

Homework 2

Prithviraj Kadiyala

February 16, 2018

1 First Answer

Matlab Code

```
1  clc ;
2
3  %Suzi1 open and read image
4  suzi=fopen( 'Suzi1.bin' , 'r' );
5  s=fread(suzi ,[256,256] , 'uchar' );
6  s=s' ; %Transpose Image
7  figure(1);
8  colormap(gray(256)); %Converting into a Gray scale image
9  image(s); % Display image
10 title( 'Original Suzi Image' );
11 axis ( 'image' );
12 axis off;
13 print ( figure(1) , 'OriginalSuzi' , '-dpng' );
14
15 % Creating a histogram of the image
16 h=sum(hist(s,256) ');
17 figure(2);
18 bar(h);
19 title( 'Histogram Image of Suzi1 image' );
20 print ( figure(2) , 'Histogram' , '-dpng' );
21
22 t=92; %T=92.
23 J=s; %Making Dimensions of J equal to s
24 for i=1:255 % Loop for making values less than threshold to one
25     .
26     for j=1:255
27         if s(i,j)>=t
28             J(i,j)=0;
```

```

29         J(i,j)=255;
30     end
31 end
32 end
33
34 figure(3);
35 colormap(gray(256)); %Converting into a grayscale image
36 image(J) %Display the thresholded image
37 title('Image at Threshold T=92');
38 axis('image');
39 axis off;
40 print (figure(3), 'Threshold=92', '-dpng');
41
42 %Blob_Coloring
43 R=zeros(256,256);
44 k=1;
45 extrarow=zeros(1,256); %Adding the extra Row
46 extracolumn=zeros(257,1); %Adding the extra Column
47 J=[extrarow;J];
48 J=[extracolumn J];
49 for i=2:256
50     for j=2:256
51         if J(i,j)==255 && J(i,j-1)==0 && J(i-1,j)==0 %Check
                    condition 2 white blobs (Left and Top)
52             R(i,j)=k; %R is the blob label
53             k=k+1; % k is the blob number
54         end
55         if J(i,j)==255 && J(i,j-1)==0 && J(i-1,j)==255 %Check
                    condition 1 white blob (Top)
56             R(i,j)=R(i-1,j);
57         end
58         if J(i,j)==255 && J(i,j-1)==255 && J(i-1,j)==0 %Check
                    condition for 1 white blob (Left)
59             R(i,j)=R(i,j-1);
60         end
61         if J(i,j)==255 && J(i,j-1)==255 && J(i-1,j)==255 %Check
                    condition for the case 4
62             R(i,j)= R(i-1,j); % Both neighbouring blobs are the
                    same
63             if R(i,j-1)~=R(i-1,j)
64                 if R(i,j-1)< R(i-1,j)
65                     kill=R(i-1,j);
66                     keep = R(i,j-1);

```

```

67         else
68             kill=R(i,j-1);
69             keep=R(i-1,j);
70         end
71         for ii=1:i-1 % Loop for changing the blob
                        number R to the lowest
72             for jj=1:256
73                 if R(ii,jj)==kill
74                     R(ii,jj)=keep;
75                 else
76                     if R(ii,jj)> kill
77                         R(ii,jj)=R(ii,jj)-1;
78                     end
79                 end
80             end
81         end
82         ii=i;
83         for jj=1:j
84             if R(ii,jj)== kill
85                 R(ii,jj)= keep;
86             else
87                 if R(ii,jj) > kill
88                     R(ii,jj)=R(ii,jj)-1;
89                 end
90             end
91         end
92         k=k+1;
93     end
94 end
95 end
96 end
97 figure(4);
98 colormap(gray(256)); %Converting into grayscale
99 image(R) %image after blob coloring
100 title('Resulting Image after Blob Coloring');
101 axis('image');
102 axis off;
103 print (figure(4),'BlobColored','-dpng');
104
105 %Blob Counting%
106 count=zeros(1,k); % Making a matrix for count with k number of
                        columns
107 for p= 1:k

```

```

108     for i=1:255
109         for j=1:255
110             if R(i,j) == p
111                 count(1,p)=count(1,p)+1; %Counting the blobs
112             end
113         end
114     end
115     p = p+1;
116 end
117 %Maximum of all count
118 maximum = -50; % Any value compared will be stored
119 display(k);
120 for i = 1: k
121     if count(i) > maximum
122         disp(count(i));
123         disp(maximum);
124         maximum = count(i);
125         index = i; %This index refers to the index of the
                    matrix with highest number of blobs
126     end
127 end
128 disp(maximum);
129 disp(index);
130
131 %Minor Region Removal
132 for i = 1:255
133     for j = 1:255
134         if R(i,j) ~= index % If small blob then kill
135             J(i,j) = 0;
136         end
137     end
138 end
139 figure(5);
140 colormap(gray(256)); %Converting into grayscale
141 image(J) %Display image
142 title('Image after Minor Region Removal');
143 axis('image');
144 axis off;
145 print (figure(5), 'MinorRegionRemoved', '-dpng');
146
147 K=J;
148 for i=1:255
149     for j=1:255

```

```

150         if J(i,j)==255 %When there is a present pixel then
               replace with the original Suzi pixel
151             K(i,j)=s(i,j);
152         elseif J(i,j)==0
               K(i,j)=255;
153         end
154     end
155 end
156 end
157 figure(6);
158 colormap(gray(256)); %Converting into grayscale
159 image(K) %Display image
160 title('Image K');
161 axis('image');
162 axis off;
163 print (figure(6), 'ImageK', '-dpng');

```

1(c):Yes, a chain code can be used to represent main contour in the obtained contour image. That's because the contour obtained can be represented by storing sequence of direction codes and it would traverse according to contour if at all initial coordinate is specified.

Original Suzi Image



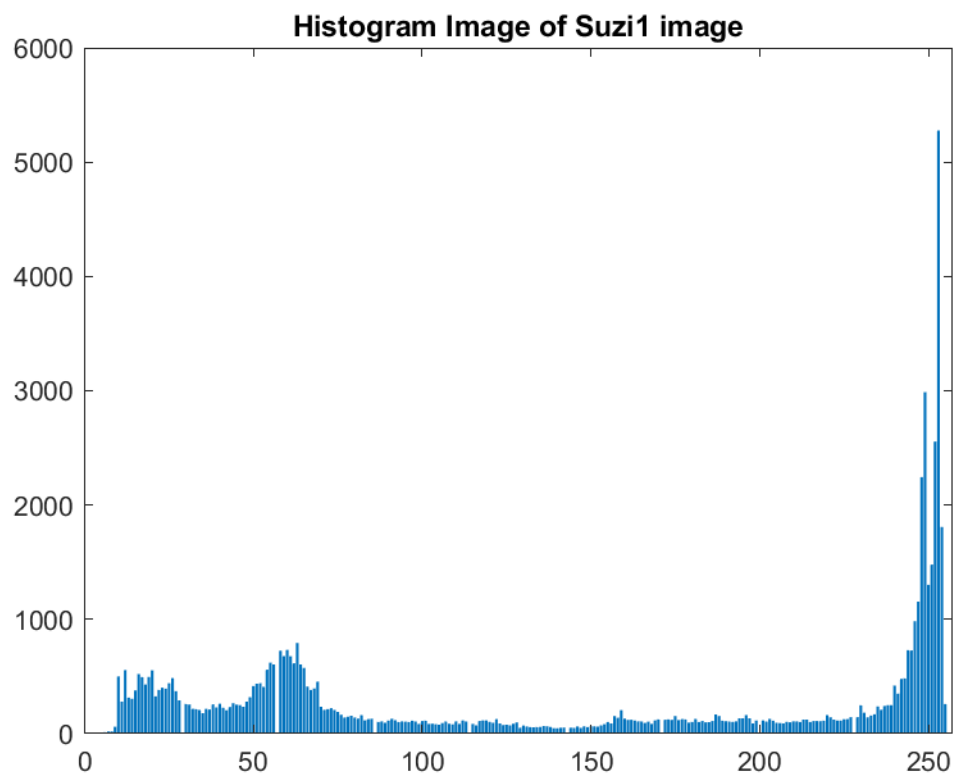


Image at Threshold T=92



Resulting Image after Blob Coloring



Image after Minor Region Removal





2 Second Answer

Matlab Code

```
1  clc;
2
3  %HW2 ct_scan
4  %open and read image
5  ct_scan=fopen('ct_scan.bin','r');
6  s=fread(ct_scan,[256,256],'uchar');
7  s=s'; %Transposeing the Matrix
8  figure(1);
9  colormap(gray(256)); %Converting into a Gray scale image
10 image(s); % Display image
11 title('Original ct-scan Image');
12 axis('image');
13 axis off;
14 print (figure(1),'OriginalCTScan','-dpng');
15
16 %Creating a histogram of the image
```

```

17 h=sum(hist(s,256)');
18 figure(2);
19 bar(h);
20 title('Histogram Image of ctscan image');
21 print (figure(2),'CTHist','-dpng');
22
23 %Thresholding the Image
24 t=90 %Threshold value by trial as T=90.
25 J=s; %Equating J to s
26 for i=1:255 % Loop for making values less than threshold to one
    .
27     for j=1:255
28         if s(i,j)<=t
29             J(i,j)=255;
30         else
31             J(i,j)=0;
32         end
33     end
34 end
35 figure(3);
36 colormap(gray(256)); %Converting into a grayscale image
37 image(J) %Display image
38 title('Image at Threshold T=90');
39 axis image;
40 axis off;
41 print (figure(3),'T=90CT','-dpng');
42
43 %Blob_Coloring
44 R=zeros(256,256);
45 k=1;
46 extrarow=zeros(1,256); %Adding Extra Row
47 extracolumn=zeros(257,1); %Adding Extra Column
48 J=[extrarow;J];
49 J=[extracolumn J];
50 for i=2:256
51     for j=2:256
52         if J(i,j)==255 && J(i,j-1)==0 && J(i-1,j)==0 %Check
            condition for the case 1 - 2 background blobs (Left
            and Top)
53             R(i,j)=k; %R-blob label
54             k=k+1; % k-blob number
55         end
56         if J(i,j)==255 && J(i,j-1)==0 && J(i-1,j)==255 %Check

```

```

57         condition for the case 2 – 1 background blob (Top)
           R(i , j)=R(i-1,j) ;
58     end
59     if J(i , j)==255 && J(i , j-1)==255 && J(i-1,j)==0 %Check
           condition for the case 3 – 1 background blob (Left)
           R(i , j)=R(i , j-1);
60     end
61     if J(i , j)==255 && J(i , j-1)==255 && J(i-1,j)==255 %Check
           condition for the case 4 – 0 background blobs
62         R(i , j)= R(i-1,j) ; % Both neighbouring blobs are the
           same
63         if R(i , j-1)~=R(i-1,j)
64             if R(i , j-1)< R(i-1,j)
65                 kill=R(i-1,j) ;
66                 keep = R(i , j-1);
67             else
68                 kill=R(i , j-1);
69                 keep=R(i-1,j) ;
70             end
71         for ii=1:i-1 % Loop for changing the blob
           number R to the lowest
72             for jj=1:256
73                 if R(ii , jj)==kill
74                     R(ii , jj)=keep ;
75                 else
76                     if R(ii , jj)> kill
77                         R(ii , jj)=R(ii , jj)-1;
78                     end
79                 end
80             end
81         end
82     end
83     ii=i ;
84     for jj=1:j
85         if R(ii , jj)== kill
86             R(ii , jj)= keep ;
87         else
88             if R(ii , jj) > kill
89                 R(ii , jj)=R(ii , jj)-1;
90             end
91         end
92     end
93     k=k+1;
94 end

```

```

95         end
96     end
97 end
98 figure(4);
99 colormap(gray(256)); %Converting into grayscale
100 image(R) %image after blob coloring
101 title('Resulting Image after Blob Coloring');
102 axis('image');
103 axis off;
104 print (figure(4), 'BlobColoringCT', '-dpng');
105
106 %Blob Counting%
107 count=zeros(1,k); % Making a matrix for count with k number of
    columns
108 for p= 1:k
109     for i=1:255
110         for j=1:255
111             if R(i,j) == p
112                 count(1,p)=count(1,p)+1; %Counting the blobs
113             end
114         end
115     end
116     p = p+1;
117 end
118 %Maximum of all count
119 maximum = -50; % Takes the compared value in
120 display(k);
121 for i = 1: k
122     if count(i) > maximum
123         disp(count(i));
124         disp(maximum);
125         maximum = count(i);
126         index = i; %This index refers to the index of the
            matrix with highest number of blobs
127     end
128 end
129 disp(maximum);
130 disp(index);
131
132 %Minor Region Removal
133 for i = 1:255
134     for j = 1:255
135         if R(i,j) ~= index % If small blob kill it

```

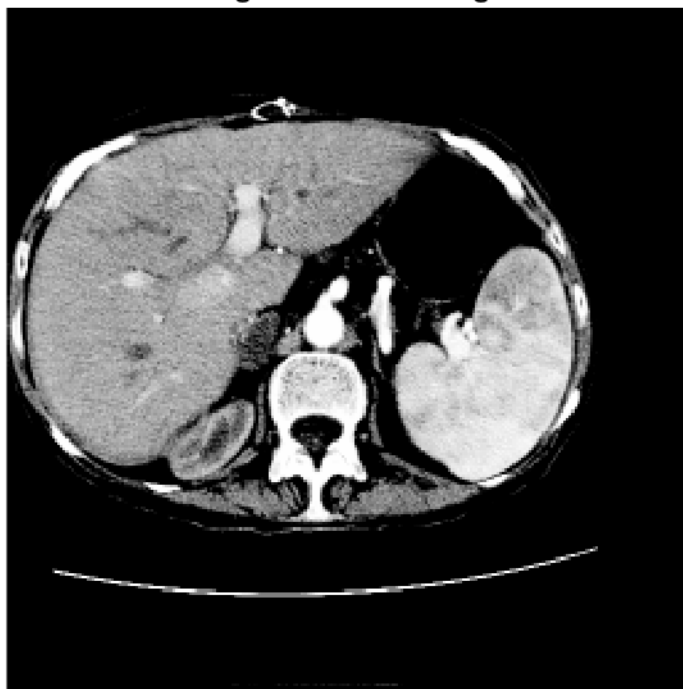
```

136         J(i,j) = 0;
137     end
138 end
139 end
140 figure(5);
141 colormap(gray(256)); %Converting into grayscale
142 image(J) %Display image
143 title('Image after Minor Region Removal');
144 axis('image');
145 axis off;
146 print (figure(5), 'BlobRegionRemovalCT', '-dpng');
147
148 K=J;
149 for i=1:255
150     for j=1:255
151         if J(i,j)==255 %If the pixel is logical true then
152             replace with original CT_Scan image
153             K(i,j)=s(i,j);
154         elseif J(i,j)==0
155             K(i,j)=255;
156         end
157     end
158 end
159 figure(6);
160 colormap(gray(256)); %Converting into grayscale
161 image(K) %Display image
162 title('Image K');
163 axis('image');
164 axis off;
165 print (figure(6), 'ImageKCT', '-dpng');

```

Output Images:

Original ct-scan Image



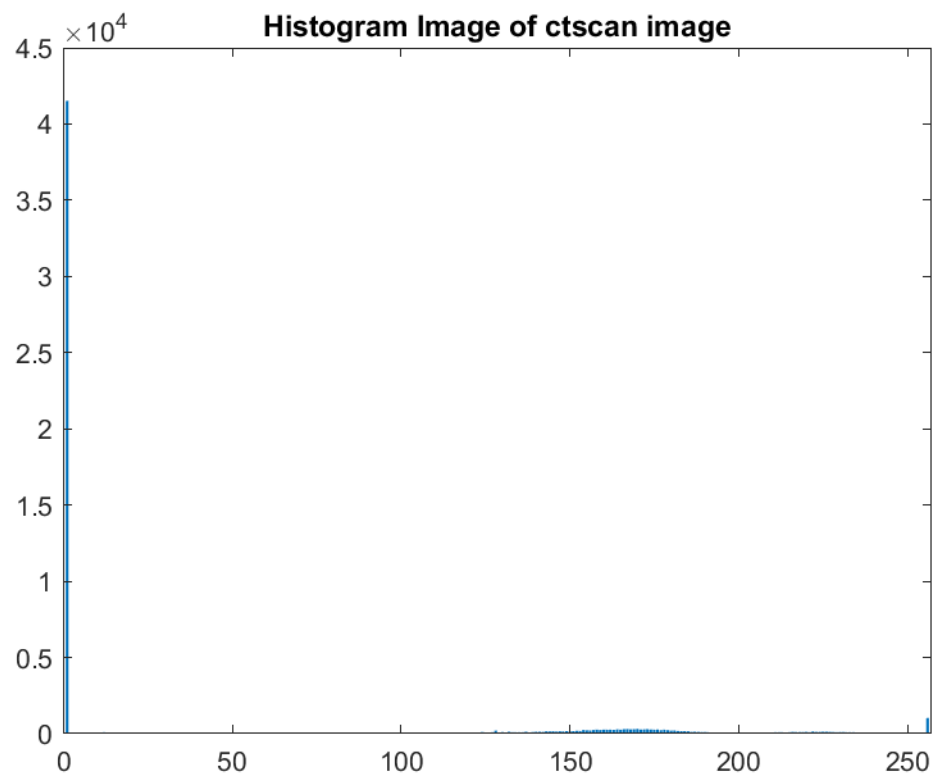
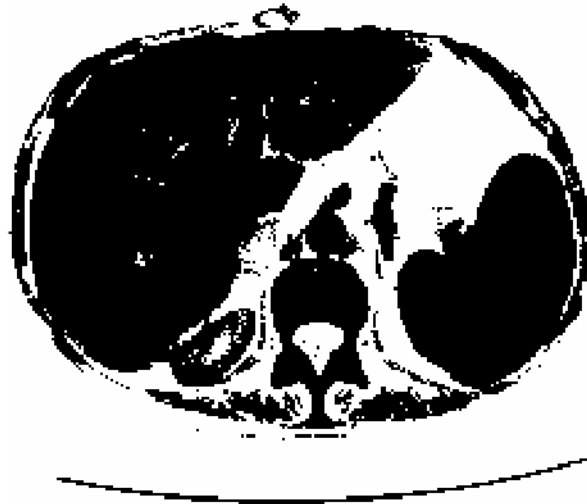


Image at Threshold T=90



Resulting Image after Blob Coloring



Image after Minor Region Removal

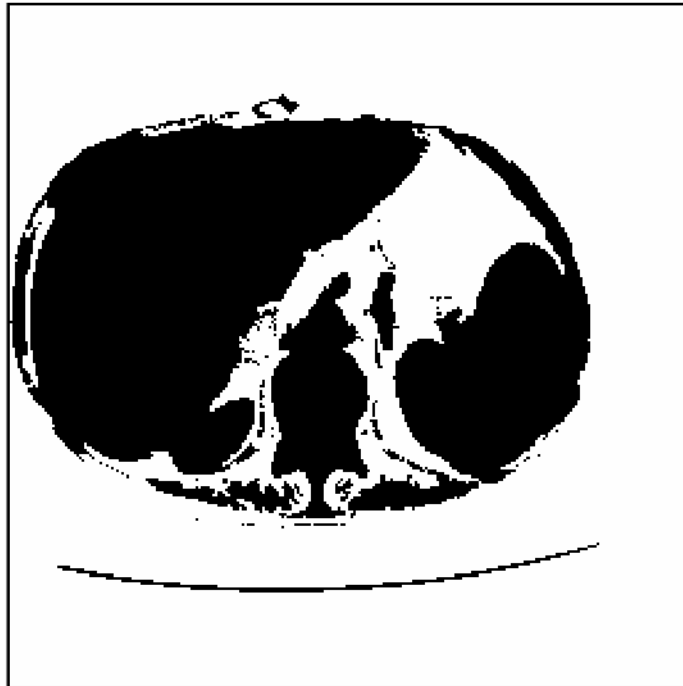


Image K

