EE-569

INTRODUCTION TO DIGITAL IMAGE PROCESSING

HOMEWORK #1

DIXITH REDDY GOMARI

USC ID-3098766483

gomari@usc.edu

Problem 1: Basic Image Manipulation

1.1 Motivation:

- The first part of the homework #1 consists of Image manipulation methods like mirroring, resizing, image compositing and change of color spaces etc.
- The techniques used here are basic techniques used in digital image processing.
- Techniques like Image compositing is used in film industry where scenes are shot with a blue or green screen behind and then later changed to required background.
- Other techniques like applying Sepia filter which is widely used in camera apps to filter the image and layer blending techniques used in apps like Prisma.

1.2 Approach and Procedure:

1. Image Mirroring, Resizing and Compositing

- For the image mirroring method, I have traversed from the first pixel of the first row and swapped its position to the last positon of the same row.
- The size of the image is unchanged in this process.
- The second technique involved resizing of an image from converting a 300 x 300 image to 200 x 200, 400 x 400 and 600 x 600 sizes respectively.
- This technique uses the reverse mapping procedure to find the pixel intensity of the output image from the respective input image.
- If F(p',q') is the intensity of the pixel of the output image at position p',q' and let us say we are resizing an image of size a x b to size c x d, then by reverse mapping F(p',q') will be present at (a/c)*p' and (b/d)*q' position of the input image.
- As decimal value positions do not exist in a digital image, in order to find the intensity value, we need to perform bilinear interpolation technique.
- Let us say the reverse mapped pixel exists between 4 coordinates of the input image say F(p,q), F(p,q+1), F(p+1,q) and F(p+1,q+1) then $\Delta x = \left(\left(\frac{a}{c}\right) * p'\right) - p$ and $\Delta y = \left(\left(\frac{b}{d}\right) * q'\right) - q$.
- Then F(p',q') is given by $F(p', q') = (1 - \Delta x)(1 - \Delta y)F(p, q) + \Delta x(1 - \Delta y)F(p, q + 1) + (1 - \Delta x)\Delta yF(p + 1, q) + \Delta x\Delta yF(p + 1, q + 1)$.
- The third operation we were asked to perform is Image compositing, where the blue background of the dog image should be replaced with the beach image and the dog should be placed at [1100,400] of the beach image.
- In this process we needed to identify the pixels of dog image where the color was blue. I have set a range for the RGB channel for which we get the background as color blue.
- For this range of pixels, I have replaced the dog image with the beach image.

1.2.1 Results:

Figure 1 Original Image

Figure 2 Mirrored Image

Figure 3 200 x 200 of Mirrored image

Figure 4 400 x 400 of Mirrored Image

Figure 5 600 x 600 of Mirrored Image

Figure 6 Original Beach Image

Figure 7 Dog on the Beach

1.2.2 Discussion

- The bilinear interpolation reduces sharp edges in image resizing which is undesirable for vision. Another way to do resizing without decreasing the quality of the image considerably is bi-cubic interpolation.
- Challenges faced in the image compositing was correctly classifying the range of color for the background which needs to be removed and replaced by the beach image.

2. Color Space Transformation

2a. CMY(K) color space

- Color space or color model is an abstract mathematical model which describes the range of colors. Some of the color models are RGB, CMY etc.
- One of the color model is the CMY model which is a subtractive model which produces color by reducing the light from white.
- CMY color spaces can be produced from the RGB image as follows

$$C = 1 - R$$
$$M = 1 - G$$
$$Y = 1 - B$$

- The procedure followed here is normalizing the RGB components to a range between 0 and 1 and for respective gray scale images use the above equations.
- After finding the respective values renormalize it to a range of 0 and 255 to obtain the gray scale images.

1.2.3 Results

Figure 8 Original Parrot Image

Figure 9 C grayscale image

Figure 10 M grayscale image

Figure 11 Y grayscale image

Figure 12 Original Building Image

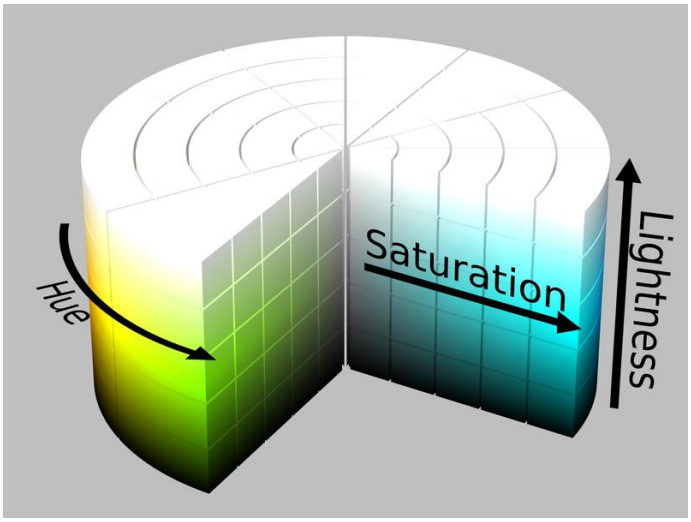
Figure 13 C grayscale Image

Figure 14 M grayscale image

Figure 15 Y grayscale image

2b. HSL color Space

- HSL stands for Hue, Saturation and Lightness and this color model is a cylindrical color model.
- In each cylinder, the angle around the central vertical axis corresponds to "hue", the distance from the axis corresponds to "saturation", and the distance along the axis corresponds to "lightness".



- Given an RGB image, it can be convered to its corresponding HSL grayscale channels using the following equations.

$$M = \max(R, G, B)$$
$$m = \min(R, G, B)$$
$$C = M - m$$

$$H=\begin{cases} 0 & C=0 \\ 60\left(\frac{G-B}{C}\bmod 6\right) & M=R \\ 60\left(\frac{B-R}{C}+2\right) & M=G \\ 60\left(\frac{R-G}{C}+4\right) & M=B \end{cases}$$

$$L=\frac{M+m}{2}$$

$$S=\begin{cases} 0,\ L=0 \\ \frac{C}{2L},0<L<0.5 \\ \frac{C}{2-2L},otherwise \end{cases}$$

1.2.4 Results

Figure 16 Original Cat Image

Figure 17 Hue grayscale Image

Figure 18 Saturation grayscale image

Figure 19 Luminance grayscale image

Figure 20 Original Dolphin Image

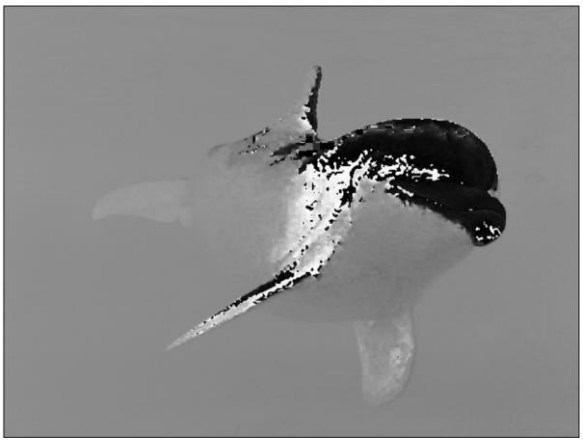


Figure 21 Hue grayscale Image

Figure 22 Saturation grayscale image

Figure 23 Luminance grayscale image

1.2.5 Discussion

- Hue is an angular parameter which ranges from 0 to 360 degrees which represents color.
- Saturation is the intensity or purity of a hue and its value ranges from 0 to 100%
- Lightness represents the relative degree of black or white mixed with a given hue and its value ranges from 0 to 100%.

2c) Sepia Filtering

- Sepia filtering is used to give warmth and special effect to images.
- Sepia filtering is done by converting an RGB image to grayscale image and then applying a special filter which gives the warmth to the image.
- RGB image is converted to a gray scale image by using the following equation.

$$I = 0.21\mathrm{R} + 0.72\mathrm{G} + 0.07\mathrm{B}$$

- Then Sepia filter is applied to the gray scale image using a special weighted matrix.

$$\begin{bmatrix} S_F(R) \\ S_F(G) \\ S_F(B) \end{bmatrix} = \begin{bmatrix} 0.393 & 0.769 & 0.189 \\ 0.349 & 0.686 & 0.168 \\ 0.272 & 0.534 & 0.131 \end{bmatrix} \begin{bmatrix} I \\ I \\ I \end{bmatrix}$$

1.2.6 Results

Figure 24 Beach Original Image

Figure 25 Grayscale Image (Intermediate Result)

Figure 26 Sepia Filter applied on Beach Image

1.2.7 Discussion

- The challenge occurred while applying the sepia filter was that few pixel intensities were going above 255 which resulted in inaccurate values at those pixel positions.
- I have resolved this issue by clipping those pixel intensities down to 255 which gave the perfect output.

3) Layer Blending Mode Implementation (Bonus)

- Layer Blending techniques is a very popular technique to produce special effects for an image and used in popular apps like prisma.
- The technique I have used here is the screen layer blending technique.
- With Screen blend mode the values of the pixels in the two layers are inverted, multiplied, and then inverted again. The result is a brighter picture.

- Let f(a,b) be the pixel intensity value of the resulting image. It can be calculated by

$$f(a,b) = 1 - (1-a)(1-b)$$

Where a is the top layer intensity and b is the bottom layer intensity.

1.2.8 Results

Figure 27 Top Layer

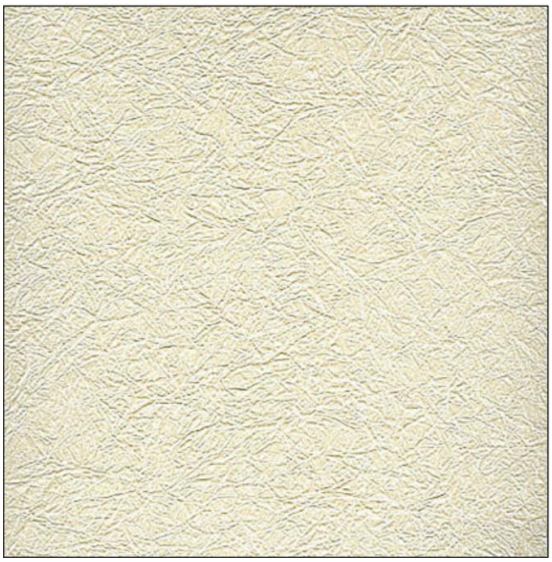


Figure 28 Bottom Layer

Figure 29 Screen Layer Blending Method

Problem 2: Histogram Equalization.

2.1 Motivation:

- The quality of the image can be improved by manipulating the histogram of the images. This process is called as histogram equalization.
- Through adjustment of the histogram, the pixels of the image can be better distributed on the histogram.
- The quality of the image can be improved if the spread of pixels is done from small range to larger range.
- The method is useful in images with backgrounds and foregrounds that are both bright or both dark.
- Histogram equalization can be used in photography which improves the quality of the photographs significantly, but it is truly helpful for the photographs which have false colors.

2.2 Procedure and approach

2.2.1 Histogram Enhancement for Grayscale Images

2.2.1.a Method A) The linear transfer-function-based histogram enhancement method

- Initially calculate the histogram of the grayscale intensities of the given image.
- Find the minimum gray-value F_{min} and the maximum gray-value of F_{max} in the histogram of the given grayscale image. The first non-zero gray -value in the histogram is set to F_{min}, and the last non-zero gray-value in the histogram is set to F_{max}.
- Let us say we want to spread the histogram over a new range (G_{min},G_{max}), then the transfer function is given by:

$$G = H(F) = G_{min} + \left(\frac{G_{max} - G_{min}}{F_{max} - F_{min}} \right) . (F - F_{min})$$

Gmax = Max value after processing (In this case it is 255)
Gmin = Min value after processing (In this case it is 0)
Fmax = Max value before processing
Fmin = Min value before processing

2.2.1.b Method B) The cumulative-probability-based histogram equalization method

- In the first step find the number of pixels for each intensity for a given image.
- Calculate the probability of each pixel intensity by dividing the number of pixels of each intensity by total number of pixels for the given image.
- Now find the cumulative sum of the probabilities found in the previous step.
- Renormalize these values for a desired range between 0 and 255 for the Histogram equalized output image.

2.2.2 Results:

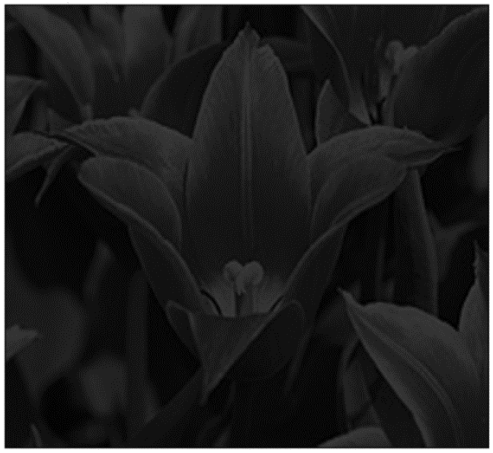


Figure 1 Original tulip dark image

Figure 2 Histogram of tulip_dark

Figure 3 Histogram equalization by method A

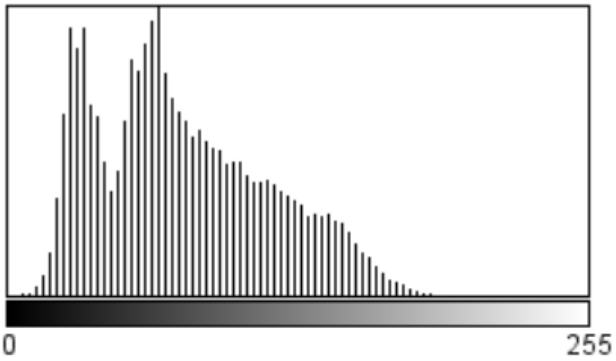


Figure 4 Histogram after method A

Figure 5 Histogram equalization by method B

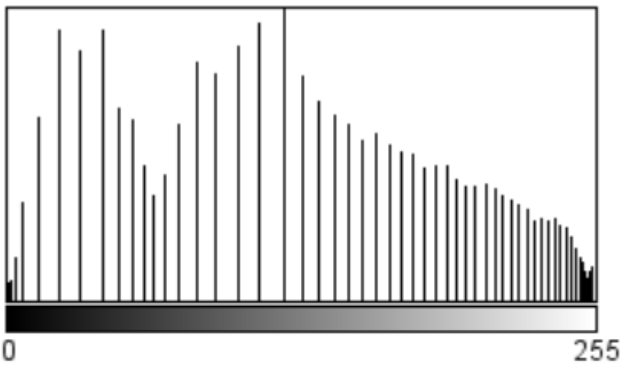


Figure 6 Histogram after method B

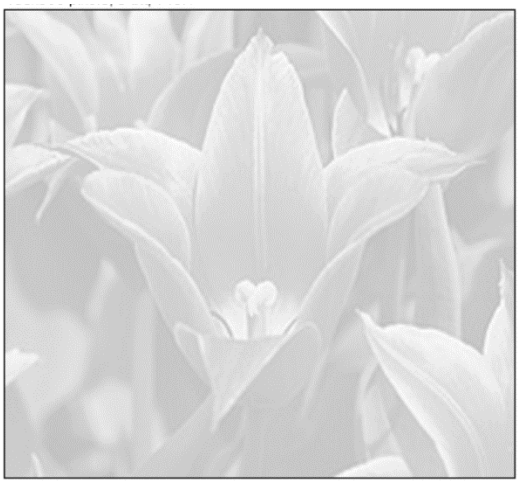


Figure 7 Original bright tulip image

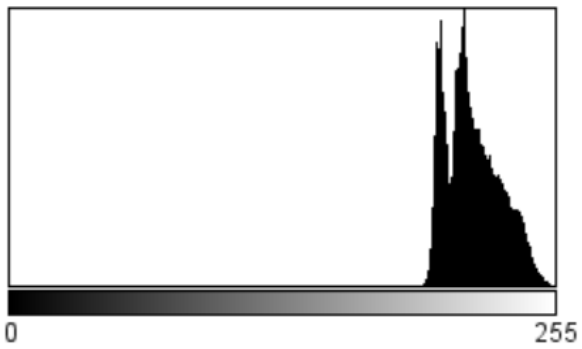


Figure 8 Histogram of bright tulip image

Figure 9 Histogram equalization by method A

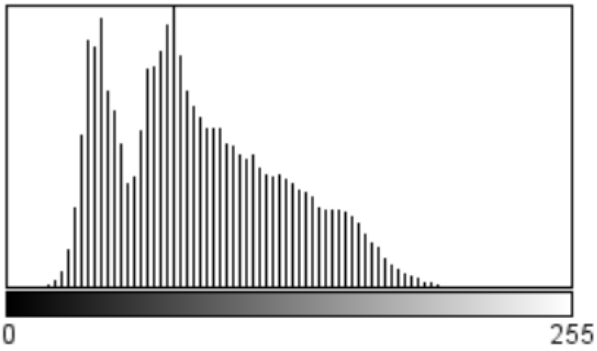


Figure 10 Histogram after method A

Figure 11 Histogram equalization by method B

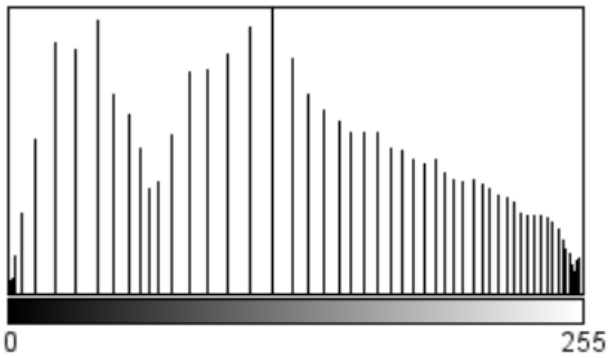


Figure 12 Histogram after method B

Figure 13 Original tulip mix image

Figure 14 Histogram of tulip mix

Figure 15 Histogram equalization by method A

Figure 16 Histogram after method A

Figure 17 Histogram equalization by method B

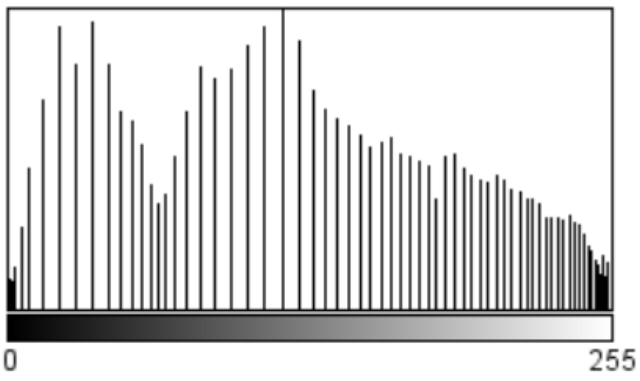


Figure 18 Histogram after method B

2.2.3 Discussion

- We can see that histogram equalization by method A for tulip mix is not effective as it is already in the range of 0 and 255.
- As we can see that histogram of the tulip mix is spread over 2 places, which we can use as an advantage to perform method A on those ranges separately.
- For the first region find the F_{min} and F_{max} and specify its G_{min} as 0 and G_{max} as 150, and then traverse in the reverse direction for finding F_{min} and F_{max} of the second region and specify its G_{min} as 128 and G_{max} as 255.
- Now repeat the method A for these 2 regions separately by keeping a threshold pixel of let's say 127 and then applying it for 2 regions separately.
- The resultant image and its histogram are as shown

Figure 19 Image when Method A performed separately

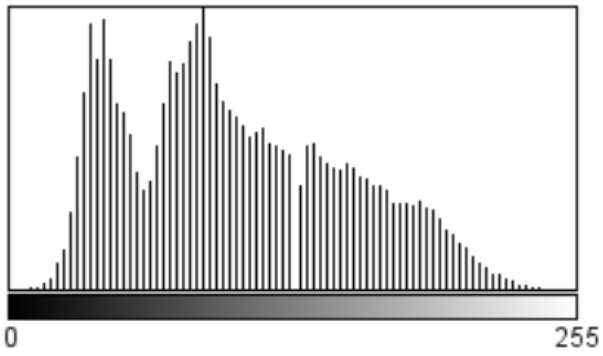


Figure 20 Histogram of the above image

2.2.4 Histogram Enhancement for Color Images

2.2.4.a Method A) The linear transfer-function-based histogram enhancement method

- This method is same as method A used for gray images.
- Convert an RGB image into 3 separate grayscale channels i.e., the R Channel G channel and B channel.
- Then repeat the method A process for all 3 channels separately and merge those values into the final RGB image.
- The results are as shown:

2.2.4.b Results

Figure 21 Original Bedroom image

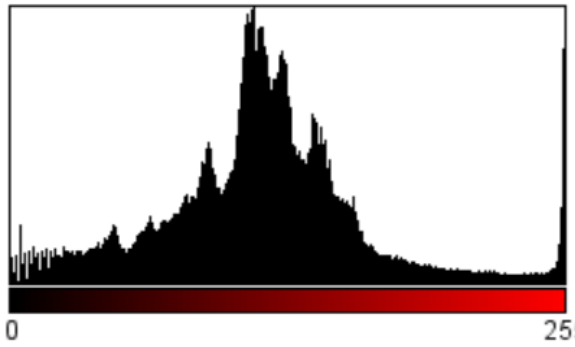


Figure 22 Histogram of R channel

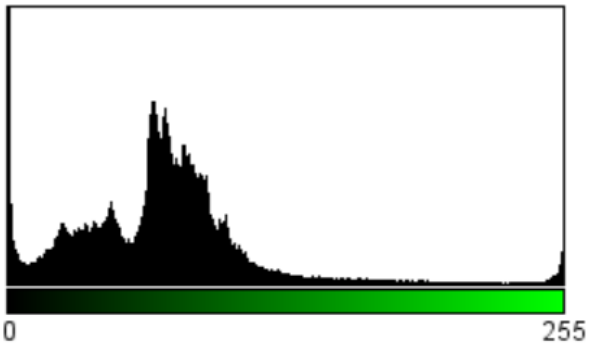


Figure 23 Histogram of G channel

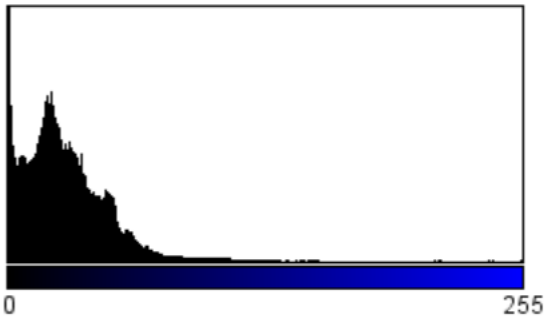


Figure 24 Histogram of B channel

Figure 25 Bedroom image after method A

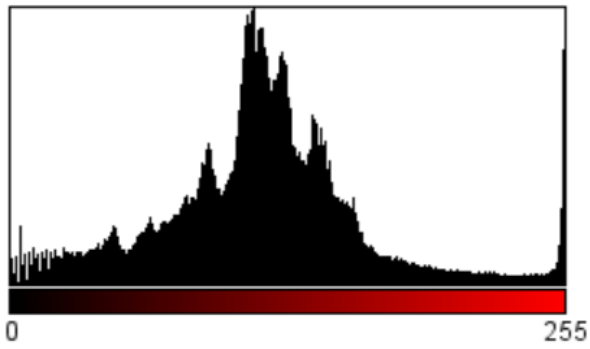


Figure 26 Histogram of R channel

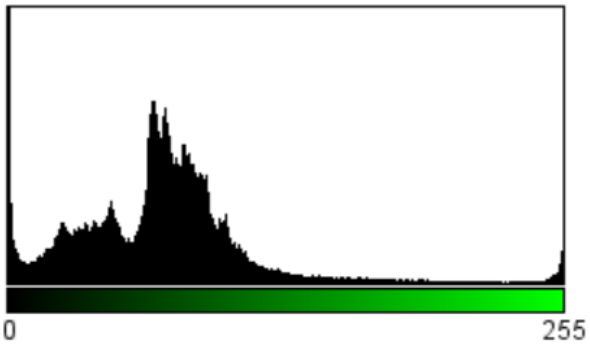


Figure 27 Histogram of G channel

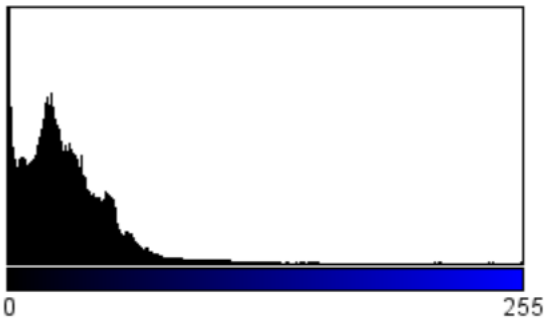


Figure 28 Histogram of B channel

2.2.4.c Discussion

- As we can see the method A is not effective in contrast enhancement of the image as the range of pixels in these channels range from 0 to 255.
- So expanding it in the same range would not help in the enhancement of the image.
- Method B is a good alternative to improve the enhancement of color images.

2.2.4 d) Method B) The cumulative-probability-based histogram equalization method

- The cumulative probability method for color images is similar to the one with gray scale images.
- Initially the color images are to be converted to their respective gray scale channels
- Then repeat method B i.e., find the probability of occurrence of each pixel and then find the cumulative sum.
- Then multiply with the range which we would like to equalize our histogram.
- Repeat this process with the other two channels and then store these values in the output image.

2.2.5 Results

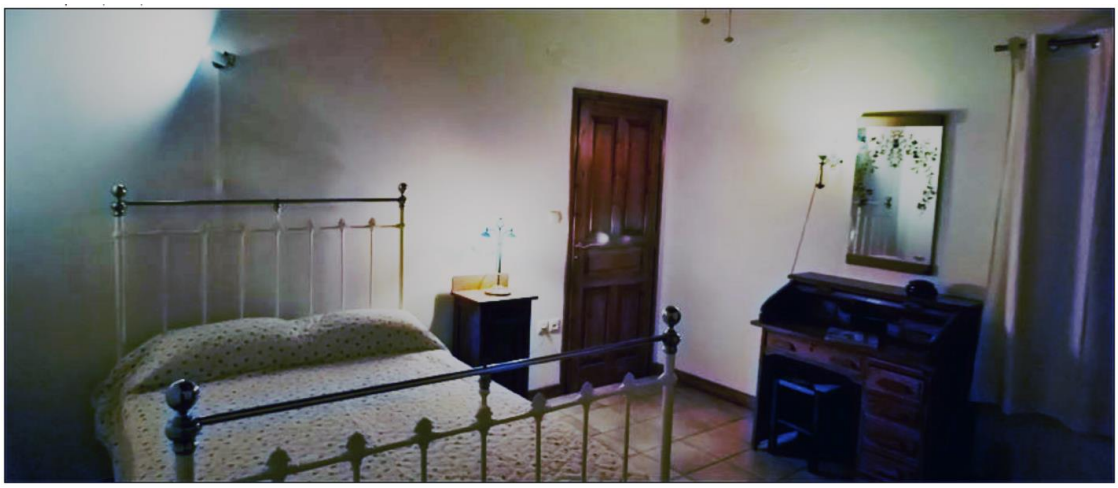


Figure 25 RGB equalization

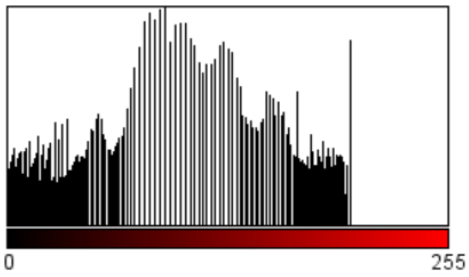


Figure 26 Histogram of R channel

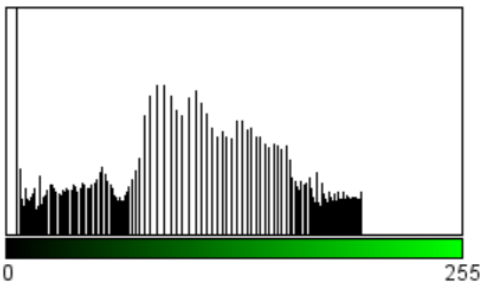


Figure 27 Histogram of G channel

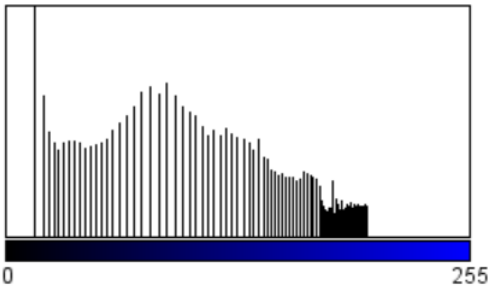


Figure 28 Histogram of B channel

2.2.6 Discussion

- When we compare the histograms of the original image and the RGB equalized image we can see that color is not preserved.
- Our objective of equalization is to increase the contrast of the image but not the color values.
- It means that we have to preserve the Hue and Saturation of the image while increasing its luminance.
- In order to do that we have to convert RGB image to HSL color space and then preserve Hue and Saturation.
- Consider Luminance of the image and find the cumulative sum of the luminance and multiply with the desired range of value.
- This preserves the color of the image and increases the contrast of the image. The results are shown below.

2.2.7 Results

Figure 29 HSL enhanced image

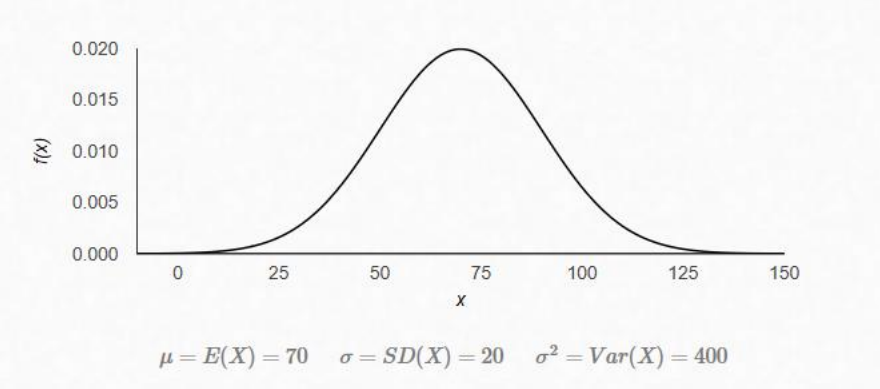
2.3 Histogram Transform

- Histogram transform also called as histogram matching or histogram specification is the transformation of the image so that its histogram matches with the specified histogram.
- First, find the number of pixels for each intensity for its three respective RGB channels.
- Then find the probability of occurrence of these pixels for their channels respectively.
- Find the cumulative sum of each channel separately.
- Produce the specified histogram that the given image needs to be transformed to.
- Let C1(i,j) be the cumulative sum array of the image, and C2(i,j) is the cumulative pdf array of the specified histogram, in this case it's a Gaussian function given by

$$p\left(\frac{x}{\mu},\sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- Then the pixels of the image are mapped from C2(i,j) when C1(i,j)>=C2(i,j).
- The pixels will be replaced by that of the mapped Gaussian values. This in turn increases the contrast of the images.

Given a Gaussian pdf with mean $\mu = 70$ and standard deviation $\sigma = 20$



The histogram of the transformed image should match the histogram of the above Gaussian.

2.3.1 Results

Figure 30 Forest 1 original image

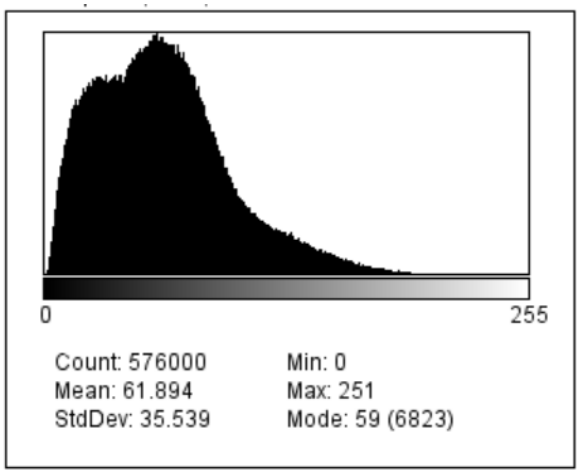


Figure 31 Histogram of forest 1

Figure 32 Transformed image of forest1

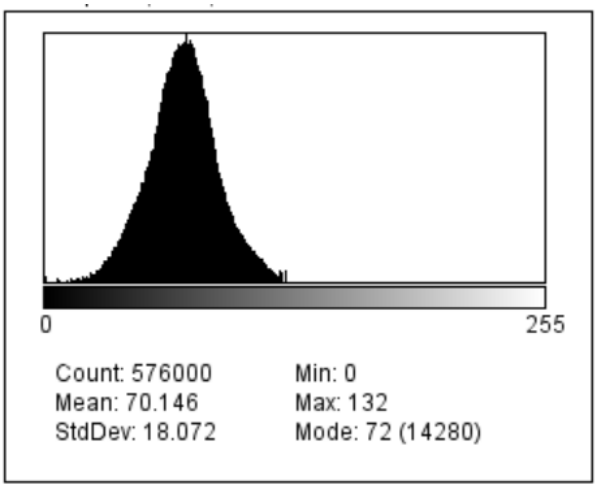


Figure 33 Histogram of the transformed image

Figure 34 Forest 2 original image

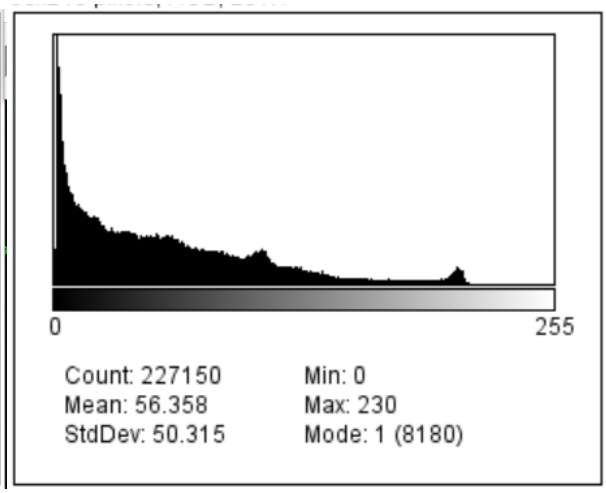


Figure 35 Histogram of forest 2

Figure 36 Transformed image of forest2

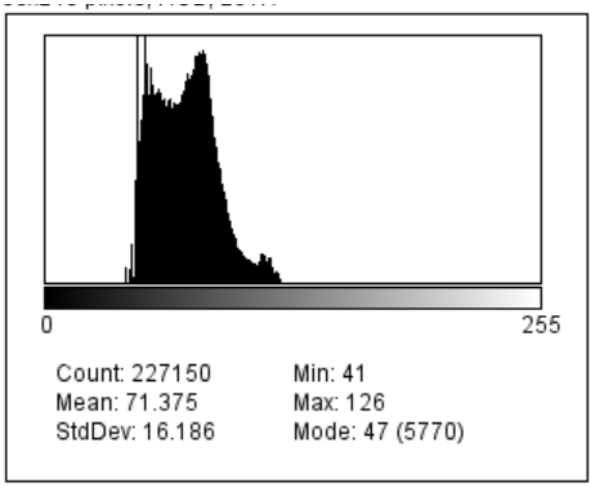


Figure 37 Histogram of the transformed image

2.3.3 Discussion

- I have implemented the histogram matching for the images forest 1 and forest 2, where its histograms are shown.
- As you can see that the mean of the transformed image matches with the histogram of the Gaussian with mean 70.
- This technique can be used to normalize the images obtained at same local illumination.

Problem 3: Noise Removal

3.1 Motivation

- Noise is an unavoidable characteristic in any form of signal processing.
- The noise in the image is caused due to various factors like the equipment that is used to take the photographs, environment etc.
- Noise can be classified into uniform noise and impulsive noise which includes salt and pepper noise.
- To denoise an image there are several filters that can be used to remove different kinds of noise.
- The quality of a denoised image can be found by the Peak Signal to Noise ratio(PSNR), that is given by

$$\text{PSNR (dB)} = 10\log_{10}\left(\frac{\text{Max}^2}{\text{MSE}}\right)$$

$$\text{where MSE} = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(Y(i,j)-X(i,j)\right)^2$$

X : Original Noise-free Image of size $N \times M$

Y : Filterd Image of size $N \times M$

Max: Maximum possible pixel intensity = 255

3.2 Procedure and Approach

3.2.1 Mix Noise in Color Image

- Take the noisy image and identify the type of noises in each channel of the image.
- If an image contains different types of noises in different channels it is called a mixed noise.
- To get a proper denoised image I applied various filters in various combinations.

The filters that I applied are:

Mean Filter:

Mean filter is best used for reducing impulsive noise and it can be implemented as follows:

- Let us assume a window size of 3X3, and the coordinates of the window ranges from (i-1,j-1) to (i+1,j+1) with center pixel placed at (i,j).
- The center pixel is modified by element by element multiplication of the window which is 1/9{{1,1,1},{1,1,1},{1,1,1}} for 3X3 window and the image where the pixel needs to be modified.

Median Filter:

Median filter is used to reduce Salt and pepper noise and is implemented as follows:

- Instead of assuming a weighted window, we directly manipulate the image to obtain a denoised image.

- Assume a window 3X3 around the pixel we want to modify, then the center pixel is replaced by the median value of the other pixels around it.

Gaussian Filter

Gaussian Filter is used to remove a special kind of noise called the Gaussian noise.

- It is similar to that of the mean filter except that its weight in the mask differs.
- The weight matrix for a Gaussian filter of 3X3 window is 1/16{{1,2,1},{2,4,2},{1,2,1}}.
- Same thing should be repeated as we did with the mean filter.

3.2.2 Results

Figure 1 Pepper Image

Figure 2 Noisy Pepper Image

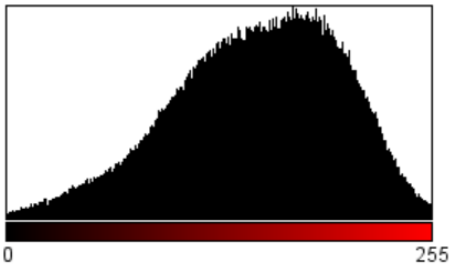


Figure 3 Histogram of R

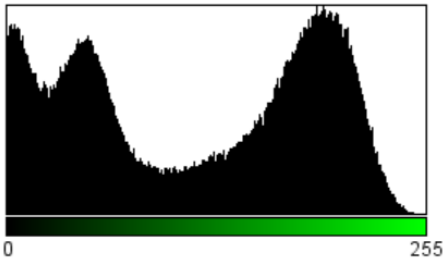


Figure 4 Histogram of G

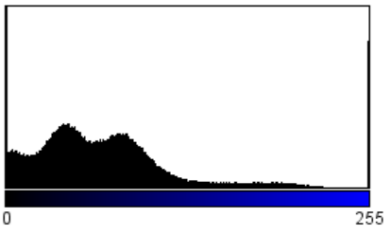


Figure 5 Histogram of B

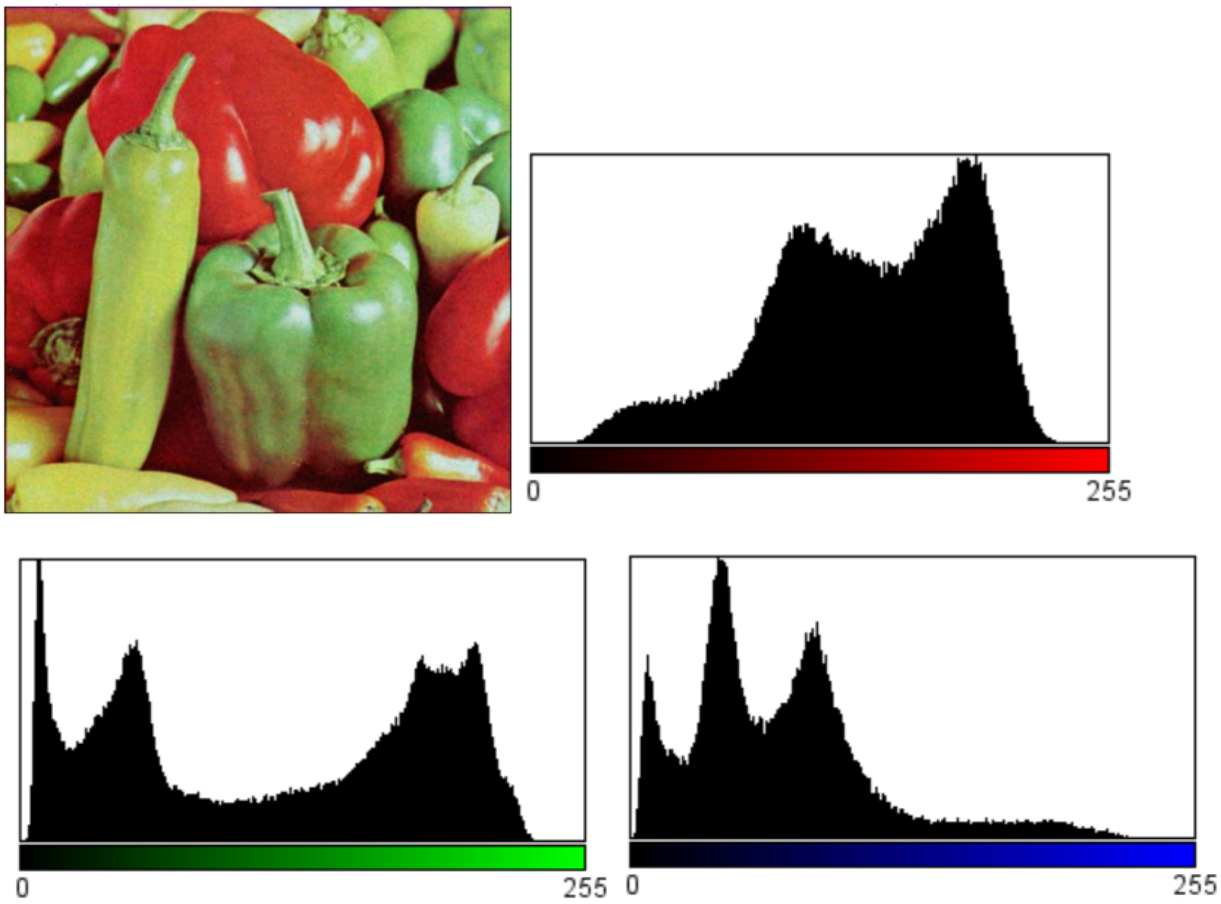
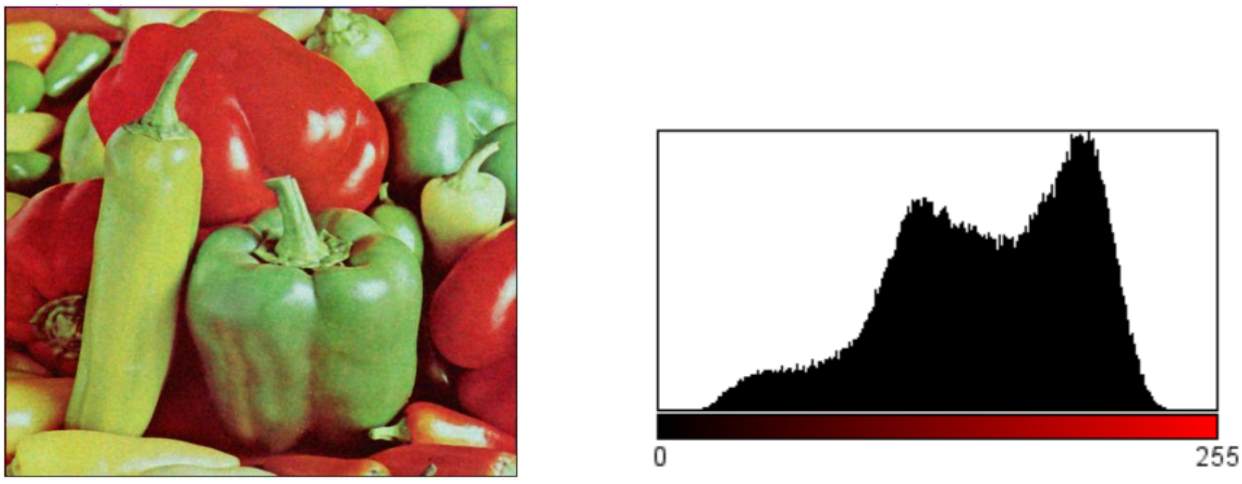


Figure 6 Denoised Image with its RGB histograms

The above denoised image is produced by mean filter on green channel, Gaussian filter on red channel and median filter on blue channel.

It produces PSNR of 27.17 for red channel,28.47 for green channel and 29.75 for blue channel.



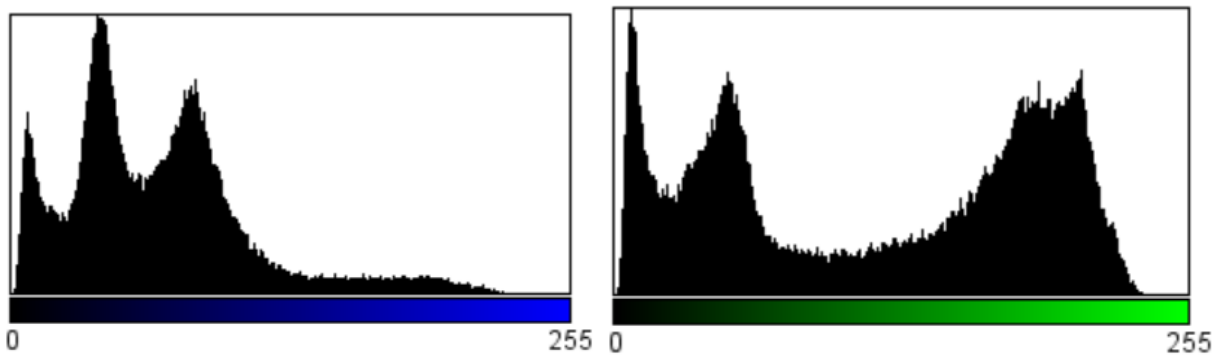


Figure 7 Denoised Image and histograms (2)

The above denoised image is produced by median filter on green channel, Gaussian filter on red channel and median filter on blue channel.

It produces PSNR of 27.17 for red channel,30.81 for green channel and 29.75 for blue channel.

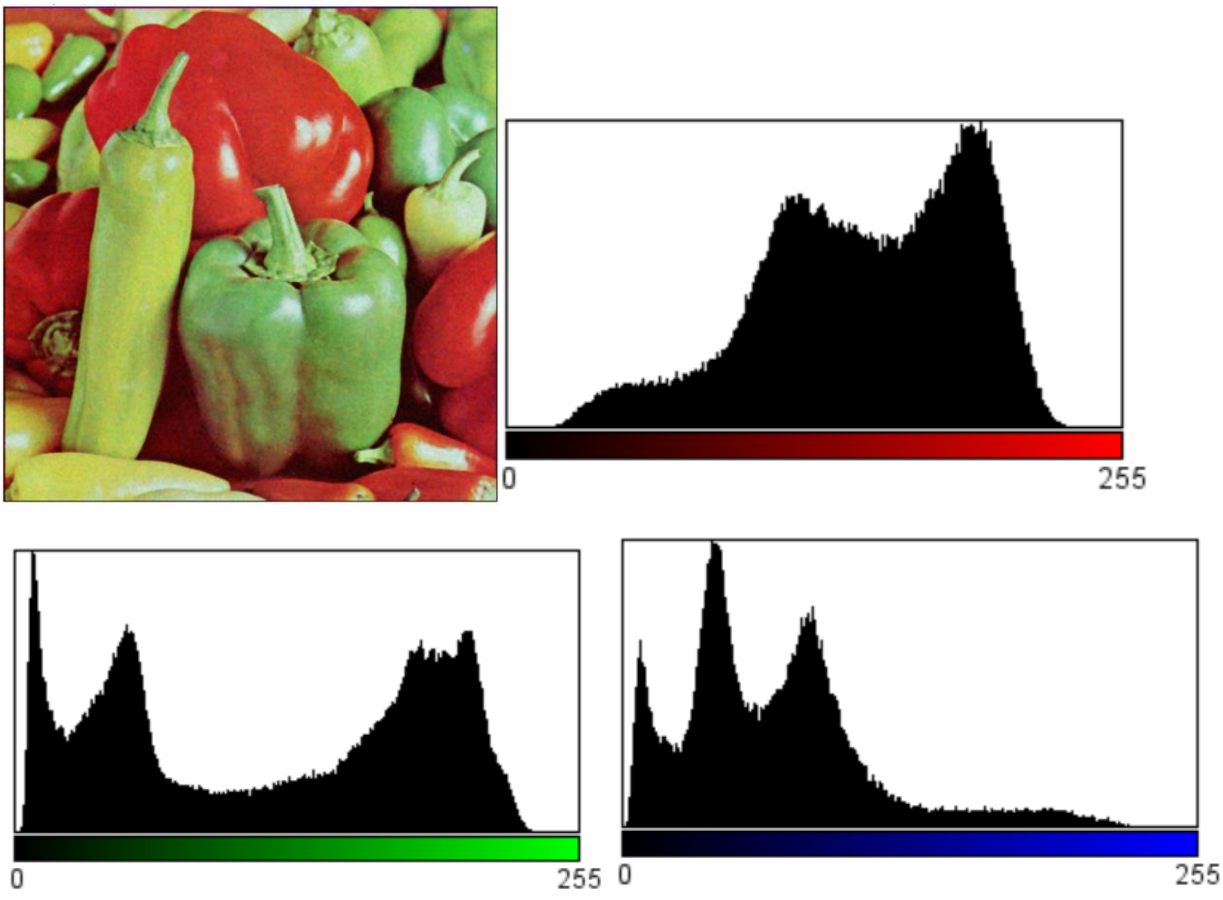


Figure 8 Denoised Image and histograms (3)

The above denoised image is produced by Gaussian filter on green channel, Gaussian filter on red channel and median filter on blue channel.

It produces PSNR of 27.17 for red channel,29.85 for green channel and 29.75 for blue channel.

3.2.3 Discussion

- We can see in the histograms of the original image that red channel is dominated by the Gaussian noise and blue channel has peaks at 0 and 255 which indicates it has impulse noise (Salt and pepper noise)
- Performing same kind of denoising process on the whole image will not produce a good output if the image has mixed noise.
- As red channel is dominated by Gaussian noise Gaussian filter is effective in removing the noise in the red channel.
- Blue channel is dominated by salt and pepper noise so, mean or median filter gives good results.
- As we can see that for different combinations of these filter cascading we get effective PSNR values.
- Window sizes also have an effect on the PSNR values of the denoised images.
- For mean and median filter smaller window sizes are preferred as they smoothen the filter when large window sizes are taken which is not acceptable.
- Whereas for Gaussian filter larger window sizes are preferred as they assign weights according the distance from the center pixel.
- The pixels far from the center pixel are given less weight and the nearer ones are given more weight.
- Cascading can be done in any order as we are operating on the 3 channels separately, it would not a difference if the order of cascading is changed.
- To improve the performance of the current algorithm identifying the type of noise correctly and assigning appropriate window sizes for different type of noises is important.
- Maintaining same window size may not be effective for all kinds of noises.
- Various other filters can be used with different weights which reduce the smoothening of the edges will also improve its performance.

3.2.4 Non Local Means Filter

- Non local means filter as the name suggests works on the pixels which are non-local to the center pixel we want to modify.
- Non local means compares each pixel value in the image to the pixel we want to modify and then assigns weights according to the similarity of the pixel values to the center image.
- Weights are given according to the similarity with the center pixel.
- Let N(v,i) be the value of the center pixel we are comparing we want to modify and N(v,j) be all the pixels in the neighborhood.
- Then the weights assigned is given by

$$w(i,j) = \exp\text{^}(-\frac{\sum(N(v,i)-N(v,j))^2}{h^2})/z(i)$$

- Where h is the standard deviation and varies from 0.85 to 100 and z(i) is the sum of all exponential terrms of the euclidean distances of the neighbourhood.
- Then the center pixel is modified according to this formula Image=W(I,j)*v(I,j) where W is the weight assigned to the center pixel of the patch that is v(I,j)

NLM can be visualised by this figure

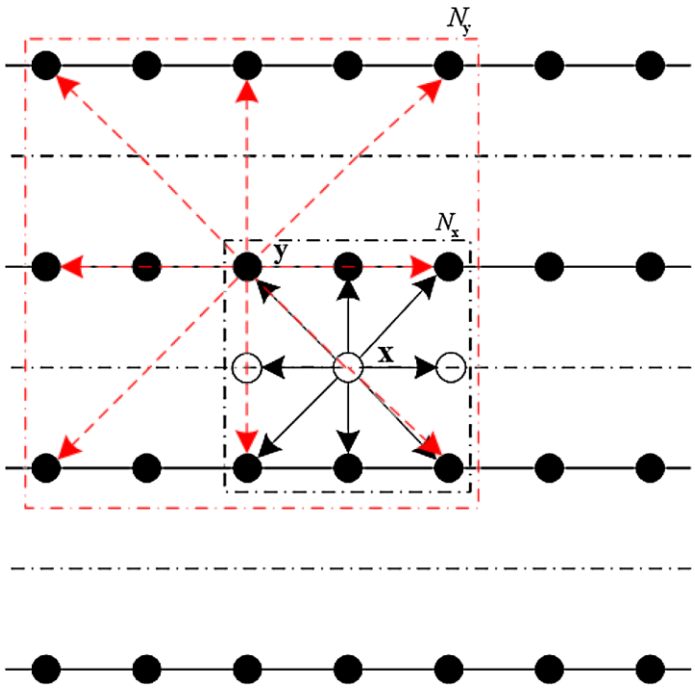


Figure 9 Visualization of NLM

3.2.5 Results

Figure 10 Denoised Image for h=20

Figure 11 Denoised Image for h=100

Figure 12 Denoised image for h=60

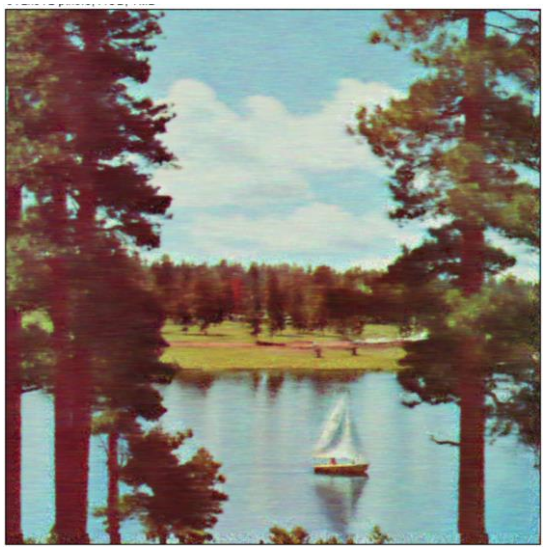


Figure 13 Denoised image for h=100

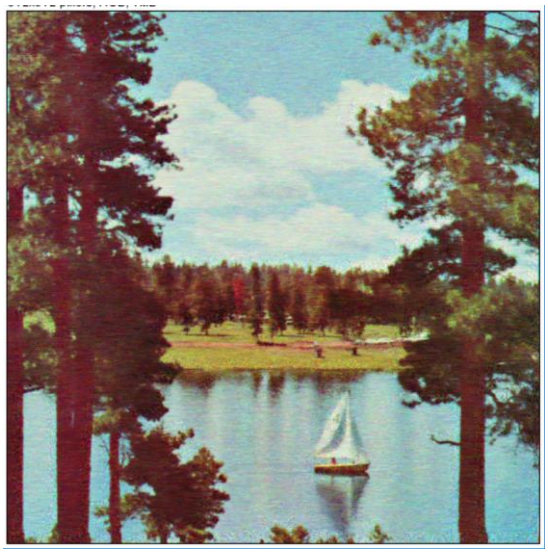


Figure 14 Denoised Image for h=60

3.2.6 Discussion

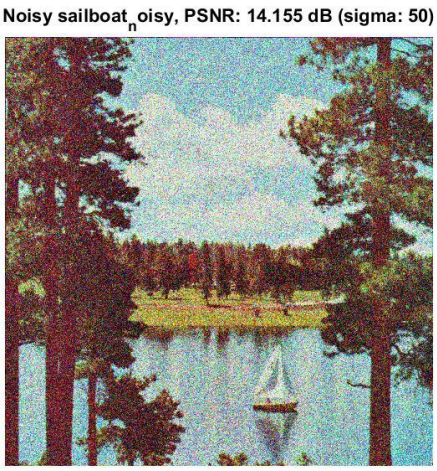
- As we can see that increasing the standard deviation denoises the image in a much better way.
- Search window also plays an important role in denoising the image.
- As the search window increases it compares with more number of neighbors resulting in much better local mean filtering.
- Increase in the search window increases the PSNR as it compares with more number of neighborhood pixel.
- Increase in the patch size decreases the PSNR as it computes the Euclidean distance which increases the Euclidean distance.

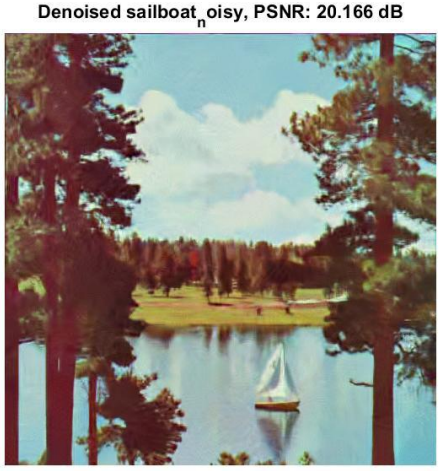
3.2.7 BM3D

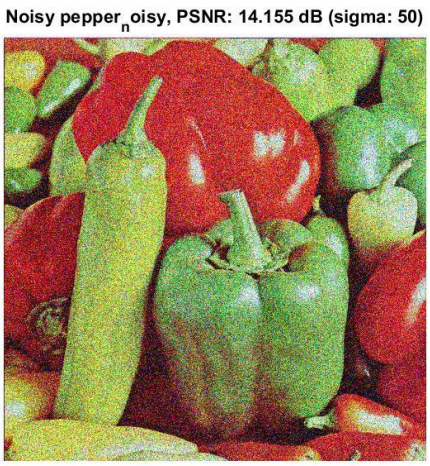
- BM3D is an approach based on filtering in 3D transform.

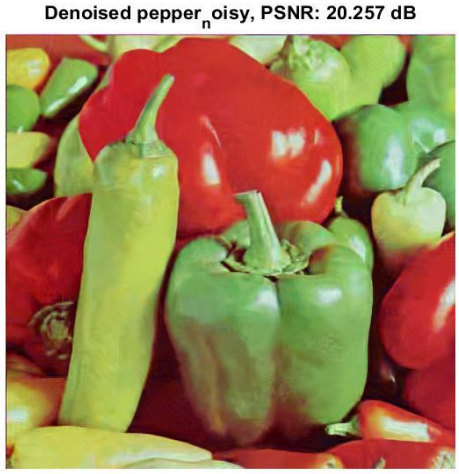
- Similar blocks are matched with block sliding mechanism which forms a 3D stacks.
- Block sliding and block matching is done to match the similar blocks of images through the entire image.
- Then the similar blocks are combined together to form a 3D array.
- The elements in the array exhibit high level of correlation due to similarity between the blocks.
- Apply unitary transform which is a 3D transform, which use this similarity between the blocks of images and reduces the noise in the images.
- Then the inverse 3D transform gives the estimates for all the matched blocks.
- Then repeat the experiment for all the sliding blocks which gives the estimates for all the similar blocks.
- Then the weighted average of these blocks will give the effective block values to the image.

3.2.8 Results









3.2.9 Discussion

- BM3D compares the similar blocks instead of similar pixels in the neighborhood.
- When compared to the NLM algorithm it is robust and time complexity is less.
- The complexity of comparing all the neighborhood pixels is much greater when compared to the blocks of similar images.
- 3D unitary transform is much faster than the Euclidean distance comparison as it computes all the stacks of similar blocks at one go.

References:

[1] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2. IEEE, 2005, pp. 60–65.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transformdomain collaborative filtering,” Image Processing, IEEE Transactions on, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] [Online]. Available: http://www.cs.tut.fi/~foi/GCF-BM3D/

[4] [Online] http://images.google.com/

[5] [Online] http://sipi.usc.edu/database/

[6] EE569_HW1_2017Spring_v5

[7] http://www.rapidtables.com/convert/color/hsl-to-rgb.htm