Matt Hurt Final Project

Contents

Counting the value of coins

The purpose of the this program shwos the user the value of all coins. When the image is read in it gets converted into a black and white image and gets cleaned up with the imfill. The motion function gets called to eliminate any stand alone pixels. The label variable is set to automatically choose how to label the objects in the image. To ensure the label function is working correctly, image is set to equal the first image it lables. The end result shows the user the original image and a new image that shows the user the final result.

```
clear all;
close all;
clc;
file = input('Please enter the imge file name:', 's');
```

```
Error using input
Cannot call INPUT from EVALC.

Error in coin_counter (line 16)
file = input('Please enter the imge file name:', 's');
```

Prepiring the image for edge detection

```
normalizedThresholdValue = 0.4; % In range 0 to 1.
image = imread(file);
image2 = rgb2gray(image);
resize = imresize(image2, [500, 300]);
h = fspecial('motion', 1, 1);
filtered = imfilter(resize,h);
```

Canny edge detection

```
[h,w] = size(filtered);
cannyEdgeDetect = edge(filtered, 'canny', 0.5);
fillCannyImage = imfill(cannyEdgeDetect, 'holes');
figure, imshow(filtered);

binaryImage = im2bw(fillCannyImage, normalizedThresholdValue);% One way to threshold to binary
cleanImage = imfill(binaryImage, 'holes');
label = bwlabel(cleanImage,8);
thresholdValue = normalizedThresholdValue * max(max(image)); % Gray Levels.
```

Print the results

```
figure,
subplot(2,2,1), imshow(fillCannyImage);
```

```
title('Canny edge detection');
subplot(2,2,2), imshow(cannyEdgeDetect)
title('Edge Detection');
subplot(2,2,3), imshow(label)
title('All objects that were labeled');


% Print the entire array of images recorded at different positions.  This
% is displaying how bwlabel() organized the data.
% figure,
subplot(3,3,1), imshow(label==1), title('Object 1');
subplot(3,3,2), imshow(label==2), title('Object 2');
subplot(3,3,3), imshow(label==3), title('Object 3');
subplot(3,3,4), imshow(label==4), title('Object 4');
subplot(3,3,5), imshow(label==5), title('Object 5');
subplot(3,3,6), imshow(label==6), title('Object 6');
subplot(3,3,7), imshow(label==7), title('Object 7');
subplot(3,3,8), imshow(label==8), title('Object 8');
subplot(3,3,9), imshow(label==9), title('Object 9');
```

Assign each blob a different color to visually show the distinct blobs.

```
clean = imfill(binaryImage, 'holes');
redLabels = label2rgb(clean, 'hsv', 'k', 'shuffle'); % pseudo random color labels
colorLabels = label2rgb(label, 'hsv', 'k', 'shuffle');
coloredLabels is an RGB image.
figure,
subplot(2,2,1); imshow(redLabels);
title('Binary image is passed through rgb space');
subplot(2,2,2); imshow(colorLabels);
axis image; % Make sure image is not artificially stretched because of screen's spect ratio.
caption = sprintf('Psudo colored labels, \nfrom label2rgb().\nBlobs are numbered \nfrom top to bottom, \nthen from left to right.');
title(caption);
```

Get all the blob properties.

```
L = bwconncomp(clean);
blobMeasurements = regionprops(L, 'all');
numberOfBlobs = size(blobMeasurements, 1);

% bwboundaries() returns a cell array, where each cell contains the row/col
% coordinates for an object in the image.  Plot the borders of all the
% coins on the original grayscale image using the coordinates returned by
% bwboundaries.
figure,
subplot (1,2,1); imshow(resize);
title('Input coin image');
subplot (1,2,2); imshow(label);
```

Draw a boundary around the coins

```
title('Outlines, from bwboundaries()');
axis image; % make sure image is not artificially stretched because of screen's aspect ratio.
hold on;
boundaries = bwboundaries(binaryImage);
numberOfBoundaries = size(boundaries, 1);
for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    plot(thisBoundary(:,2), thisBoundary(:,1), 'g', 'LineWidth', 2);
end
hold off;
```

Get metrics of each blob / "object"

PixelIdxList is a linear index of all the pixels in each region. For example, blob #3 might have 4000 pixels in it. PixelIdxList is the index of the original image that locates each of the 4000 pixels in that blob #3. Linear index is a list from 1 to the number of pixel in the image where the upper left is pixel #1, then it goes down row by row until it gets to the end of the first column, then it moves over to the next column and goes down those rows, and so on until it gets to the last row in the last column. The calculations are printed out to the console.

```
textFontSize = 14;
labelShiftX = -7;
blobECD = zeros(1, numberOfBlobs);
% Print header line in the command window.
fprintf(1, 'Blob #            Mean      Area    Perim    Centroid  Diameter  ECD\n');
% Loop over all blobs printing their measurements to the command window.
```

```
for k = 1 : numberOfBlobs  % Loop through all blobs.
    % Find the mean of each blob.
    thisBlobsPixels = blobMeasurements(k).PixelIdxList; % Get list of pixels in current blob.
    meanGL = mean(image(thisBlobsPixels)); %Find mean intensity (in original image).
    blobArea = blobMeasurements(k).Area; %Get centroid one at a time
    blobPerimeter = blobMeasurements(k).Perimeter;              % Get perimeter.
    blobCentroid = blobMeasurements(k).Centroid;                % Get centroid one at a time
    blobECD(k) = sqrt(4 * blobArea /pi); %Compute Equivalent Circular Diameter.
    fprintf(1, '#%2d %17.1f %11.1f %8.1f %8.1f %8.1f %8.1f \n', k, meanGL, blobArea, blobPerimeter, blobCentroid, blobECD(k));
    % Put the "blob number" labels on the "boundaries" grayscale image
    text(blobCentroid(1) + labelShiftX, blobCentroid(2), num2str(k), 'FontSize', textFontSize, 'FontWeight', 'Bold');
end
```

Count the values

The for loop below iterates through each blob or 'object' using regionprops and bwlabel to find the area of each blob according to its size and label the object with it's value. The final result shows the user the original image and the total value on the screen. /// Note /// I made an attempt of using a try / catch block in this section of the code however this involved the use of imcircles() and I did not fully understand how to implement. The overall idea is to identify overlaping coins and raise a warning to the user that there is an issue without stopping the program from completing its purpose.

```
coin = label;
[L, Ne] = bwlabel(double(coin));
prop = regionprops(L,'Area','Centroid');
total=0;
for n = 1:size(prop,1)
    cent=prop(n).Centroid;
    X=cent(1); Y=cent(2);
    % Dime
    if prop(n).Area<2300 %&& prop(n).Area <= 2300
        text(X-10,Y, '$0.10','Color', 'red')
        total=total+10;
    end
    % Penny
    if prop(n).Area > 2300 && prop(n).Area <= 2500
        text(X-10,Y,'$0.01','Color', 'red')
        total=total+1;
    end
    % Nickel
    if prop(n).Area >= 2500 && prop(n).Area < 3200
        text(X-10, Y, '$0.05', 'Color', 'red')
        total=total+5;
    end
    % Quarter
    if prop(n).Area >= 3700 && prop(n).Area < 3800
        text(X-10, Y, '$0.25', 'Color', 'red')
        total=total + 25;
    end
end
hold on
title(['Total of money: ',num2str(total),'Cents']);
```

Interesting observations //

This is an iteration I found that is able to separate the data between i and j. Then it is supposed to label every element according to its value based on its area measured in the image for each object. It did not behave as advertised however, the program does compile and it does return an image of one of the dimes captured in the image in figure 3. I perhaps made an error in translating the intended use however, I found the behavior useful. for j=1:max(max(label)) [row, col] = find(label==j); length = max(row)-min(row)+2; width=max(col)-min(col)+2; target=uint8(zeros([length width])); sy=min(col)-1; sx=min(row)-1; end

for i=1:size(row,1) x=row(i,1)-sx; y=col(i,1)-sy; target(x,y)= A(row(i,1), col(i,1)); end mytitle=strcat('Object Number:',num2str(j)); figure,imshow(target);title(mytitle);

Poor attempt at trying to transpose the image

```
%level = 0.5; %
%figure, imshow(im1);

%coinImage = rgb2gray(coinImage);
% coinImage = double(coinImage);
%
% SH = [-1 -2 -1
%       0 0 0
%       1 2 1];
%
% SV = [-1 0 1
%      -1 0 1
%      -1 0 1];
%
% im2 = imfilter(coinImage, SH,'corr');
```

```
% figure, imshow(im2);
% im3 = filter2(SV, coinImage);
% figure, imshow(im3);
% im4 = sqrt((im2.^2+im3.^2)/2);
% imfill(im4);
% BW = filter2(fspecial('sobel'),coinImage);
% min_matrix = min(BW(:));
% max_matrix = max(BW(:));
% K = mat2gray(BW);
% min_image = min(K(:));
% max_image = max(K(:));
%compliment = imcomplement(BW);
% [numrows, numcols] = size(coinImage);
% resizeIm = imresize(coinImage, [255 255]);
% adjIm = imadjust(resizeIm,[0.3 0.7],[]);
% im = double(adjIm);
% mx = max(max(adjIm));
% mn = min(min(adjIm));
% im2  = (adjIm - mn) ./ (mx - mn) * 255;
% BW = rgb2gray(adjIm);


%negativeImage = 127.5 - BW;
%edge(BW)
%figure, imshow(im4);


%noiseFix = filter2(fspecial('average'),BW)/200; % K-average
%noiseFix = medfilt2(BW);
%fix = imadjust(BW, [0.7 0.9], []);
%imcomplement(noiseFix)
%coin1 = imbinarize(noiseFix, level);%(BW, level);
%T=0.1;

%edgeDetect = edge(coin1,'Prewitt');
%imfill(coin1);
%imshow(edgeDetect);
```

This idea wasn't nessesary. Fixed with im2bw().

Fill image regions and holes

```
%coin2 = imfill(coin1, 'holes');
% %% Label and count connected components
% coin2 = logical([1 1 1 0 0 0 0 0
%                  1 1 1 0 1 1 0 0
%                  1 1 1 0 1 1 0 0
%                  1 1 1 0 0 0 1 0
%                  1 1 1 0 0 0 1 0
%                  1 1 1 0 0 0 1 0
%                  1 1 1 0 0 1 1 0
%                  1 1 1 0 0 0 0 0]);
% L = bwlabel(coin2,4);
% [r,c] = find(L == 2);
% [L, NUM] = bwlabel(double(coin2));
%figure, imshowpair(coinImage,coin2, 'montage');

%figure, imhist(coin1);
```

This idea is for dealing with color images /// Did not work ///

```
%Pre-process the RGB Color space
% rmat = fix(:,:,1);
% gmat = fix(:,:,1);
% bmat = fix(:,:,3);
% levelr = 0.63;
% levelg = 0.5;
% levelb = 0.4;
% i1 = im2bw(rmat, levelr);
% i2 = im2bw(gmat, levelb);
% i3 = im2bw(bmat, levelb);
% Isum = (i1&i2&i3);

% figure;
% subplot(2,2,1), imshow(i1)%(rmat);
% title('Red Plane');
% subplot(2,2,2), imshow(i2)%(gmat);
% title('Green Plane');
```

```
% subplot(2,2,3), imshow(i3)%(bmat);
% title('Blue Plane');
% subplot(2,2,4), imshow(Isum)%(coinImage);
% title('Sum of all planes');%('Original Image');

%imadjust(
% Icomp = imcomplement(Isum);
% Ifilled = imfill(Icomp, 'holes');
% figure, imshow(Ifilled);
% %%
% se = strel('disk',250);
% Iopenned = imopen(Ifilled,se);
% imshow(Iopenned);
```

Measure properties of image regions

```
%prop = regionprops(L, 'Area','Centroid');
```

Counter

```
%total = 0;
```

Labeling gone wrong...

for n = 1:size(prop,1) %For 1 to Tatal number of coins cent = prop(n).Centroid; X = cent(1); Y = cent(2); if prop(n).Area > 2000 text(X-10,Y,'5 C') total = total+5; else total =total+10; text(X-10,Y,'10 C') end end hold on

REFERENCES:

https://www.youtube.com/watch?reload=9&v=caP302v33Q8 https://www.mathworks.com/videos/color-based-segmentation-with-live-image-acquisition-68771.html https://www.youtube.com/watch?reload=9&v=1-jURfDzP1s https://www.youtube.com/watch?reload=9&v=XrC1r80-p-k https://www.youtube.com/watch?reload=9&v=RaOr4A50i4M https://www.youtube.com/watch?reload=9&v=SzYfSUDc7mU https://www.youtube.com/watch?reload=9&v=89SB1tAa6kg https://www.youtube.com/watch?reload=9&v=D1qBaFwuF4E https://www.mathworks.com/products.html?s_tid=gn_ps https://www.mathworks.com/products/image.html#imreg https://blogs.mathworks.com/steve/2016/05/09/image-binarization-im2bw-and-graythresh/ https://www.mathworks.com/help/images/ref/edge.html?searchHighlight=edge&s_tid=doc_srchtitle https://www.mathworks.com/help/images/ref/imfill.html?searchHighlight=imfill&s_tid=doc_srchtitle https://github.com/cheeyi/matlab-viola-jones https://www.youtube.com/watch?reload=9&v=8TSn_DbKtYI https://www.mathworks.com/matlabcentral/fileexchange/25157-image-segmentation-tutorial https://www.mathworks.com/matlabcentral/answers/59131-counting-coins-in-an-image https://www.mathworks.com/matlabcentral/fileexchange/14042-coin-recognition https://www.mathworks.com/matlabcentral/fileexchange/22924-count-of-coins https://www.youtube.com/watch?reload=9&v=uB3K8YrncMo https://www.mathworks.com/matlabcentral/fileexchange/26978-hough-transform-for-circles https://www.mathworks.com/matlabcentral/answers/77721-counting-australian-coins-in-matlab

---

Published with MATLAB® R2019a