# BBM 415 - Fundamentals of Image Processing
# Laboratory Assignment 2

Murat Çelik, 21827263
Department of Computer Engineering
Hacettepe University
Ankara, Turkey
b21827263@cs.hacettepe.edu.tr

November 15, 2022

## 1 Introduction

Color manipulation on photographs has become a rapidly increasing problem with the use of social media. There are different solutions to this problem. As one of these solutions, the RGB-edited picture has a difficult solution for this problem, "lab color space" was developed as an alternative [1]. This color space makes it easy to change the color effect in images.

In this assignment, we will apply color effects change on our source image depending on our target image.

## 2 Experiment

### 2.1 Part 1

In this section, we applied the basic color transfer methodology. The steps are given below in order.

1. Convert the source and target images to l$\alpha\beta$ color space representation from RGB color space.
2. For each channel of source and target images, compute means ($\mu_l$, $\mu_\alpha$ , $\mu_\beta$) and standard deviations ($\sigma_l$, $\sigma_\alpha$ , $\sigma_\beta$).
3. Subtract means from the data points for source image:

$$l^* = l - \mu_l$$

$$\alpha^* = l - \mu_\alpha$$

$$\beta^* = l - \mu_\beta$$

4. Scale new data points according to the relative standard deviations of target and source images:

$$l' = l^* \frac{\sigma_t^l}{\sigma_s^l}$$

$$\alpha' = \alpha^* \frac{\sigma_t^\alpha}{\sigma_s^\alpha}$$

$$\beta' = \beta^* \frac{\sigma_t^\beta}{\sigma_s^\beta}$$

5. Add the averages computed for the target to scaled data points.

```python
# 1. Convert the source and target images to lαβ color space representation from RGB color space
source = cv2.cvtColor(source_img, cv2.COLOR_BGR2LAB).astype("float32")
target = cv2.cvtColor(target_img, cv2.COLOR_BGR2LAB).astype("float32")

# 2. For each channel of source and target images, compute means (μl, μa, μβ ) and standard deviations (σl, σa, σβ ).
def compute_mean_std(image):
    # compute the mean and standard deviation of each channel
    (l, alpha, beta) = cv2.split(image)
    (lMean, lStd) = (l.mean(), l.std())
    (alphaMean, alphaStd) = (alpha.mean(), alpha.std())
    (betaMean, betaStd) = (beta.mean(), beta.std())
    # return the color statistics
    return (lMean, lStd, alphaMean, alphaStd, betaMean, betaStd)

(lMeanSrc, lStdSrc, alphaMeanSrc, alphaStdSrc, betaMeanSrc, betaStdSrc) = compute_mean_std(source)
(lMeanTar, lStdTar, alphaMeanTar, alphaStdTar, betaMeanTar, betaStdTar) = compute_mean_std(target)

# 3. Subtract means from the data points for source image:
(l, a, b) = cv2.split(source)
l -= lMeanSrc
a -= alphaMeanSrc
b -= betaMeanSrc

# 4. Scale new data points according to the relative standard deviations of target and source images:
l = (lStdTar / (lStdSrc + 0.000001) ) * l
a = (alphaStdTar / (alphaStdSrc + 0.000001)) * a
b = (betaStdTar/ (betaStdSrc + 0.000001)) * b

# 5. Add the averages computed for the target to scaled data points.
l += lMeanTar
a += alphaMeanTar
b += betaMeanTar
```

Figure 1: Code Line

The main code section is given above. In this section, the above-mentioned operations can be seen. Precautions have been taken for Zero-Division error. The main build was done with Opencv and Numpy libraries

### 2.1.1 Good Results

Considering the closeness of the color distributions and the harmony of the photographs, successful results were obtained in some pictures. It has been observed that the overall success is low. It is thought that more successful and efficient results can be obtained with newer techniques of paintings.


Figure 2: Succesfull Color Transformation

In Figure 2, the goal is for the source image to turn red. It has successfully accomplished this.


Figure 3: Succesfull Color Transformation

In Figure 3, the source image is expected to be like an image taken in the evening. Source image transformed successfully.

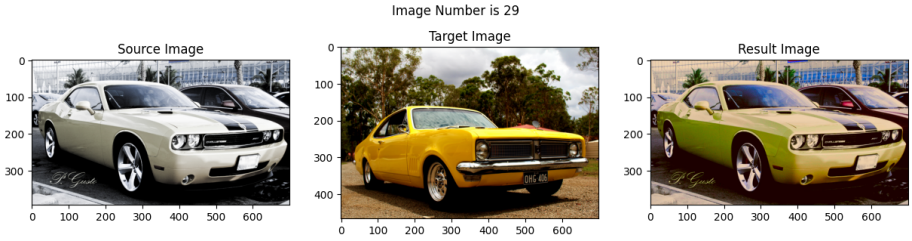


Figure 4: Succesfull Color Transformation

In Figure 4, it has been observed that a white car can be converted to a yellow car.

### 2.1.2 Bad Results

Color transfer proceeds unsuccessfully between some images. The main reasons for this are that the color distributions are so incompatible that they are not similar to each other.

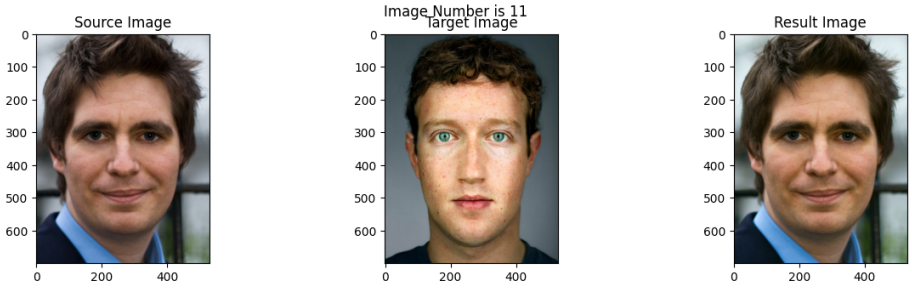


Figure 5: Unsuccesfull Color Transformation

In Figure 5, the change in the human face was limited only to skin color and an inefficient result emerged. Here, the source image was expected to take the hair color, eye color and skin color of the target image.

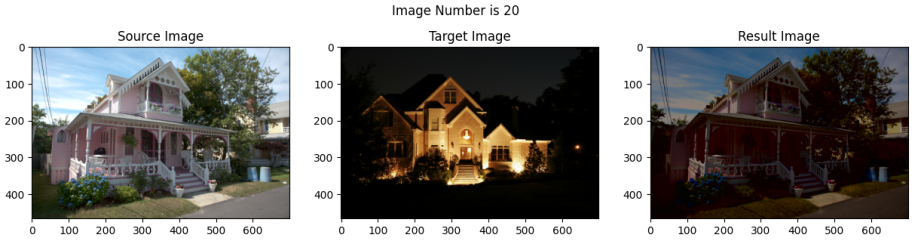


Figure 6: Unsuccesfull Color Transformation

In Figure 6, Result picture, it is expected that our house is predominantly white and the environment is dark. But a bad picture emerged where the house is dark and the surroundings are gray.

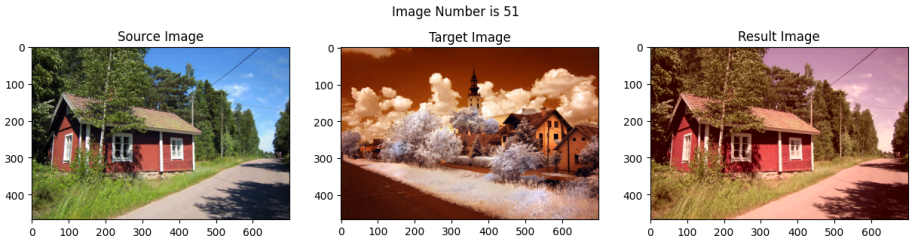


Figure 7: Unsuccesfull Color Transformation

In Figure 7, the source image is blurred due to the red tone in the target image. This produced a poor quality and bad picture as output. The main reason for this is that the target picture is taken from afar and with non-normal colors.

### 2.1.3 Histogram Results

Histogram analysis of source, target and result images has been extracted.

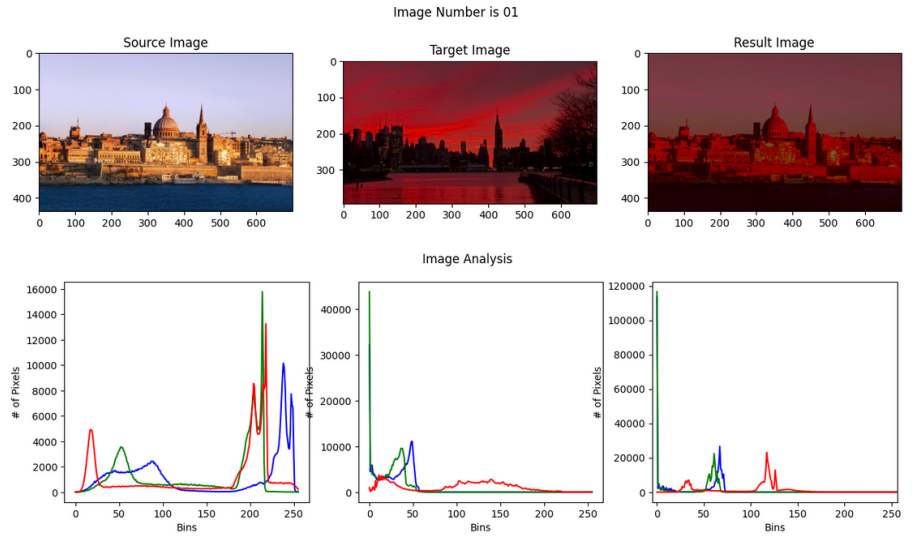


Figure 8: Histogram Analysis of Images

In Figure 8, the source image in the picture can be said to be a balanced picture in all three channels. The dominance of the red color is evident in the target picture. It was as expected that red would be superior to other color channels in the result picture.

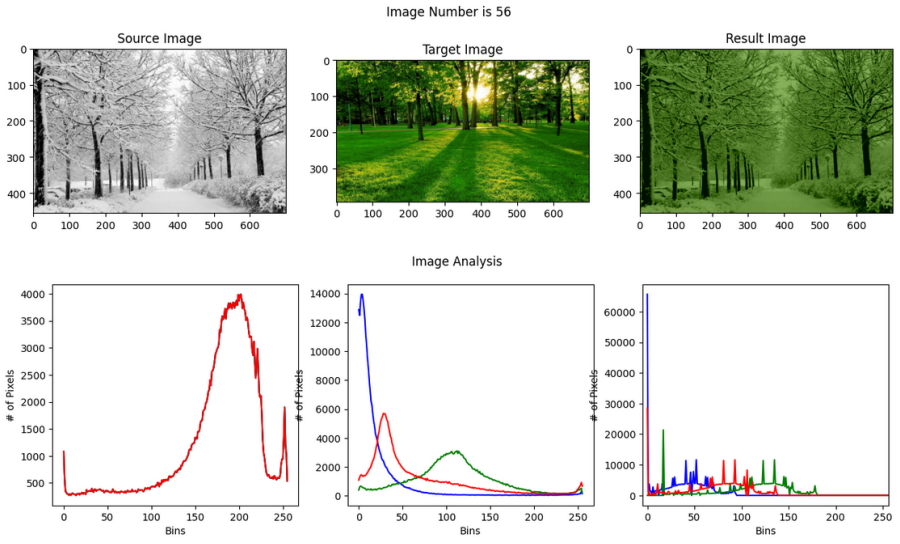


Figure 9: Histogram Analysis of Images

In Figure 9, the Source picture is from a winter period. It also has 3 color channels as strong. The second picture is of a forest and the green color is dominant. It is seen that the result image is such that it reduces the power of the 3 color channels. Here, the green color remained a little stronger than the others.

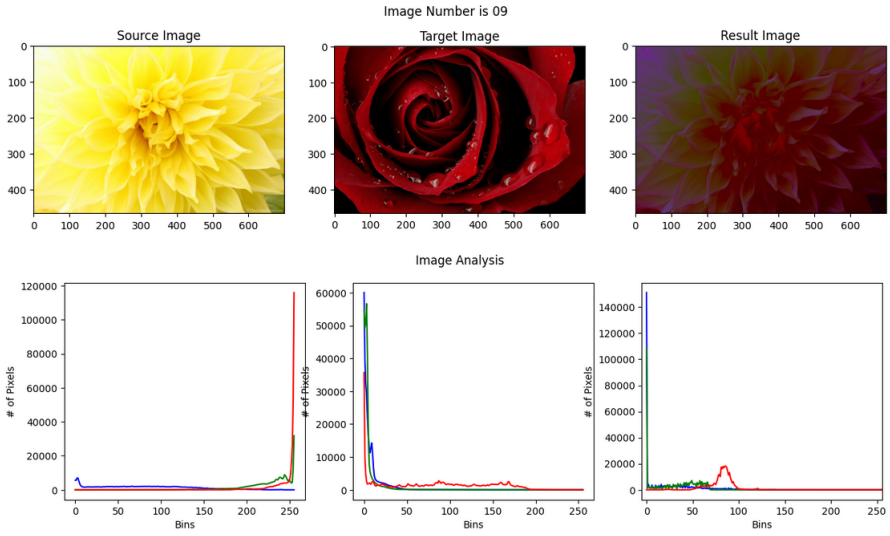


Figure 10: Histogram Analysis of Images

In Figure 10, the picture shows the process of transforming a yellow flower into red. The color channels in the picture are extremely strong due to the density of white and yellow. In our Result picture, these powers have decreased and the effect of red has been observed.

## 2.2 Part 2

In this section, we perform the function used in part 1 by using similarity detection between rectangular regions on the picture. Our main goal is to replace one part of the picture with the corresponding part of the picture in the other picture.

### 2.2.1 Experiment Details

- In order to eliminate the size inconsistency in the data, the resize operation was performed to the closest [512, 768, 1024, 1280] dimensions.
- It is divided into rectangular regions of 64, 128, and 256 sided lengths.
- Similarity was determined between rectangular regions with the sum of squared differences (SSD) technique.
- The best variables were determined with different combinations and the final output was shown in that way.

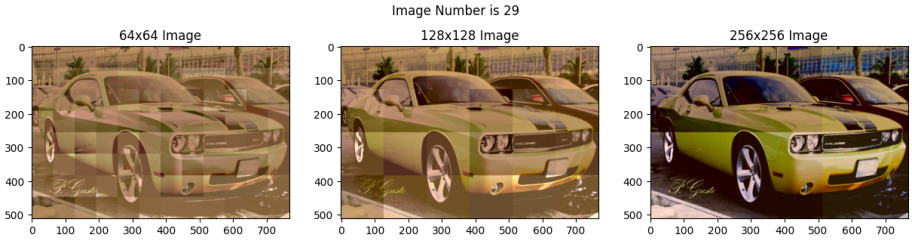


Figure 11: Comparision of Size of Rectangular

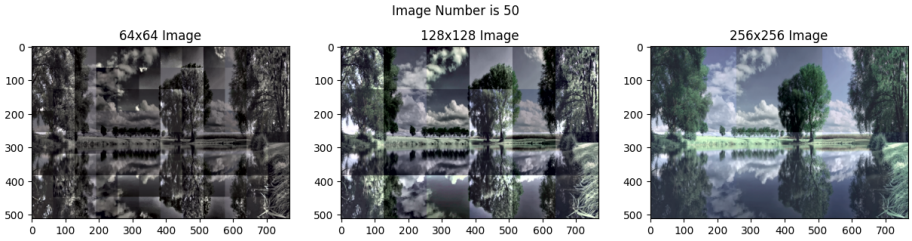


Figure 12: Comparision of Size of Rectangular

Working with rectangular regions has produced unsuccessful results. It was observed that as the regions got smaller, the picture quality decreased even more. Due to the color differences between the rectangular regions, the regions became visible in the picture, which strengthened the failure.

### 2.2.2 Experiment Results

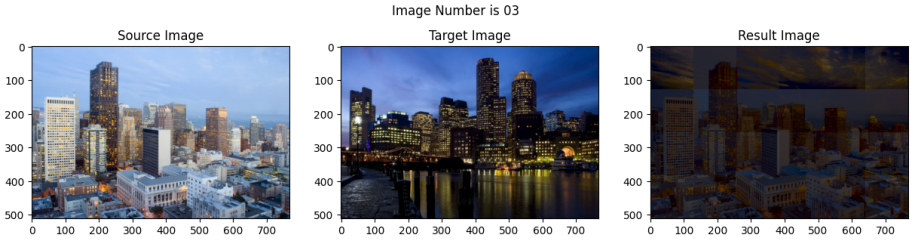


Figure 13: Comparision of Size of Rectangular

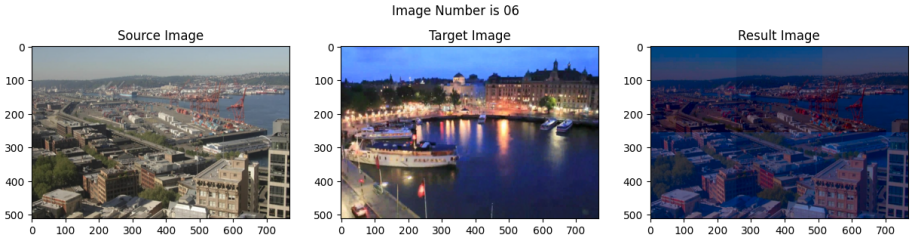


Figure 14: Comparision of Size of Rectangular

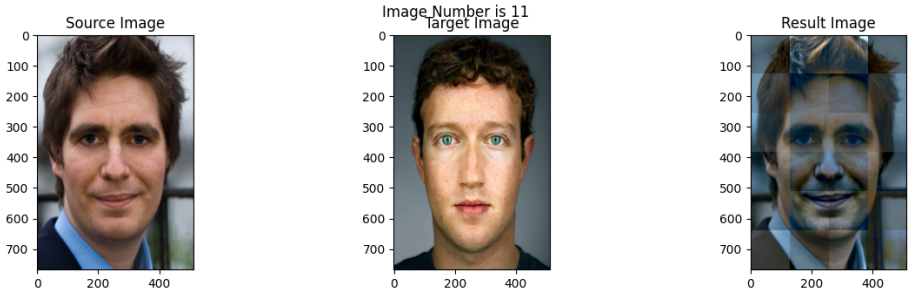


Figure 15: Comparision of Size of Rectangular

It was observed that the results were quite bad compared to part 1. Experimental steps and results were performed for each data with a jupyter notebook. The best outputs are shown.

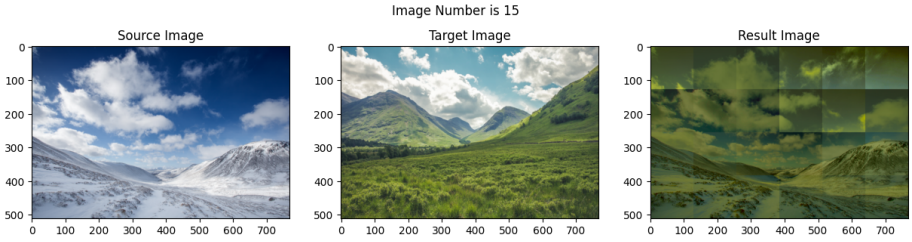


Figure 16: Comparision of Size of Rectangular

# 3 Conclusion

Overall, the experiments were performed with average success. In this assignment, I learned about the lab color space. Information about the basic color transformation technique was obtained and these were tried with different images [2]. Analysis was done with the color histogram. Research will continue with investigations on how new techniques are realized.

# References

[1] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

[2] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21:34–41, 2001.