LAB 9
Implementation of shape features

February 22, 2018

TEAM MEMBERS :

- Abhishek Singh (15BCE1009)
- Aman Hussain (15BCE1077)
- Shivam Kapoor (15BCE1339)

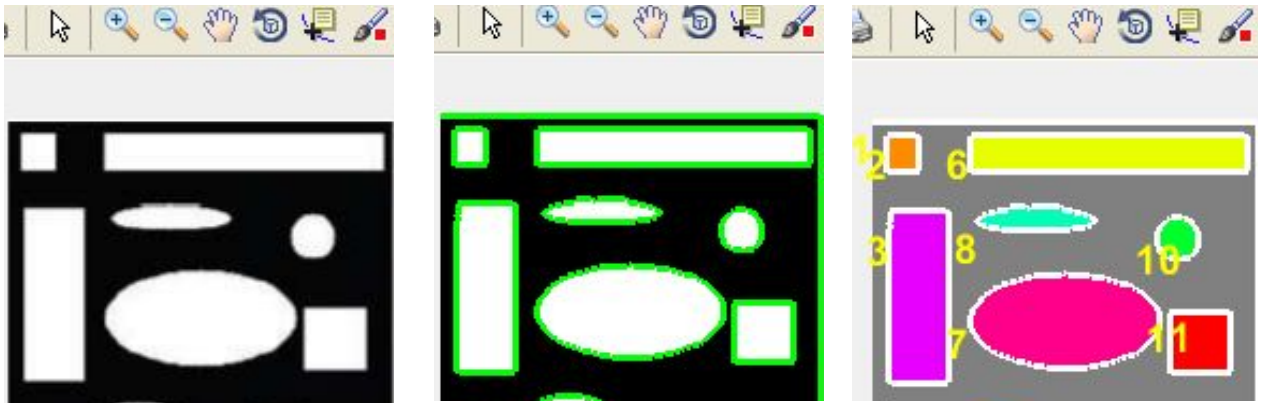
Code

```
% Clear screen
clc
clear

% Load Image
img = imread('img1.bmp');
fprintf('Original imported image - ')
imshow(img)

% Convert to Grayscale
img=im2bw(img, graythresh(img));

% Highlight boundaries
[B,L] = bwboundaries(img)
```

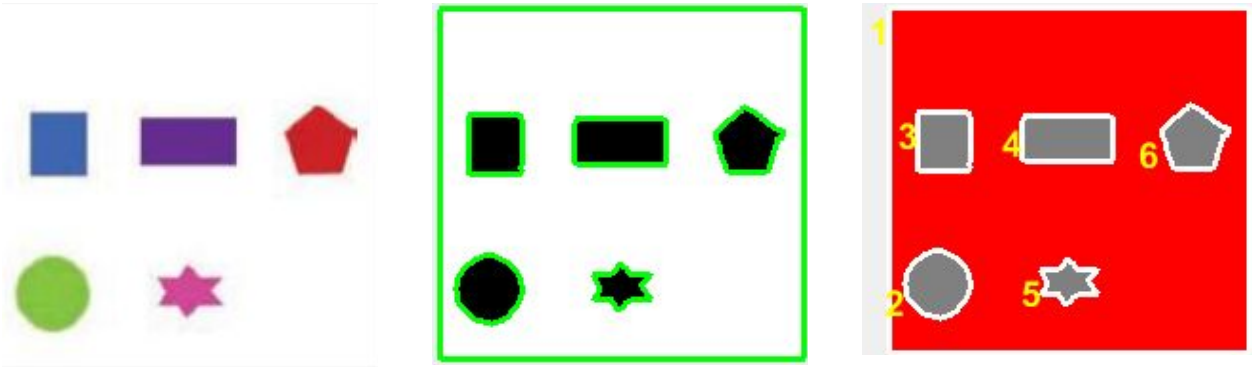
```
figure; imshow(img); hold on;
for k=1:length(B),
   boundary = B{k};
   plot(boundary(:,2),boundary(:,1),'g','LineWidth',2);
end

% Using bwlabel to label the connected regions (i.e., objects) in the test
% image, pseudocoloring them, and displaying each of them with an associated
% numerical label.
[L,N] = bwlabel(img);
RGB = label2rgb(L,'hsv',[.5,.5,.5],'shuffle');
figure; imshow(RGB); hold on;
|for k=1:length(B),
   boundary = B{k};
   plot(boundary(:,2),boundary(:,1),'w','LineWidth',2);

    text(boundary(1,2)-11,boundary(1,1)+11,num2str(k),'Color','y','FontSize',14,'FontWeight
    ','bold');

end
```

Output (img1)

Output (img2)



Output (img3)



Question 1

What is the value of N returned by bwlabel? Does it make sense to you?

- Bwlabel returns the label matrix L that contains labels for the 8-connected objects found in BW.
- N in [L,N] = bwlabel(___) returns the number of connected objects found in BW.
- Bwlabel is used to label the connected regions (i.e., objects) in the test image, pseudocolor them, and display each of them with an associated numerical label.

Code (In Continuation to above)

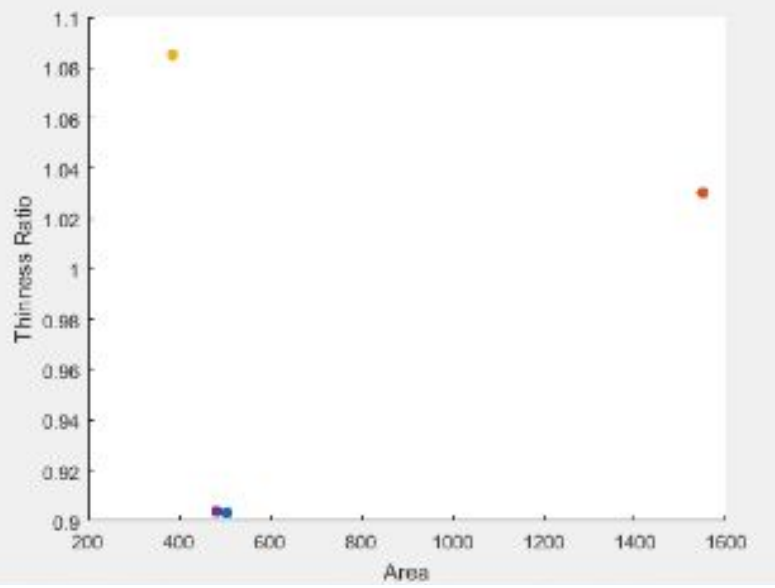
```
% Using regionprops to extract the binary features for each object in the image.
stats = regionprops(J,'all');
temp = zeros(1,N);

for k=1:N
   % Thinness Ratio
   temp(k) = 4*pi*stats(k,1).Area / (stats(k,1).Perimeter)^2;
   stats(k,1).ThinnessRatio = temp(k);

   % Aspect Ratio
   temp(k) = (stats(k,1).BoundingBox(3))/(stats(k,1).BoundingBox(4));
   stats(k,1).AspectRatio = temp(k);
end

imagefiles = dir('*.jpg');
nfiles = length(imagefiles);
```

```
% Calculate Area
areas = zeros(1,N);
for k=1:N
   areas(k) = stats(k).Area;
end


% Calculate Thinness ratio
TR = zeros(1,N);
for k=1:N
   TR(k) = stats(k).ThinnessRatio;
end

figure(); hold on;
cmap = colormap(lines(16));
for k=1:N
   scatter(areas(k), TR(k), [], cmap(k,:), 'filled'), ylabel('Thinness Ratio'), xlabel('Area');
   hold on;
end

name = cell(1,N);
for k=1:N
   if (TR(k) >0.9)
      name{1,k} = 'circle';
   else if (TR(k) >0.8)
         name{1,k} = 'square';
      else
         name{1,k} = 'other';
      end
```

```
        end
    end
```

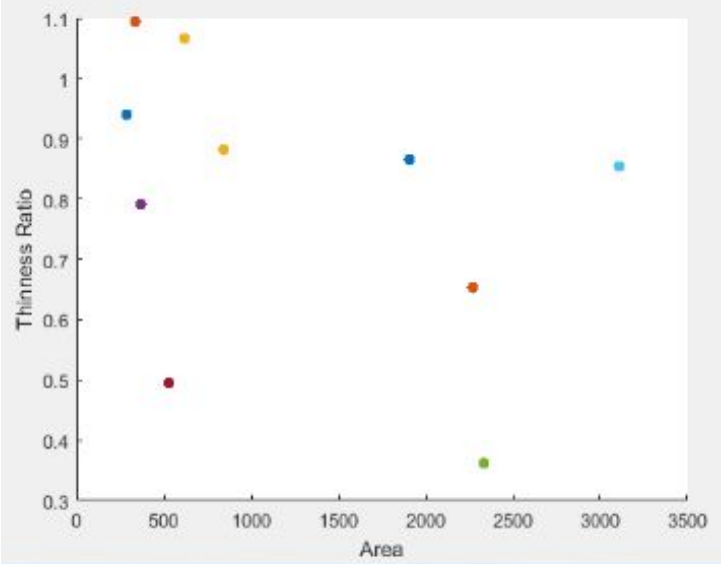
Output (img1)

Object	Area	Centroid	Orientation	Euler Number	Eccentricity	Aspect Ratio	Perimeter	Thinness Ratio
Top Left Square	509	33.5, 35	90	1	0.29166	0.9565	83.916	0.9036
Big circle	1549	89.645, 92.208	72.066	1	0.070713	0.9778	137.46	0.9030
Small Circle	387	147.06, 142.8	88.543	1	0.35643	0.9130	66.939	1.0302
Top Right Square	483	152,35	90	1	0.40786	0.9130	81.956	1.0853



Output (img2)

Object	Area	Centroid	Orientation	Euler Number	Eccentricity	Aspect Ratio	Perimeter	Thinness Ratio
Top Left Square	284	33.5, 35	90	1	0.29166	0.9565	83.916	0.9036
Big circle	2267	89.645, 92.208	72.066	1	0.070713	0.9778	137.46	0.9030
Small Circle	617	147.06, 142.8	88.543	1	0.35643	0.9130	66.939	1.0302
Top Right Square	364	152,35	90	1	0.40786	0.9130	81.956	1.0853
	2334	14.						
	3111							
	526							
	1906							
	331							



Output (img3)

```
>> stats(:,1).ThinnessRatio

ans =
0.9391

ans =
0.6538

ans =
1.0656

ans =
0.7910

ans =
0.3628

ans =
0.8541

ans =
0.4941

ans =
0.8657

ans =
1.0934
```

```
ans =
0.8831

>> stats(:,1).AspectRatio

ans =
0.8889

ans =
0.3580

ans =
1

ans =
2.6154

ans =
7.2222

ans =
1.9778

ans =
4.6667

ans =
2

ans =
1
```

ans =
1

>> regionprops('table',J,'all')

ans =

Area Centroid BoundingBox SubarrayIdx MajorAxisLength

MinorAxisLength Eccentricity Orientation ConvexHull ConvexImage ConvexArea

Image FilledImage FilledArea EulerNumber Extrema EquivDiameter Solidity

Extent PixelIdxList PixelList Perimeter PerimeterOld

_____ _____ _____ _____

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____ _____

_____ _____ _____

31033 100.21 91.986 [1x4 double] [1x183 double] [1x194 double] 226.59 216.9

0.28933 -28.741 [755x2 double] [183x194 logical] 35502 [183x194 logical] [183x194

logical] 35502 -4 [8x2 double] 198.78 0.87412 0.87412 [31033x1 double]

[31033x2 double] 734.64 750

```
>>stats(:,1).ThinnessRatio

ans =
0.7226

>> stats(:,1).AspectRatio

ans =
1.0601
```