'ko');

roundObjects = roundObjects + 1;

end

text(boundary(1,2)-35,boundary(1,1)+13,metric\_string,'Color','y','FontSize',14,'FontWeight','bold')

end

% list total number of objects

fprintf('Total number of objects: %d\n', length(B));

% list total number of round objects (using threshold)

fprintf('Total number of round objects: %d\n', roundObjects)