Final project script 1

Step 1. Localize the centroid of each coin

Your task in this step is to segment the coins from the background in the image and find the centroid of each

- Otsu threshold the image to create in initial, but noisy segmentation. Use your 'OtsuThreshold' funct bottom of the script. Store the result in **msk**.
- Filter the mask to fill all the holes and remove all the false positives. One option is to perform a dilati segmentation. You need to choose a structuring element just large enough to fill in all the holes. If y an erosion operation on **msk_dil** to separate the coins that were joined together by dilation so that a would choose a structuring element large enough to disconnect each coin. Storing the result in **msk**
- Use the built-in function `regionprops` to find the centroid of each coin as the centroid of each con closest integer, in [Nx2] matrix **centroid**, where N is the number of components and the columns co number of pixels) of each component in [Nx1] vector **component_size**.

It is recommended that you write the code in chronological order and run the script each time you create a n creating **msk**, and you will see your initial thresholding result displayed in a figure. All assessed variables ha errors while you are developing your script.

Script ?

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```
% Define the filter size we will use in step 2:
filtsize = 85;

% Creating test image 'im' by splicing together two built in images.
% Also zero-padding (adding zeros around the border) with half the
% filter size (filtsize) we will use so that the filter could be
% centered on any actual image pixel, including those near the border.
% 'coins.png' contains bright nickels and dimes on a dark background
% 'eight.tif' contains dark quarters on a bright background, so we invert it
% to match 'coins.png'
im1 = imread('coins.png');
[r,c] = size(im1);
im2 = imread('eight.tif');
[r2,c2] = size(im2);
filtsizeh = floor(filtsize/2);
im = zeros(r+r2+filtsize,c+filtsize);
im(filtsizeh+1:filtsizeh+r+r2,filtsizeh+1:filtsizeh+c) = [im1;255-im2(:,1:c)
[r,c] = size(im);
imagesc(im);colormap(gray);title('test image');axis equal;

% Initializing assessed/displayed variables as empty so that code is executa
msk=[]; msk_dil=[]; msk_dil_erd=[]; centroid=[]; component_size=[];

```

```
%%%%% 1. Localize the centroid of each coin
% Otsu threshold

[msk,~] = OtsuThreshold(im);

figure; imagesc(msk); colormap(gray); title('Otsu'); axis equal;

% Dilate

structuring_element = ones(9,9);
msk_dil = imdilate(msk, structuring_element);

figure; imagesc(msk_dil); colormap(gray); title('Dilated'); axis equal;

% Erode

structuring_element = ones(23,23);
msk_dil_erd = imerode(msk_dil, structuring_element);

figure; imagesc(msk_dil_erd); colormap(gray); title('Eroded'); axis equal;

% Connected components to get centroids of coins:

connected_comps = bwconncomp(msk_dil_erd);
props_struct = regionprops(connected_comps);
N = connected_comps.NumObjects
centroid = zeros(N,2);
component_size = zeros(N,1);
for i = 1 : N
    %centroid(i,1) = round(props_struct(i).Centroid(1));
    %centroid(i,2) = round(props_struct(i).Centroid(2));
    centroid(i,:) = round(props_struct(i).Centroid);
    component_size(i) = props_struct(i).Area;
end

centroid
component_size


%%%%%%%%%%%%%%%%%%%% Helper Functions %%%%%%%%%%%%%%%%%%%%%

function [msk,thrsh] = OtsuThreshold(img)
    % Define the Otsu threshold 'thrsh' using the histogram of img
    hst = imhist(img);
    thrsh = otsuthresh(hst)*255;

    % Apply the threshold to 'img' to make 'msk'
    msk = (img > thrsh);
end

```

▶ Run Script ?

Assessment: All Tests Passed

Submit ?

✓ **Is msk correct?**

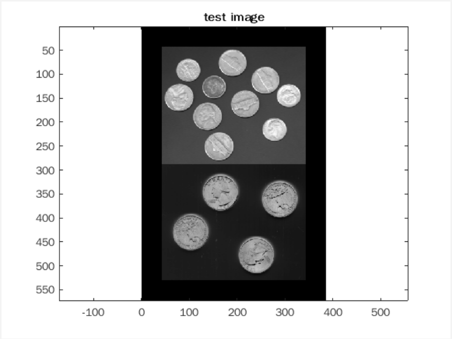
✓ **Is centroid correct?**

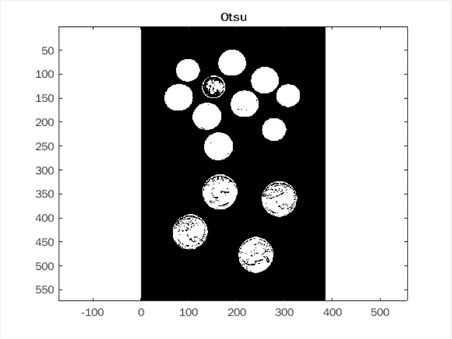
✓ **Use regionprops**

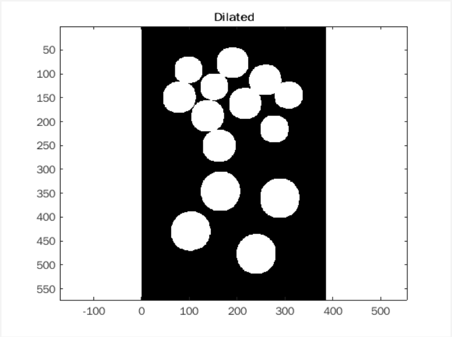
Output

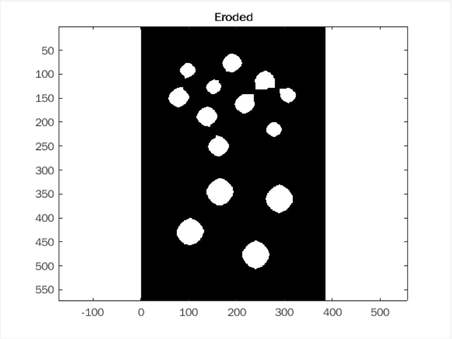
```
N =

    14


centroid =

    79   149
   103   429
    98    92
   138   188
   152   127
   165   346
   162   251
   190    77
   218   161
   240   477
   259   115
   289   360
   278   215
   306   144


component_size =

        1251
        2318
         720
        1268
```

```
         694
        2328
        1313
        1152
        1291
        2376
        1336
        2377
         733
         798
```




Otsu
50
100
150
200
250
300
350
400
450
500
550
-100
0
100
200
300
400
500


Dilated
50
100
150
200
250
300
350
400
450
500
550
-100
0
100
200
300
400
500


Eroded
50
100
150
200
250
300
350
400
450
500
550
-100
0
100
200
300
400
500