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

4C8 – DIGITAL IMAGE AND VIDEO PROCESSING

Lab 1 – Pixel Operations

Department of Electrical and Electronics

(e-Report submission)

Table of Content

Submitted by:

Rohan Taneja
19323238
Submission: 01/03/2021

1. Basic:

Starting with the basic implementation.

Initializing the picture and displaying it with original colours.

```
I = imread("jfk.png");
pic = I; figure(1); image(I);  colormap(gray(256)); axis image;
title('jfk.png - original', 'FontSize', 8);
```

1.1. Converting to grayscale and assign matrix.

```
pic = rgb2gray(pic); figure(2); imshow(pic);
title('jfk.png - grayscale', 'FontSize', 8);
```

1.2. Adding Pixel Brightness Value by 128

```
figure(3); imshow(pic+128);
title('jfk.png - added 128 pixels', 'FontSize', 8);
```

The shift towards brighter colour on addition of 128 bits compared to the original pic which results in the exceeding the colour points beyond the range of grayscale that is between [0,255] for uint8. Hence, resulting in loss of details and whiter image.

1.3. Subtracting Pixel Brightness value by 128 bits

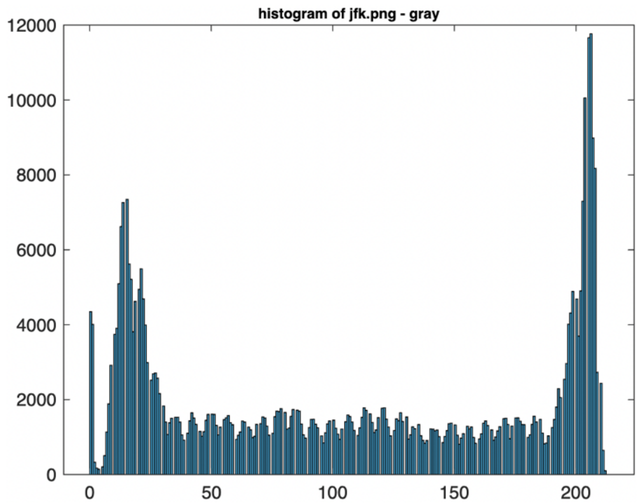
Converse to section 1.2. The subtracted value leads to shift towards black shade. Since, the values start to reach to lower bound of the grayscale range in uint8 [0,255]. Hence, resulting in darker image losing the details.

```
newpic = pic-128; figure(4); imshow(newpic);
title('jfk.png - subtracted 128 pixels', 'FontSize', 8);
```

2. Histogram

2.1. RGB to Grayscale with Histogram

```
pic = imread('jfk.png'); figure(1); image(pic);
pic = rgb2gray(pic); figure(1); imshow(pic); title('jfk.png - gray', 'FontSize', 8)
figure(2); histogram(pic, 256); title('histogram of jfk.png - gray', 'FontSize', 8)
```



3. Histogram of colour planes

3.1. R,G,B Colour Space with Histogram

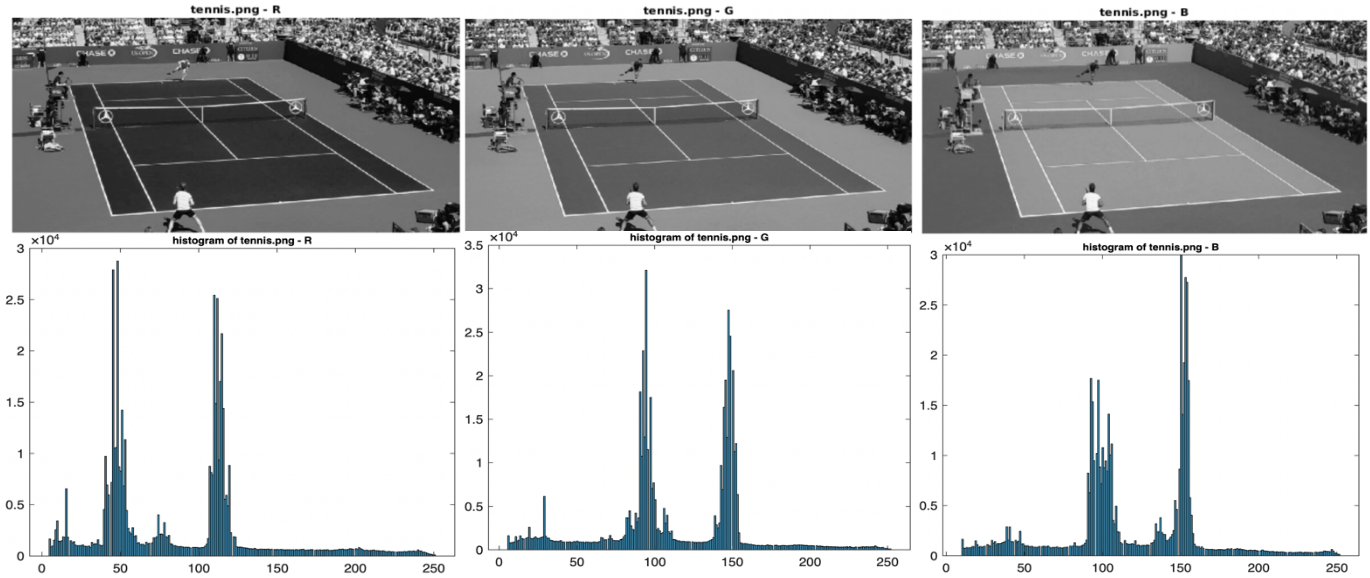
```
pic = imread('tennis.png'); figure(1); image(pic);
r = pic(:,:,1); figure(1); imshow(r); title('tennis.png - R', 'FontSize', 8)
figure(2); histogram(r, 256); title('histogram of tennis.png - R', 'FontSize', 8)
g = pic(:,:,2); figure(3); imshow(g); title('tennis.png - G', 'FontSize', 8)
figure(4); histogram(g, 256); title('histogram of tennis.png - G', 'FontSize', 8)
b = pic(:,:,3); figure(5); imshow(b); title('tennis.png - B', 'FontSize', 8)
figure(6); histogram(b, 256); title('histogram of tennis.png - B', 'FontSize', 8)
```

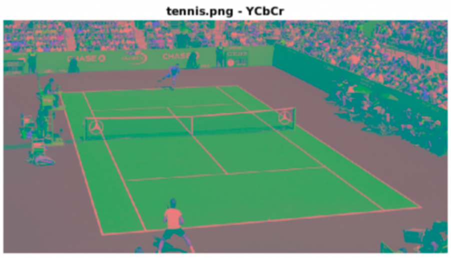
3.2. YCbCr (YUV) Color Space with Histograms

```
pic = rgb2ycbcr(pic);
y = pic(:,:,1); figure(1); imshow(y); title('tennis.png - Y', 'FontSize', 8)
figure(2); histogram(y, 256); title('histogram of tennis.png - Y', 'FontSize', 8)
cb = pic(:,:,2); figure(3); imshow(cb); title('tennis.png - Cb', 'FontSize', 8)
figure(4); histogram(cb, 256); title('histogram of tennis.png - Cb', 'FontSize', 8)
cr = pic(:,:,3); figure(5); imshow(cr); title('tennis.png - Cr', 'FontSize', 8)
figure(6); histogram(cr, 256); title('histogram of tennis.png - Cr', 'FontSize', 8)
```

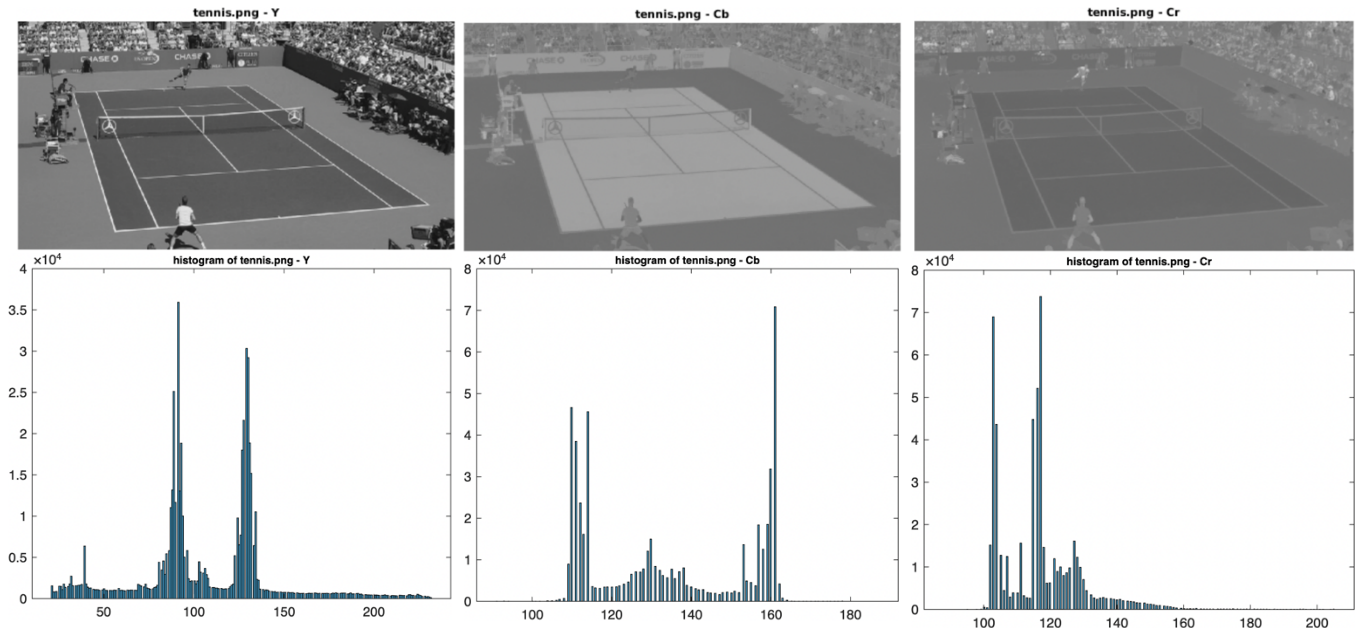
3.3. Comment on Colour Space with Histogram

As we can see in the original image. The tennis court itself is blue therefore in **the colour space RGB**, we can see the highest density corresponds for histogram – B in range [150,200]. The ground covering outside the court is green which results in dense area for histogram – G in range [100,150]. The red component is distributed unevenly and comprises on lower range [0,100].





Now, converting from RGB to colour space YCbCr (YUV-equivalent). We observe in the original picture with YCbCr. Majority of the colour shifts towards chrominance B-Y region as the tennis court covers a great area in the image. Therefore, we can observe in the histogram for Cb (Blue-Green) area denser in the range [140,180], followed by Cr (Red Green) ranging in [100,160]. The luminance Y ranges higher in [90,150] which is sensitive to human eye.

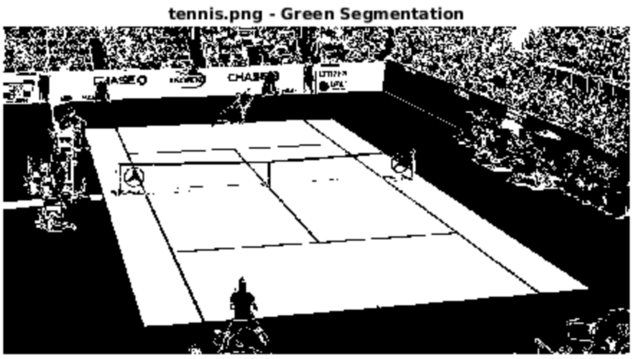


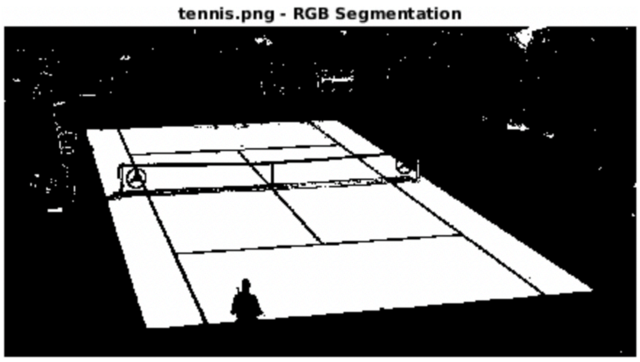
4. Segmentation

4.1. Multiple Segmentation

While performing the colour segmentation of the image – tennis.png. The mask applied is coded as follows:

```
r = pic(:,:,1);
g = pic(:,:,2);
b = pic(:,:,3);
mask = (r > 20 & r < 64) & (g > 50 & g < 130) & (b > 120 & b < 180);
figure(1);imshow(pic); title('tennis.png - Original', 'FontSize', 8)
figure(2);imshow(mask); title('tennis.png - RGB Segmentation', 'FontSize', 8)
```





The resulting segmentation is shown above. After multiples threshold trials on respective colour channels.

Only working with masking the green channel between the specified threshold – the green area surrounding the court is shifted to black, keeping the tennis court on the whiter side. Still, the banners and crowd area required some work. For the green channel, the range (50,130) masks perfectly but breaks as soon as the upper bound is shifted to 160.

When ran individually on R,G or B channel – algorithm shows major improvement with segmentation masking the green channel within the range with resulting focus over the court but the crowd area required segmentation using the other two channels. Introducing three channel segmentation improved the results significantly. The resulting mask was obtained restricting red channel in range (20,64) which significantly helped masking the surrounding of the court. For the blue channel, the range (120,180) was helpful sharpening the edges in the tennis court.