EE569 Project 1

Name: Shreyas Dinesh Patil
Email: shreyasp@usc.edu

Problem 1:

Image Demosaicing

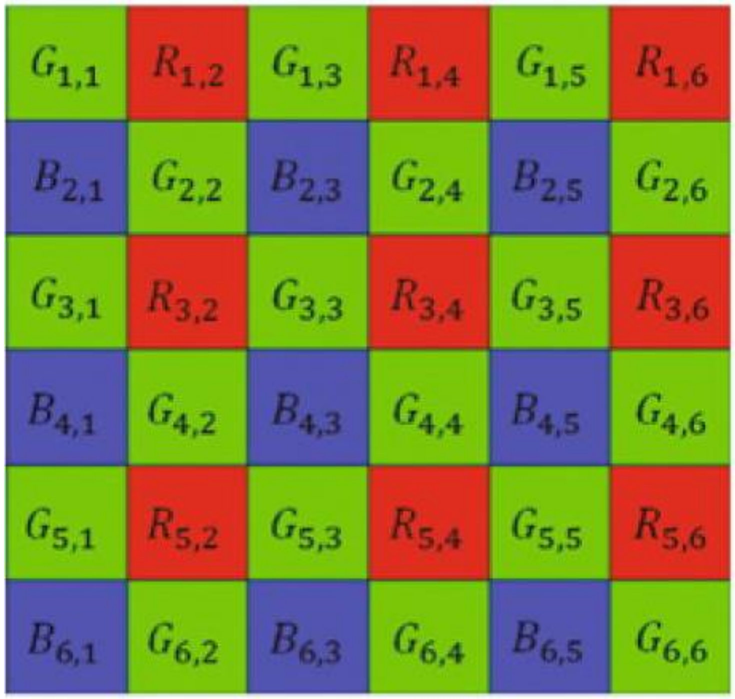
I.Motivation:

Digital cameras have replaced film-based cameras. In digital cameras at each pixel location there are at least 3 color components red, green and blue. To capture images digital camera sensors are arranged in the form of color filter array(CFA). Most of the times the filter is the Bayer's filter which is a specific pattern used for capturing single color component at each pixel location. There are many procedures that are used for reconstructing color images by constructing other color components at each pixel location. This procedure of constructing full color image from incomplete color samples output from image sensor arranged according to CFA is called Demosaicing.

2 most commonly used Demosaicing algorithms are Bilinear and Malvar-He-Cutler(MHC).

Approach for Bilinear Demosaicing:

In this method, the missing color value at each pixel is approximated by bilinear interpolation using the average of its two or four adjacent pixels of the same color based on the Bayer's pattern.


$G_{1,1}$ $R_{1,2}$ $G_{1,3}$ $R_{1,4}$ $G_{1,5}$ $R_{1,6}$
$B_{2,1}$ $G_{2,2}$ $B_{2,3}$ $G_{2,4}$ $B_{2,5}$ $G_{2,6}$
$G_{3,1}$ $R_{3,2}$ $G_{3,3}$ $R_{3,4}$ $G_{3,5}$ $R_{3,6}$
$B_{4,1}$ $G_{4,2}$ $B_{4,3}$ $G_{4,4}$ $B_{4,5}$ $G_{4,6}$
$G_{5,1}$ $R_{5,2}$ $G_{5,3}$ $R_{5,4}$ $G_{5,5}$ $R_{5,6}$
$B_{6,1}$ $G_{6,2}$ $B_{6,3}$ $G_{6,4}$ $B_{6,5}$ $G_{6,6}$

To construct Red and Blue at (3,3) we use average of adjacent Red and Blue Pixel values.

Consider the pixel location G(3,3). The Red and Blue values at (3, 3) will be calculated as,
B(3, 3) = ½(B(2, 3) + B(4, 3))
R(3, 3) = ½(R(3, 4) + R(3, 2))

Approach for MHC Demosaicing:
The green component at red pixel location is estimated as,

Gˆ(i, j) = Gˆbl(i, j) + αΔR(i, j)

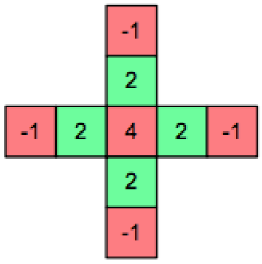
Red pixel at green pixel location,

Rˆ(i, j) = Rˆbl(i, j) + βΔG(i, j)

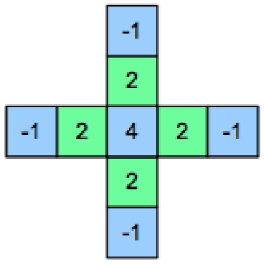
Red component at blue pixel location,

Rˆ(i, j) = Rˆbl(i, j) + γΔB(i, j)

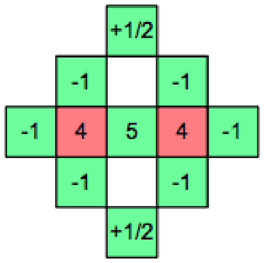
Here, the parameters α, β and γ control the weight of the Laplacian correction terms. The below eight different filters are used for interpolating the different color components at different locations.



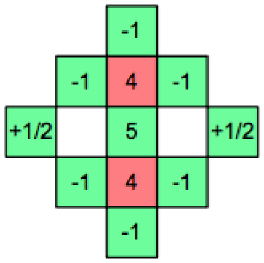
G at R locations



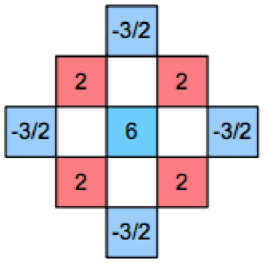
G at B locations



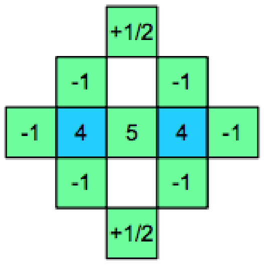
R at green in
R row, B column



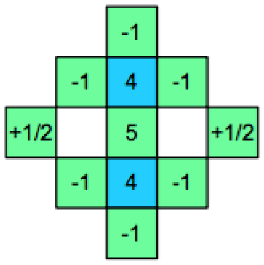
R at green in
B row, R column



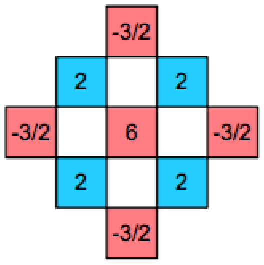
R at blue in
B row, B column



B at green in
B row, R column



B at green in
R row, B column



B at red in
R row, R column

III. Experimental Results

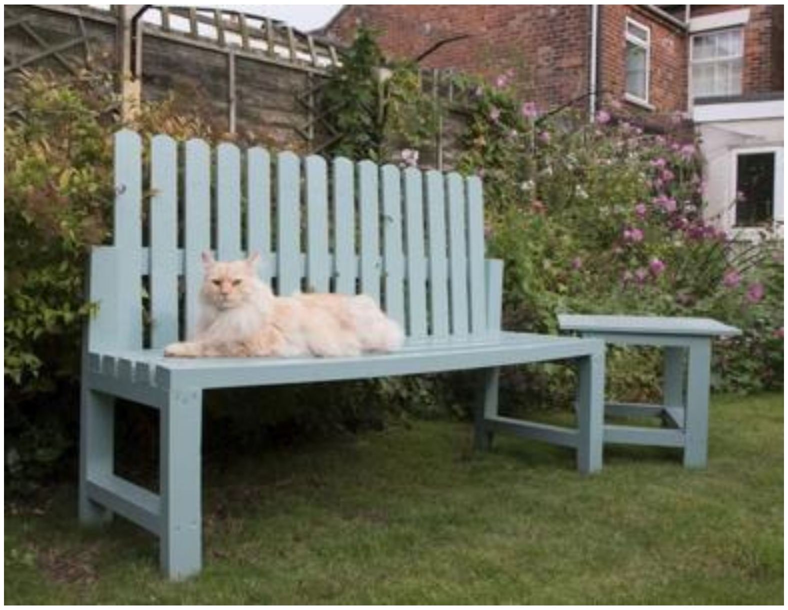


Figure1: Original image cat_ori.raw

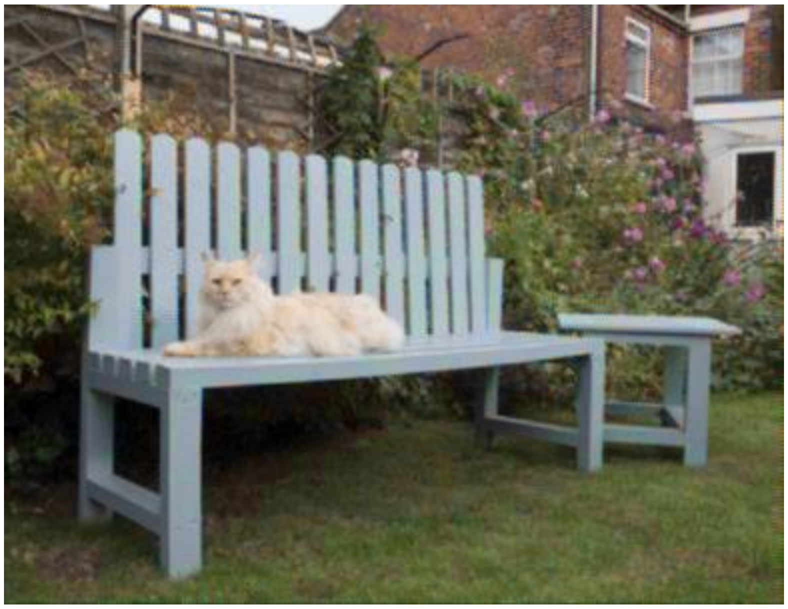


Figure2: Image after applying bilinear interpolation cat_bilinear_out.raw

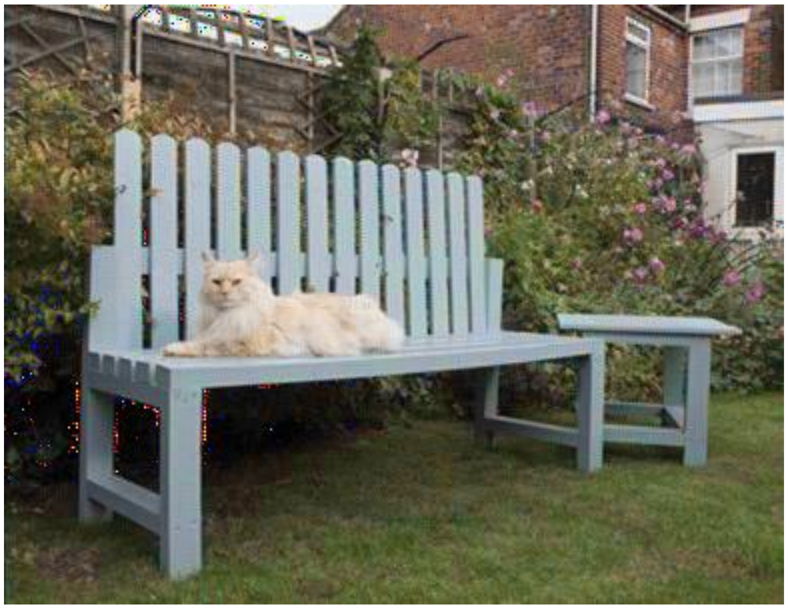


Figure3: Image after applying MHC cat_mhc_out.raw

IV. Discussion

- The visual quality of bilinear demosaicing is generally very poor. This is because visually 2 types of artifacts get generated, Zipper and false color. Zipper effect is the abrupt change in the neighboring pixels values. This can produce strong color distortions due to misalignment near edges. False colors are the colors not present in the original image. This is caused due to inconsistency among 3 color planes.
- These artifacts can be removed by using MHC demosaicing method. It yields a higher quality demosaicing result by adding a 2nd-order cross-channel correction term to the basic bilinear demosaicing result. In MHC there is dependency between all 3 channels. That dependency is implemented using linear filters with weights showing the relative dependency between neighboring pixel values. This reduces the zipper effect present in bilinear demosacing.
- Comparing MHC and bilinear results we observe MHC gives more clear and sharper image than bilinear. This is because bilinear does interpolation of channels independently of each other. MHC addresses the color channel dependency and gives closer results to the original image.

(C) <u>**Histogram Equalization**</u>

I. Abstract and Motivation

Sometimes the images background and foreground are dark or bright. This makes the objects in the images indistinguishable. Histogram equalization is a method of contrast adjustment using the image's histogram. It is used for image contrast enhancement.

II. Approach

2 methods of histogram equalization used, the transfer-function-based histogram equalization method and the cumulative-probability-based histogram equalization method.

Method A: Transfer function based histogram equalization

1) Obtain the histogram
   Count the frequency of pixels of each grayscale value (0~255)
2) Calculate the normalized probability histogram
   Divide the histogram by total number of pixels
3) Calculate the CDF
4) Create the mapping-table
   Mapping rule: x to CDF(x)* 255

Method B: Cumulative-probability-based histogram equalization

1) Sort image pixel values in ascending order
2) Calculate bucket size (size = total no. of pixels/256)
3) Assign all pixels to buckets from start.
4) Assign each pixel the value corresponding to the bucket number.

III. Experimental Results

1) Histogram Plotting

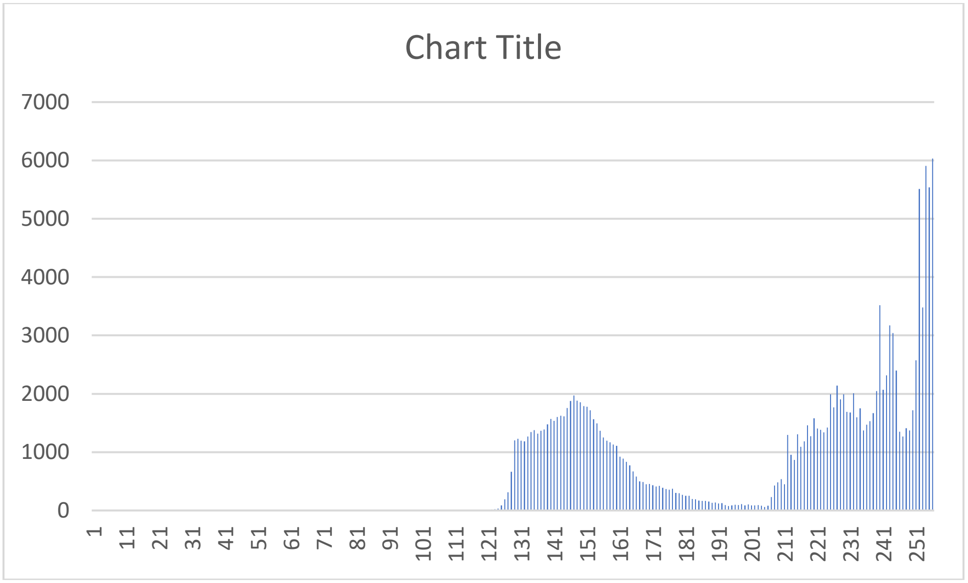


Figure1: Input histogram of rose_bright.raw

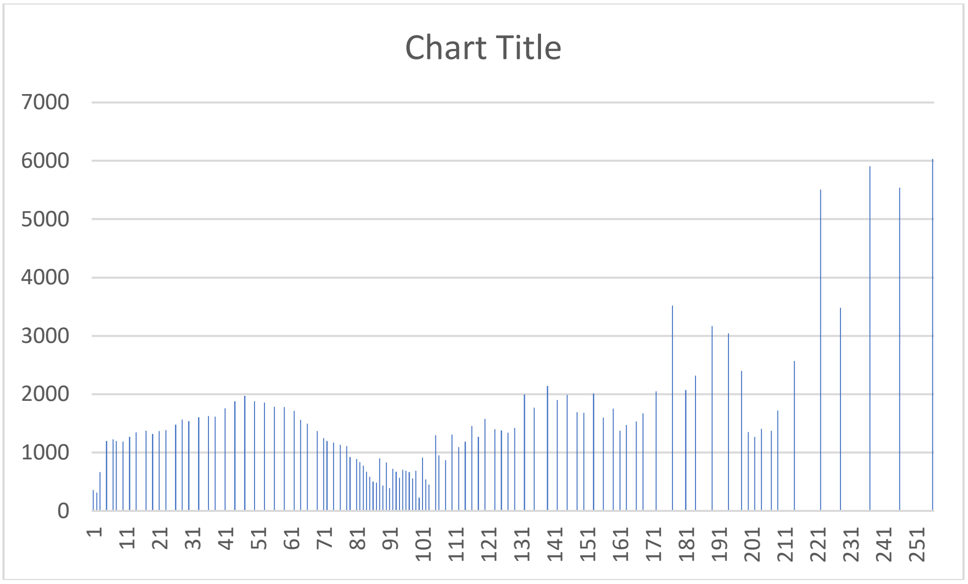


Figure2: Output histogram of rose_bright.raw using method A

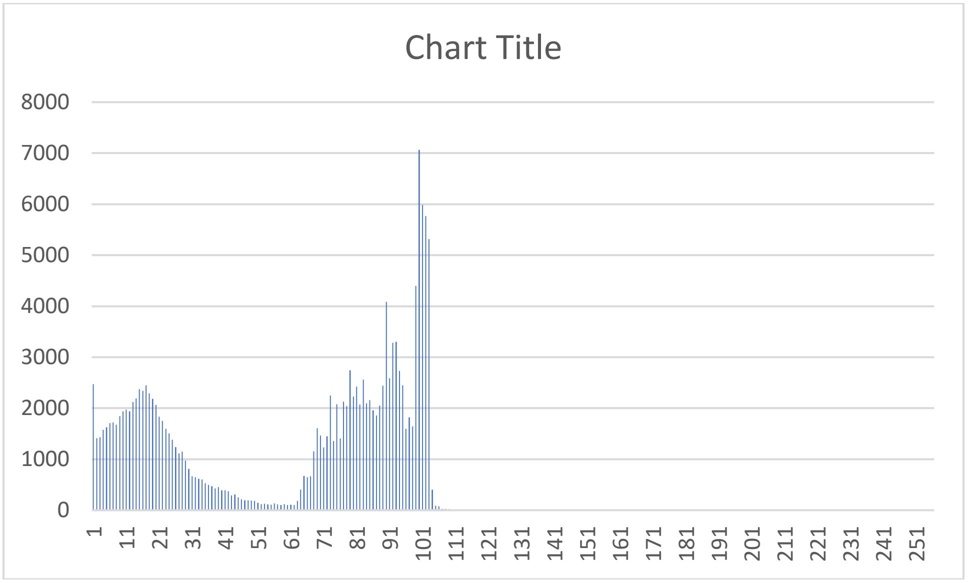


Figure3: Input histogram of rose_dark.raw

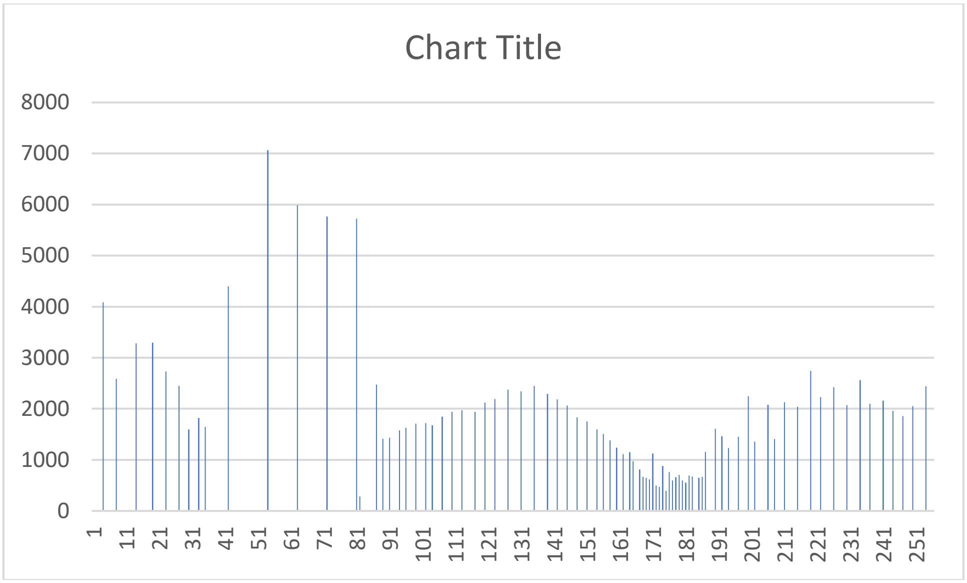


Figure 4: Output histogram of rose_dark using method B

(2)

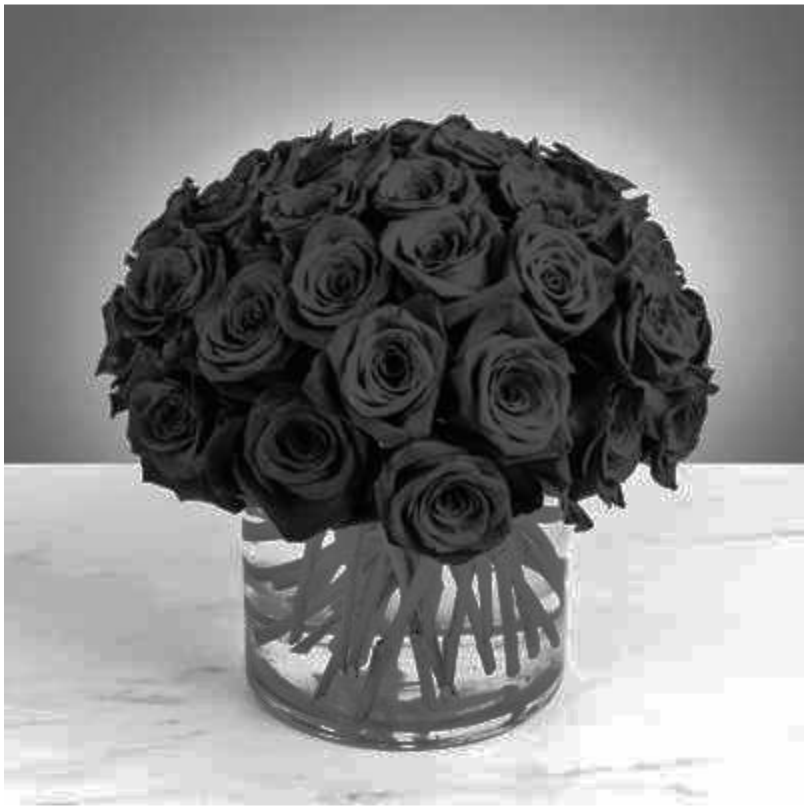


Figure1: Image after method A rose_brightA_out.raw

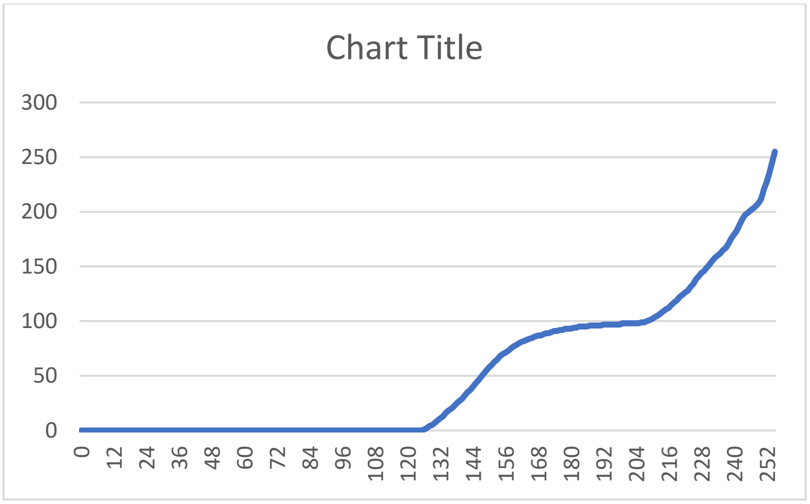


Figure2: rose_bright transfer function

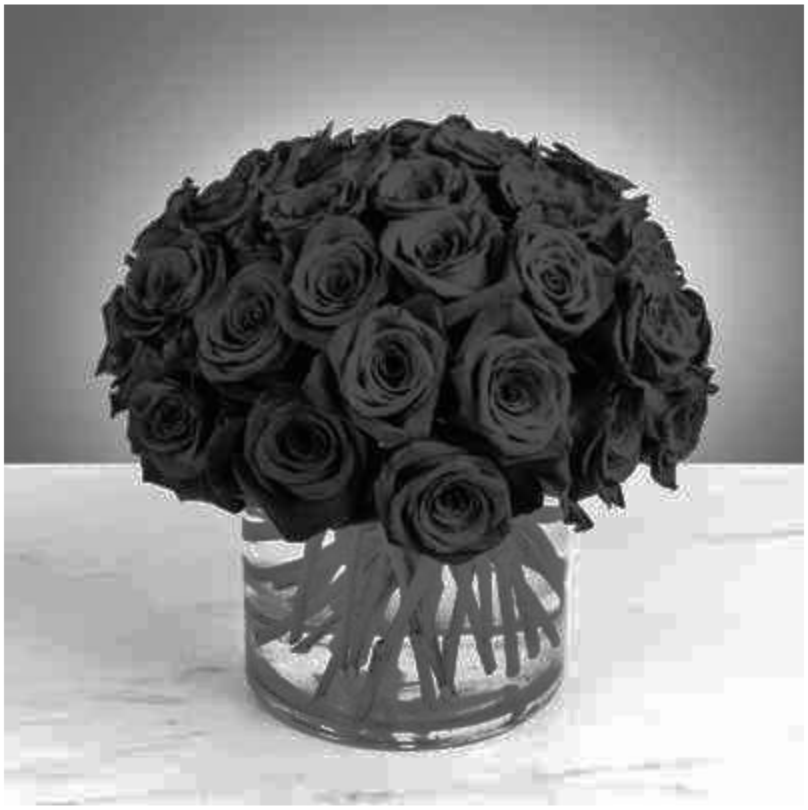


Figure3: Image after method A rose_darkA_out.raw

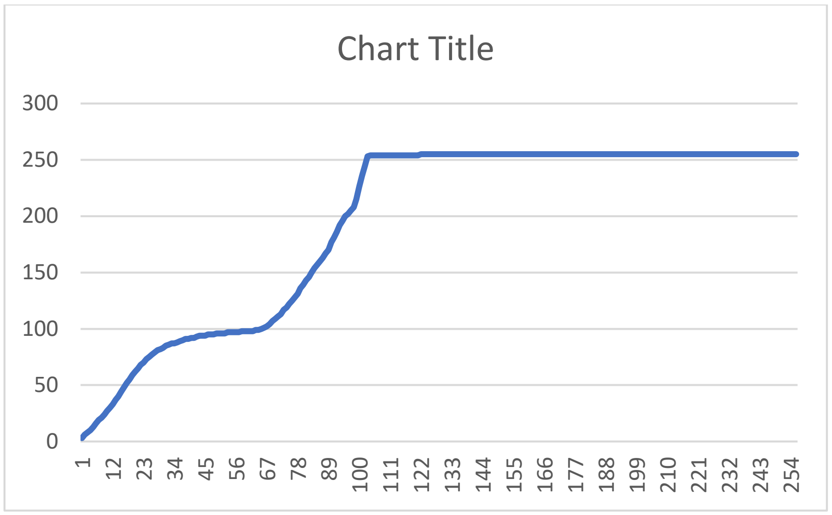


Figure4: rose_dark transfer function

(3)

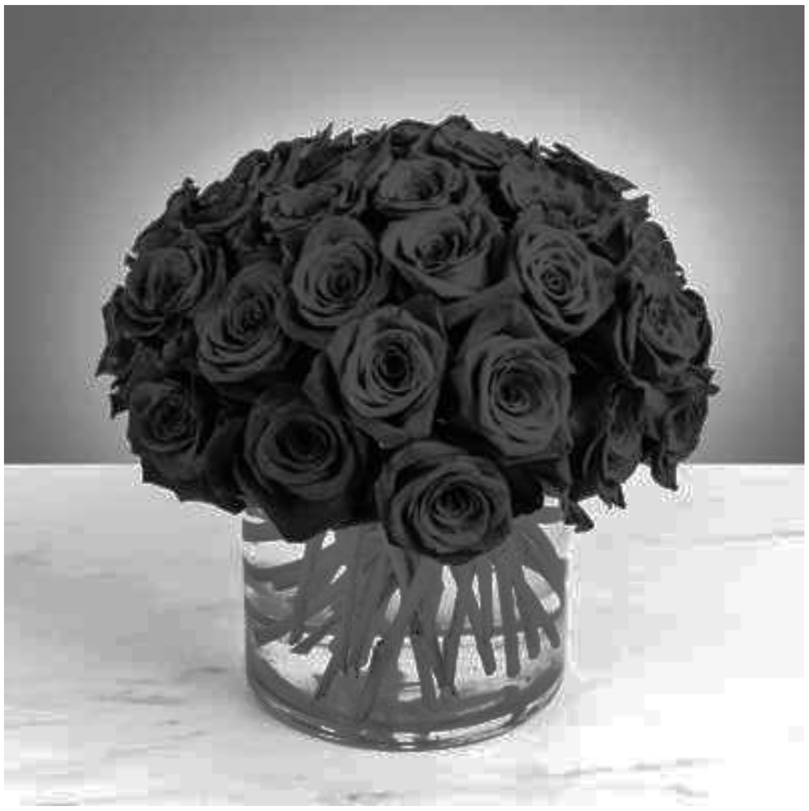


Figure1: Image after method B rose_brightB_out.raw

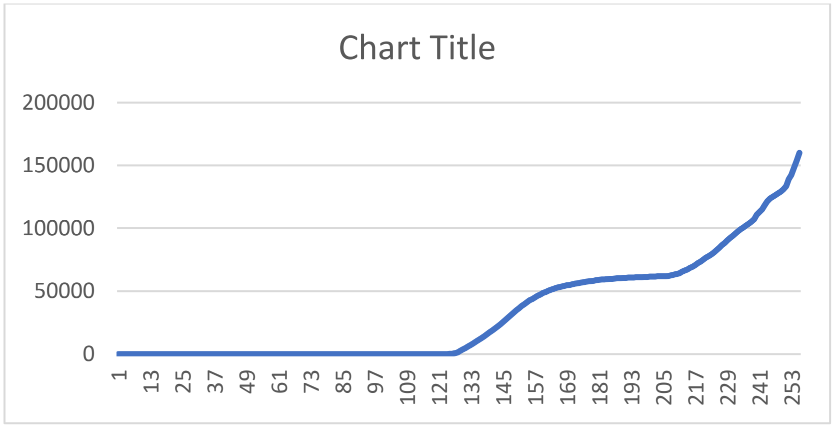


Figure2: Cumulative histogram rose_bright.raw

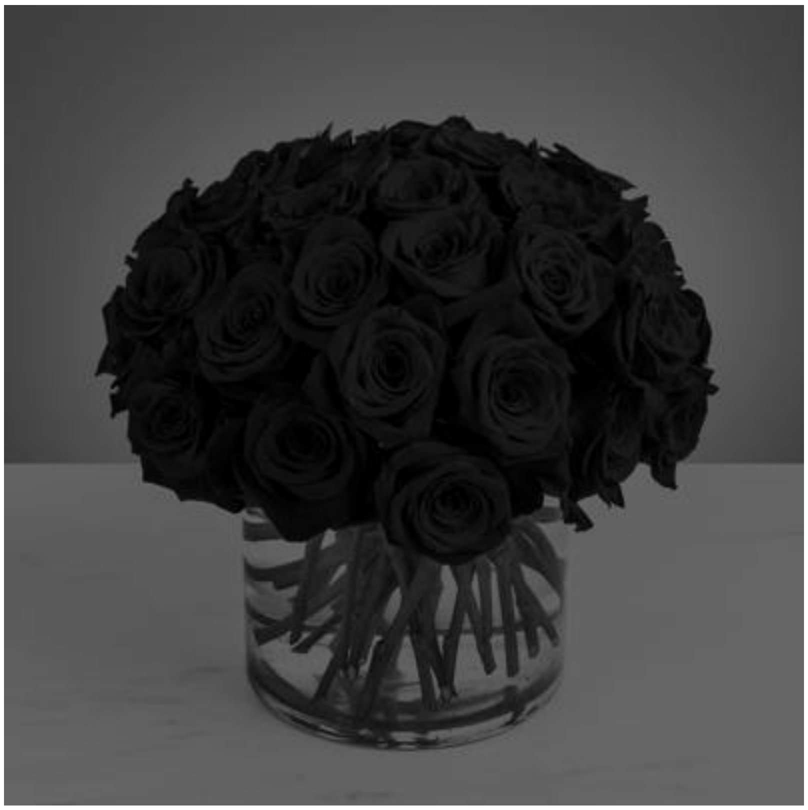


Figure3: Original rose_dark.raw image

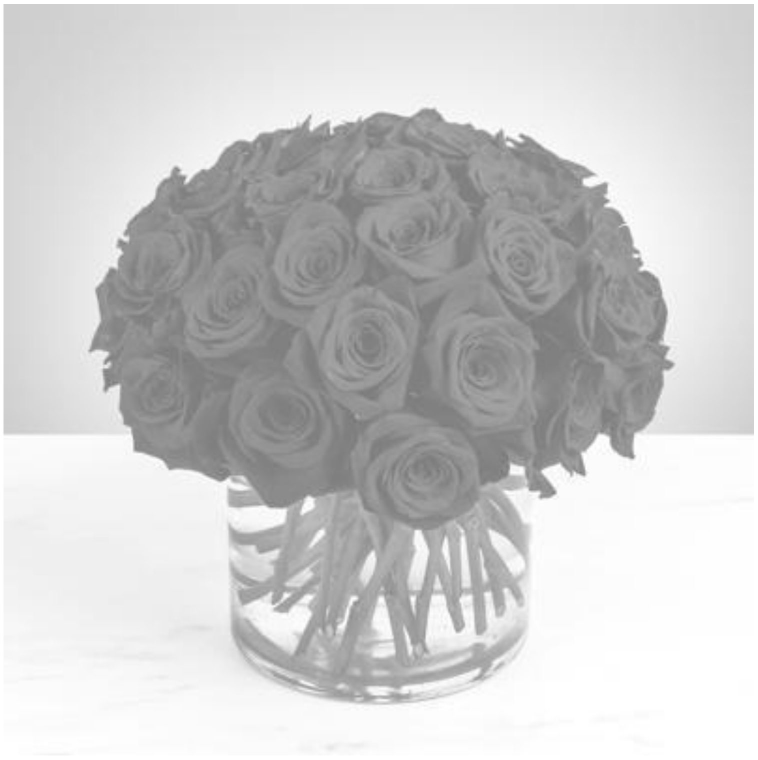


Figure4: Original rose_bright.raw image

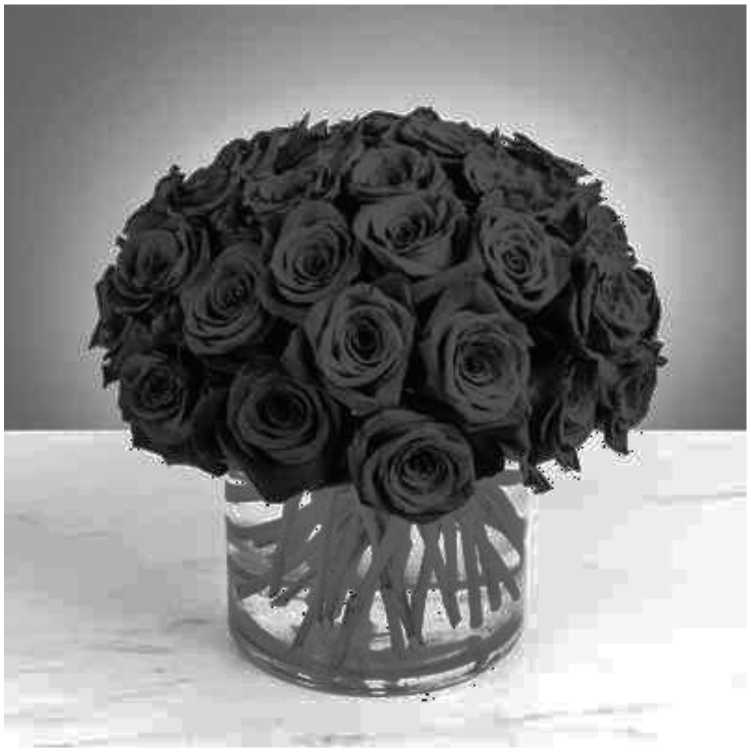


Figure5: Image after method B rose_darkB_out.raw

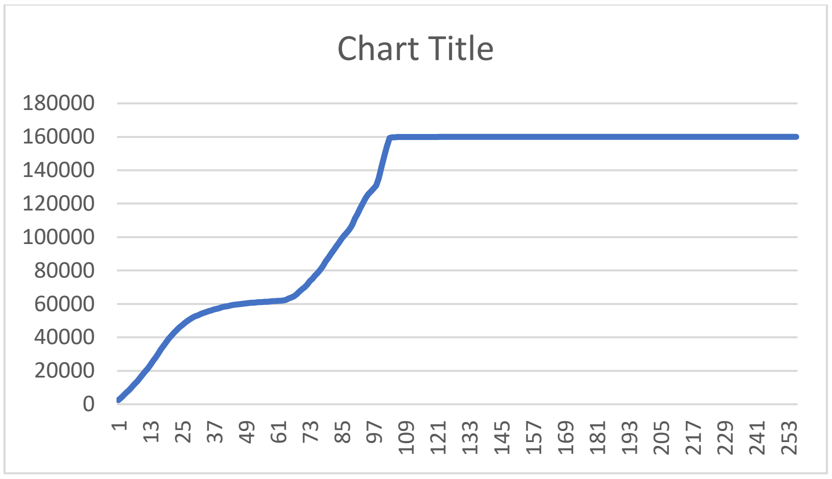


Figure6: Cumulative histogram rose_dark.raw

(5)

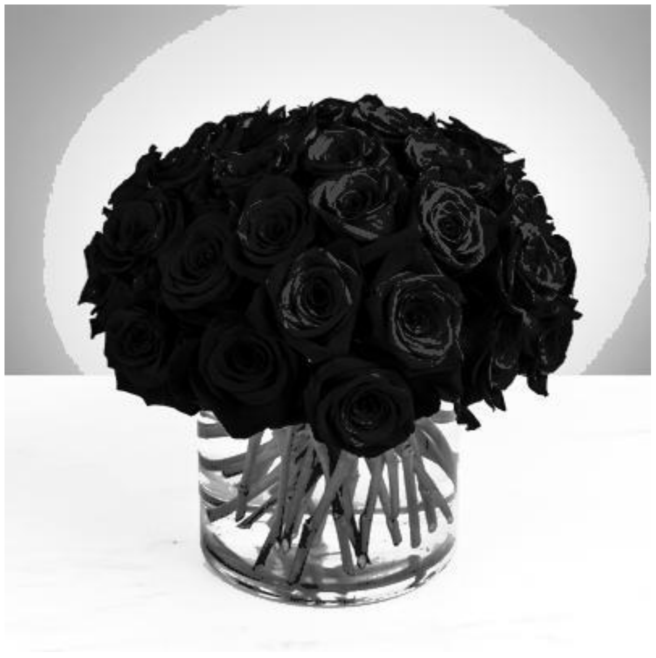


Figure1: Original rose_mix.raw image

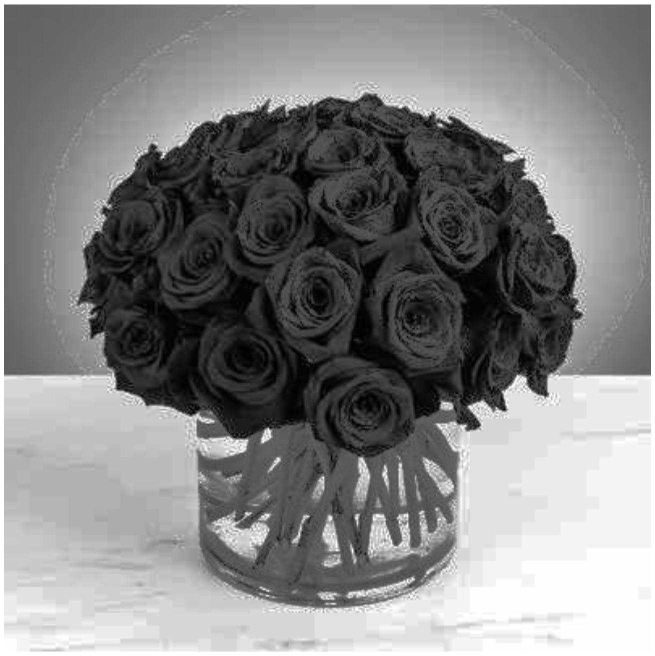


Figure2: Image after applying method A on rose_mix.raw

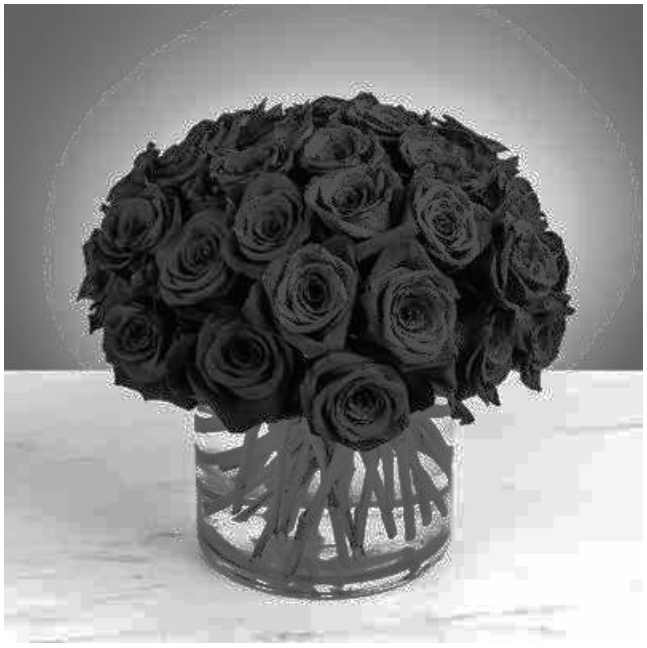


Figure2: Image after applying method B on rose_mix.raw

Discussion:

- As you can observe from above images, using method A on dark and bright rose images enhances the contrast between the rose and its background. Thus the rose pot is now clearly visible compared to the original image. Similar results have been obtained using method B.
- From the histogram of inputs and comparing those with the output image histogram we can see that the earlier concentrated pixel values near 0 for dark rose and near1 for bright rose have been spread throughout the gray scale.
- From transfer function for test images rose_dark and rose_bright we can see the enhancement of images. The number of lower pixel values in case of rose_dark have increased and higher values have remained constant which shows the equalization of pixel values. Same is the case with rose_bright where number of higher pixel values having been redistributed to pixels with lower values.

Q5A5: Yes the result is similar as in previous part.

Problem 2:

Image Denoising

(a) <u>Gray-level image</u>

I. Approach

The pepper image with uniform noise can be removed with uniform weighted and gaussian weighted low pass filters which are linear filters. For uniform filter the pixel value at the center of the filter is replaced with the average of its neighboring pixels in the filter.

In gaussian filter the neighboring pixel values are weighted according to the following expression,

W(I, j , k, l) = (1/sqrt(2*pi))*exp{(-[i-k]^2 – [j-l]^2)/(2*(a^2))}. Here (i, j) are the filter center coordinates and (k, l) are the neighboring pixel coordinates in the filter.

Bilateral filter is a non-linear filter which preserves edges while denosing.

Bilateral filter has coefficients given by the below expression.

$$w(i,j,k,l) = exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_c^2} - \frac{\|I(i,j)-I(k,l)\|^2}{2\sigma_s^2}\right)$$

(i, j) is the center pixel location in the filter.

(k, l) is the neighboring pixel location in the filter

I(i, j) – intensity value at the center of the filter

I(k, l) – intensity value of neighboring pixel in the filter

The non-local mean filter utilizes the pixel value from a larger region rather the mean of a local window centered around the target pixel.

The weight function for the filter is,

$$f(i,j,k,l) = \exp\left(-\frac{\left\|I(N_{i,j})-I(N_{k,l})\right\|_{2,a}^2}{h^2}\right)$$

The inside expression are calculated using the below expressions,

$$\left\| I(N_{i,j}) - I(N_{k,l}) \right\|_{2,a}^{2} = \sum_{n_1,n_2 \in N} G_a(n_1,n_2)(I(i-n_1, j-n_2) - I(k-n_1, l-n_2))^2$$

$$G_a(n_1,n_2) = \frac{1}{\sqrt{2\pi}a} \exp\left(-\frac{n_1^2+n_2^2}{2a^2}\right)$$

N(x, y) is the window centered around location (x, y). N is the local neighborhood and n1, n2 belongs to N.

III. Experimental Results

(1) The type of embedded noise is uniform.

(2) Uniform and gaussian weighted filter results

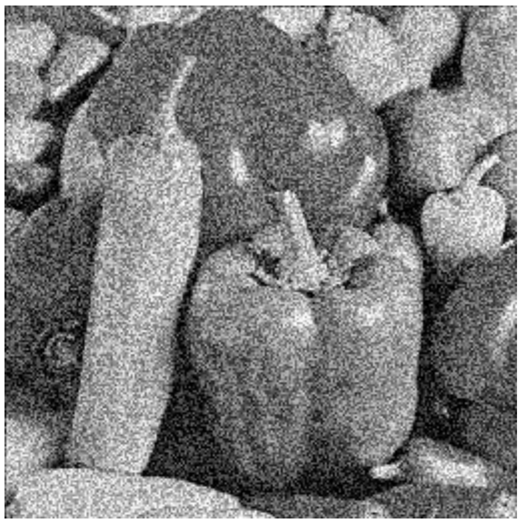


Figure1: Original image of pepper with uniform noise

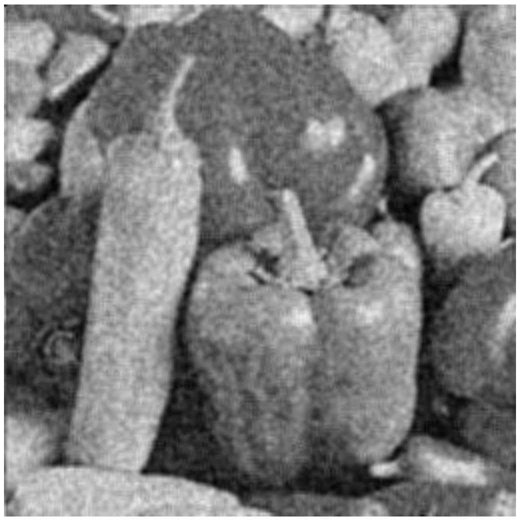


Figure2: Image after applying (3x3) uniform filter

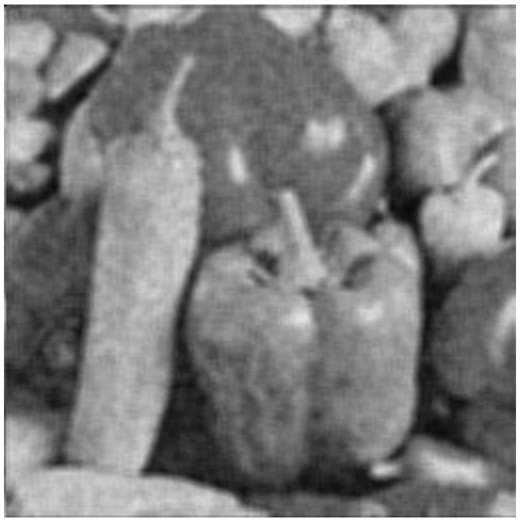


Figure3: Image after applying (5x5) uniform filter

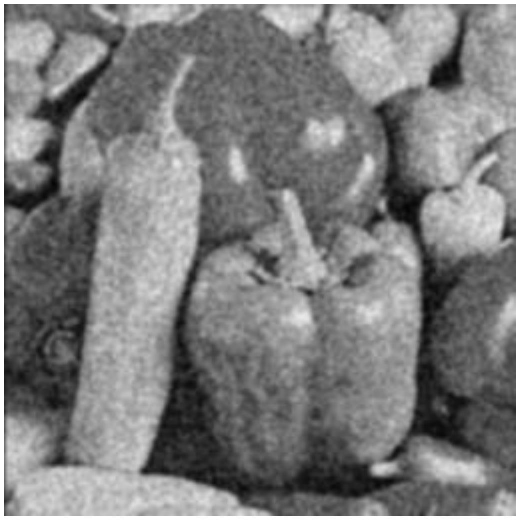


Figure4: Image after applying (5x5) gaussian filter

(3) Bilateral filter results

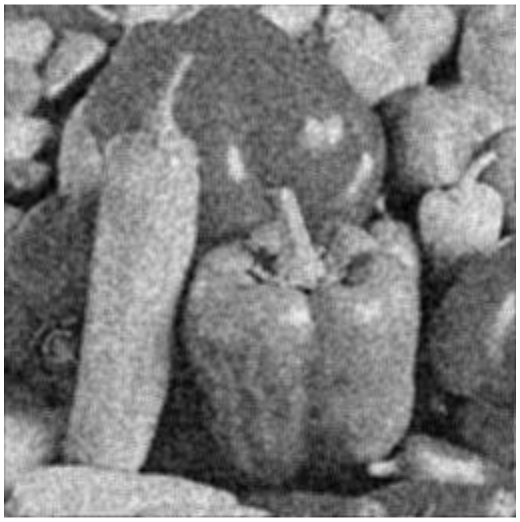


Figure1: Image after applying (3x3) bilateral filter

(4) Non-local mean filter results

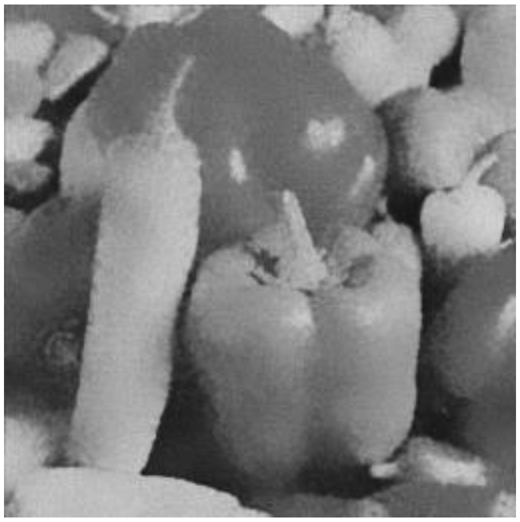


Figure1: Image after applying (3x3) non-local mean filter with spread parameters a=1 and s=100

IV. Discussion

1) From the above results we can see that uniform weighted filter blurs the image and also smooths the edges. The border of the image has been blurred by uniform filter, but gaussian filter has preserved the image edges and border with less denoising compared to uniform filter.
2) Bilateral filter output has preserved the image border almost perfectly and also preserved the edges in the images. The weights of bilateral filter not only depend on position of neighboring pixels (as is the case in the gaussian filter) but also on their intensity values.
3) The non-local mean filter utilizes the pixel value from a larger region rather the mean of a local window centered around the target pixel. Thus this filter takes into account pixels far away from the center pixel being considered by the window. This creates global dependency of pixel values and their location with the center pixel. The result obtained is clean and has noise removed along with edges being preserved. The drawback is that the filtered image looks little unrealistic.

(b)Color Image

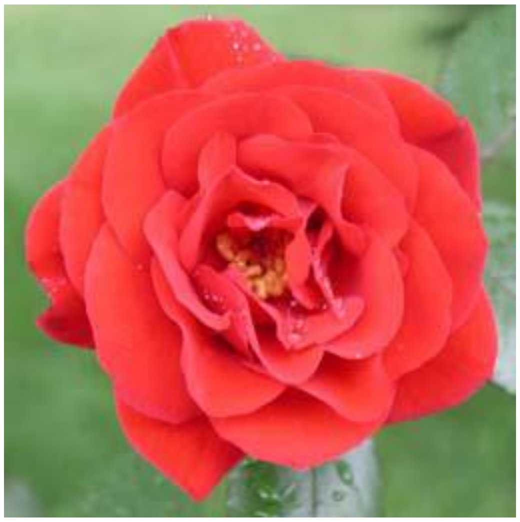


Figure1: Original rose_color image

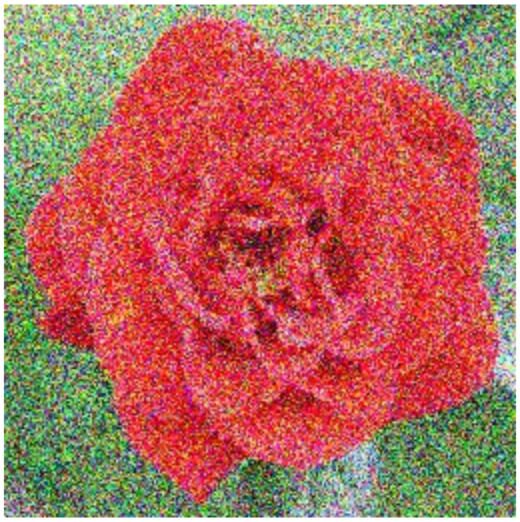


Figure2: Original rose_color image with mixed noise

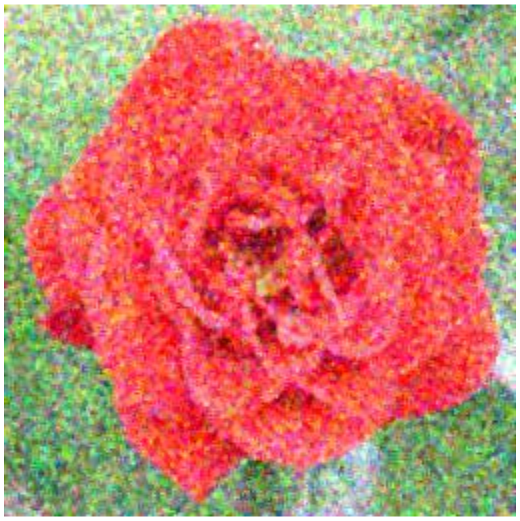


Figure3: rose_color image after applying (3x3) median filter

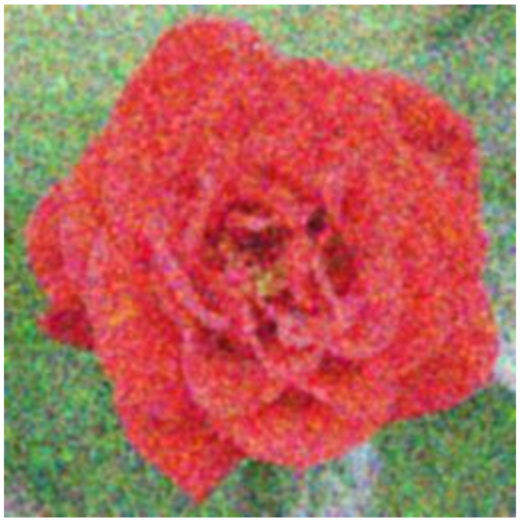


Figure4: rose_color image after applying (5x5) gaussian filter

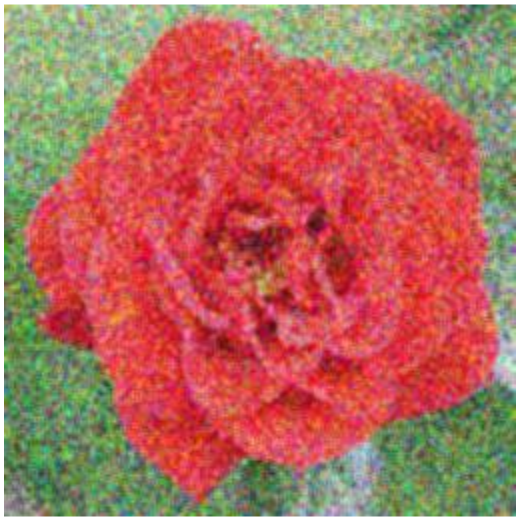


Figure5: rose_color image after applying (5x5) uniform filter

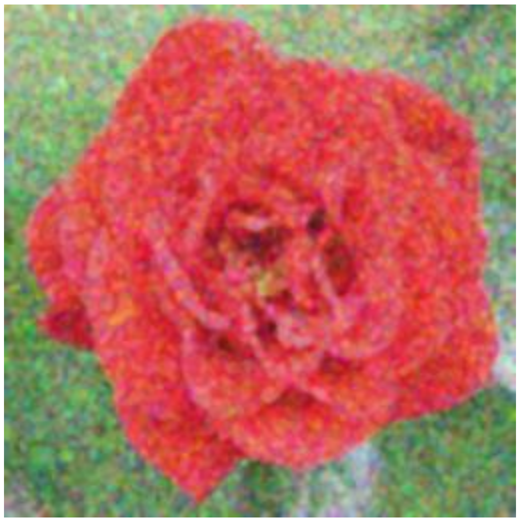


Figure6: rose_color image after applying (3x3) median filter on (5x5) gaussian filtered image.

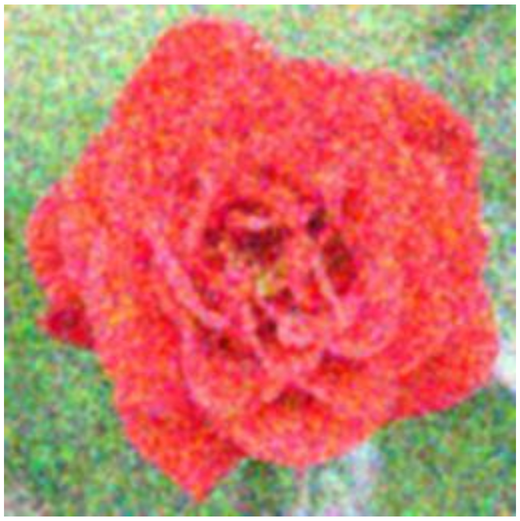


Figure7: rose_color image after applying (5x5) gaussian filter on (3x3) median filtered image.

IV. Discussion

(1) Yes, we should perform denoising on individual channels separately for both noise types.

(2) I have used gaussian, uniform and median filters to remove mixed noise. Uniform and gaussian filters help in removing uniform noise effectively and median filters in impulse noise.

(3) No the filters cannot be cascaded in any order. We can observe that applying first (3x3) median filter and then (5x5) gaussian filter results in a better image than applying (3x3) median filter on (5x5) gaussian filter.
Median cascaded with gaussian is near original image in the sense of visual quality.

(4) Bilateral and non-local mean filters reduce the blurring effect and preserve the edges in the image. They can be effective in removing uniform noise. Impulse noise can be removed with median filter. Median filter cascaded by bilateral or non-local mean filter will yield better results because the the impulse noise will be reduced after applying median filter and thus impulse noise cannot mislead non-local mean or bilateral filter while considering larger neighborhood filter.

(C) <u>Shot noise:</u>

Approach and Motivation:

Sometimes noise is introduced in the dark parts of the captured image. This is generally the case with electronic camera image sensor**.** A common way to remove shot noise is using Anscombe root transformation.

The steps to remove shot noise using Anscombe root transformation:

1) First apply Anscombe root transformation at each pixel location. The transformation at each pixel location is given by,
   f(z) = 2*sqrt(z + 3/8);
   the noise variance is stabilized by applying the Anscombe root transform. Therefore we can treat the image as mixed with gaussian noise and unit variance.
2) Then any filter that removes gaussian noise is used on the transformed image
3) Finally take the inverse transform of the filtered image.

IV. Experimental results

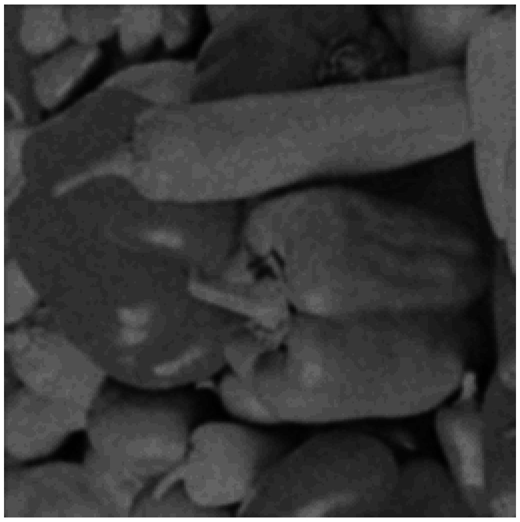


Figure2: Results of shot noise denoising using (5x5) gaussian low pass filter