COL783 - Digital Image Processing *Assignment 1 report*

Tushar Bansal - 2018EE10510

Rakshit Rao - 2018EE10833

Part 1:

There are two parts to the assignment. The first part of the assignment requires implementation of parts from the paper on **color quantization**. The methods and algorithms used for color image quantization for frame buffer display are:

1. Uniform Quantization
2. Popularity Algorithm
3. Median Cut Algorithm
4. Floyd-Steinberg Dithering Method

The task of color quantization is to select and assign a limited set of colors for representing a given color image with maximum fidelity. In the general case, the original image I contains a set of m different colors $C = \{C_1, C_2, \ldots, C_m\}$, where m could be only a few or several thousands, but at most 2^{24} for a 3×8 bit color image. The goal is to replace the original colors by a (usually much smaller) set of colors $C' = \{C'_1, C'_2, \ldots, C'_n\}$, where $n < m$. The difficulty lies in the proper choice of the reduced color palette C' such that damage to the resulting image is minimised.

Uniform Quantization

In uniform quantization each axis of the color space is treated independently. Each axis is then divided into equal sized segments. The number of these regions is dependent on the scheme used for dividing the color space. In our assignment we divide the red and green axis into 8 segments each and the blue axis into 4 resulting in 256 regions. Each one of these regions will produce a color for the color map. Once the color space has been divided each of the original colors is then mapped to the region which it falls in. The

representative colors for each region is then the average of all the colors mapped to that region.

The original and final images are shown below:

ORIGINAL IMAGE	FINAL IMAGE
	
	
	

Non-Uniform quantization

Non-Uniform quantization does not treat the individual color component separately as does scalar quantization, but each color vector $C_i = (r_i, g_i, b_i)$ or pixel in the image is treated as a single entity. Starting from the set of original color tuples $C = \{C_1, C_2, \ldots, C_m\}$, the task of non-uniform quantization is

1. to find a set of n representative color vectors $C' = \{C'_1, C'_2, \ldots, C'_n\}$ and
2. to replace each original color C_i by one of the new color vectors $C'_j \in C'$,

where n is usually predetermined ($n < m$) and the resulting deviation from the original image shall be minimum.

1. Popularity Algorithm

The popularity algorithm selects the K most frequent colors in the image as the representative set of color vector C'. Each pixel C_i is then replaced by the closest representation color vector in C'; ,i.e., the quantized color vector with the smallest distance in the 3D space.

The origianal and final images are shown below:

ORIGINAL IMAGE_1	ORIGINAL IMAGE_2
	
	

ORIGINAL IMAGE_1	ORIGINAL IMAGE_2
4 colors	4 colors
	
64 colors	64 colors
	
256 colors	256 colors

2. Median Cut Algorithm

Median cut is an algorithm to sort data of an arbitrary number of dimensions into series of sets by recursively cutting each set of data at the median point along the longest dimension. The algorithms is described below:

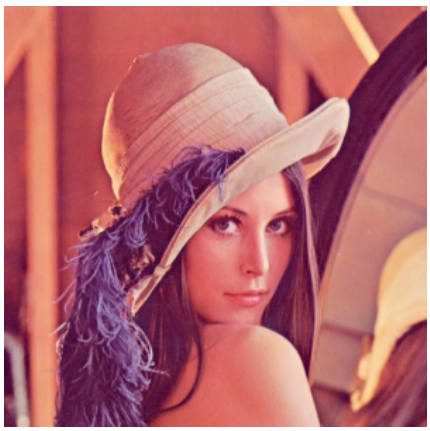
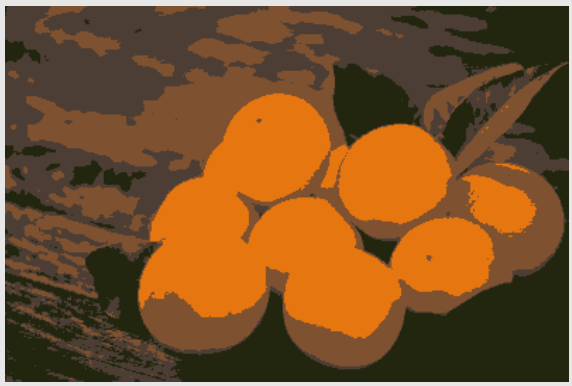
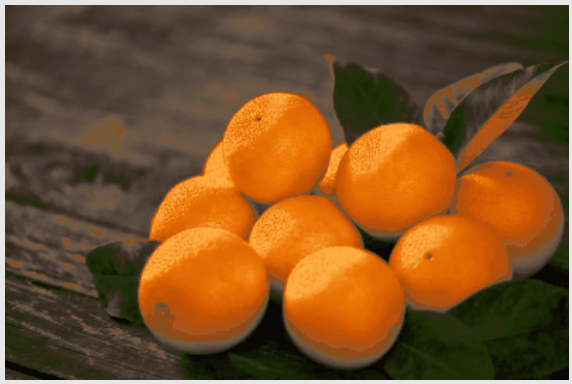
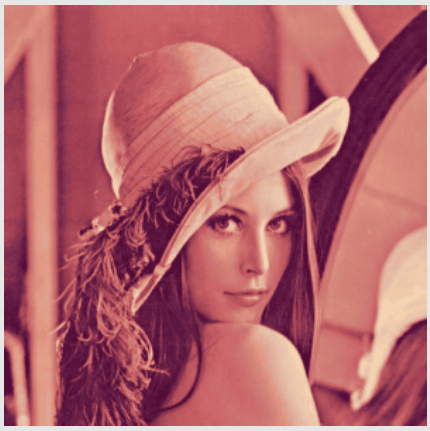
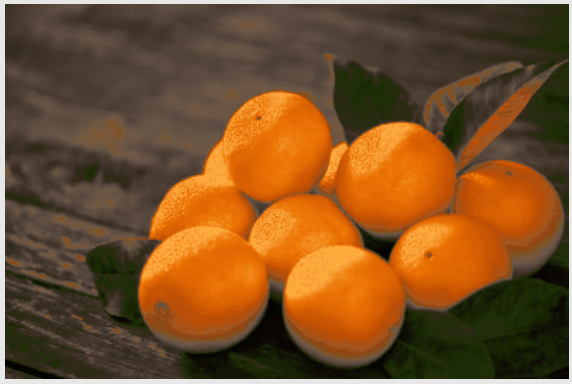
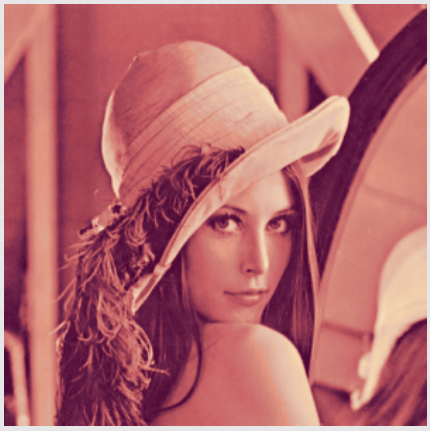
a. Move all pixels into a single large box.

b. Find the color channel (red, green, or blue) in the image with the greatest range.

c. Sort the pixels by that channel values.

d. Find the median and cut the region by that pixel.

e. Repeat the process for both boxes until you have the desired number of colors.

ORIGINAL IMAGE_1	TARGET IMAGE_1
	
	
4 colors	4 colors
	
64 colors	64 colors
	
256 colors	256 colors

3. Floyd Steinberg Dithering

The Floyd-Steinberg dithering algorithm is an example of an error-diffusion technique. In the algorithm the residual quantization error of a pixel is pushed onto its neighbouring pixels, to be dealt with later. It spreads the debt out according to the distribution (shown as a map of the neighbouring pixels):

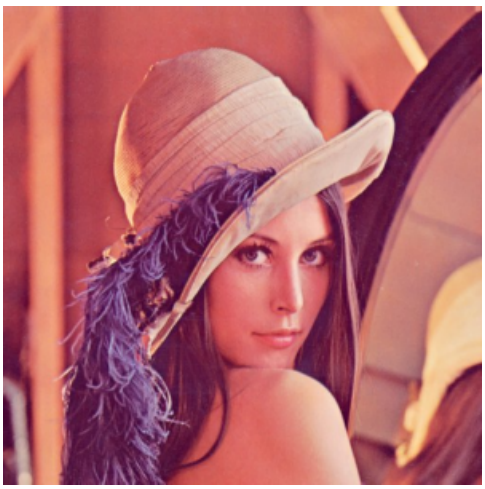
$$\begin{bmatrix} & & * & \frac{7}{16} & \cdots \\ \cdots & \frac{3}{16} & \frac{5}{16} & \frac{1}{16} & \cdots \end{bmatrix}$$

The pixel indicated with a star (*) indicates the pixel currently being scanned, and the blank pixels are the previously-scanned pixels. The algorithm scans the image from light to left, top to bottom, quantizing pixels values one by one. Each time the quantization error is transferred to the neighbouring pixels, while not affecting the pixels that already have been quantized. The pseudo algorithm taken from wikipedia is shown below:

```
for each y from top to bottom do
    for each x from left to right do
        oldpixel := pixel[x][y]
        newpixel := find_closest_palette_color(oldpixel)
        pixel[x][y] := newpixel
        quant_error := oldpixel - newpixel
        pixel[x + 1][y    ] := pixel[x + 1][y    ] +
quant_error × 7 / 16
        pixel[x - 1][y + 1] := pixel[x - 1][y + 1] +
quant_error × 3 / 16
        pixel[x    ][y + 1] := pixel[x    ][y + 1] +
quant_error × 5 / 16
        pixel[x + 1][y + 1] := pixel[x + 1][y + 1] +
quant_error × 1 / 16
```

Popularity algorithm with floyd-steinberg method

ORIGINAL IMAGE_1	TARGET IMAGE_1

ORIGINAL IMAGE_1	TARGET IMAGE_1
	
	
4 colors	4 colors
	
64 colors	64 colors

ORIGINAL IMAGE_1	TARGET IMAGE_1
	
256 colors	256 colors

Median Cut algorithm with floyd-steinberg method

ORIGINAL IMAGE_1	TARGET IMAGE_1
	
	
4 colors	4 colors

ORIGINAL IMAGE_1	TARGET IMAGE_1
	
64 colors	64 colors
	
256 colors	256 colors

Part 2:

The second part of the assignment requires implementation of **transferring of color to grey level images** using the algorithm described in the paper given in reference [2] both using global method and using swatches.

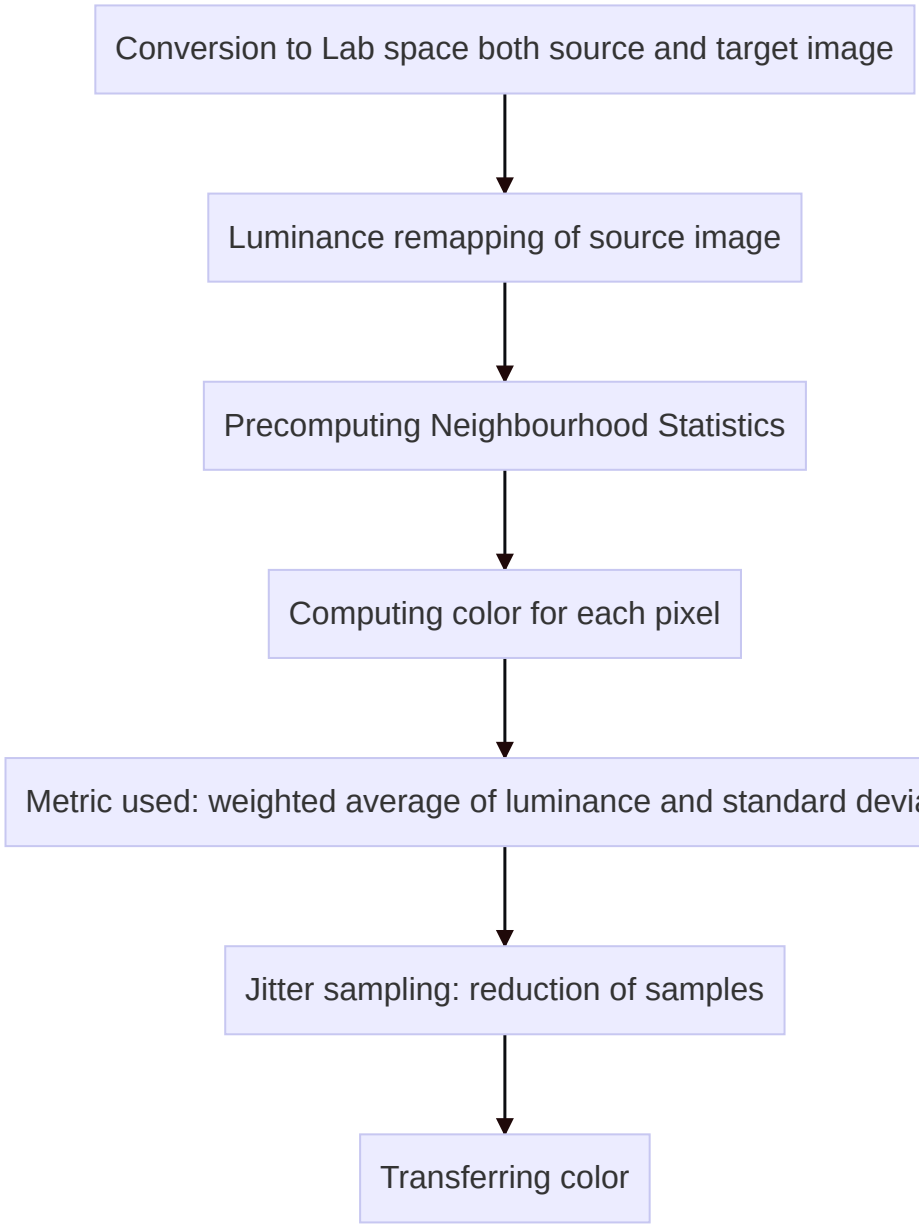
The task of “colorizing” a grayscale image involves assigning three-dimensional (RGB) pixel values to an image which varies along only one dimension (luminescence or intensity). Since different colors may have the same luminescence value but vary in hue or saturation, the problem of colouring grayscale images has no inherently “correct” solution. The task is to create a considerable good colored image without much human interference. The complete description of the process us describes below:

1. In this method we require a Source image using which the algorithm will transfer the color to the Target image (Grayscale Image).
2. In this method, the grayscale image is represented by a one dimensional distribution, hence only the luminescence channels can be matched between the

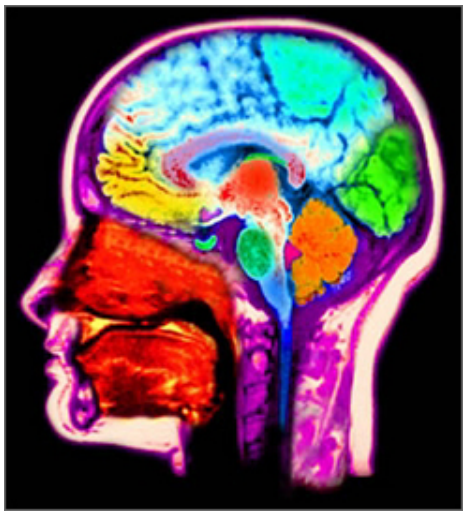
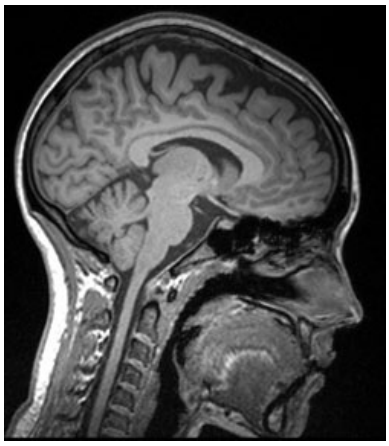
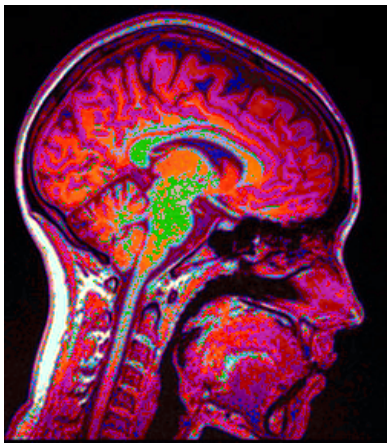
two images.

3. Because a single luminescence value could represent entirely different parts of an image, the statistics within the pixel's neighbourhood are used to guide the matching process. Once a pixel is matched, the color information is transferred, but the original luminescence value is retained.
4. In difficult cases, a few swatches can be used to aid the matching process between the source and the target image.
5. After color is transferred between the source and the target swatches, the final colors are assigned to each pixel in the grayscale image by matching each grayscale image pixel to a pixel in the target swatches using a distance metric.
6. The statistic used for matching is w_1*(lum_diff)^2 + $(1 - w_1)$ * (std_diff)^2, where w_1 is a hyper parameter in $[0, 1]$. lum_diff denotes the difference between luminescence and std_diff denotes the standard deviation.
7. Brute forcing method takes a lot of time if the images are larger, and so we use Jittered sampling approach (reducing number of pixels).
8. Thus, we have a total of 3 hyper-parameters: w_1 , number of samples in Jittered sampling (N) and pixel neighbourhood size (P).

Global Image Matching : The flow chart depicts the processes used in the following algorithm. This method is used to colorize a grayscale image without any human intervention.

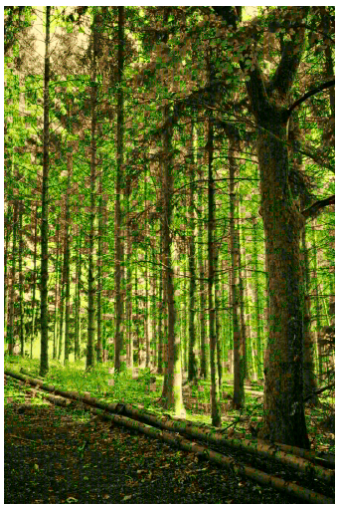


SOURCE IMAGE	TARGET IMAGE	RESULTANT IMAGE
		

SOURCE IMAGE	TARGET IMAGE	RESULTANT IMAGE
		
		

Swatches: In order to allow more user interaction in the color transfer procedure and to improve results, swatches are used between corresponding regions in the two images. The first step is to use the general procedure described above to transfer color, but now only between the corresponding swatches. This allows the user to selectively transfer colors between the source and target swatches. The second step is the texture transferring in which the L_2 distance is used to find texture matches. Note, at this stage we no longer search the color image for texture matches but only search for matches within the colorized swatches in the target image.

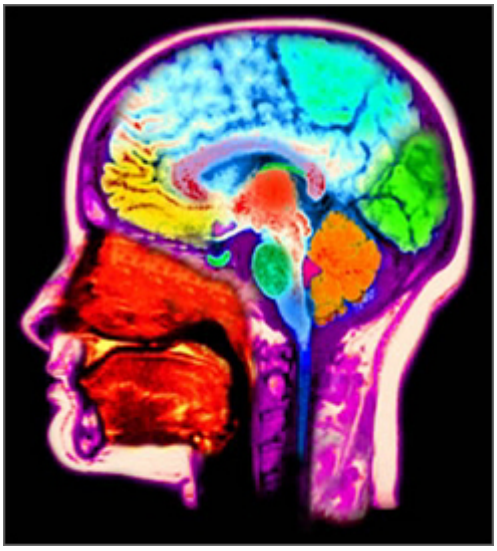
Number of swatches used : 2

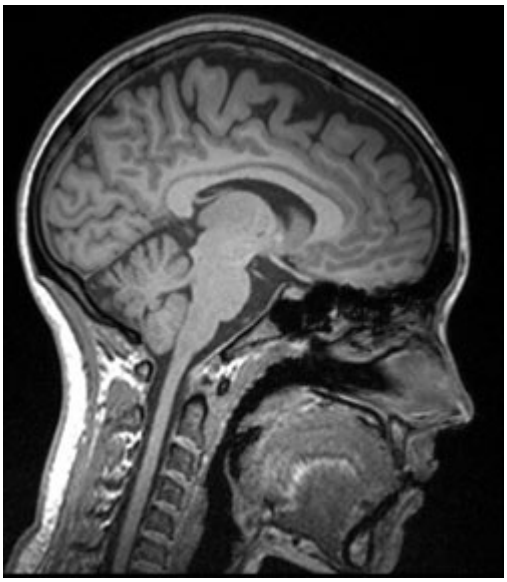
SOURCE IMAGE	TARGET IMAGE	RESULTANT IMAGE
		

Number of swatches used: 3

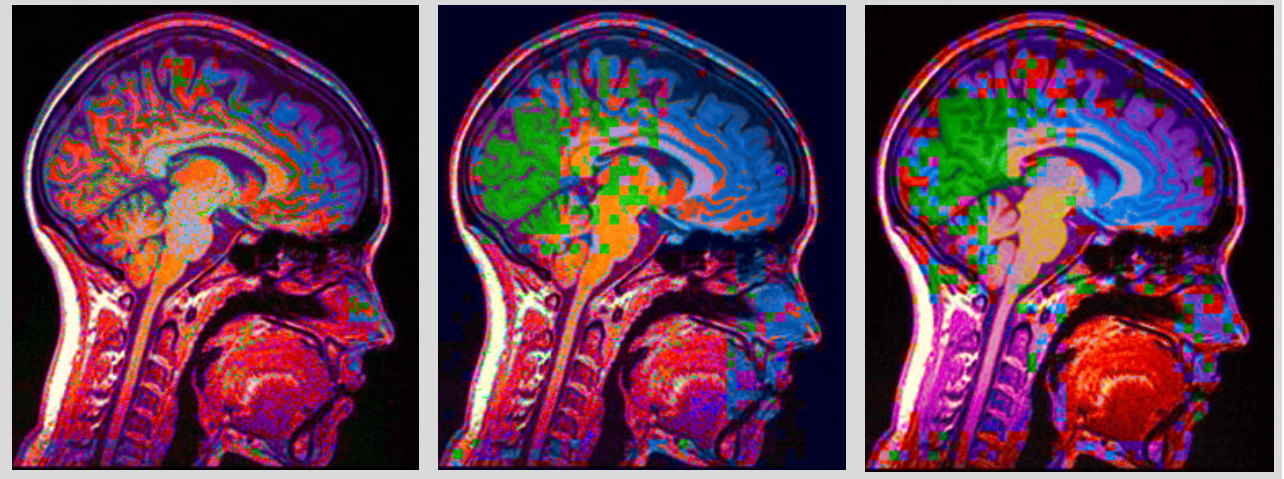
SOURCE IMAGE	TARGET IMAGE	RESULTANT IMAGE
		

Variation of number of swatches in a single image:

Source Image: 

Target Image: 

Resultant Images:



No. of swatches: 1 | Number of Swatches: 3 | Number of Swatches: 5

References Paper:

1. COLOR IMAGE QUANTIZATION FOR FRAME BUFFER DISPLAY
2. Transferring Color to Greyscale Images

References:

1. Reference for color quantization techniques: HTML pages that contain info regarding all color quantization techniques
2. Another reference for color quantization techniques: More advanced median-cut algorithms
3. Wikipedia Floyd-steinberg dithering: Nice description of floyd-steingberg dithering techniques
4. Principles of digital image processing: The theory part of the report is taken from this book with slight modification.