UPPSALA
UNIVERSITET

Computer-Assisted Image Analysis I

`Lab4_Anam_Kumar`

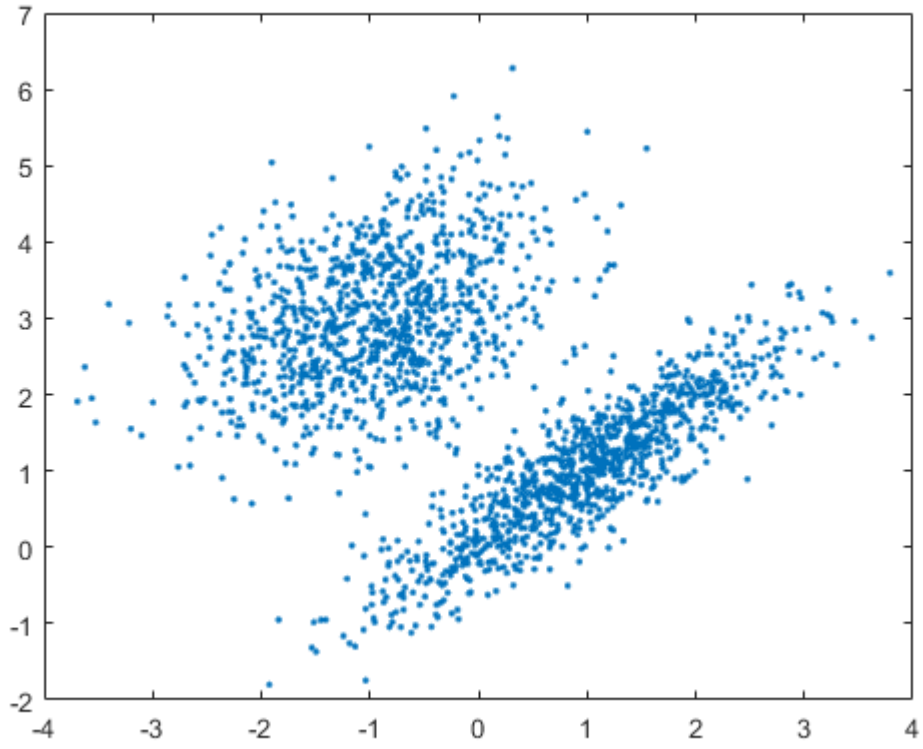
Md Tahseen Anam

Lokesh Kumar

January 3, 2020

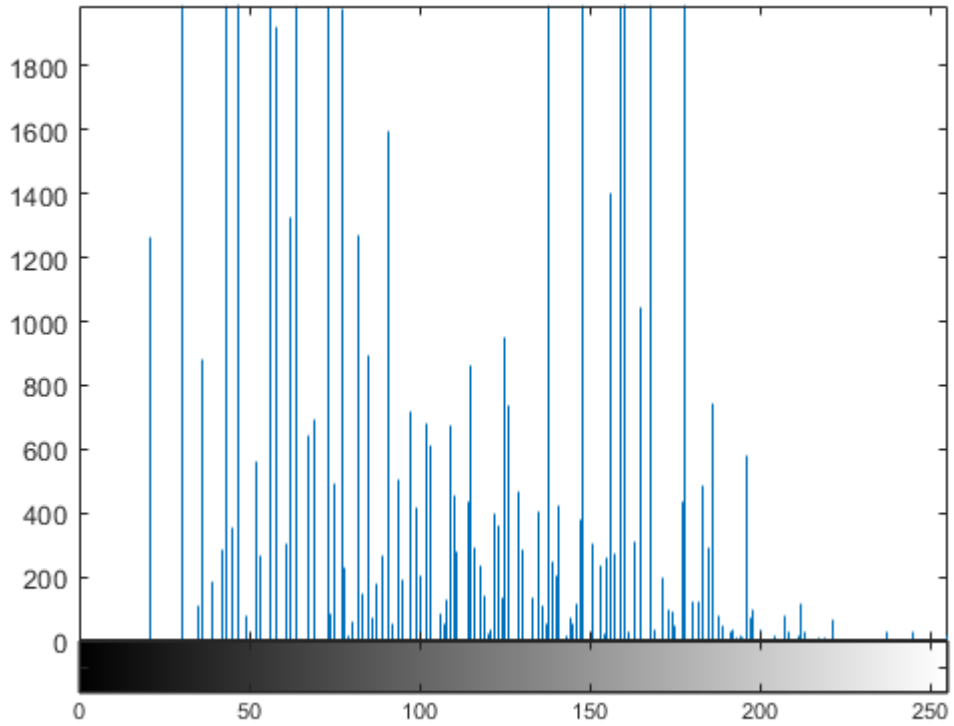
1 Exercise 1

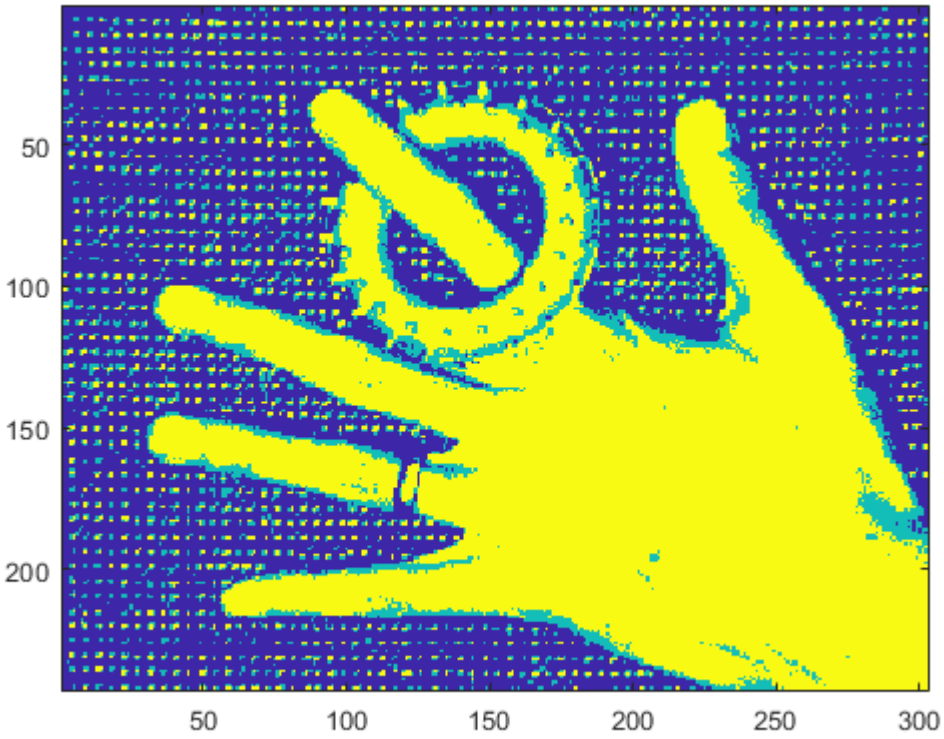
No, the two classes are not possible to separate by using either the x or y co-ordinate as a single feature. Because if we use only x co-ordinate then we will get vertical lines, using only y co-ordinates as a single feature will give us horizontal lines. This is not possible to separate because we need atleast two co-ordinates to separate the classes.



2 Exercise 2

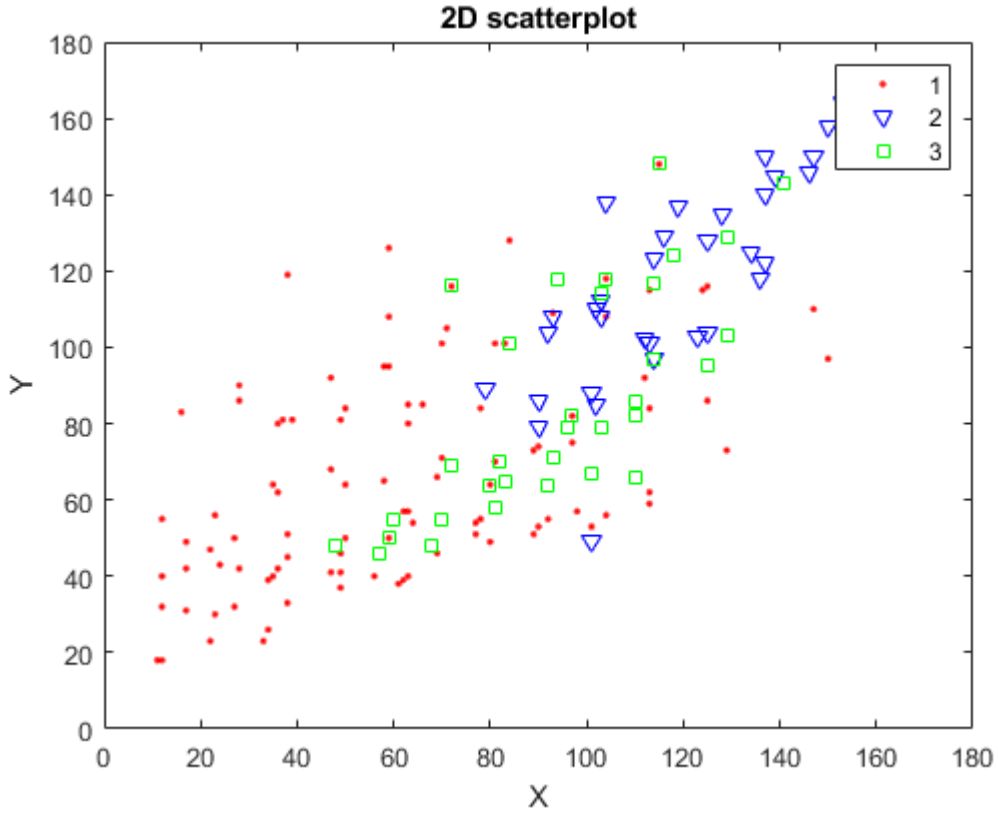
No, because for multiple objects the gray scale values overlap. That is why, in this case multiple thresholding can not successfully classify three classes. We can observe this behavior from the histogram given below.

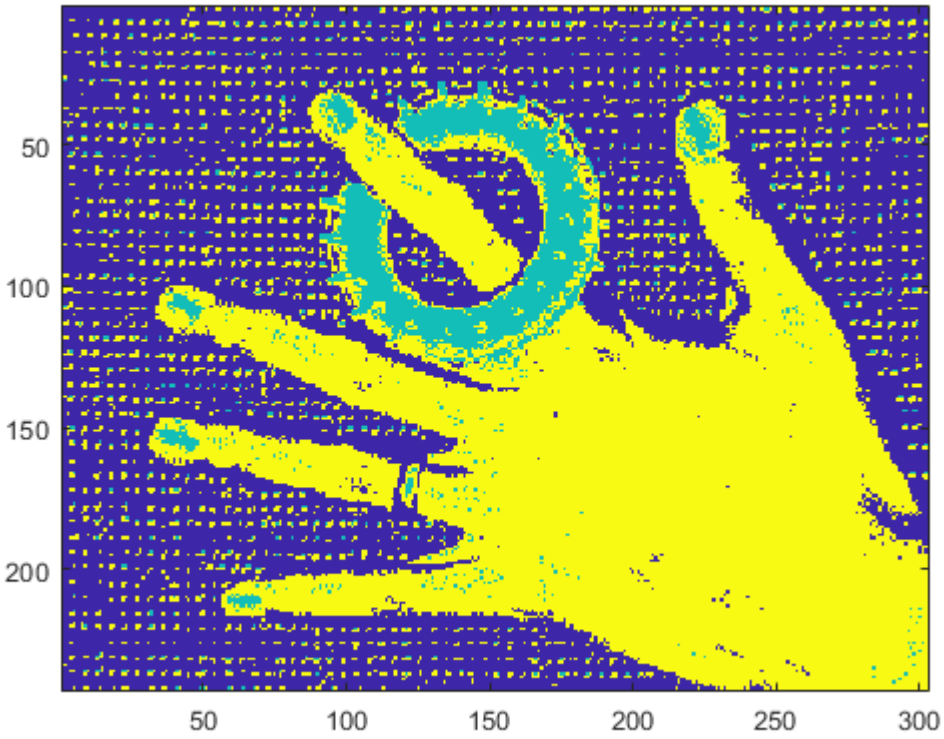



50
100
150
200
50
100
150
200
250
300

3 Exercise 3

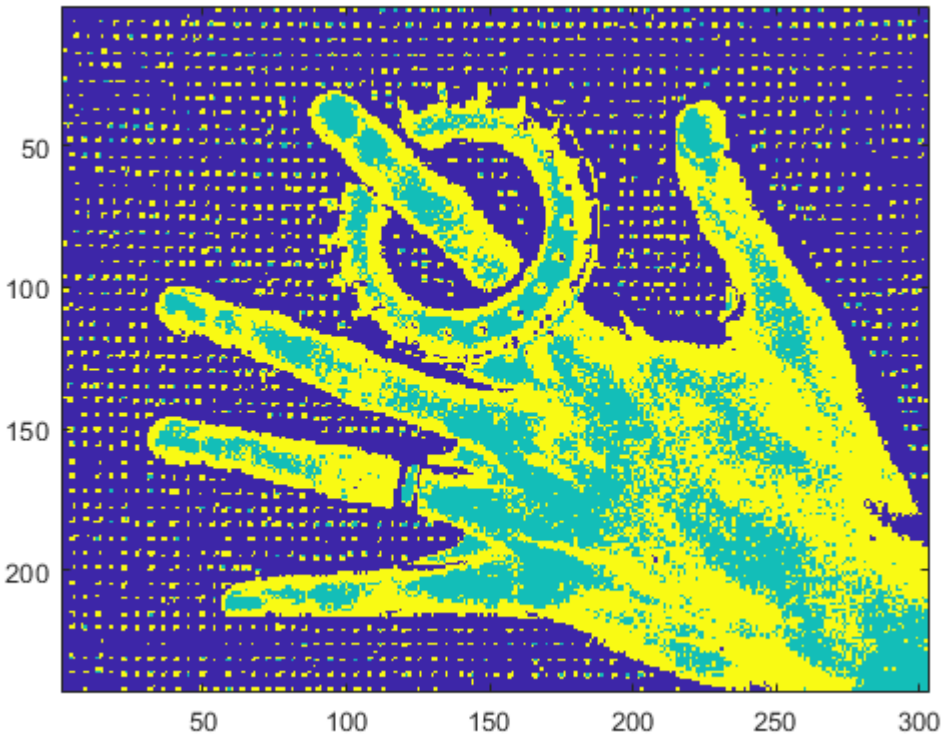
We are using linear classification model. The linear classification allows us to draw lines between classes. The assumption is that we can draw lines between different RGB bands to differentiate between classes.



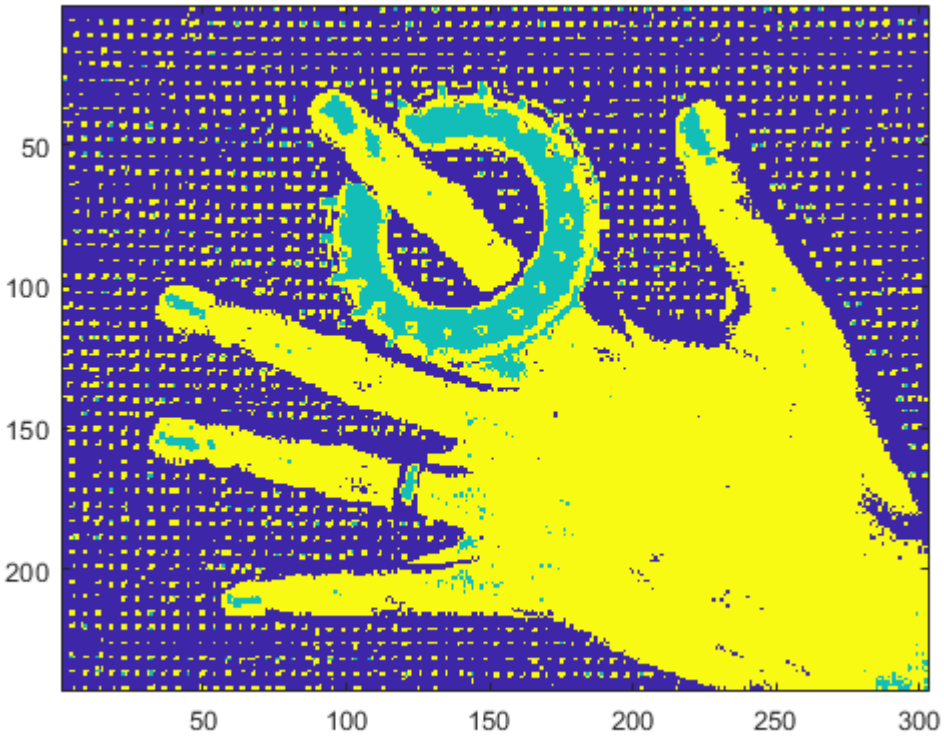

50
100
150
200
50
100
150
200
250
300

4 Exercise 4

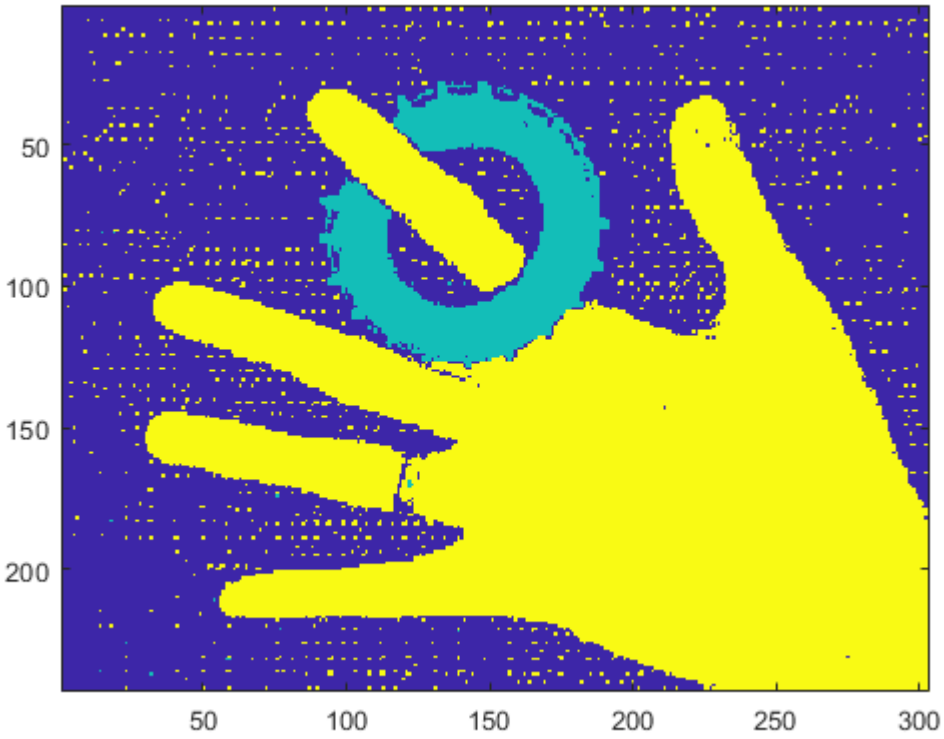
No, the result did not improve when we used classification on our grayscale image. We got the image below using grayscale image.

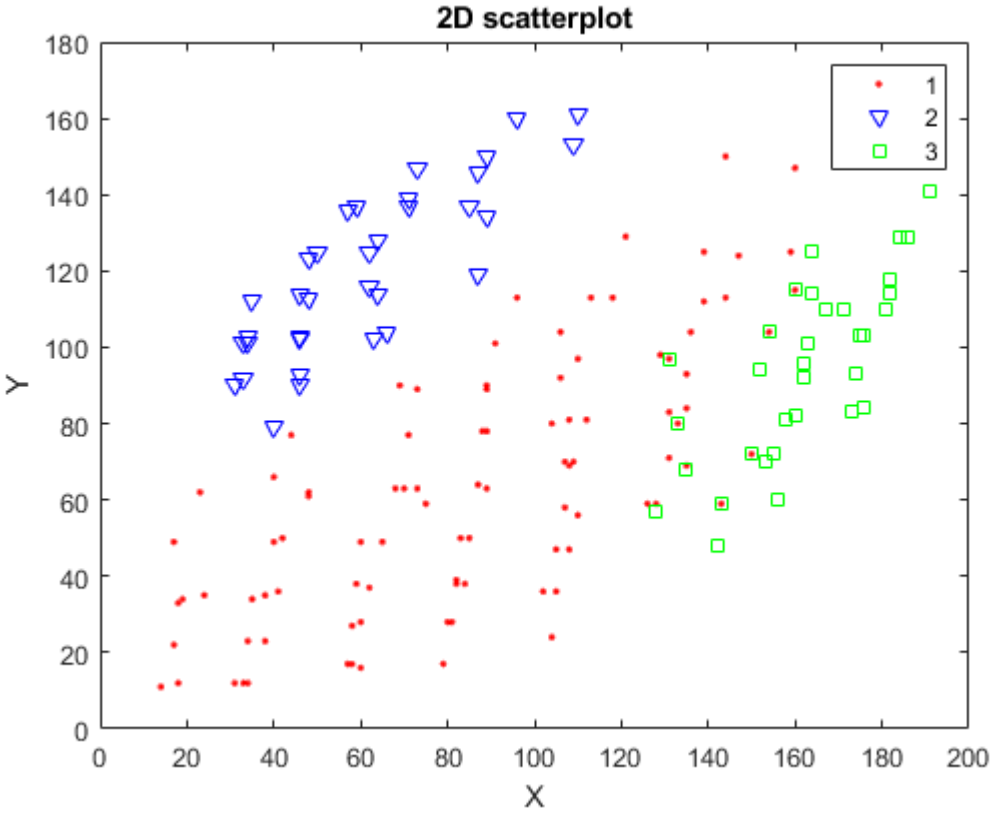


The result got improved when we used single band image instead of grayscale image. We have used green channel to create our single band image. Our classification has separated the object from the hand. This is possible as the object and the hand have different RGB values whereas they have same grayscale values in the grayscale image.



Incorporating pairs of bands will improve the classification even more. If we use full RGB information then the classification gets even better as all three channels can be used as discriminative features.


50
100
150
200
50
100
150
200
250
300


2D scatterplot
1
2
3
Y
X
0
20
40
60
80
100
120
140
160
180
200

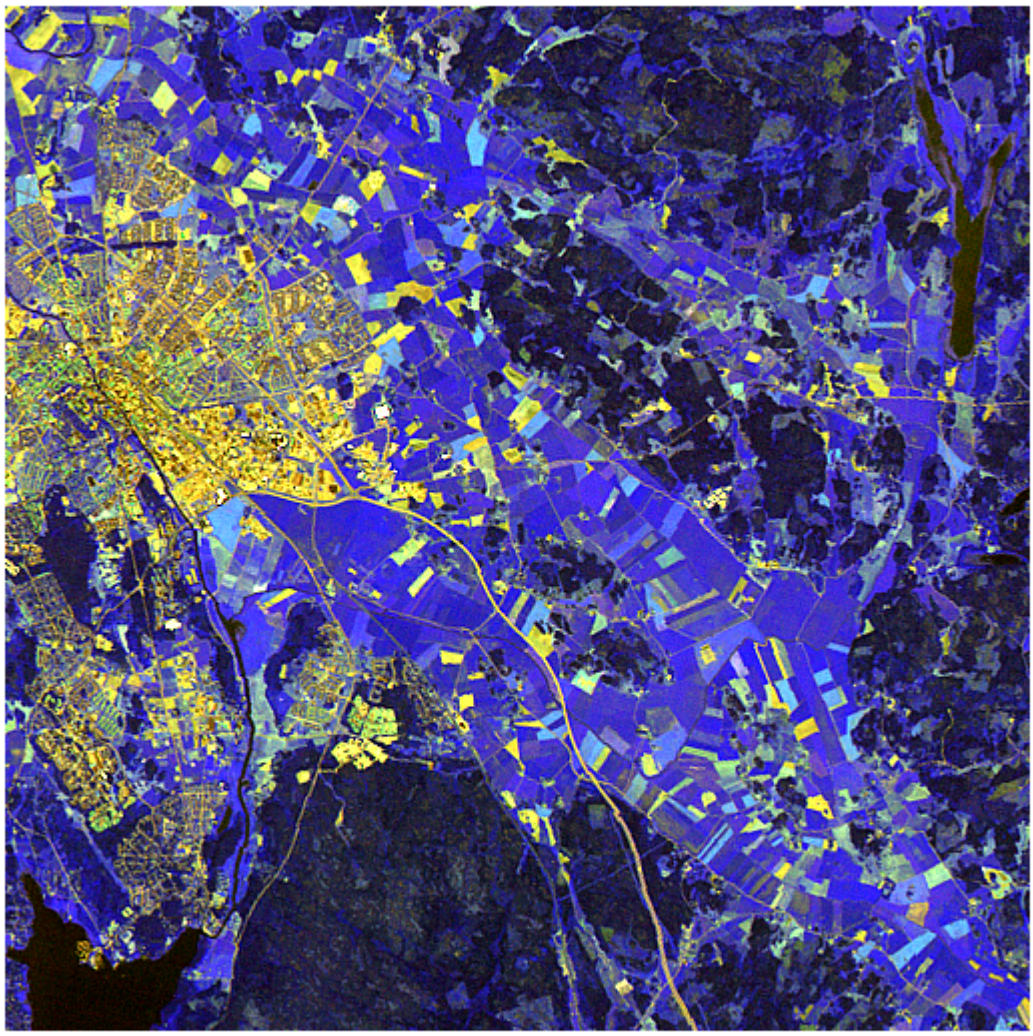
5 Source code of exercises 1,2,3,4

```
close all;
load cdata;
figure(1);
plot(cdata(:,1),cdata(:,2),'.');
%plot(cdata(:,1),'.');
%plot(cdata(:,2),'.');

I = imread('handBW.pnm');
figure(2);imshow(I);
figure(3);imhist(I);
% Read the image
% Show the image
% Show the histogram

figure(4);
mtresh(I,100,135);

I2 = imread('hand.pnm');
% Read the image
figure(5);
imshow(I2);
% Show the image
R = I2(:,:,1);
% Separate the three layers, RGB
G = I2(:,:,2);
B = I2(:,:,3);
figure(6);
plot3(R(:),G(:),B(:),'.') % 3D scatterplot of the RGB data

label_im = imread('hand_training.png'); % Read image with labels
figure(7);imagesc(label_im);
% View the training areas

I3(:,:,1) = G;% Create an image with two bands/features
I3(:,:,2) = B;
[data,class] = create_training_data(I3,label_im); % Arrange the
    training data into vectors
figure(8);
%imshow(I3);
scatterplot2D(data,class);
% View the training feature vectors

Itest = im2testdata(I3);
C = classify(double(Itest),double(data),double(class));
ImC = class2im(C,size(I3,1),size(I3,2));
figure(9);
imagesc(ImC);
```

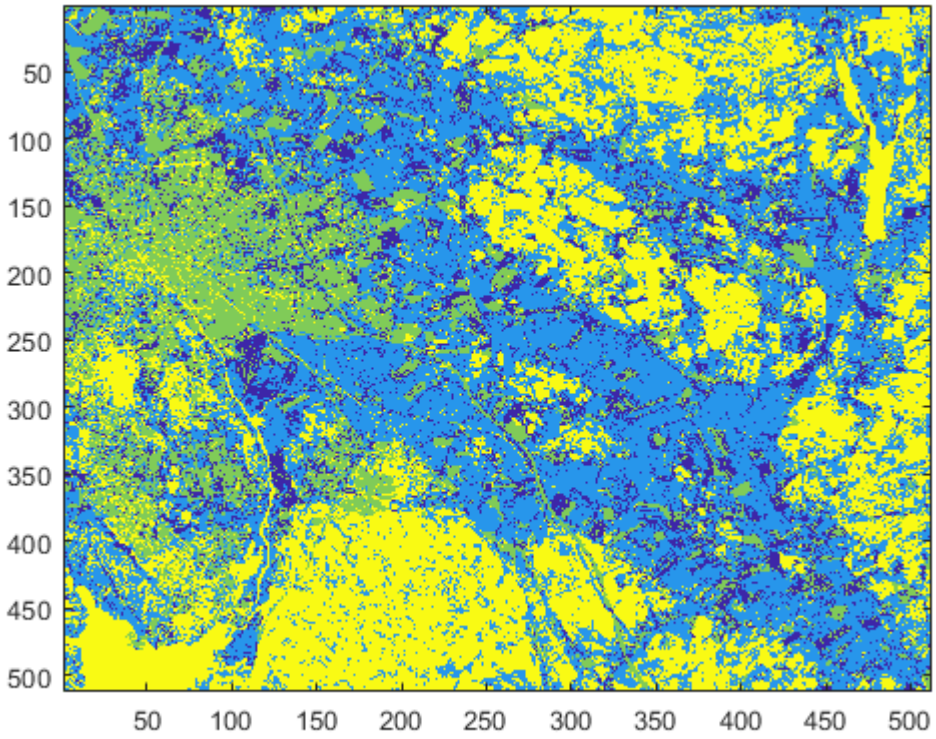
```
47
48 I4(:,:,1) = G;% Create an image with single bands/features
49 [data,class] = create_training_data(I4,label_im); % Arrange the
       training data into vectors
50 figure(10);
51 %imshow(I3);
52 scatter(data,class);
53 % View the training feature vectors
54
55 Itest = im2testdata(I4);
56 C = classify(double(Itest),double(data),double(class));
57 ImC = class2im(C,size(I4,1),size(I4,2));
58 figure(11);
59 imagesc(ImC);
60
61 I5 = rgb2gray(I2);
62 [data,class] = create_training_data(I5,label_im); % Arrange the
       training data into vectors
63 figure(12);
64 %imshow(I3);
65 scatter(data,class);
66 % View the training feature vectors
67
68
69 Itest = im2testdata(I5);
70 C = classify(double(Itest),double(data),double(class));
71 ImC = class2im(C,size(I5,1),size(I5,2));
72 figure(13);
73 imagesc(ImC);
74
75
76 I6(:,:,1) = R;% Create an image with two bands/features
77 I6(:,:,2) = G;
78 I6(:,:,3) = B;
79 [data,class] = create_training_data(I6,label_im); % Arrange the
       training data into vectors
80 figure(14);
81 %imshow(I3);
82 scatterplot2D(data,class);
83 % View the training feature vectors
84
85 Itest = im2testdata(I6);
86 C = classify(double(Itest),double(data),double(class));
87 ImC = class2im(C,size(I6,1),size(I6,2));
88 figure(15);
89 imagesc(ImC);
```

6 Exercise 5

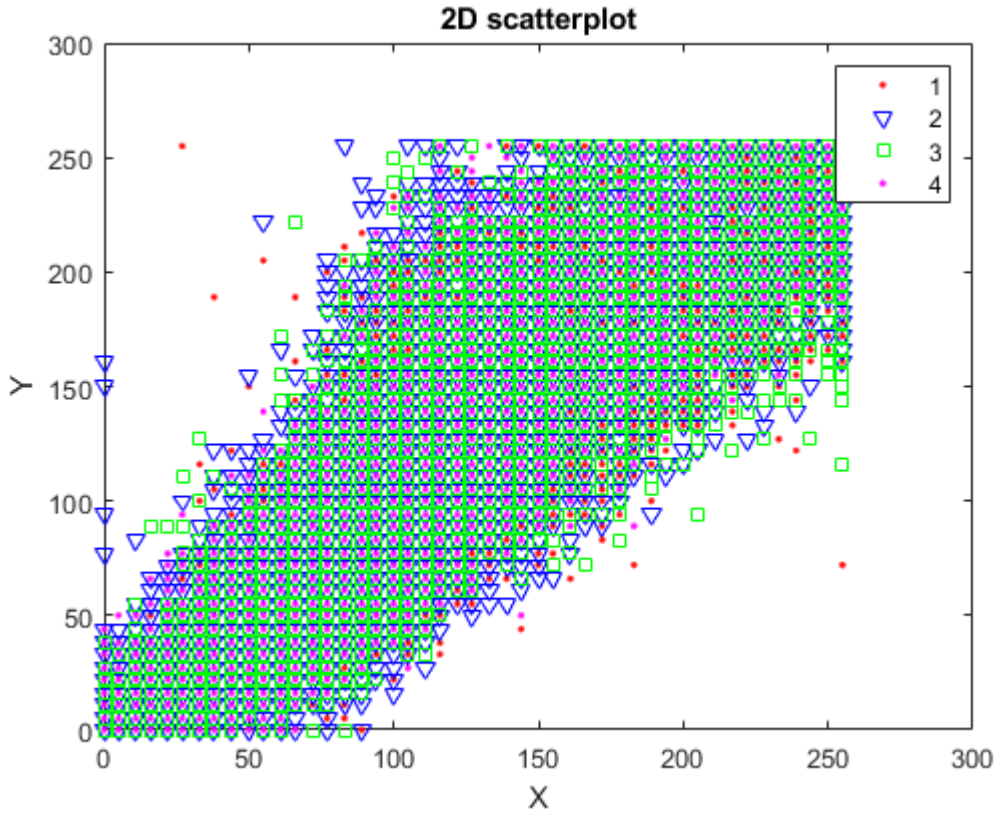
We have used bands 1, 3 and 4. Because we can discriminate buildings, forest and field using these bands.



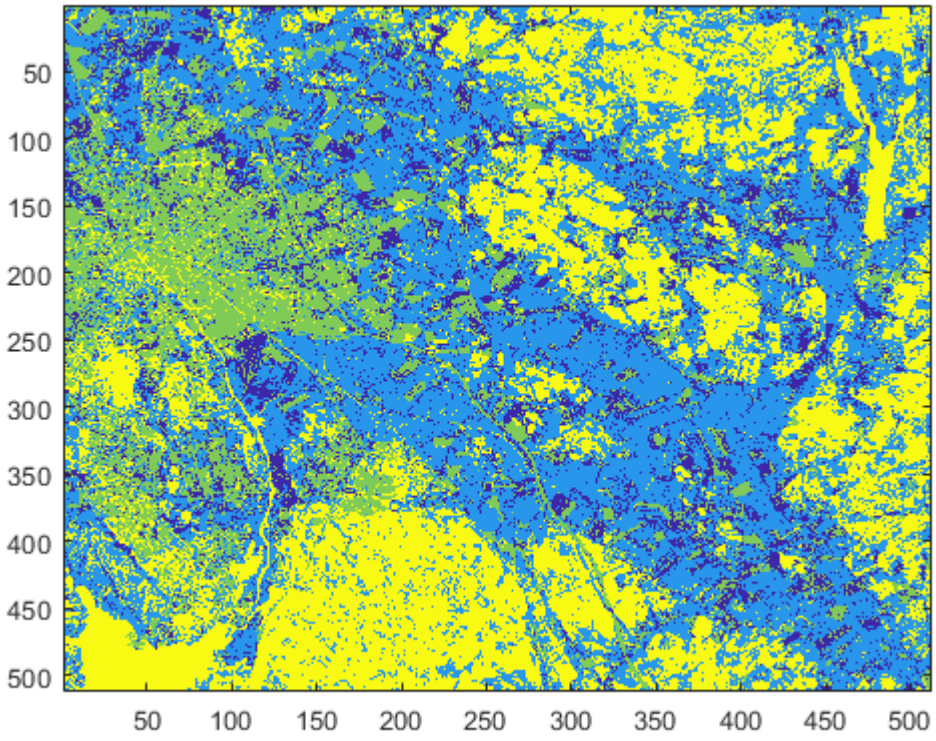
We have used quadratic classifier to do our classification and classified using the selected bands 1,3 and 4 and got the following result.

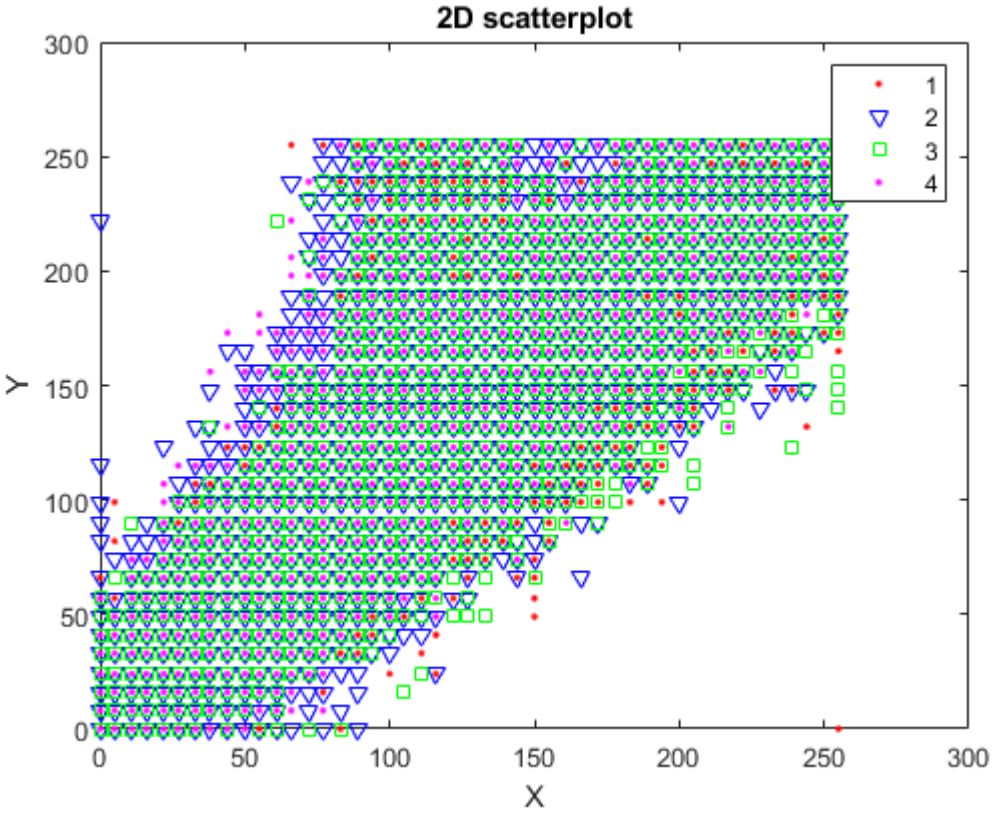


Here is our scatterplot figure.



If we use all bands for classification, we do not see much difference. Here is the result of classification using all bands.


50
100
150
200
250
300
350
400
450
500
50
100
150
200
250
300
350
400
450
500


2D scatterplot
300
250
200
150
100
50
0
Y
0
50
100
150
200
250
300
X
1
2
3
4

7 Exercise 6

Using all bands for classification may overfit the image and the result may remove some details. It will also slow down the classification process.

8 Source code of exercises 5,6

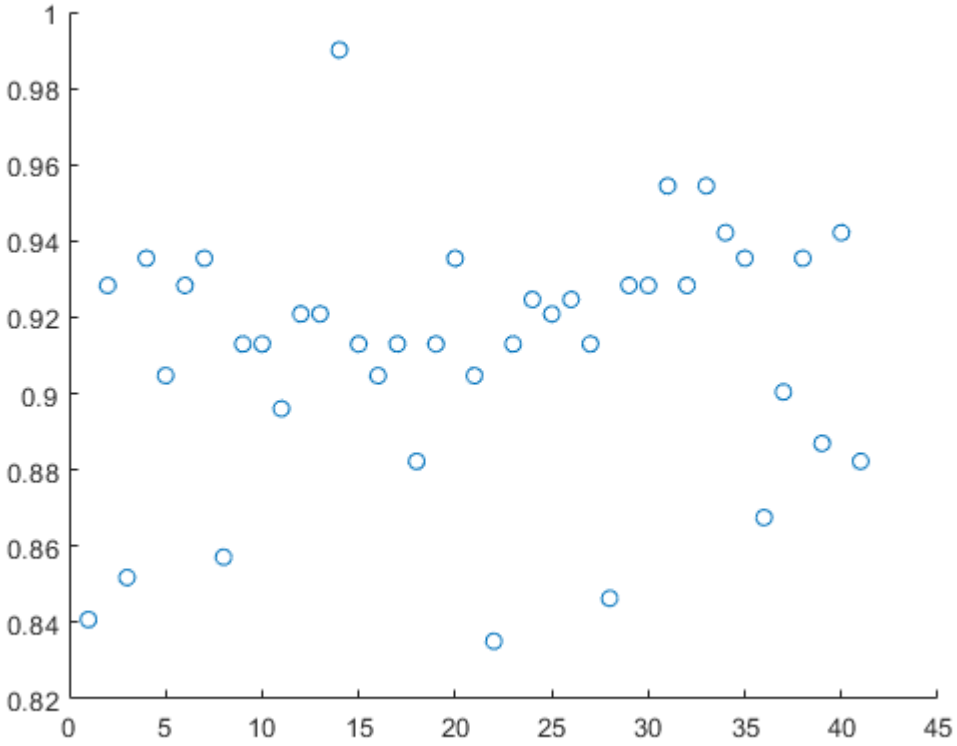
```
close all;
clear all;
load cdata;
load landsat_data;
I2 = landsat_data;
figure;
imshow(landsat_data(:,:,[1,3,4])./255);

% Show the image
R = I2(:,:,1);
% Separate the three layers, RGB
G = I2(:,:,3);
B = I2(:,:,4);


label_im = imread('hand_training.png'); % Read image with labels
figure(7);imagesc(label_im);
label_im = imresize(label_im, [512,512]);
label_im(1:100,1:100) = 1;
label_im(101:200,101:200) = 3;
label_im(101:200,201:300) = 4;
label_im(201:512,301:512) = 2;


I6(:,:,1) = R;% Create an image with three bands/features
I6(:,:,2) = G;
I6(:,:,3) = B;
[data,class] = create_training_data(I6,label_im); % Arrange the
    training data into vectors
figure(14);
%imshow(I3);
scatterplot2D(data,class);
% View the training feature vectors

Itest = im2testdata(I6);
C = classify(double(Itest),double(data),double(class),'quadratic');
ImC = class2im(C,size(I6,1),size(I6,2));
figure(15);
imagesc(ImC);

```

```
40 [data,class] = create_training_data(I2,label_im); % Arrange the
       training data into vectors
41 figure(16);
42 %imshow(I3);
43 scatterplot2D(data,class);
44 % View the training feature vectors
45
46 Itest = im2testdata(I2);
47 C = classify(double(Itest),double(data),double(class),'quadratic');
48 ImC = class2im(C,size(I2,1),size(I2,2));
49 figure(17);
50 imagesc(ImC);
```

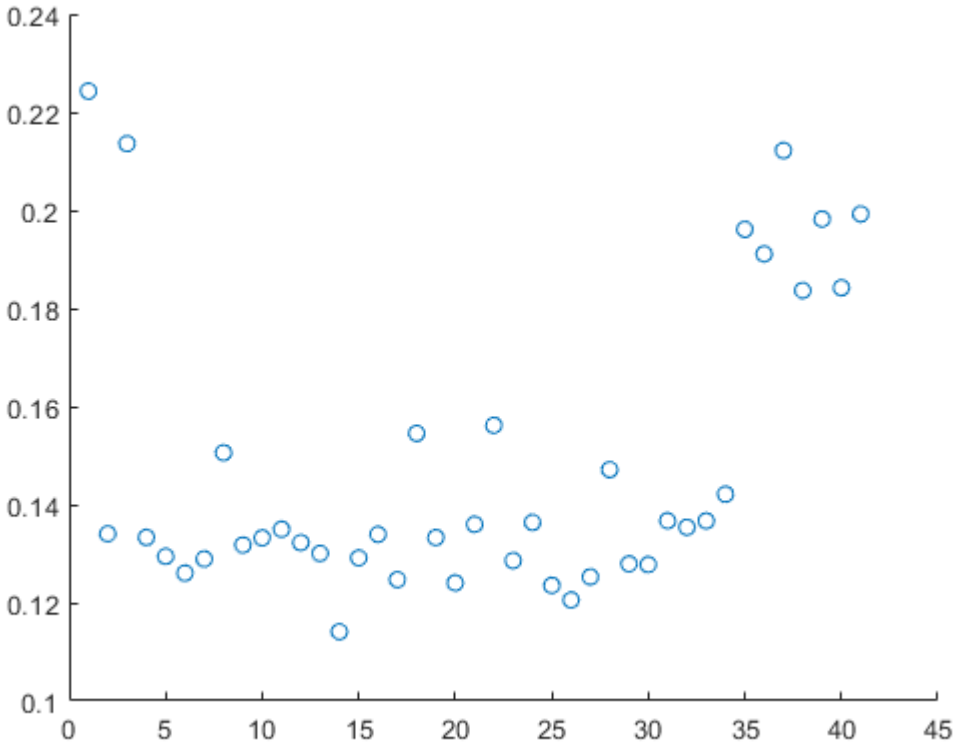
9 Exercise 7

Using all bands for classification may overfit the image and the result may remove some details. It will also slow down the classification process.

This is the resulted Entropy.



Below is the resulted Uniformity.



From the above images of entropy and uniformity, we can say that outliers are the ones with low entropy and high uniformity. So, those objects should be removed. Real virus should have higher entropy because of the randomness of nearly same grayscale value. Images with fewer different grayscale values have higher uniformity. So, real viruses should have lower uniformity.

10 Source code of exercises 7

```
close all;

%% Read image and template from disk

% Read image
I = imread('viruses.tif');

% Read template
template = imread('virusTemplate.tif');

% Show image and template
figure('name','Original image');imshow(I);
figure('name','Template');imshow(template);

%% Template matching

% Select step length (How many pixels the template is moved in each
% iteration)
stepSize = 10;

% Do the template matching
ccimg = templatematching(I,template,stepSize);

% Show the resulting correlation image
%figure('name','correlation coefficients');imshow(ccimg,[]);colormap
    (copper);colorbar;


%% Find localmaxima in the correlation image

maxima = imextendedmax(ccimg,0,4);

% Shrink objects to points
maxima = bwmorph(maxima,'shrink',inf);

% Calculate the correlation coefficients for the maxima
maxvals = maxima .* ccimg;

% Pick the corresponding correlation values and threshold them sung
    Otsu's
% method.
h = hist(maxvals(maxvals > 0),128);
h2 = imfilter(h,[1 1 1 1 1] ./ 5);
thresh = graythresh(h2);

% Create a binary image keeping only those with a correlation value
    that
```

```
45 % lies above the threshold.
46 maxvals(maxvals < thresh) = 0;
47 maxvals(maxvals ~= 0) = 1;
48
49 % Label the binary image and obtain their positions
50 maxlbl = logical(maxvals);
51 maxcentroids = regionprops(maxlbl,'centroid');
52
53 % Create a circular mask which will be used to cutout individual
     objects
54 % from the original image.
55 [h,w] = size(template);
56 d = max([h w]);
57 mask = imcircle(d);
58
59 % Cut out all objects using the mask and store it in a cell array
60 % named "cutouts".
61 nrOfObjects = length(maxcentroids);
62 cutouts{nrOfObjects} = [];
63 figure('name','Segmentation result');imshow(I);hold on;
64 for i = 1 : nrOfObjects
65     % The position corresponding to the centroid has to be
         calculated since
66     % the centroid position depends on the step size (stepSize) used
         .
67     realpos = [((maxcentroids(i).Centroid(1)-1)*stepSize+1) ((
         maxcentroids(i).Centroid(2)-1)*stepSize+1)];
68
69     cutouts{i} = I(realpos(2):realpos(2)+d-1,realpos(1):realpos(1)+d
         -1);
70     cutouts{i} = cutouts{i} .* uint8(mask);
71
72
73     rectangle('Position',[realpos(1),realpos(2),d,d],...
74         'Curvature',[1,1],...
75         'edgecolor', 'w',...
76         'linewidth', 1 );
77 end
78 hold off;
79
80 %% Adding one noisy disk and one gray disk
81
82 %refimg = noise(100 .* ones(size(mask)),'mu',1);
83 %cutouts{1,end+1} = uint8(refimg.* mask);
84 %cutouts{1,end+1} = uint8(mask).*128;
85
86 %% Collect all cutouts in an image
87
88 imagesPerRow = 10;
```

```
89 cols = imagesPerRow;
90 rows = ceil(size(cutouts,2) / cols);
91
92 objectMap = uint8(zeros(size(mask).*[rows cols]));
93 counter = 1;
94
95 for i = 1 : rows
96     for j = 1 : cols
97         if counter <= size(cutouts,2)
98             objectMap((i-1)*d+1:(i-1)*d+d,(j-1)*d+1:(j-1)*d+d) =
                  cutouts{counter}(:,:);
99             counter = counter + 1;
100        end
101    end
102 end
103
104 figure('name','Cutouts of segmented objects');imshow(objectMap);
105
106 %% cleaning the workspace from surplus variables
107 keep('I','template','mask','cutouts','nrOfObjects');%'objectMap');
108
109 %% Texture analysis
110 %
111 % I             - original image
112 % template      - the template used in the template matching
      procedure
113 % cutouts       - a cell array with all the segmented objects
114 % nrOfObjects   - the number of objects in 'cutouts'
115 % mask          - the binary mask used to cutout the objects
116 % objectMap     - a compiled image of all the objects in 'cutouts'
117 %
118
119
120 % Place for your own texture analysis.
121
122 figure,imshow(cutouts{1});
123
124 for n = 1:nrOfObjects
125     gcm = graycomatrix(cutouts{n},'Offset',[0 1; -1 1; 1 -1 ; -1
          -1]);
126     en(n)  = entropy(gcm);
127     energy = graycoprops(gcm, 'Energy Correlation');
128     un(n) = mean(energy.Energy);
129 end
130
131
132 figure;
133 title('Uniformity');
134 scatter([1:41],un);
```

```
135
136 figure;
137 title('Entropy');
138 scatter([1:41],en);
```