UNIVERSITY OF TEXAS AT AUSTIN

NEU 380E Natural Image Statistics

Haoqi Wang (haoqiwang@utexas.edu, ID: hw9335)

May 12, 2020

1 INTRODUCTION

Natural images are devoted to displaying elements around us, such as landscapes, wildlife, plants, man-made objects, etc. Although it seems that the number of natural images is infinite, they are very rare in image space and form a complicated distribution. Studying the statistical structure of natural scenes can be significant for the understanding of visual perception. The goal of this project is to analyze natural image statistics through calculating the conditional expectation of each color channel value of natural scenes, which is later used for the estimation of a missing color channel. Especially, we wanted to see if the mean contrast is correlated with the conditional expectation of color and helps the estimation. The measurements reveal some interesting configurations of RGB color space, which may play important roles in the vision system.

2 METHODS

This project was coded in python and all the codes and results were uploaded to github.

2.1 Preparation

2.1.1 Preparation of CSV files

In order to store and calculate the conditional expectation of RGB value, we first generated comma-separated values (CSV) files. Here, we considered mean contrast as 10 separate levels from 0 to 0.9. For example, for the color blue, it needs 10 CSV files to store the blue values given red and green values as well as their mean contrast. What is more, it needs another 10 CSV files to record how many times that specific value occurred for the calculation of expectation. For each CSV file, it is a matrix with the dimension of 256×256 and every cell is initialized as 0, where the row and column indices stand for green and red values, respectively. In summary, we need 20 files for each color channel and 60 files in total. More details and code can be found here.

2.1.2 Preparation of images

We downloaded natural scene collection from the Natural Scene Statistics in Vision Science database, which contains 9 image sets and 1204 images in total. Around 30 images were randomly picked for later color estimation and the rest were utilized for image statistics analysis. All the original images are 16 bits with PPM format. We divided the color value of each channel of each pixel by 2^8 to transform them into 8 bits images with jpg format. More details can be found here.

2.2 Image analysis

After preparation of CSV files and images, we began image analysis.

2.2.1 Mean contrast calculation

Consider a pixel X_5, it has 8 pixels around it as shown in figure 2.1. To simplify the calculation, we omitted boundaries since they do not have 8 neighbors.

X_1	X_2	X_3
X_4	X_5	X_6
X_7	X_8	X_9

Figure 2.1: Illusration of pixels

First, the root mean square (RMS) contrast of pixel X_5 was calculated through equation (2.1) and (2.2). X_i stands for the R/G/B value of ith pixel.

$$C_X = \sqrt{\frac{1}{9}\sum_{i=1}^{9}(\frac{X_i - \overline{X}}{\overline{X}})^2} \tag{2.1}$$

, where

$$\overline{X} = \frac{1}{9}\sum_{i=1}^{9} X_i \tag{2.2}$$

Then the mean contrasts were calculated through equations (2.3), (2.4), (2.5). We kept one decimal, so there would be 9 possible values from 0 to 0.9. Note that because of limited computer precision, sometimes the mean contrast could be slightly higher than 1 and we counted these as 0.9.

$$C_{G,B} = \frac{1}{2}(C_G + C_B) \tag{2.3}$$

$$C_{R,B} = \frac{1}{2}(C_R + C_B) \tag{2.4}$$

$$C_{R,G} = \frac{1}{2}(C_R + C_G) \tag{2.5}$$

2.2.2 Pixel information extraction

At the beginning of image analysis, all the CSV files were imported and grouped into 6 256 × 256 × 10 three-dimensional arrays. For instance, for the color blue, it has 2 three-dimensional arrays. One is for recording values and another one for recording how many times the values occurred. Then, we iterated each pixel in each image. For each pixel, the RGB values were extracted and the mean contrasts were computed. The computed mean contrast was used to locate the "page" of an array and the position was given by the other two color values. For example, for color blue of a pixel, $C_{R,G}$ is calculated according to its surrounding pixels, and the row and column indices are given by the green, red values, respectively. Next, we would calculate the average value. Here, we first multiplied the value and the times the value occurred. And plus this new input value then divided it by the times the value occurred plus one. Finally, the new averaged value was rounded to an integer and we updated two arrays. In this way, we can keep track of information and calculate average conveniently. What is more, it helps to save memory and combine all the contrasts later. Python could be very slow while performing large amounts of for loops, so we imported jit from numba to accelerate the program. More details can be found here. Including importing CSV files and extracting pixel information and exporting CSV files, each image with a resolution of 4284 × 2844 takes about 14s. We also recorded the relevant information in a txt file while running the program. Algorithm 1 describes the image analysis methodology in detail.

Algorithm 1 Pixel information extraction

for each image **do**
    import CSV files and group them into 3D arrays as RG, RG record, RB, RB record, GB, GB record
    read image
    for each pixel [i, j] (except boundaries) **do**
        r = image[i, j, 0], g = image[i, j, 1], b = image[i, j, 2]
        calculate C_R, C_G, C_B according to (2.1)
        calculate $C_{G,B}$, $C_{R,B}$, $C_{R,G}$ according to (2.3), (2.4), (2.5)
        locate values, RG cell = RG[g, r, $C_{R,G}$], RG record cell = RG record[g, r, $C_{R,G}$]
        update RG, RG[g, r, $C_{R,G}$] = (RG cell × RG record cell + b) / (RG record cell + 1)
        update RG record, RG record[g, r, $C_{R,G}$] = RG record cell+ 1
        same for RB, GB
    end for
    export CSV files
end for

2.3 Statistics visualization

2.3.1 Heat map and 3D surface plot

We utilized the heat map to visualize statistics with different contrast. For a matrix, the indices of row and column provide the position and the heat value is given by the color value, from 0 to 255. What is more, we averaged all the contrast matrices to generate a matrix not considering contrast. We also plotted heat maps and 3D surfaces to visualize the result.

2.3.2 Calculate MSE

For each color, we have 10 contrast matrices. We calculated the mean squared value (MSE) for each contrast and plotted them.

2.4 Image recovery

2.4.1 Color removal

Before recovering color, we first removed color by setting the color channel matrix as 0.

2.4.2 Color estimation

The predicted value is the conditional expectation given the other two colors and the mean contrast, namely, $E(R|G,B,C_{G,B})$, $E(G|R,B,C_{R,B})$, $E(B|R,G,C_{R,G})$, which were stored in CSV files after training. We need to extract information from the matrices we recorded to recover the image. For example, if we are going to recover red, we applied (2.6). For a pixel, the green and blue values were extracted first. Then, green and blue values of the pixels around it were extracted and used to calculate mean contrast. This is how we find the estimation of the red value. The rest green and blue values are similar and equations are given by (2.7), (2.8).

$$\widehat{R}_{opt} = R(G,B,C_{G,B}) \tag{2.6}$$

$$\widehat{G}_{opt} = G(R,B,C_{R,B}) \tag{2.7}$$

$$\widehat{B}_{opt} = B(R,G,C_{R,G}) \tag{2.8}$$

For recovering images without considering contrast, the process is similar but we do not need to calculate mean contrast. More details of removing and estimating colors can be found here.

2.4.3 Image comparison

To quantify and evaluate the quality of recovered images, we compared the original image and recovered image by calculating the MSE. The equation is given by (2.9).

$$MSE = \frac{1}{n}\sum_{i}^{n}(\widehat{x}_i - x_i)^2 \tag{2.9}$$

, where x_i is the predicted color value and n is the number of predictions.

2.5 Image display

The transformed images look dark on screen, so we utilized gamma correction to make it look brighter. Here is how we corrected. First the original 8 bits image was scaled to $I_{in} \in (0,1)$, then gamma correction was applied with equation (2.10)

$$I_{out} = I_{in}^{\gamma} \tag{2.10}$$

, where we chose $\gamma = 1/2$. After the correction, I_{out} was then scaled back to 8 bits image to view. This is only done when we display the images.

3 Results

3.1 Pre-test

Before we started training, we tested the algorithm on a random natural scene image from the internet.

3.1.1 Statistics visualization

The result of image analysis can be found here. We did not estimate boundaries since the image was analyzed without considering boundaries. The optimal estimation of missing color are shown as figure 3.1, 3.2, 3.3, given the observed color values on the horizontal and vertical axes. We can find the data for high contrast is quite sparse. This situation may improve after we analyze more images. The combination of contrast levels is shown as figure 3.4.

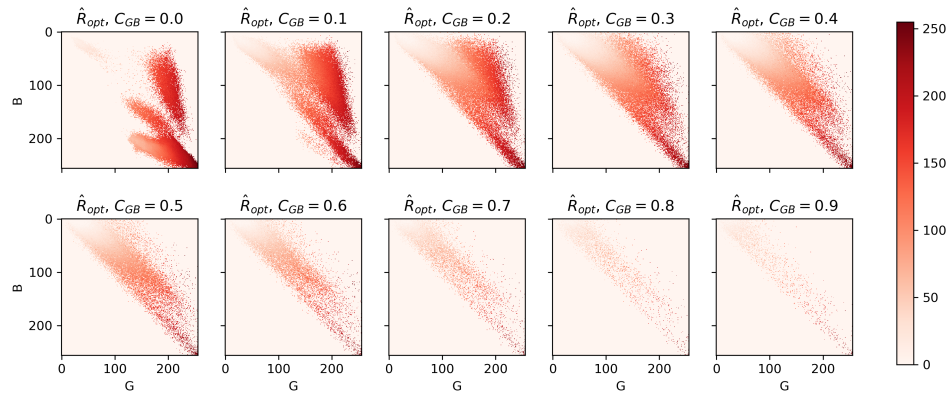


Figure 3.1: Estimation of red channel

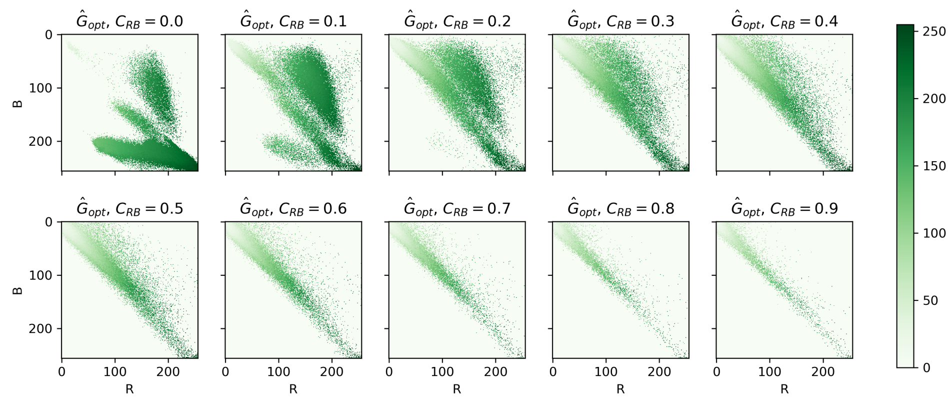
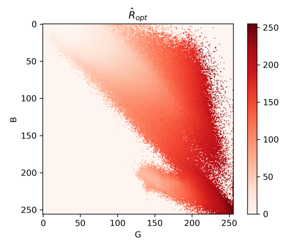
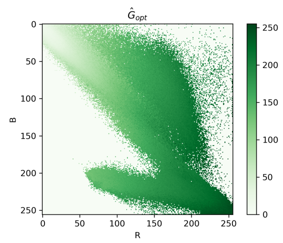


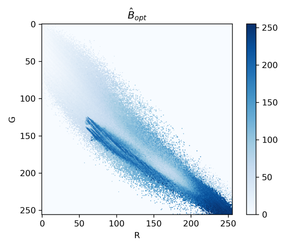
Figure 3.2: Estimation of green channel



(a) Estimation of red channel



(b) Estimation of green channel



(c) Estimation of blue channel

Figure 3.4: Estimation of a missing color channel

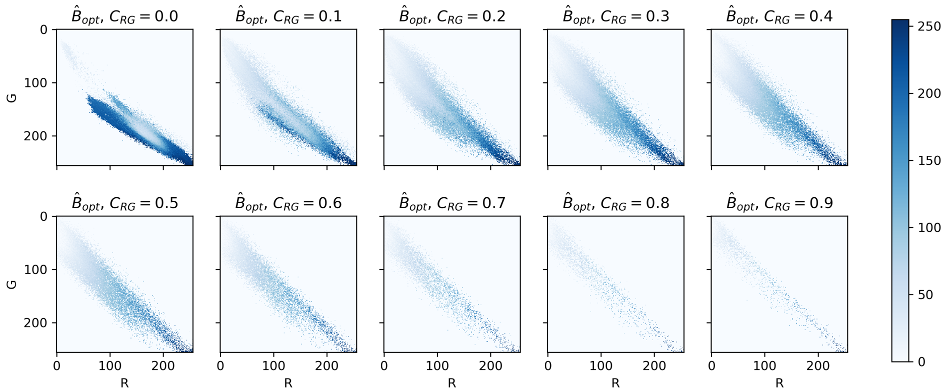


Figure 3.3: Estimation of blue channel

3.1.2 Image recovery

We simply recovered images from the data we just extracted on the same image. The recovered image is shown as figure 3.5. The first column is the original image. The second column is the image removed red, green, blue color channel, respectively. The third column is the recovered image considering mean contrast levels. The fourth column is the recovered images not considering mean contrast levels. The MSE of recovered images considering mean contrast levels seems smaller than that of not considering mean contrast levels. The estimation of blue seems worse compared to the other 2 channels.

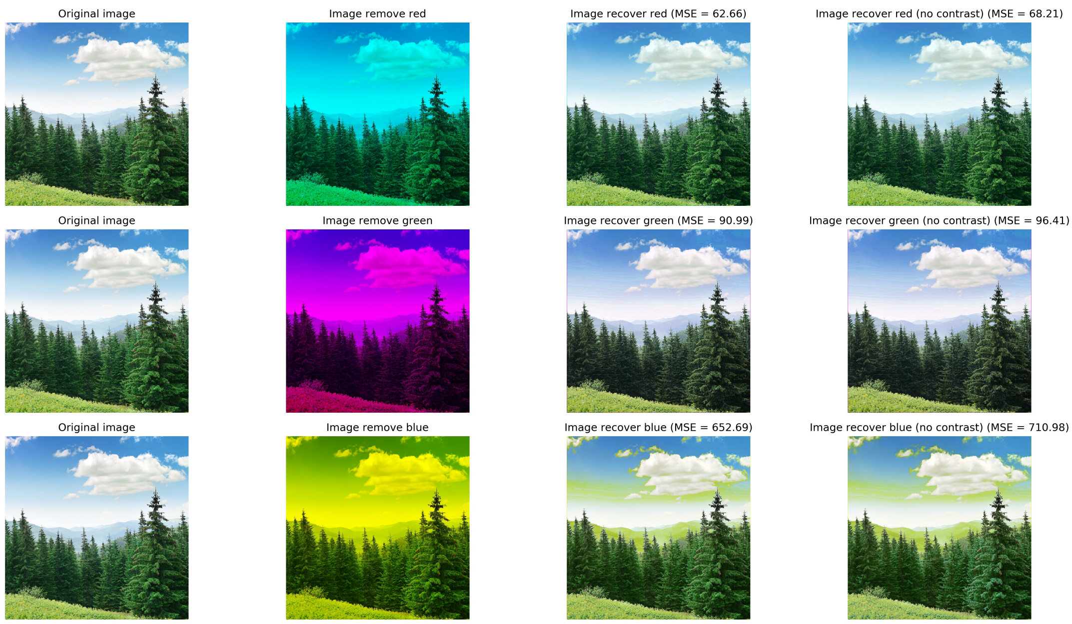


Figure 3.5: Pre-test image recover

This pre-test demonstrated the correctness of our algorithm.

3.2 Experiment

Here we analyzed all the training images.

3.2.1 *Statistics visualization*

The result of image analysis can be found here. The optimal estimation of missing color is shown as figure 3.6, 3.7, 3.8, given the observed color values on the horizontal and vertical axes. For all the colors, the configuration of different contrast seems similar except that it becomes sparse when the mean contrast is high. For color red and green, the configurations seem close while different from blue.

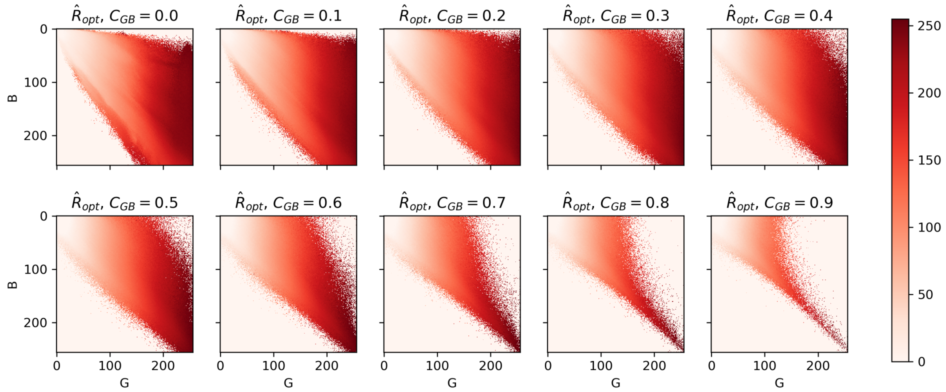


Figure 3.6: Estimation of red channel

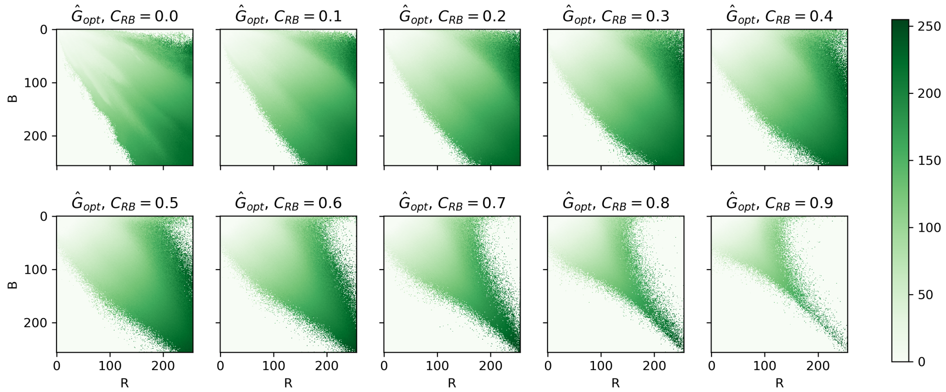


Figure 3.7: Estimation of green channel

The MSE of 10 contrasts is shown as figure 3.9. It seems interesting that the MSE of red is the highest while blue is the lowest. Perhaps this is related to some properties of these colors in natural scenes.
The heat map of the combined contrast plot is shown as figure 3.10 and the surface plot is shown as figure 3.11. From the 3D surface plot, we can see the difference of structure more obviously. For color blue, its peak lies on the diagonal. For red and green, they seem more like a plane.

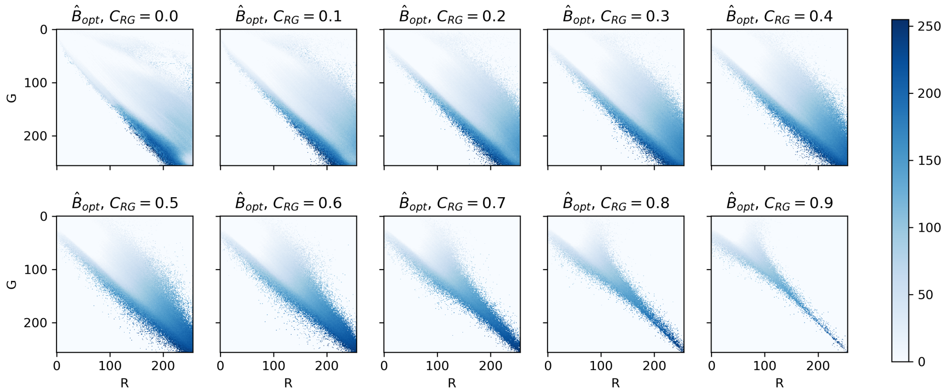


Figure 3.8: Estimation of blue channel

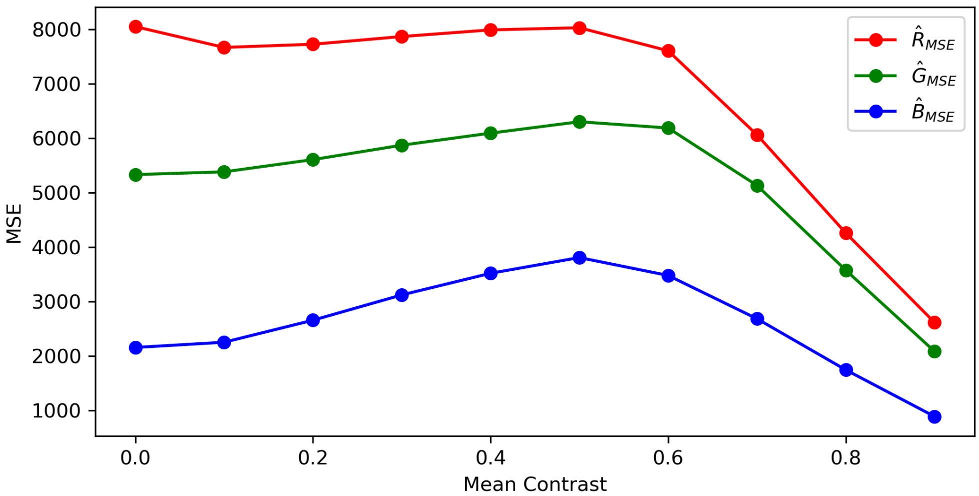
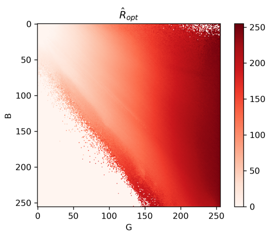
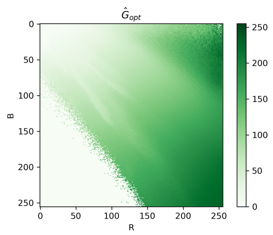


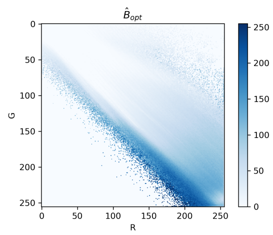
Figure 3.9: MSE contrast



(a) Estimation of red channel

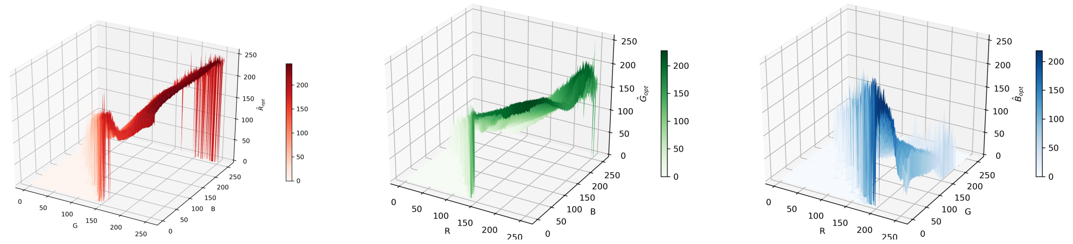


(b) Estimation of green channel



(c) Estimation of blue channel

Figure 3.10: Estimation of a missing color channel



(a) (b) (c)

Figure 3.11: Estimation of a missing color channel surface plot

3.2.2 *Image recovery*

Here we picked several images to test. Some results are shown as figure 3.12, 3.13. We cannot see much difference from recovered image considering or not considering contrast. From the MSE value, considering contrast is slightly lower than that of not considering contrast for most of times. But it is not clear why sometimes considering contrast makes even worse prediction.

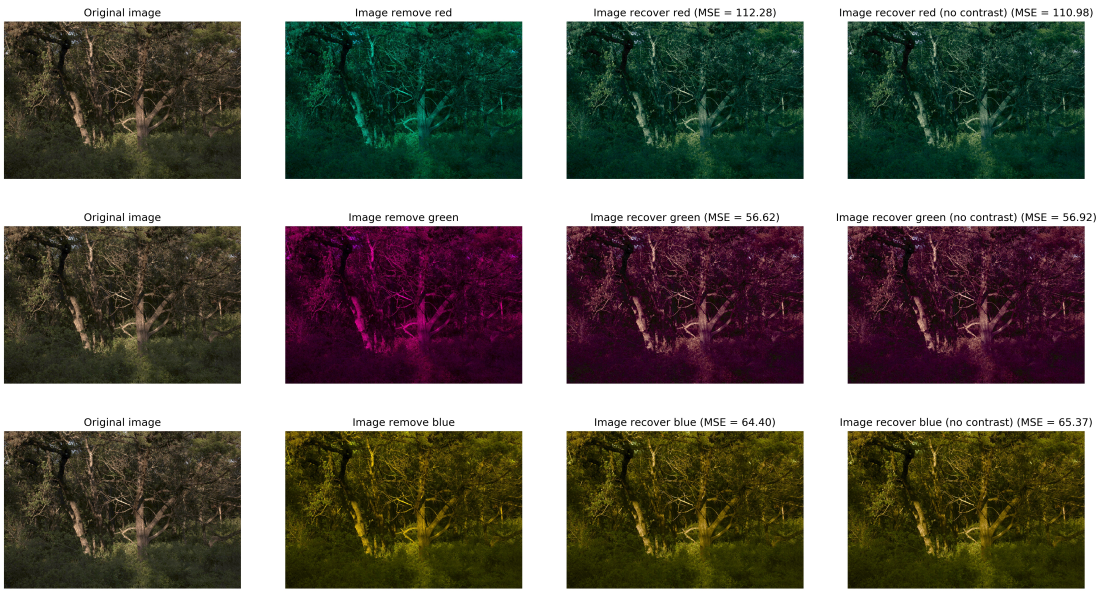


Figure 3.12: Image recovery

4 Conclusion

In this project, we extracted color information and use the information to predict the missing color channel. From the estimation plots, we found that the configurations of different mean contrast for a color are similar, which indicates the mean contrast does not have a great influence on estimation. From the recovered images, we noticed that the estimated results seem close. Therefore, the mean contrast information does not help much for color prediction. The color does not have obvious correlation with the mean contrast of the other two color channels.

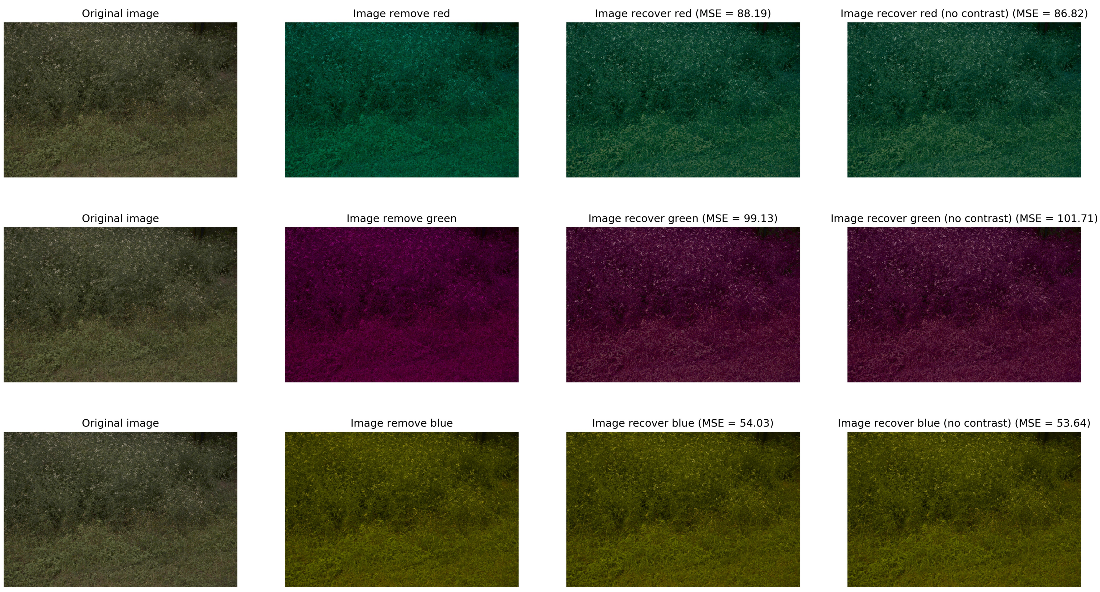


Figure 3.13: Image recovery