# CompE565, Fall 2021

# Homework 1: Basic Digital Image Processing Operations

**Prepared by**

*Jacob Bananal (820416396)*

E-mail: *jbananal2509@sdsu.edu*

Computer Engineering

San Diego State University

# Table of Contents

# List of figures

## Introduction

For this homework assignment, I was introduced to the basics of reading, displaying, and operating images in Matlab. With that being said, it allowed me to get a better understanding on what the RGB and YCbCr color spaces are and by coding it and displaying each band separately along with comparing it to the original image. Not only that, I was able to experience different kinds of options when it came down to image compression. It gave me a clear picture on how it can affect the quality along with the color of the given photo we used depending whether it has been either upsampled or subsampled.

## Procedural Section

*(1) Read and display the image using Matlab.*

*(2) Display each band (Red, Green and Blue) of the image file.*

I was asked to read an image file and display the image by using the Matlab commands “imread” to read the image file, then “imshow” to display the image on the window. With the given picture that was required to read the image, the data is read into a three dimensional array that correlates with the row, column, and color band index of that image. Therefore, with the three dimensional array, I got to extract the red, green, and blue bands separately by setting the third dimension to a specific value related to the band where the first two dimensions is set to “:” because it reads the rows and columns. In this case of RGB, I used the command:

```
red_band = origImg( :, :, 1);
```

*(3) Convert the image into YCbCr color space.*

*(4) Display each band separately (Y, Cb and Cr bands)*

For steps 3 & 4, I converted the RGB image into YCbCr color space. I converted the RGB image using the Matlab command “rgb2ycbcr.” After converting the image to YCbCr, we displayed the bands individually. With the same method that was used in step 2, I was able to use the three dimensional array to display each of the y, Cb, an Cr band.

*(5) Subsample Cb and Cr bands using 4:2:0 and display both bands*

I was asked to subsampled the Cb and Cr bands using 4:2:0 where I would reduce the pixels of the image file by half. I accomplished this by using a FOR statement, then towards the end, I use the MATLAB command:

```
Cb_420 = CbSub(1:2:end, 1:2:end);
```

```
Cr_420 = CrSub(1:2:end, 1:2:end);
```

*(6) Upsample and display the Cb and Cr bands using Linear Interpolation and Simple Row/Column Replication*

For this part, I was asked to upsampled and display the Cb and Cr bands using linear interpolation and row & column replication where I used two FOR statements. In the first for statement, it would simply be upsampled then the second for statement is finding the linear interpolation and the row & column replication.

*(7) Convert the image into RGB format*

I converted the images back into RGB format by using the Matlab command “ycbcr2rgb.” With this command, I was able to accomplish converting YCbCr to RGB format without any issues.

*(8) Display the original and reconstructed images*

For this part, we were asked to display the original image that was given to us along with comparing it to the reconstructed images that were in step 7. I displayed these images using the subplot command along with the “imshow” display command.

*(10) Measure MSE between the original and reconstructed images (obtained using linear interpolation only)*

For this part, I calculated the MSE between the original and reconstructed image of upsampling using linear interpolation of each band by using the formula that was given to us in the lecture:

$1 NM'2\ MSE = NM \sum\sum[f(j,k) - f\ (j,k)]$

With this given formula, I was able to convert it to code which you will see below:

```
mse = sum(sum(pixdiffence.^2))/(totalRows*totalColumns);
```

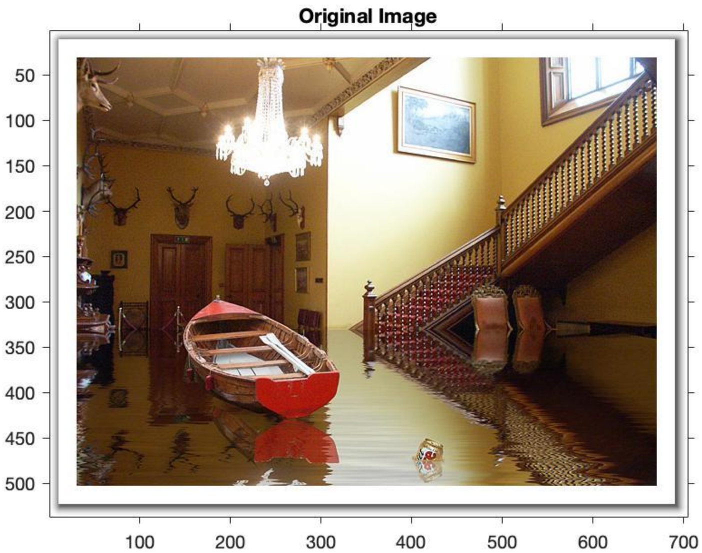
Before I used this equation, I had to figure out the pixels difference.

*(11) Compression Ratio*

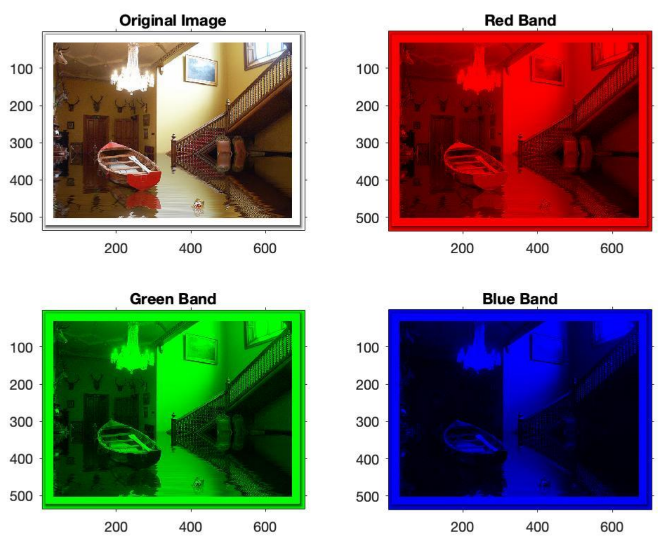
For the compression ratio, I achieved this by subsampling the Cb and Cr components/bands then dividing it with the original image size using the code:

```
ratio = origHouseSize/subsampHouseSize;
```

## Results (Pictures are in order starting from 1 to 8):

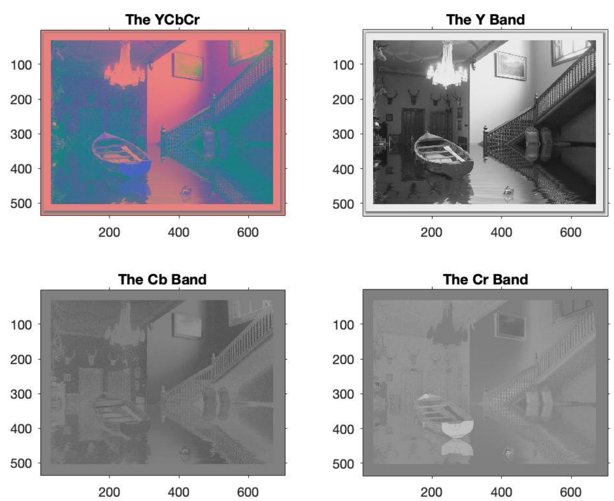



The first image above is displaying the original image.

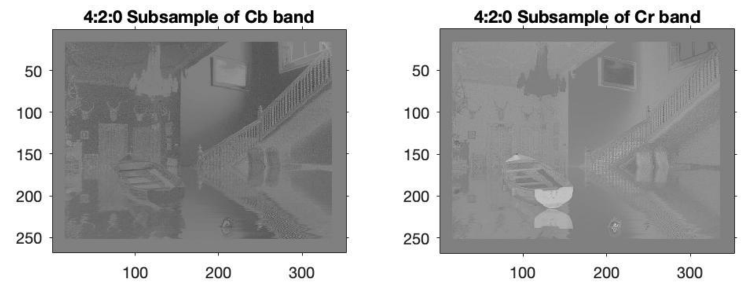



The second image above is displaying the original image along with the RGB.

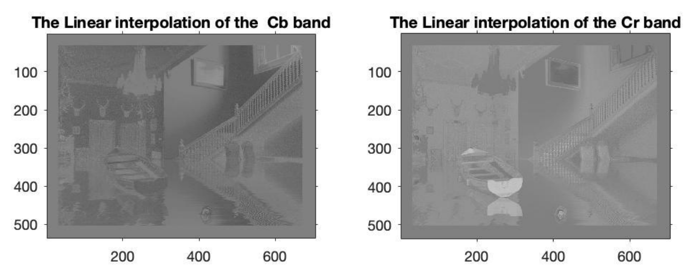

The first third above is displaying the conversion of the RGB to YCBCr and then the other way around.



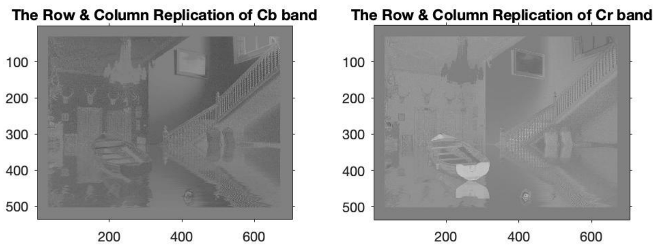

The fourth image is simply displaying the YCbCr along with the y, Cb, and Cr components.



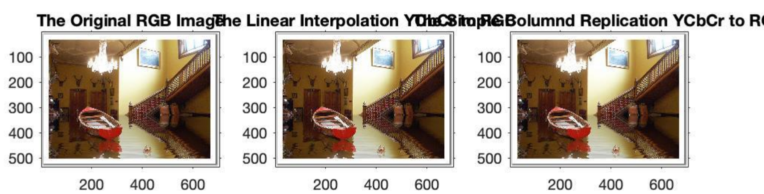

The fifth image above is the 4:2:0 subsample of the Cb and Cr band



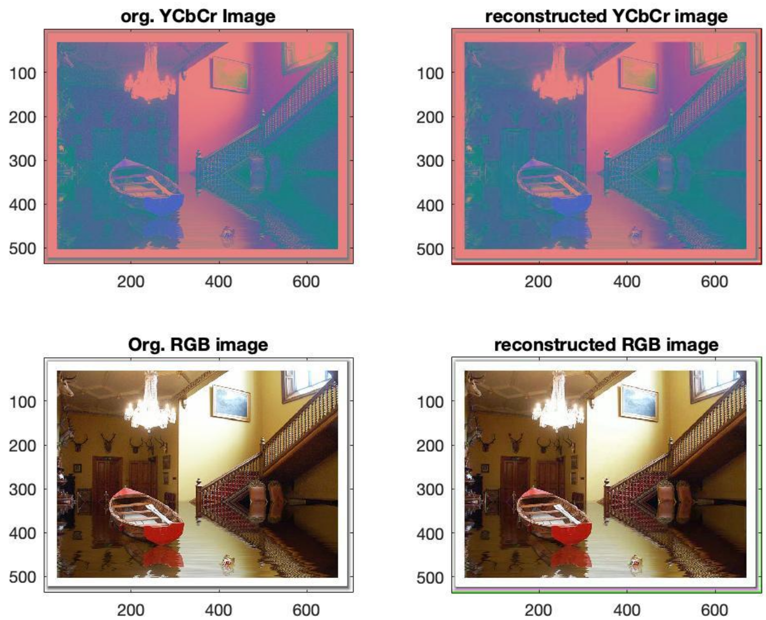

The sixth image above is the linear interpolation of both the Cb and Cr band.




The seventh image is the row & column replication of the Cb and Cr band.




Then the eighth image is the comparison of the RGB image along with the linear interpolation & the row/column replication of the YCbCr to RGB.




The ninth image is the original YCbCr image then the reconstructed images that were used in step 7.

Calculated MSE results and Compression Ratio:

The MSE of Red Band 0.166153

The MSE of Green Band 0.121202

The MSE of Blue Band 0.197822

The Compression Ratio is 2.396849

These were the values I got using the equations I learned from the lecture and transferring it over to MATLAB.

## Conclusion

As a result, I learned a lot about reading, displaying, and being able to manipulate images that I was given. I felt that I needed to work more on the image compression and get more of a better understanding in terms of operating the quality and color of the image. I wanted to see if I can change it to a different color rather than the RGB like we were told to. One thing I wish I can learn more about was trying to transfer the equations from the lecture notes and videos and coding it in MATLAB. I thought that was more of a struggle for me but the rest turned out to be understandable. Last of all, I thought understanding and being able to calculate the compression ratio took me since I was trying to see how it reads the bytes of the original image and the reconstructed image then uses to calculate the compression ratio by dividing each other.

## References

Mathworks Website Links

[1] MathWorks. (2021). “Image Processing Toolbox” (R2021b).

https://www.mathworks.com/help/images/index.html?category=index&s_tid=CRUX_topnav