Assignment 2

Özhan Taşdemir, 21827866
Department of Computer Engineering
Hacettepe University
Ankara, Turkey
b21827866@cs.hacettepe.edu.tr

November 18, 2022

1 Introduction

Color transfer is a practical method to change the appearance of a source image according to the color pattern of a target image. Often this means removing a dominant and undesirable color cast, such as the yellow in photos taken under incandescent illumination.

2 Experiment

2.1 Part 1

In this assignment, part 1 focuses on implementing the color transfer method of Reinhard et al and and analyzing RGB histograms of before and after images.

To implement Reinhard et al we need to perform 5 steps

Step 1: Converting the RGB format source and target images to lab color space

2. Compute means (µl, µa, µb) and standard deviations (l, a, b) for each channels.

3. Subtract means from the data points for target image:

4. According to the relative standard deviations of target and source images, scale new data points

5. Add the averages computed for the source to scaled data points and get the color transferred image.

2.2 Image 1

!!I have red-green color blindness. For this reason I replaced green line in histogram plot with yellow!!

Figure 1: Source Image 1

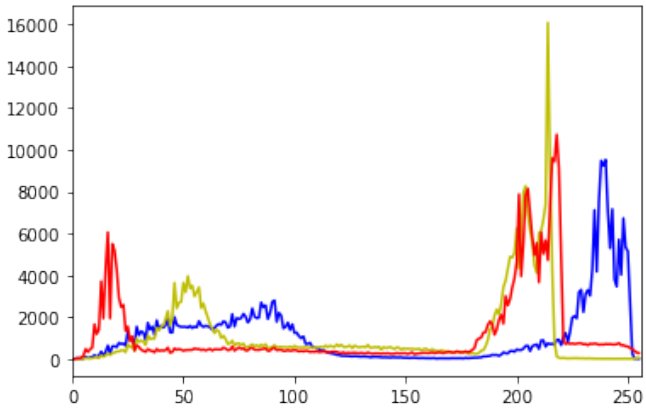

Figure 2: Image 1 Histogram plot

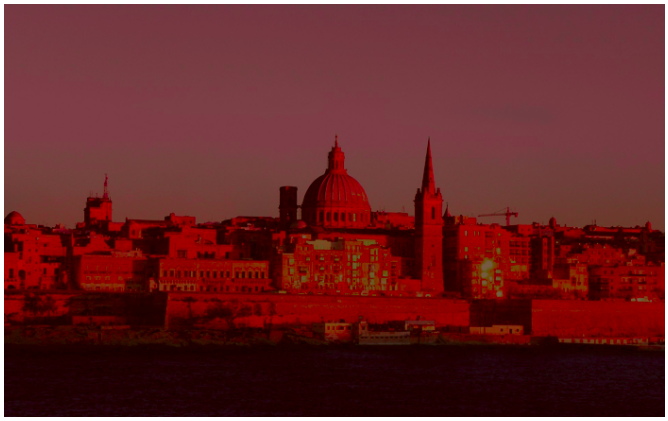

Figure 3: Result Image 1

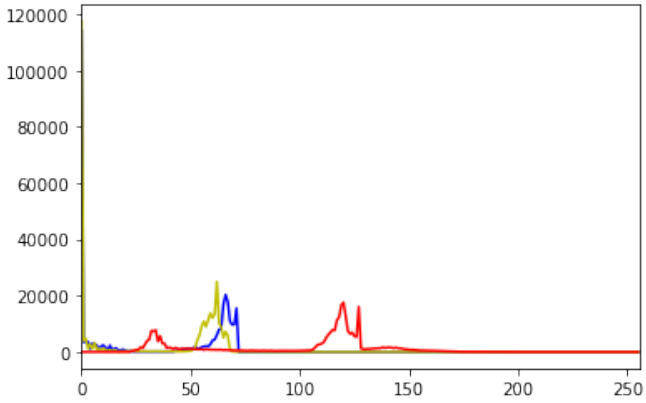

Figure 4: Result 1 Histogram plot

Source Image 1 includes both dark and bright areas. Sea and shadows are dark. The sun hits building and gives a brightness. Also sky is look relatively bright. For this reasons colors placed mostly left and right sides of the histogram plot.

Target image looks dark and red. After color transfer also resulting image looks dark and red. Color transfer performed correctly. As we can see right side of the histogram plot is completely empty because image does not has a bright zones. There is a huge blue and green peak at 0 (from plot its hard to see actually because of the blue line, yellow line and the border of plot is lined up). Because image look red. So blue and green pixel need to be dark (value 0).

2.3 Image 2

Figure 5: Source Image 2

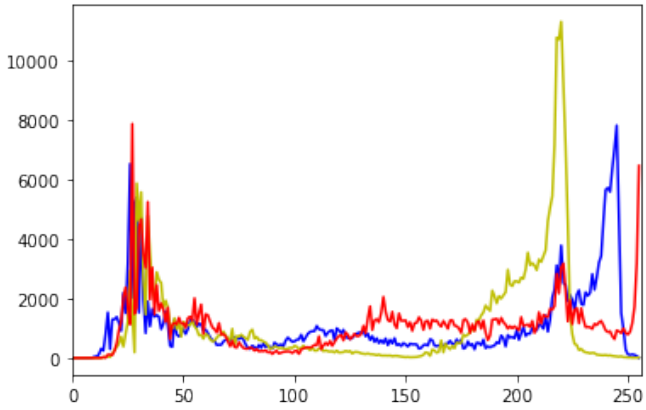

Figure 6: Source 2 Histogram plot

Figure 7: Result Image 2

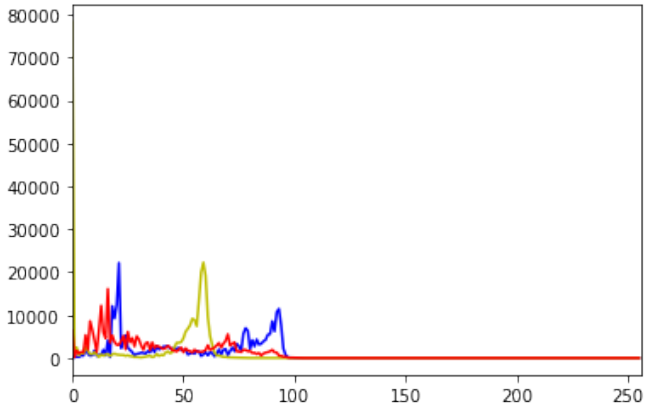

Figure 8: Result 2 Histogram plot

We can analyze Image 2 with dividing it to two sections. Lower part of the image really dark and the upper part is bright. Transition between this part gives us a little mid-bright color pixels(75 to 175)

Our target image is very similar to source image. Because of this color transfer works perfectly. It also creates same vibe with target image. The photo now looks like it was taken at a later time from original. In the result image we have low light. From histogram we can see there is no bright pixels(100 to 255). We have a lot blue pixels. To create sunrise look also we have a some red and green.

2.4 Image 3

Figure 9: Source Image 3

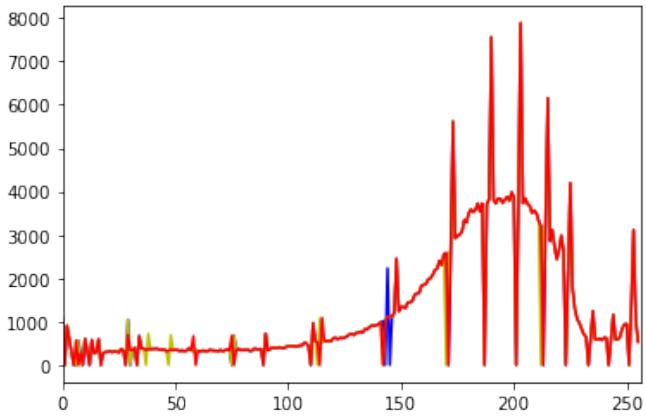

Figure 10: Source 3 Histogram plot

Figure 11: Result Image 3

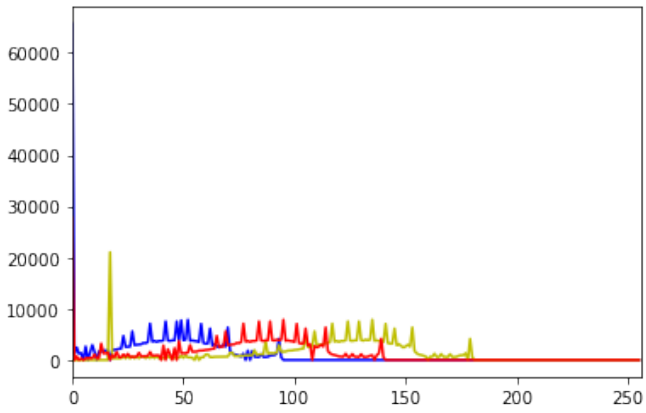

Figure 12: Result 3 Histogram plot

In the third image histogram plot draws our attetion. Three color channels are almost lined up with each other. Reason for this we have onlt white and black color tones. We can create white and black tones with using same amount of red, blue and green.

Target images contains green trees and lot of grass. For this reason our result image becomes green. We have a huge peak at pixel value 0 for blue and red to create this green look.

Figure 13: Source Image 4

Figure 14: Result 4

Figure 15: Target Image

2.5 Image 4 (Failed Example)

For Source image 4, result does not look as expected. I expected hard changes in the sky and plants. Colors of the target image edited aggressively and does not look natural. lab color space minimizes correlation between channels for many natural scenes. I think the reason image does not look natural and the lab color space did not worked.

2.6 Part 2

For Part 2, I first resized the pictures to the same sizes. The different sizes of the pictures adversely affected the results. I've also resized the images to be multiples of the tile size so that I can divide them into equal parts. I divided the images into equal parts with numpy's reshape function. Then I kept the NCC scores in an array by comparing all the tiles with each other. I determined the highest score and performed the color transfer process that we used in the first part. I chose 50px as the tile size. I made my choice experimentally. I had inconsistent results with small numbers and bad results with high numbers. I found 50 pixels to be the ideal number

Table 1: Methods

Method Name	Input(s)	Output(s)	Info
calculate_ncc	img_1, img_2	product	returns ncc score
myround	pixel, base	returns integer	calculates size
color_transfer_blocks	no inputs	images	transfers colors

I did not expect to get effective results from Part 2. SSD and NCC algorithms are not advanced enough to detect similarity in images. They can detect some similarities, but these are not enough for us to get good results.

For first image we have a nice color transfer but there is no smooth transition between tiles. This makes image look like Minecraft.

At Image 2 algortithm founds similar regions and transfer colors. Darkness of the image is as expected but some bright tiles destroys composition.

For Image 3 algorithm produces unexpected result. Target image is a dark city. But our result is not dark. Simply our result is blue version of the source image. We can say algorithm having a problem finding similar tiles.

Figure 16: Source Image 1

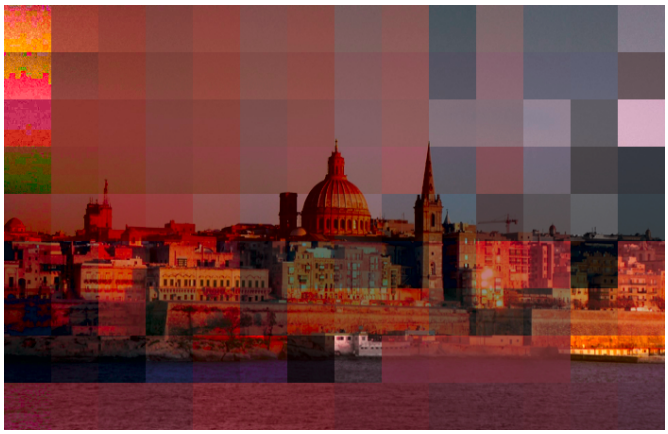

Figure 17: Result 1

Figure 18: Source Image 2

Figure 19: Result 2

Figure 20: Source Image 3

Figure 21: Result 3

3 Conclusion

In this assignment we implement the color transfer method of Reinhard et al. Using this method, we performed the color transfer process quickly and effectively. In the second part, we also identified similar regions and performed color transfer between those regions.