Color Image Processing:

The human visual system can distinguish hundreds of thousands of different color shades and intensities, but only around 100 shades of grey. Therefore, in an image, a great deal of extra information may be contained in the color, and this extra information can then be used to simplify image analysis, e.g. object identification and extraction based on color.

The modules implemented are: Color model transformations, pseudo coloring, smoothing and sharpening of color images.

Contribution:

- Guru, Rupa – Color Transformations, GUI, Report
- Sakitha, Vishal – Pseudo Coloring, Smoothing and Sharpening, Report

Color Model Transformations:

- **RGB to CMYK**
  The RGB values of a color image are in the range of 0 to 255. To convert them into CMYK, RGB values are initially normalized into 0 to 1 range. This can be achieved by division of rgb values by 255.0. Then c, m, y, k values can be calculated by taking the negative of 1 for the above normalized values. These values can be mapped to CMYK values of an image by normalizing them with the minimum value among c, m, y and k values. These values cannot represent an image. To present it in a CMYK color space, the c, m, y, k values need to be multiplied by a constant. We used 100 as a constant value to multiply to c, m, y, k values.
- **CMYK to RGB**
  Since values were multiplied with a constant to convert to cmyk, the c, m, y, k values are divided with a constant value to get them into range of 0 to 1. To calculate the values of rgb, the c, m, y values are normalized using the k value and mapped with respective r, g, b values. But the resultant values are observed to be in the range of 0 to 1. RGB values are in the range of 0 to 255. Therefore, the result need to be mapped with rgb by multiplying the c, m, y values with 255. The resultant values represent r, g, b values.
- **RGB to HSV**
  The RGB values of a color image are in the range of 0 to 255. To convert them into HSV, RGB values are initially normalized into 0 to 1 range. This can be achieved by division of rgb values by 255.0. Calculate the maximum and minimum values among these normalized rgb values. Hue can be calculated based on maximum value among rgb.
  Calculations involved:
  Value of Saturation = (Max(r,g,b) – Min(r,g,b)) / Max
  Value of Intensity = Max
  If r is the maximum value, Hue = ((G - B) / MAX-MIN) % 6
  If g is the maximum value, Hue = 2 + ((B - R) / MAX-MIN)
  If b is the maximum value, Hue = 4 + ((R - G) / MAX-MIN)

- **HSV to RGB**
  The HSV values of an image are normalized into 0 to 1 range values.
  Calculations:
  C = V * S
  X = C * (1 -|(H/60°) mod 2 -1|)
  M = V – C
  R = (R' + m) * 255, G = (G' + m) * 255, B = (B' + m) * 255

$$(R', G', B') = \begin{cases} (C, X, 0) & , 0^\circ \leq H < 60^\circ \\ (X, C, 0) & , 60^\circ \leq H < 120^\circ \\ (0, C, X) & , 120^\circ \leq H < 180^\circ \\ (0, X, C) & , 180^\circ \leq H < 240^\circ \\ (X, 0, C) & , 240^\circ \leq H < 300^\circ \\ (C, 0, X) & , 300^\circ \leq H < 360^\circ \end{cases}$$

  Output:

RGB

CMYK

HSV

Pseudo Coloring:

Given a black and white image, human eye can simply distinguish between only around 2 dozens of black and white shades or intensity shades whereas given a color image, we can distinguish among thousands of color shades. So, given a black and white image or an intensity image, if we go for pseudo color processing techniques that is to assign different colors to different ranges of intensity values, interpretation of such image is more convenient than the interpretation of an ordinary or simple intensity level image.

- **Intensity slicing:** The basic purpose of Intensity Slicing technique is to assign different colors for different intensity ranges in a black and white image.
- **Pseudo color transformation:** Gray scale image corresponds to a single plane, we convert that to 3 different planes that is R, G and B - red green and blue planes and those red, green and blue planes when combined, give you an interpretation of a color image.

Color is not assigned to different intensity ranges but the color is decided by the corresponding transformation functions. These transformation functions, we have generated fR - the red component, fG - the green component and fB - the blue component and when you combine this red component, green component and blue component; the corresponding color image is generated.

Color image that is being generated it is a pseudo colored image. Obviously, it is neither a full color image nor the color of the original image. So, the only purpose is the different intensity regions will appear as different colors in our colored image. So, this coloring is again a pseudo coloring, it is not the real coloring.

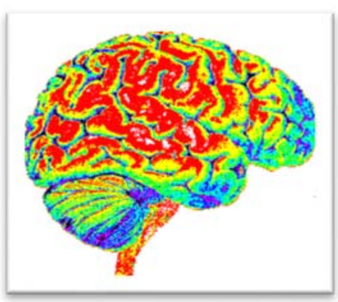
- **Applications:**
  a) If you have looked at the screen which the security people checks on the screen, the image appears in particular form where you find that the background has appeared as red, the different garments bags, they have appeared as blue. Now again, this is a pseudo coloring technique in which you can distinguish between different objects present in this particular image.

  b) The transformation functions are usually sinusoidal functions. Pseudo coloring purpose is that you define different bands of input intensity values and the different bands are given to different objects. For example, a band somewhere is for identification of say an explosive, a band somewhere is for identification of the garments bags and so on.

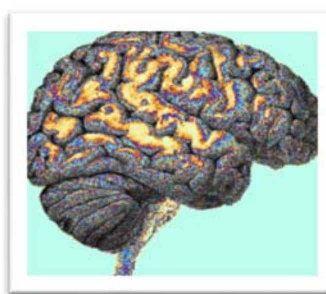
  c) Pseudo color image gives us the advantage that we can distinguish between different objects present in the image from its color appearance.

Output:

Grey scale input



Intensity Slicing



Color transformation

Smoothing and sharpening

Modify value based on the characteristics of surrounding pixels. In image processing, it is both necessary to remove image noise and improve the definition of image texture and details. Image sharpening is the preprocessing step that approaches the latter target whereas image smoothing deals with the former.

- **Smoothing:** In the real life, the images always contain noise, hence in many cases a smoothing step is needed to achieve certain goals.

  Smoothing is carried over a neighborhood size of 5 by 5 using **Gaussian low pass filter**. Our smoothing method presented good results, since it smooths well the noise without losing details and edge information. However, if the level of noise is high, it produced some artifacts in the image.

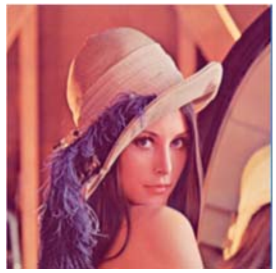
- We performed smoothing of RGB and only Intensity component of HSI:
  Smoothing only Intensity component of HSI is observed to be faster compared to smoothing an RGB image.
- **Sharpening:** The image sharpening is within the topic of color image enhancement, which encompasses all the techniques whose purpose is for the improvement of the image visual appearance and highlight details of the image.
  Example: To improve an image taken in poor light conditions.

  Gaussian High Pass filter is used for sharpening an input image to enhance its edges.

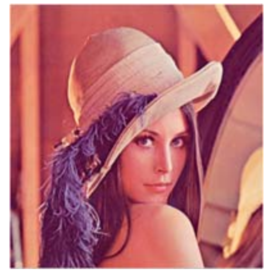
- For simultaneous application of smoothing and sharpening, the order in which we carry the operations can greatly change the output. If we sharpen before smoothing, we can increase the relevance of image noise, which will complicate the smoothing task. If, by contrast, we smooth before sharpening, we may have information loss in the smoothing process that the sharpen method could not recover. In general, the second approach usually provides better outcomes, however, it is still not an optimal solution.

Output:

Input



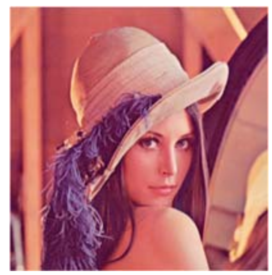
RGB Smoothing



RGB Sharpening

Input

HSI Smoothing



HSI Sharpening