Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

EEU44C08

4C8 DIGITAL IMAGE AND VIDEO PROCESSING

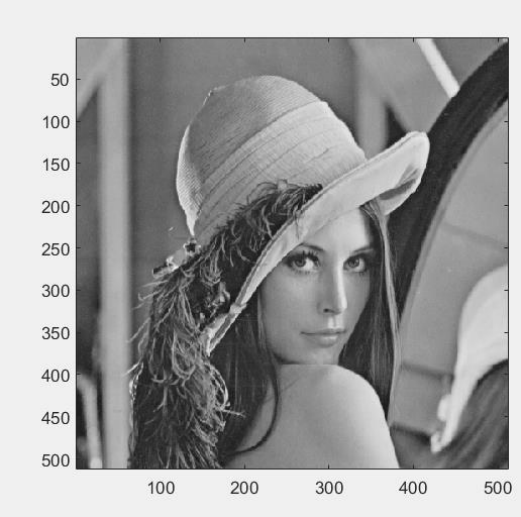
Submitted By

Harsh Dhingra

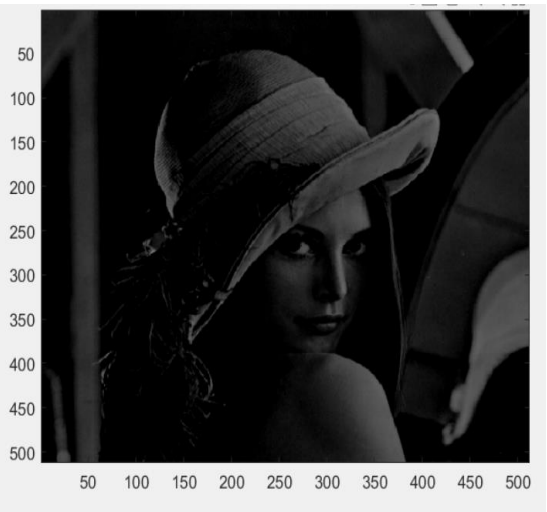
19323904

Q1 BASIC IMAGE LOADING AND DISPLAY

When adding 128 to the pixel, i.e. to the pic's brightness value, we can observe that much of the image turns white. The reason for this is – pixel value determines the brightness level of a grayscale image, adding 128 to the pixel results in increase brightness or pixel value, i.e., pixel approaches more near 255 (255^{th} value is considered as white), Whereas on subtracting the pixel comes near 0 (0^{th} value is considered as black), that is why much of the image turns black on subtracting 128 from pixel value.



a)After adding 128



b) After decreasing 128

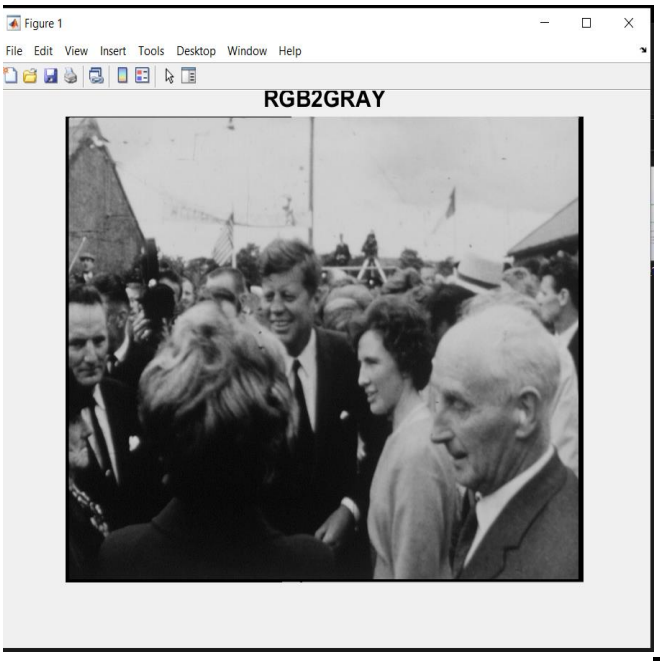
Editor - C:\Users\harsh\Desktop\LAB1\Q1.m

Q1.m | Q2.m | Q3.m | q4.m | +

```
1 -  I = imread('lena512.png');
2 -  figure(1); image(I); colormap(gray(256)); axis image;
3    %I = double(imread('lena512.png'))/255; figure(2); image(I); axis image;
4
5 -  I1 = I + 128;
6 -  figure(2);image(I1);colormap(gray(256));shg;
7
8 -  I2 = I -128
9 -  figure(3);image(I2);colormap(gray(256));shg;
10
11
```

c)Code for question 1

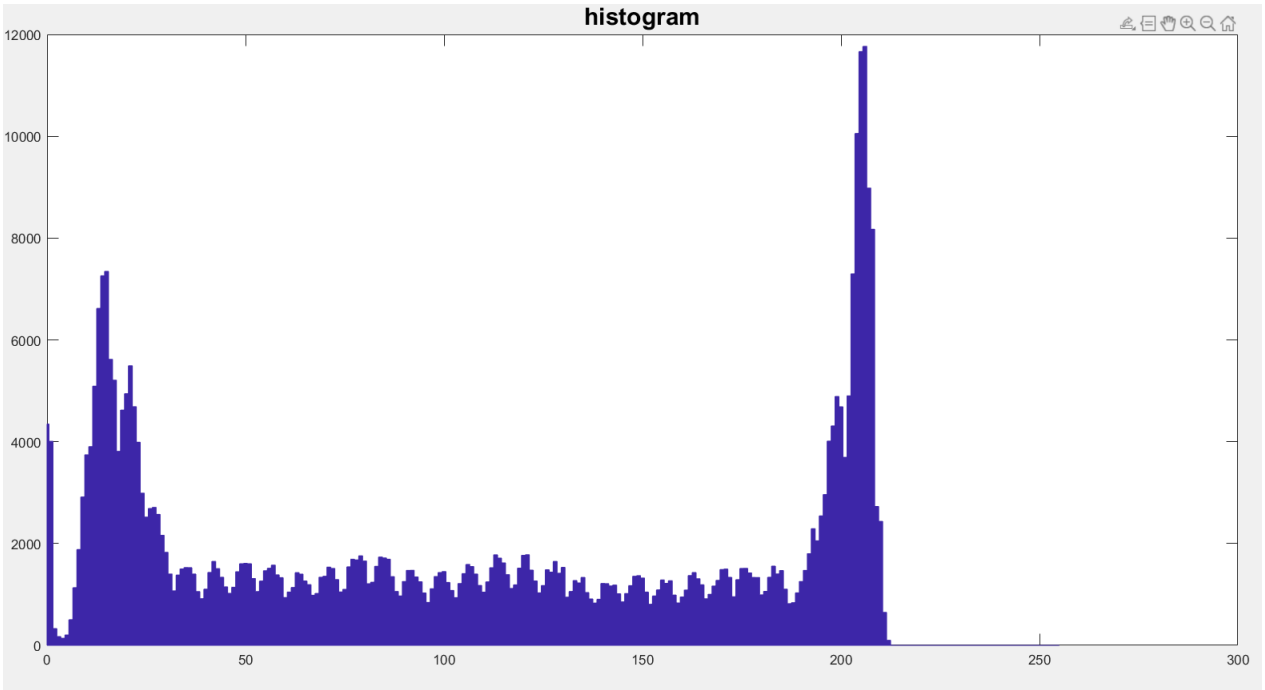
Q2 Histograms



d) after RGB2GRAY

```
Q1.m | Q2.m | Q3.m | q4.m | +
1 -  I = imread('jfk.png');
2
3 -  img = rgb2gray(I);
4 -  figure(1)
5 -  imshow(img);
6 -  title('RGB2GRAY', 'FontSize', 18)
7 -  figure(2)
8 -  hist(img(:),0:255)
9 -  title('histogram', 'FontSize', 18)
10
```

e) Code for Q2



f) Histogram

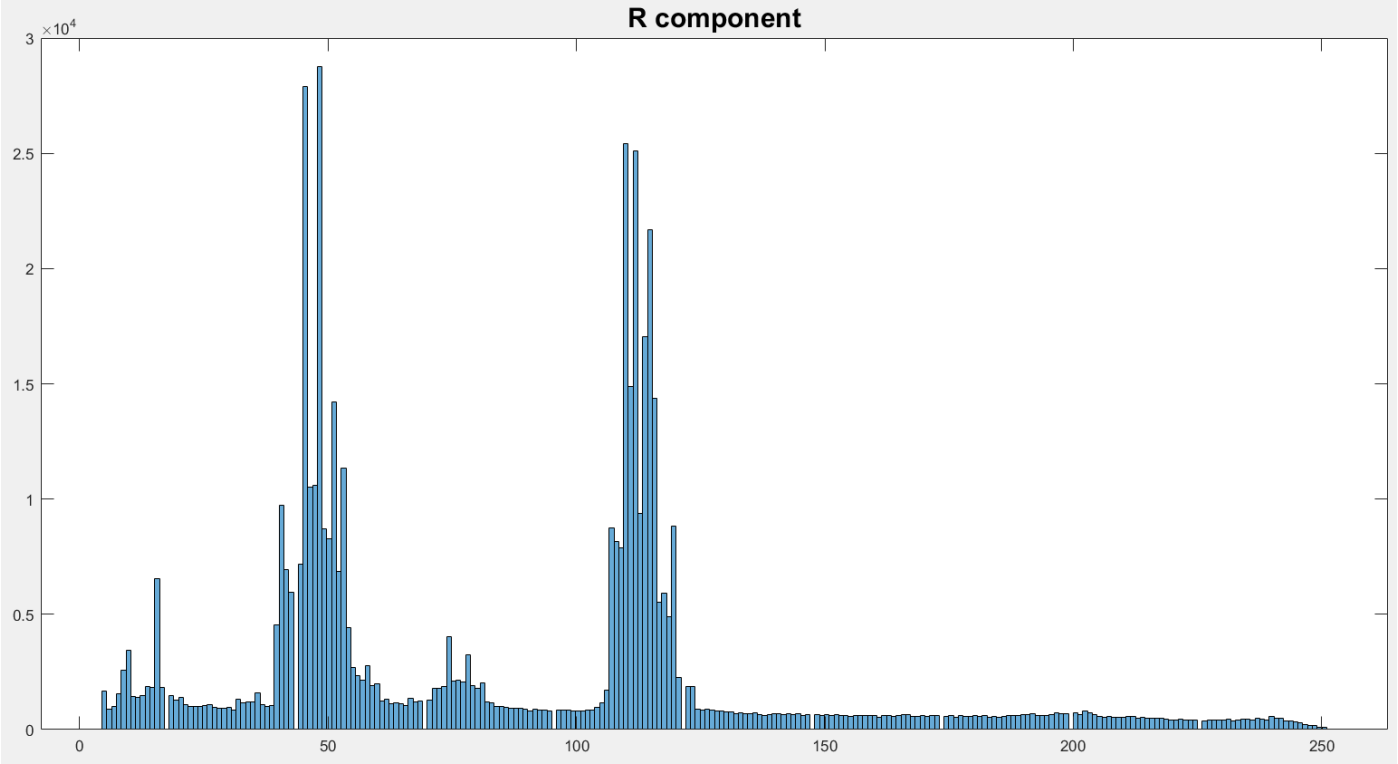
Q3Histograms of color planes

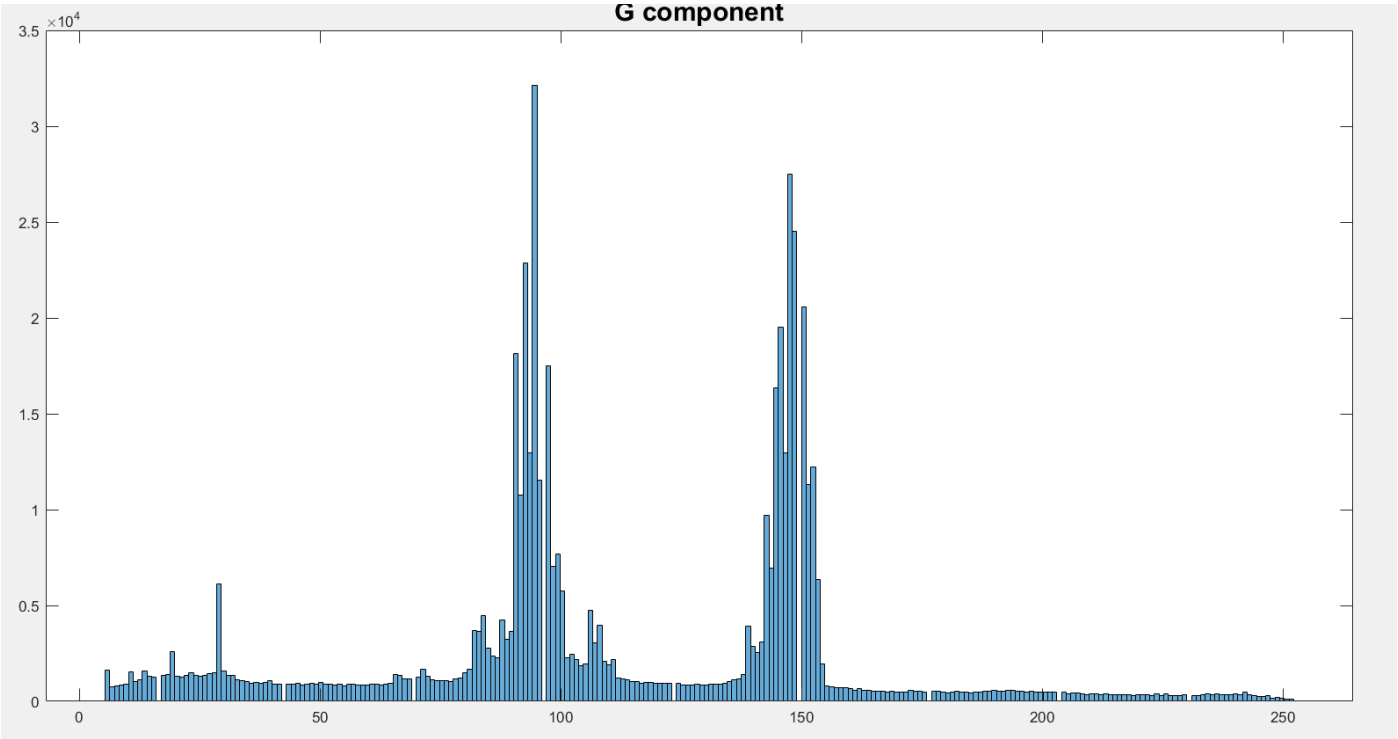
In the original image, it can be seen that green and blue take most of the picture red component is in the range 38 to 56, green is concentrated in range 77 to 110 while blue is from 132 to 160.

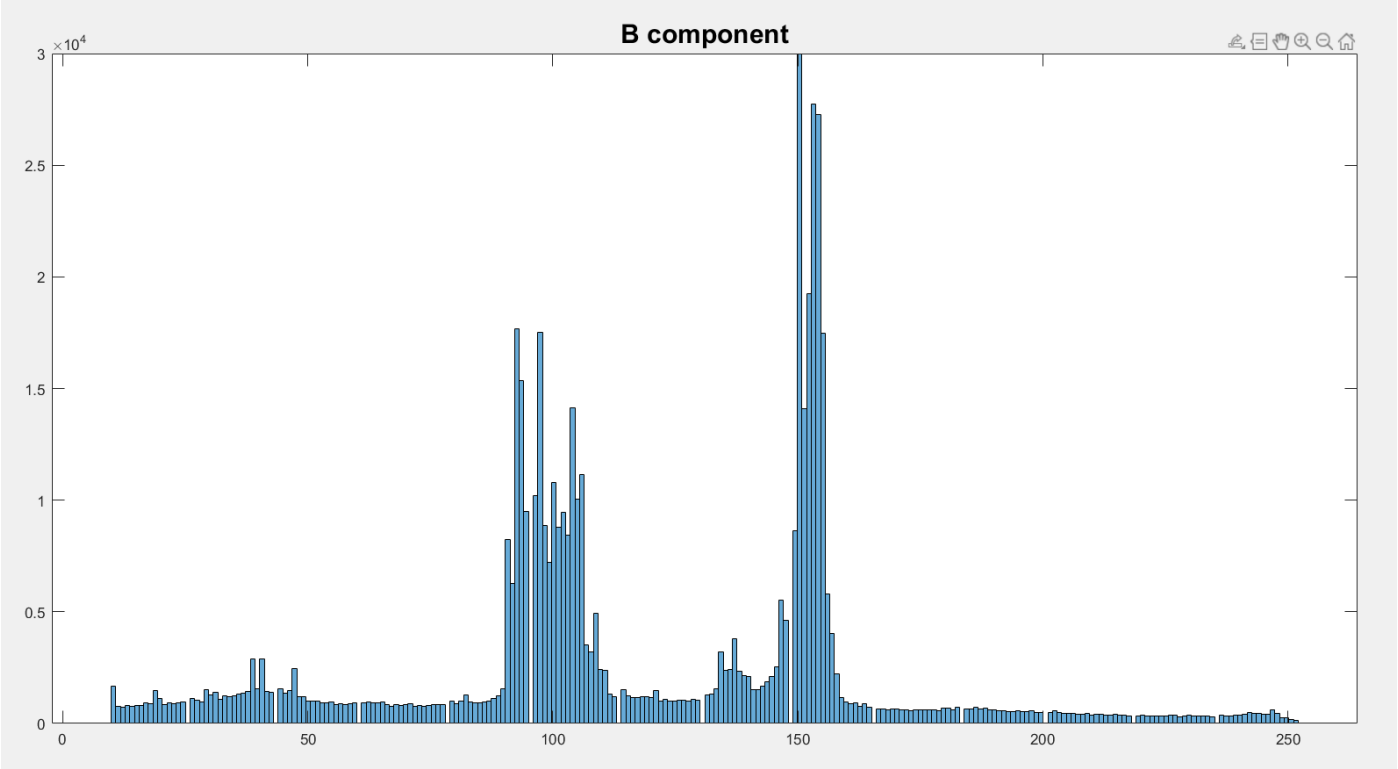
After the rgb2ycbcr, y component is from 78 to 100, cb component is concentrated from 151 to 161 while the cr component is concentrated from 113 to 120

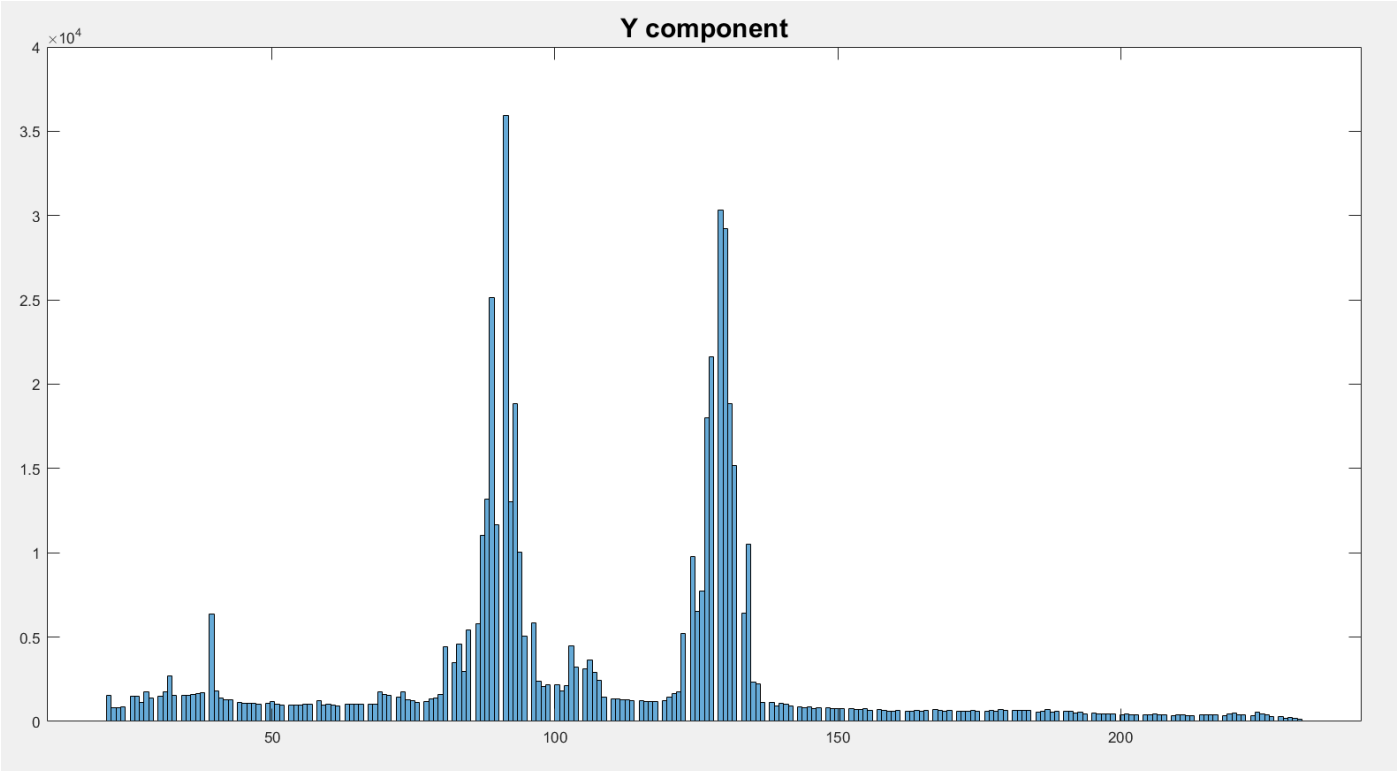
```
I = imread('tennis.png');


figure(2)
histogram(I(:, :, 1), 256)
title('R component', 'FontSize', 18)

figure(3)
histogram(I(:, :, 2), 256)
title('G component', 'FontSize', 18)

figure(4)
histogram(I(:, :, 3), 256)
title('B component', 'FontSize', 18)

img = rgb2ycbcr(I);

figure(5)
histogram(img(:, :, 1), 256)
title('Y component', 'FontSize', 18)


figure(6)
histogram(img(:, :, 2), 256)
title('Cb component', 'FontSize', 18)

figure(7)
histogram(img(:, :, 3), 256)
title('Cr component', 'FontSize', 18)
```

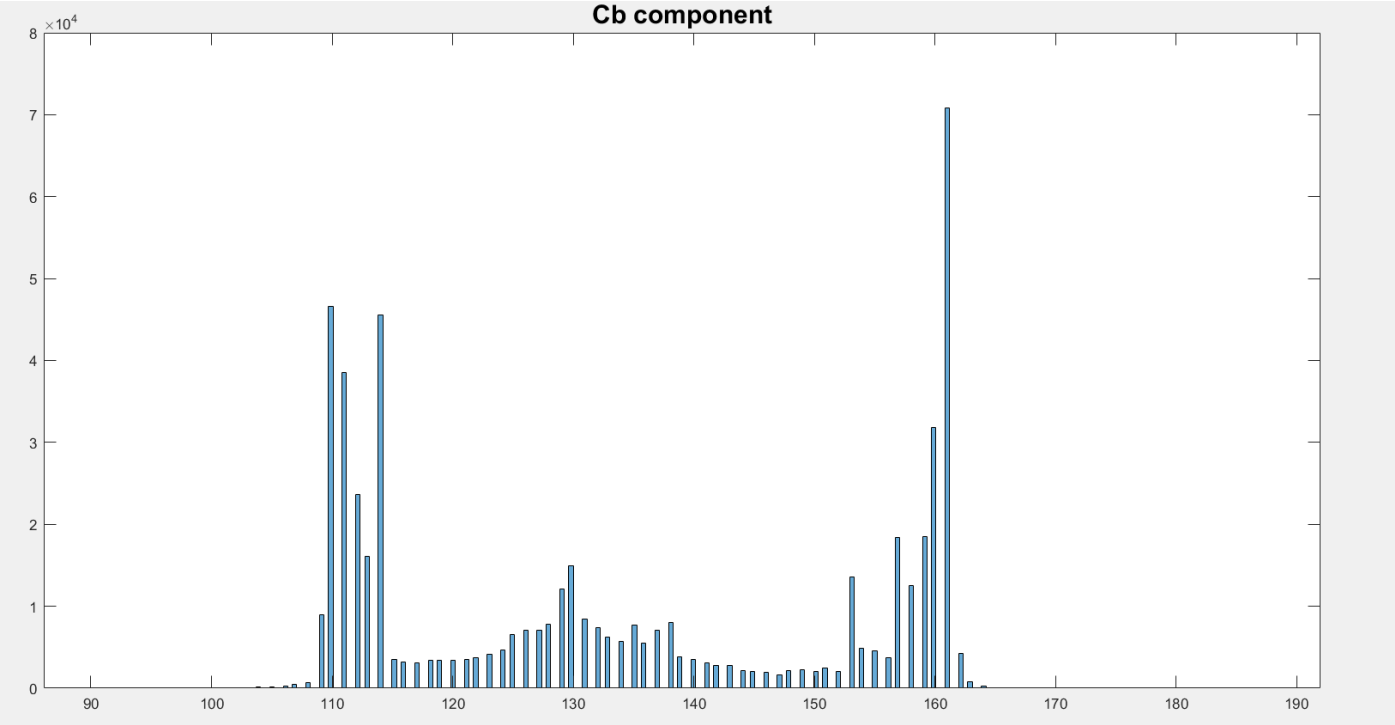
g) Code for Q3

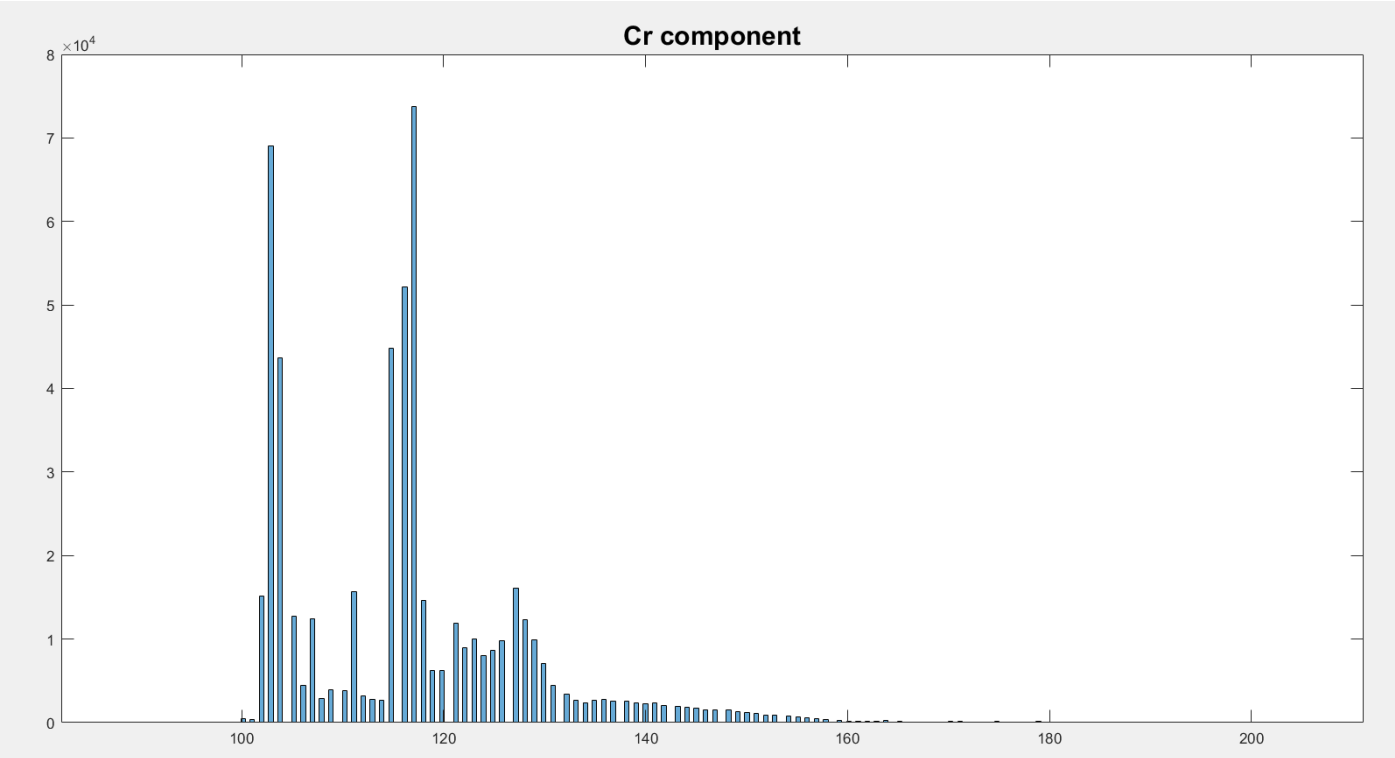

R component
×10^4
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250


G component
×10^4
3.5
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250


B component
×10^4
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250


Y component
×10^4
4
3.5
3
2.5
2
1.5
1
0.5
0
50
100
150
200


Cb component
×10^4
0
1
2
3
4
5
6
7
8
90
100
110
120
130
140
150
160
170
180
190


Cr component
×10^4
0
1
2
3
4
5
6
7
8
100
120
140
160
180
200

Q4 Segmentation

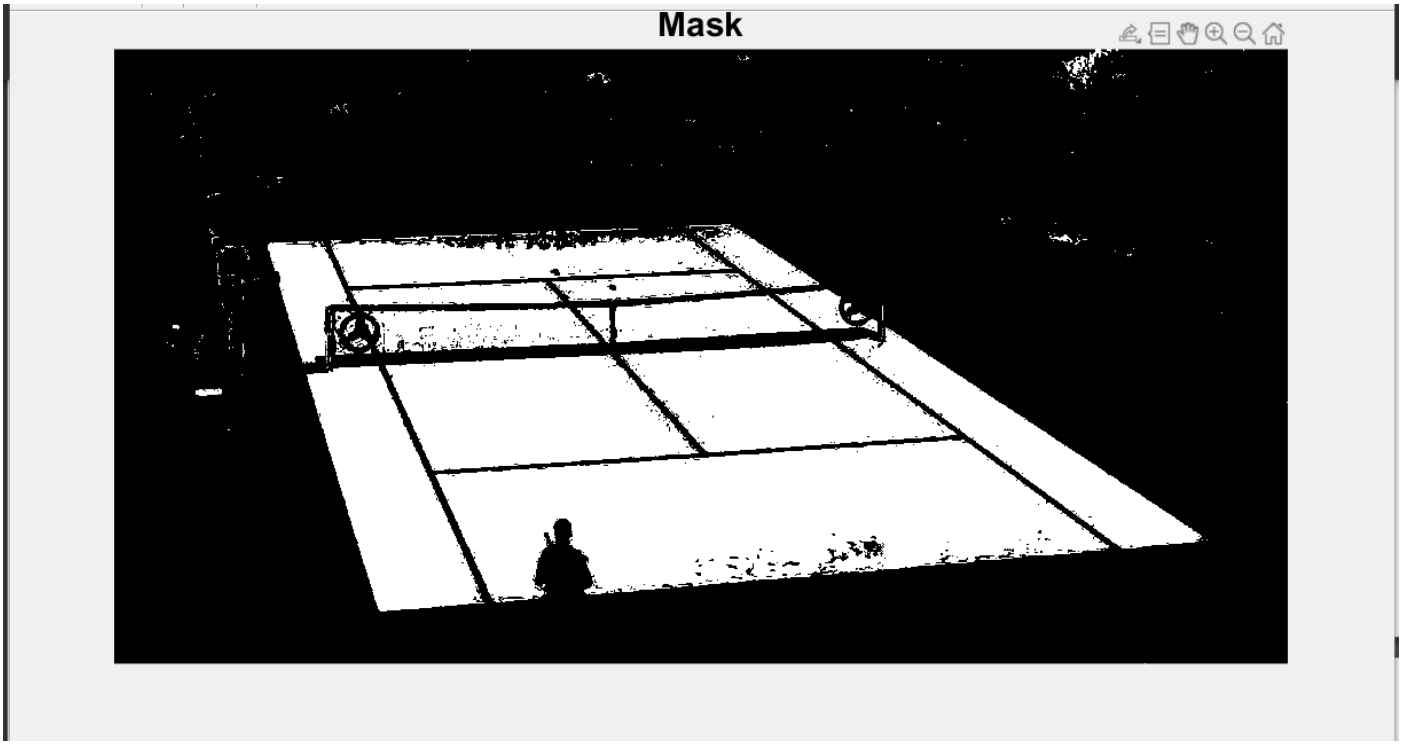
The threshold values used are 38 to 56 for red, 77 to 110 for green, and 132 to 160 for blue. After the rgb segmentation, we can see the court. When we use the green channel segmentation, we can see the court is properly visible, and the board behind it is also visual. The algorithm works in the given threshold ranges of RGB anything above that won't work.

Since a image consists all three colors its better to use RGB segmentation to achieve better result.

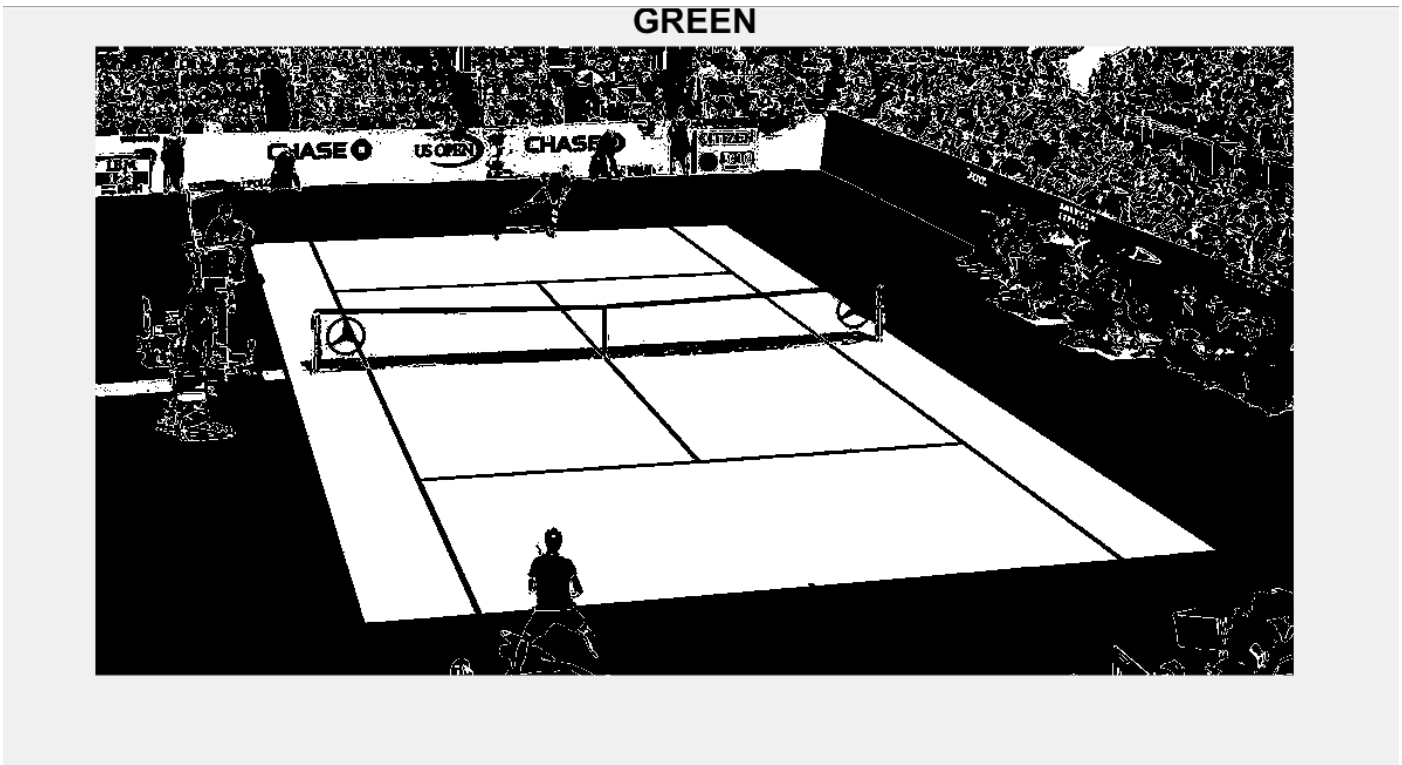
Q1.m | Q2.m | Q3.m | q4.m

```
pic = imread('tennis.png');

r = pic(:, :, 1);
g = pic(:, :, 2);
b = pic(:, :, 3);

mask = (r >= 38 & r <= 56 & g >= 77 & g <= 110 & b >= 132 & b <= 160);

figure('Name', 'MASK')
imshow(mask)
title('Mask', 'FontSize', 18)

figure('Name', 'RED')
imshow((r >= 38 & r <= 56))
title('RED', 'FontSize', 18)

figure('Name', 'GREEN')
imshow((g >= 77 & g <= 110))
title('GREEN', 'FontSize', 18)


figure('Name', 'BLUE')
imshow((b >= 132 & b <= 160))
title('BLUE', 'FontSize', 18)
```

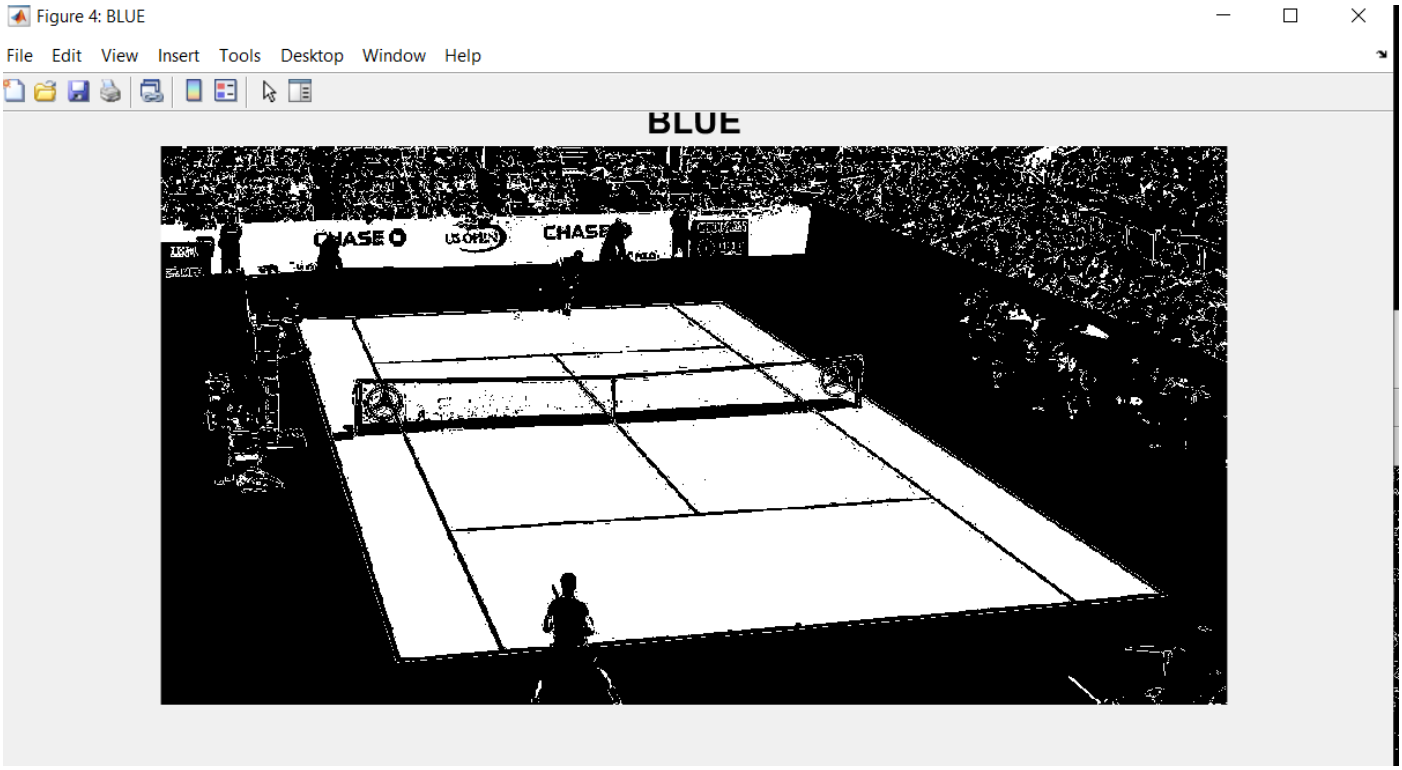
h) Code for Q3



i) Mask Resultant Image



j)Green Resultant Image



j)Blue Resultant Image

h) Red Resultant Image