EE569 – HOMEWORK – 1 REPORT

SHILPASHREE RAO

USC ID: 5636765972

EMAIL ID: shilpasr@usc.edu

PROBLEM 1: BASIC IMAGE MANIPULATION

a) Color Space Transformation

1. Color – to – Grayscale Conversion:

- **Motivation:** Grayscale / Black and white / monochrome images have pixel values in between the weakest and strongest intensity values namely black and white respectively. These images are composed of shades of gray.
  Each pixel of a color image is represented by triple intensities or combination of Red, Green and Blue channel intensities. These three channels can be represented by a single grayscale value.

- **Approach and Procedures:** Color to Grayscale conversion can be done using various algorithms. Three of the important algorithms are discussed here, namely, the Lightness method, Average Method and the Luminosity method. The formulae for the listed methods are shown as below in Figure 1.

  Lightness: Y = max((R, G, B) + min(R, G, B)) / 2
  Average: Y = (R + G + B) / 3
  Luminosity: Y = 0.21R + 0.72G + 0.07B

  Figure 1: Formulae for algorithms to convert images from Color to Grayscale
  (Source: Discussion slides)

  To compute the above outputs, each corresponding pixel values of R, G and B color channels are accessed and the above operations are performed.
  The lightness method averages the most and least significant pixel values of every location in an image. The average method simply averages the 3 channel pixel values. The luminosity method assigns weights to every channel pixel intensities and adds them up together. The Green channel pixel value is given more weightage as compared to Red and Blue.

- **Results:** The Lightness, Average and Luminosity outputs for the Color Tiffany image is as shown below in Figure 2, Figure 3 and Figure 4 respectively.

Figure 2: Lightness output

Figure 3: Average Output

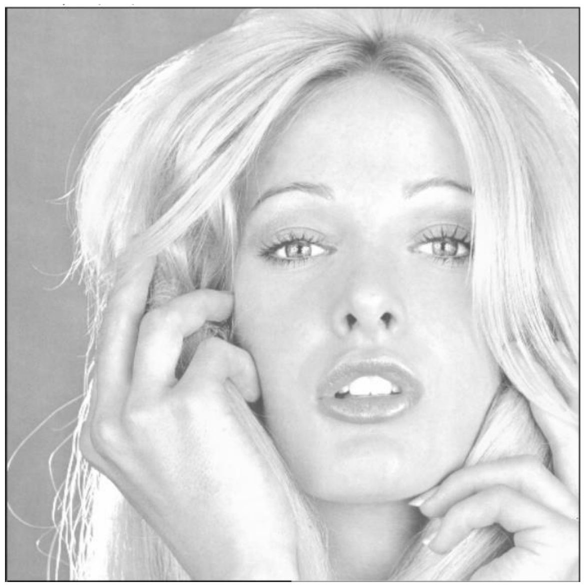


Figure 4: Luminosity Output

- **Discussion:** On observation, the Luminosity output is better as compared to the other two methods. The Average output visually looks better than the Lightness output. The brightness of the Luminosity output is higher than that of average and lightness output. The contrast of the image is reduced in the lightness method. The Green intensity value is weighted higher for the Luminosity output, which is more sensitive to human perception and hence accounts as the best method for color to grayscale conversion.

2. CMY(K) Color Space:

- **Motivation:** The CMY(K) color space is used for printing applications. The reason being, white paper emits light and CMY(K) model adds pure black(K), that can absorb light with low consumption of ink for printing. The CMY stands for Cyan, Magenta and Yellow which follows subtractive color mixing, i.e., transposition of RGB model. When the CMY colors are mixed, an imperfect black or perfect gray is generated. Here, the CMY color channel output images are displayed individually, which are obtained by operating on RGB image.

- **Approach and Procedures:** The CMYK color space can be obtained from the RGB color space. Each channel of the image is taken, and every pixel intensity is subtracted by 255, which is the maximum intensity value in the image. This can also be done by normalizing the pixel intensities between 0 and 1 and then subtract 1 by these intensities as shown in Figure 5. The value obtained is then multiplied by 255 to output the channel image.

- 1. Normalization all the pixel values to [0, 1]
- 2. $$\begin{cases} C = 1 - R \\ M = 1 - G \\ Y = 1 - B \end{cases}$$

Figure 5: RGB to CMY(K) conversion procedure

- **Results:**

Figure 6: Bear output for C channel

Figure 7: Bear output for M channel

Figure 8: Bear output for Y channel

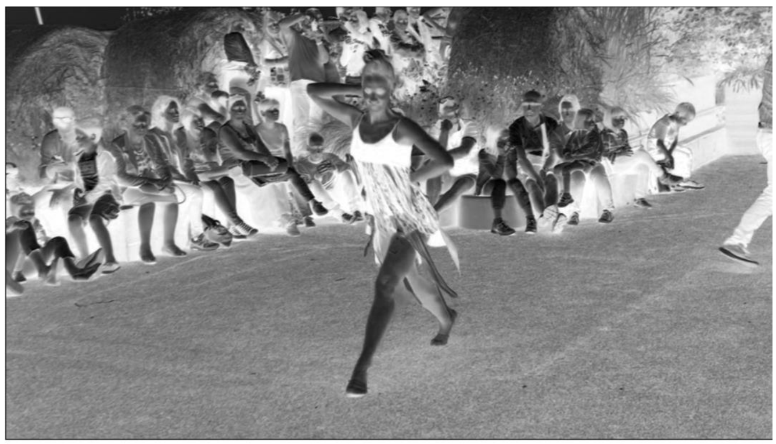


Figure 9: Dance output for C channel

Figure 10: Dance output for M channel

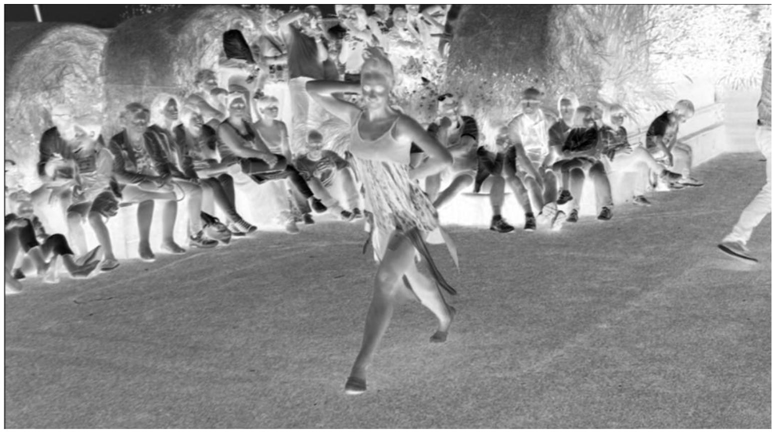


Figure 10: Dance output for Y channel

- **Discussion:** The C, M, Y channel outputs are observed to be on the darker side. The Y channel output is brighter as compared to the M and C channel outputs. Hence this technique can be utilized for printing for lower ink consumption.

b) Image Resizing via Bilinear Interpolation:

- **Motivation:** Image Zooming and shrinking without much loss is an important application in image processing. While resizing, or zooming as in the case of this problem, the new pixels are to be estimated close to the pixel values from the original image for an enhanced human perception.

- **Approach and Procedures:** In the Figure given below, the value of F(p', q') has to be interpolated or evaluated. First the ratio of the size of original image is to the size of the image to be resized is taken. For every new pixel location in the resized image, the ratio is multiplied with its co-ordinates, to estimate the corresponding co-ordinates in the original image. This value however, will generate an offset. Depending on the generated offset, the value of the location is interpolated to allocate a new pixel intensity for the resized image.

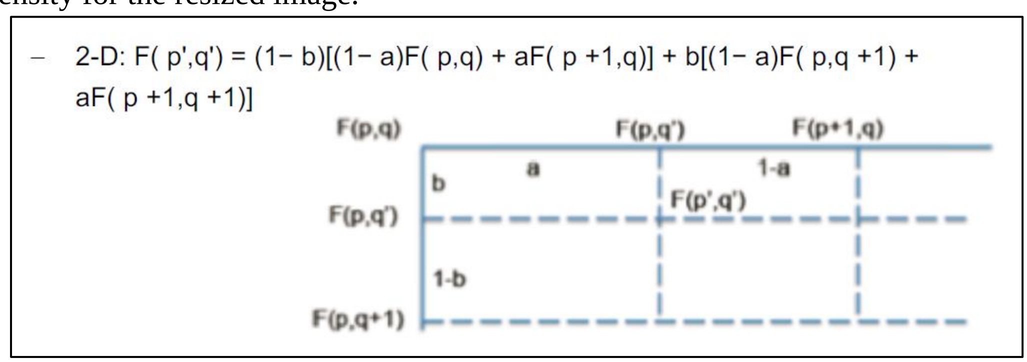


Figure 11: 2D Bilinear Interpolation for Image zooming/ Resizing (Source: Discussion Slides)

Shilpashree Rao
USC ID: 5636765972
Email id: shilpasr@usc.edu

- **Results:** The output after performing Bilinear Interpolation on 512 X 512 image is as shown below. The image is resized to 650 X 650.

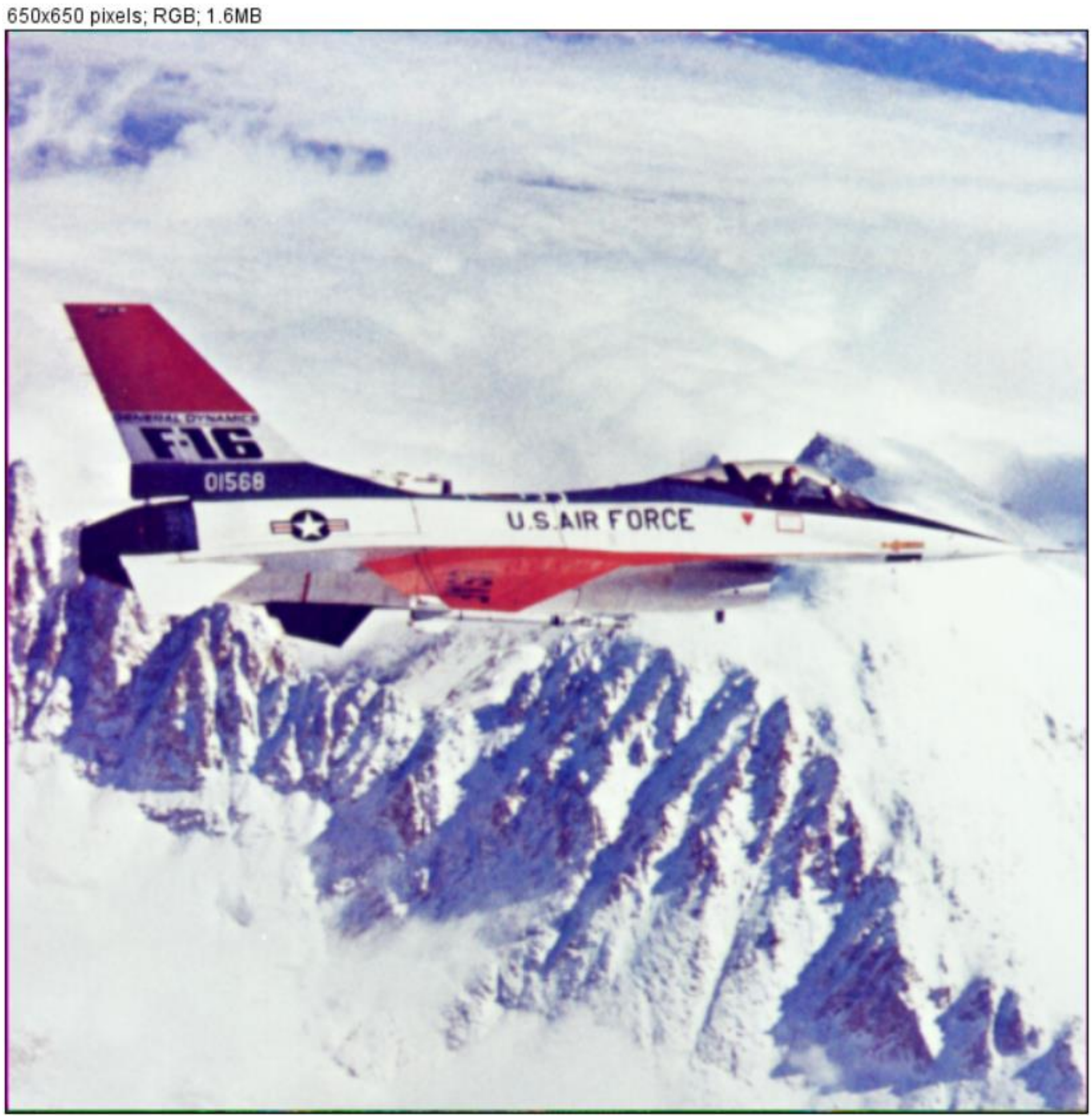


Figure 12: 650 X 650 Bilinear Interpolation Resized output

- **Discussion:** The Bilinear Interpolation accounts for the offset or non-integral zoom factor to output a good resized estimation of the original image. Certain amount of visual distortion is reduced and the output looks satisfying to the human eye. This method has proved to be better than nearest-neighbor interpolation which causes some pixels to appear more than the others due to a higher frequency in the neighborhood.
  The Bilinear Interpolation method is a non-adaptive resizing technique and hence gives a blurry resized output. We can use adaptive interpolation algorithms for a better visual quality by studying the image properties.

PROBLEM 2: HISTOGRAM EQUALIZATION

a) Histogram Equalization:

- **Motivation:** Images can be captured in poor illumination resulting in poor quality images or poor distribution of pixel intensities in an image. Such images have pixel values concentrated to a very narrow range which becomes difficult for human perception. Image enhancement can be done using contrast enhancement techniques such as histogram equalization methods wherein the pixel values are distributed to a wider range, enhancing visual quality and human perception.
  Two methods namely the Transfer function based and Cumulative Histogram based methods are discussed here.

- **Approach and Procedures:**
  - **Transfer Function based histogram equalization method:** In this method, a transfer function is computed to map the original pixel value distribution to a desired/ a uniform distribution of pixel values in every channel of an image. The steps to calculate the transfer function is as shown below.

Pixel Intensity	1	2
No. of pixels	1	3
Probability	.0625	.1875
Cumulative probability	.0625	.25
C.P * 20	1.25	5

Method to compute Transfer Function

Firstly, histogram for each channel of an image is computed giving us the pixel intensities and the corresponding number of pixel values. The probability density function of pixel values is calculated. This is given by the ratio of Total number of pixels for a pixel value by the Total number of pixels.

The pdf values are further used to calculate the CDF values. Sum of the current pdf and the previous pdf values gives the CDF for a given pixel intensity value. This CDF value is multiplied by 255 to renormalize the pixel values that are ranged between 0 and 1.

- **Cumulative probability based histogram equalization method/ Bucket filling algorithm:** The main goal of this algorithm is to distribute equal number of pixels into all the pixel intensity values. Firstly, the histogram for each channel is obtained. The number of pixels per bin/ per intensity value is given by (Height X Width)/256. The pixels from the histogram distribution are uniformly sorted and distributed into each bin to get an equalized image.

- **Results:** Per channel histograms of the input original image, enhanced images using Method A, Method B and corresponding per channel Transfer functions and Cumulative Histograms are as shown below.

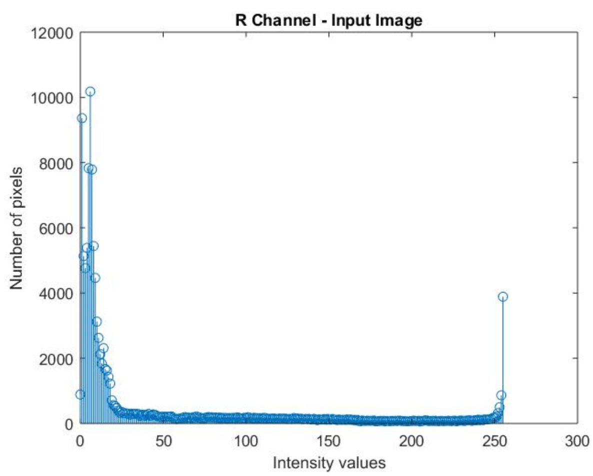


Figure 13: Original Desk Image Histogram – R Channel

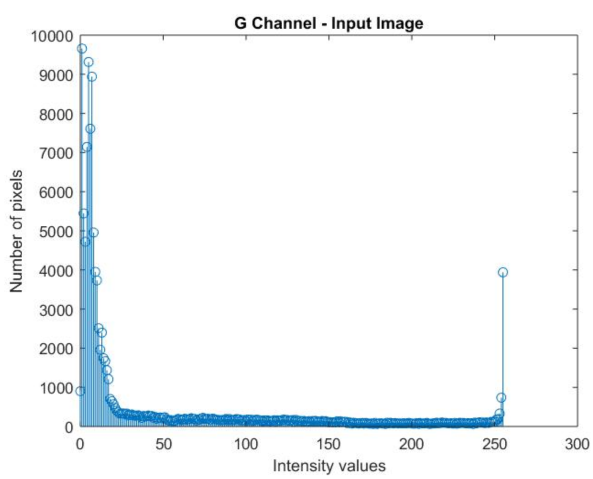


Figure 14: Original Desk Image Histogram – G Channel

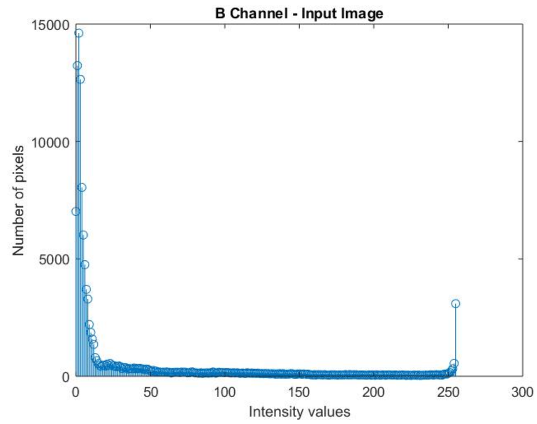


Figure 15: Original Desk Image Histogram – B Channel

Figure 16: Enhanced Image – Method A

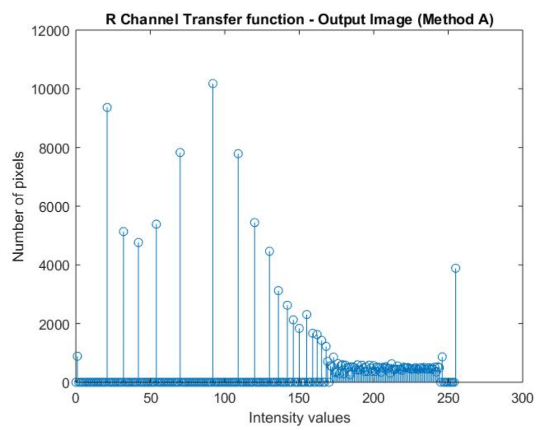


Figure 17: Method A Enhanced Desk Transfer Function – R Channel

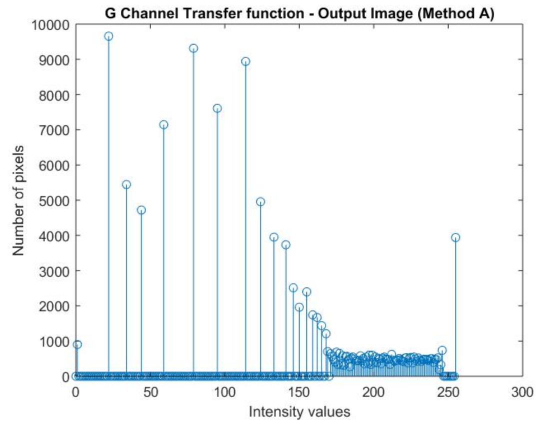


Figure 18: Method A Enhanced Desk Image Transfer Function – G Channel

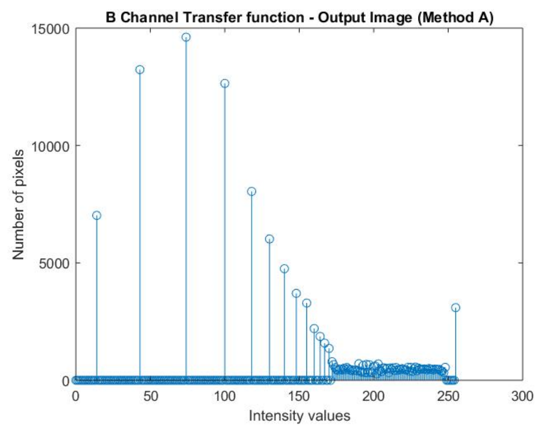


Figure 19: Method A Enhanced Desk Image Transfer Function – B Channel

Figure 20: Enhanced Image – Method B

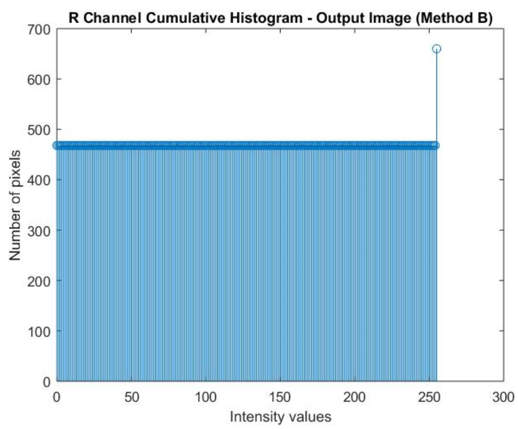


Figure 21: Method B Enhanced Desk Image Cumulative Histogram – R Channel

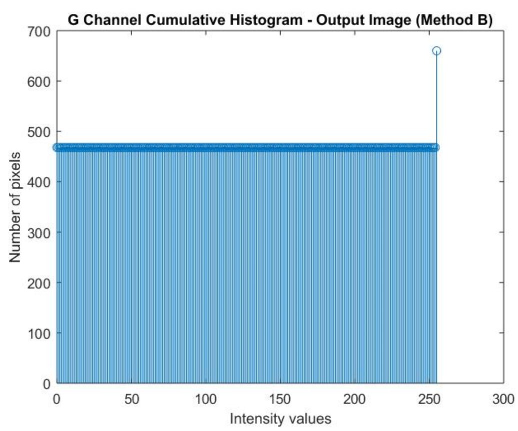


Figure 22: Method B Enhanced Desk Image Cumulative Histogram – G Channel

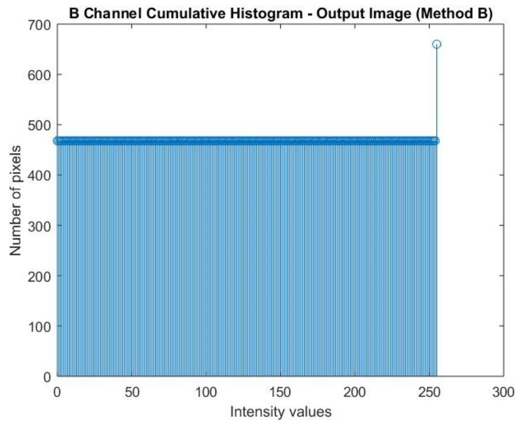


Figure 23: Method B Enhanced Desk Image Cumulative Histogram – B Channel

- **Discussion:** By observing the above histogram outputs, we can see that the input image histogram has more pixel values concentrated towards the darker side or pixel intensity values between 0 and 50. This makes the image look dark and unappealing to human eye. This is improved by the Transfer function and CDF based equalization methods as shown in their respective histograms. The pixel values are better distributed in the histograms and the image is better viewed in the enhanced images. In the Transfer function based method, many of the pixel values are not utilized which causes pixelating in the output enhanced image. This method is consistent and has an easy, fast algorithmic approach. The cumulative based method has equal distribution of pixel values and the output is visually similar to Transfer function based method with minor differences. This image is smoother with less pixilation. However, this method is computationally inefficient. Both the methods do not give us a very satisfying enhanced image output. An alternative adaptive histogram equalization method can be adopted to tackle this issue wherein multiple histograms are utilized to evaluate the enhanced image.

b) Image Filtering – Creating Oil Painting Effect:

- **Motivation:** The oil painting effect is a motivation to reduce the color space. In an RGB image, we have 256 levels in each channel which is 256^3 colors in total. These many colors are reduced to 64 colors by having 4 colors in each channel. A reduced color space gives us an oil painting effect visually.

- **Approach and Procedures:** To achieve the oil painting effect, color quantization and image filtering is performed. Firstly, the histogram of each channel of the RGB image is obtained. Pixels per bin is calculated by (Height X Width)/ 4. The number of pixels are counted from pixel value 0 in the histogram and a threshold is set when number of pixels is greater than or equal to the number of pixels per bin. The last bin may have lesser number of pixels than the number of pixels per bin. This way we get four bins per channel. Mean of the pixel intensities is calculated in every bin and all the pixels in that bin is replaced by the mean value. This way we get 4 colors per bin, resulting in 4^3 colors for the entire image, giving us the color quantized image. To get the filtered image, the color quantized image is padded with (N-1)/2 layers of pixels with the value > 256, where N is the window size. In my code, the padding layer is initialized with the value 300. The window is traversed on the quantized image ignoring the padding, and all the pixel values in the window expect the ones which have the value 300 are passed into a function that returns the pixel value which has the highest frequency. All the pixels in the window are replaced by this value that has highest frequency. This way we get a filtered image.

- **Results:** The Quantized image for Star wars and Trojan images, 64 color outputs with N (window size) = 3, 5, 7 are as shown below. The outputs for 512 colors are also shown.

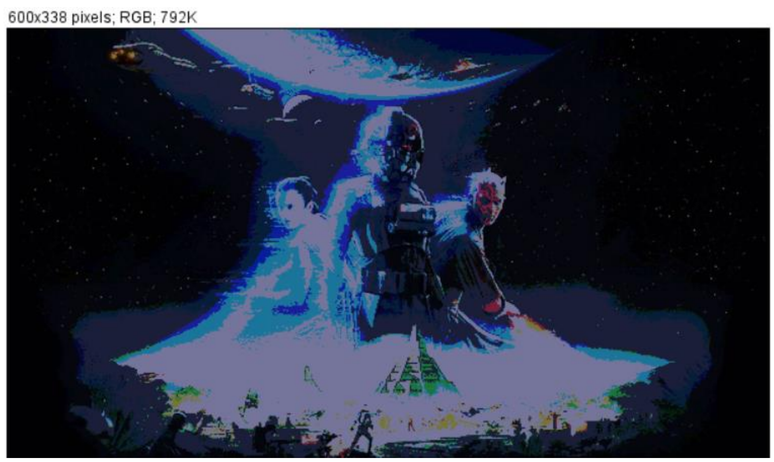


Figure 24: Quantized Star Wars Image – 64 colors

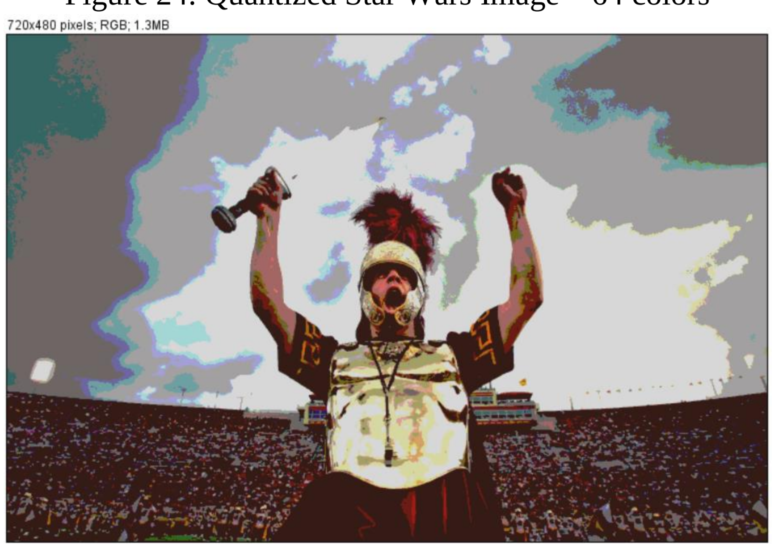


Figure 25: Quantized Trojan Image – 64 colors

Figure 26: Oil painted Star Wars Image – 64 colors – N = 5

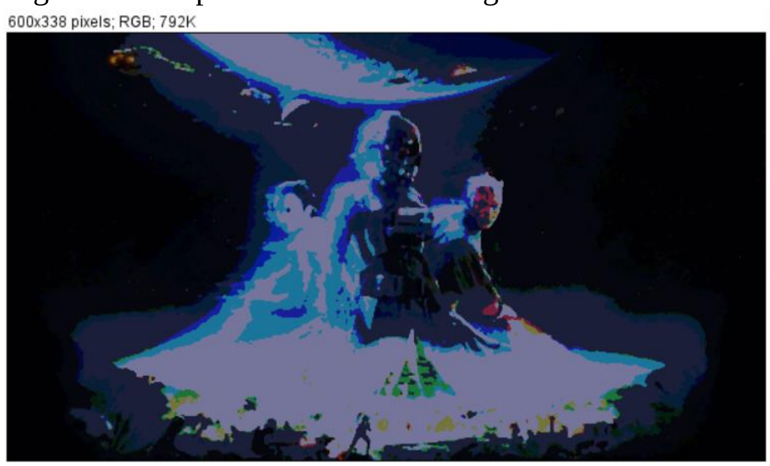


Figure 27: Oil painted Star Wars Image – 64 colors – N = 3

Figure 28: Oil painted Star Wars Image – 64 colors – N = 7

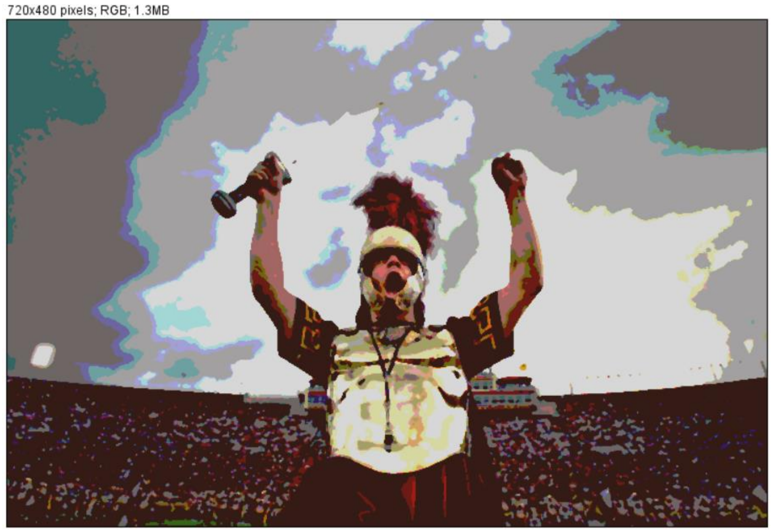


Figure 29: Oil painted Trojan Image – 64 colors – N = 3

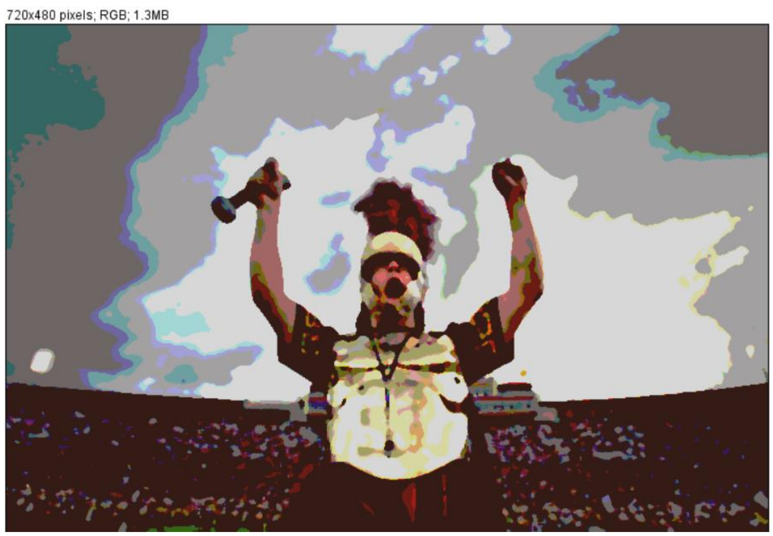


Figure 30: Oil painted Trojan Image – 64 colors – N = 5

Figure 31: Oil painted Trojan Image – 64 colors – N = 7

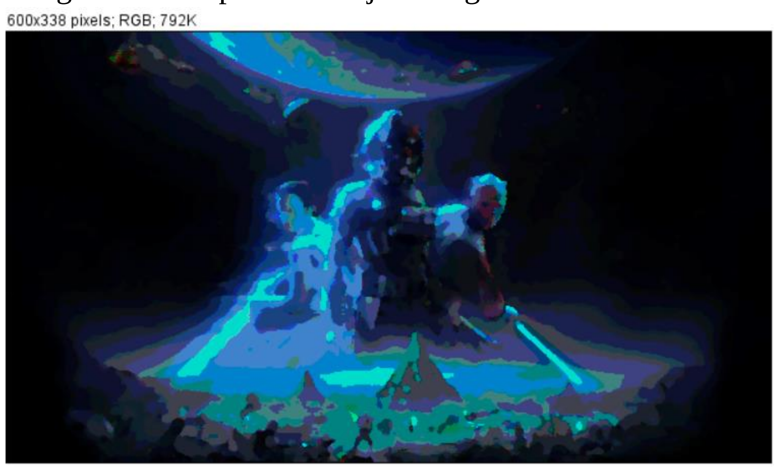


Figure 32: Oil painted Star Wars Image – 512 colors – N = 5

Figure 33: Oil painted Trojan Image – 512 colors – N = 5

- **Discussion:** (Note: Choosing the threshold is explained in Approach and Procedure above)
From the above images we can observe that the 64 color space has comparatively lesser color definitions as compared to 512 color space. Smaller window size is observed to give a better result. In this case, 3X3 window size gives better result as compared to 5X5 and 7X7. This can be explained by the fact that smaller window area is replaced by the most frequent occurring color which gives a better color definition and sharpness to the image. 512 colors defines more colors and this is more evident in the Star wars output. Higher color space results in a detailed color output.

c) Image Filtering – Creating Film Special Effect:

- **Motivation:** Given the problem statement and example outputs, the goal is to achieve a similar film effect by developing an algorithm. This can be said as a reverse contrast enhancement, wherein a stretched image histogram is narrowed down to a range of pixel intensities. By observing the given example, mirroring and color inversion on the input image has to be performed followed by applying reverse contrast enhancement that is inferred by observing the histograms of the given example input and film effect output.

- **Approach and Procedures:** To achieve the film effect, the given input image is mirrored, i.e, the location of the pixel values are reversed for each channel. Color inversion is performed on this image by computing 255-P, where P is the pixel value on each color channel. Film effect is realized on this color inverted image by mapping the pixel values from 0-250 to 75-250 for R channel, mapping the pixel values from 0-250 to 25-200 for G channel and mapping the pixel values from 0-250 to 25-175 for B channel. This is done by calculating new pixel values using the formula,

$$New\ pixel\ value = \frac{p}{255}(Max - Min) + Min$$

Where, p is the pixel value in the color inverted image for each channel, Max is the end pixel value of the new range and Min is the start pixel value of the new range.

- **Results:** The mirrored image, color inverted image and the Final Film effect image are as shown below.

Figure 34: Mirrored Girl Image

Figure 35: Color Inverted Girl Image

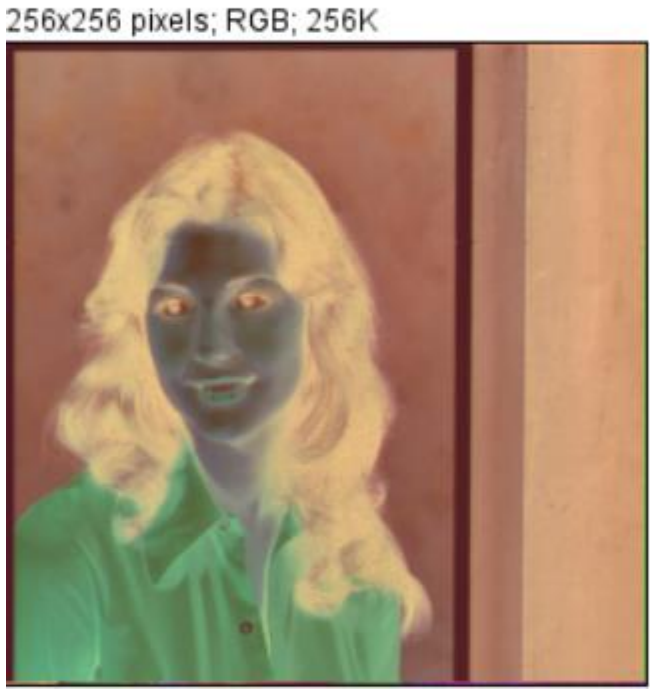


Figure 36: Final Film effect Girl Image

- **Discussion:** The film effect image has Red intensities higher than the Blue and Green intensities. This is because the range of the R channel pixels range between 75 and 250 in the film effect image. The Blue and green channel intensities are distributed almost equally.

PROBLEM 3: NOISE REMOVAL

a) Mix noise in color image:

- **Motivation:** Noise in an image acts like an outlier which deteriorates the image quality. The signals consist of random variations of magnitude of pixel intensities. The 3 major noise in the images are Salt, Pepper and Gaussian noise. In this problem, we explore various noise removal techniques such as the Median filter, Mean filter, Principal component analysis and BM3D filter for comparative study. Filtering is performed on every channel of the RGB image. The filters for each channel are decided based on the type of noise in the respective channels. The performance of the filters is evaluated using the Peak Signal to Noise Ratio (PSNR). Higher the PSNR, better is the performance of the filter.

- **Approach and Procedures:** The input image is padded with (N-1)/2 layers of pixel intensity value equal to 300, where N is the window size. The window is traversed through the Noisy padded image, ignoring the padding and the pixel values are sent to a function to return the median value of the pixel values passed. All the pixels in the window are replaced by this median value. Similarly, the mean filter is applied to the image.

- **Results:** The best output obtained for Median + Mean filter with window size (5X5) is as shown below. The R, G, B histogram outputs after applying this filter are also shown below.

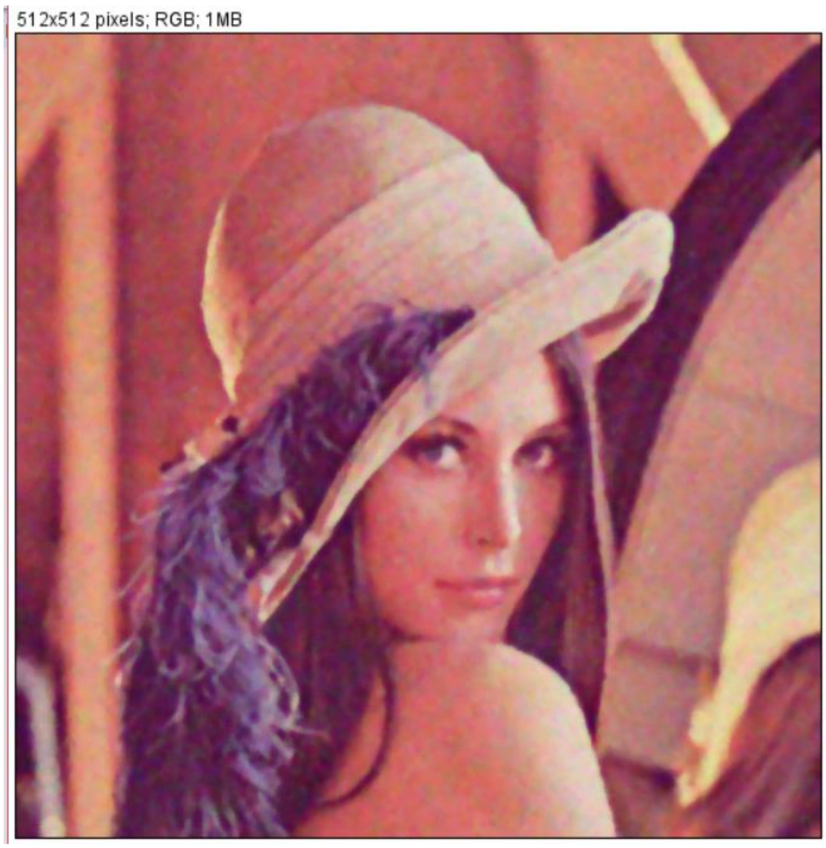


Figure 37: Median + Mean output (5X5)

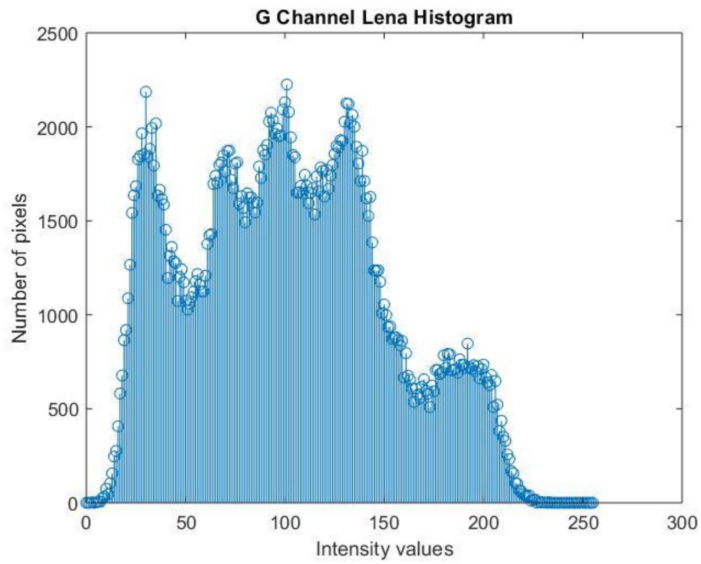


Figure 38: Histogram of the denoised image – G Channel

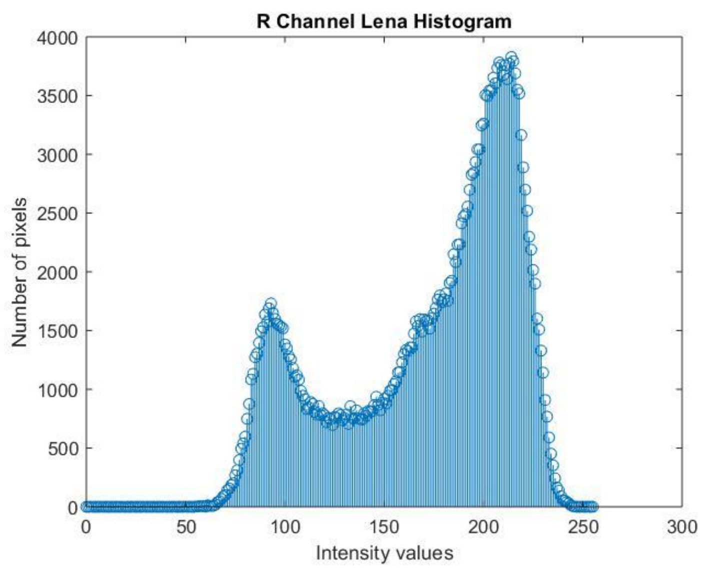


Figure 39: Histogram of the denoised image – R Channel

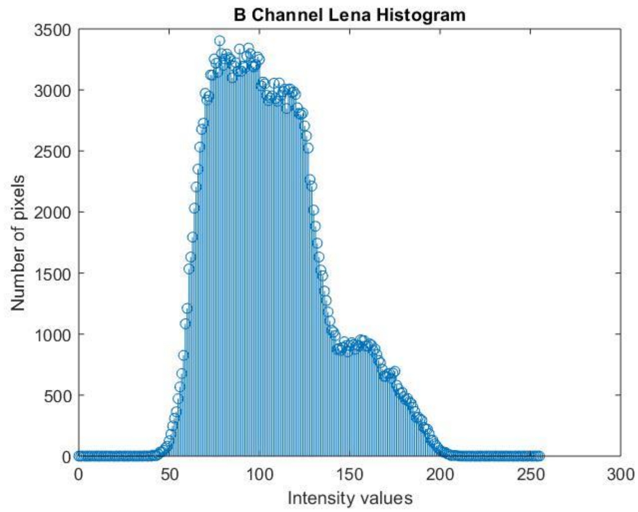


Figure 40: Histogram of the denoised image – B Channel

Filter and window size	PSNR (dB)
No Filter	16.8828
Median(3X3)	24.8599
Median(5X5)	26.0841
Median(7X7)	26.0247
Median(5X5) + Mean(3X3)	26.3151
Median(5X5) + Mean(5X5)	26.3189
Median(5X5) + Mean(7X7)	22.2962

- **Discussion:**

On observing the R, G, B channel images of Noisy Lena and their corresponding histograms as shown below, we can infer that all the channels have salt and pepper noise. Salt noise is more evident in R channel as shown. Also, R and B channels have Gaussian noise as observed from the histograms. Based on this observation, median filter is applied on all channels and mean filter is applied on R and B channels with different window sizes. From the above observation in the Results section, the effect of combining different filters and their respective window sizes are shown. The best result is obtained when Median is applied first followed by Mean filter with window size of 5 for both filters. The Median filter removes the outlier noise such as the salt and pepper noise. This makes it suitable for Mean filter as the image will be devoid of outlier noises. Hence applying Mean filter after Median works best. Mean filter removes the Gaussian noise, thus improving the PSNR. We can observe that, as the filter size is increased from 3 to 5 for Median filter, PSNR also increases. But when the filter size is increased to 7, the PSNR decreases. Hence, the optimum window size is 5 for Median filter. Using this filter, we apply Mean filter with window sizes 3, 5 and 7. Best result is obtained for Mean filter with window size 5. In the process of applying filters to remove noise, the sharpness of the image is lost and the contrast of the image is reduced. The mean filter causes a blurring effect on the image. This could be avoided by using overlapping windows for noise removal. Filters like the Non-local mean, PCA can also be used to enhance the performance.

Figure 41: R, G, B channel outputs

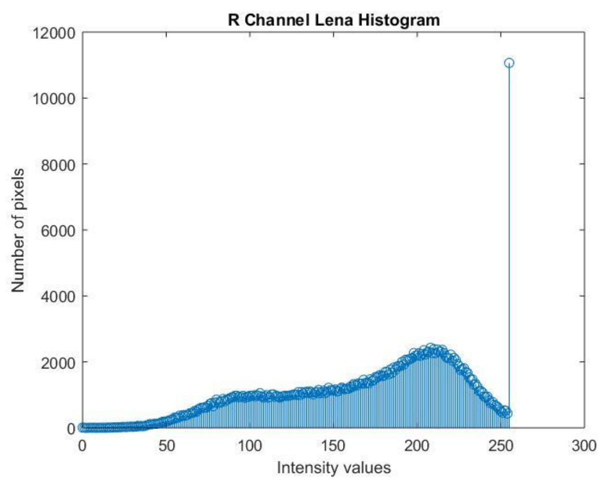


Figure 42: Noisy Image R channel Histogram

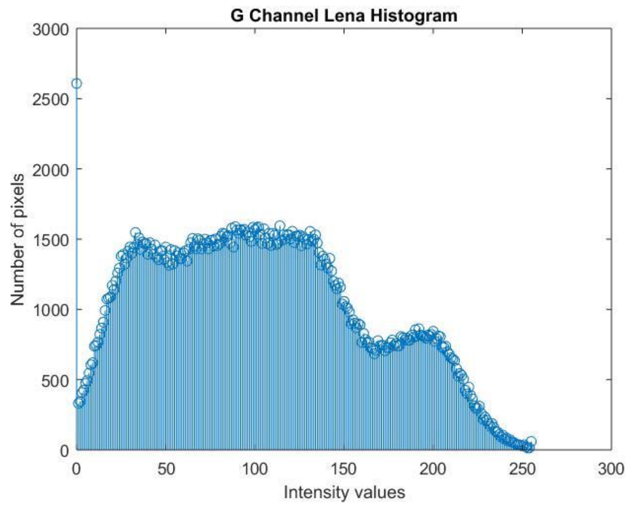


Figure 43: Noisy Image G channel Histogram

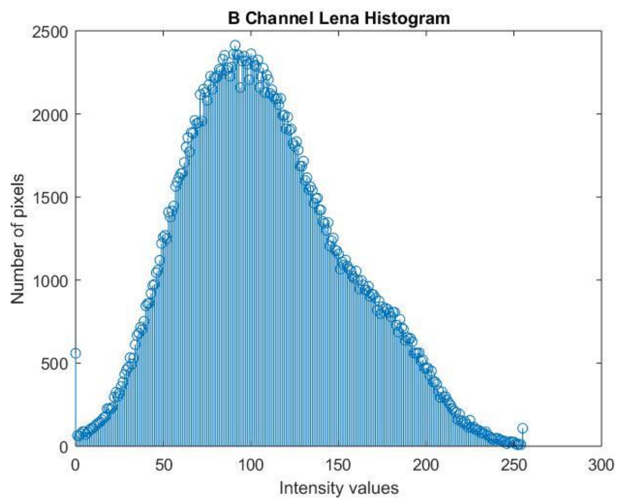


Figure 44: Noisy Image B channel Histogram

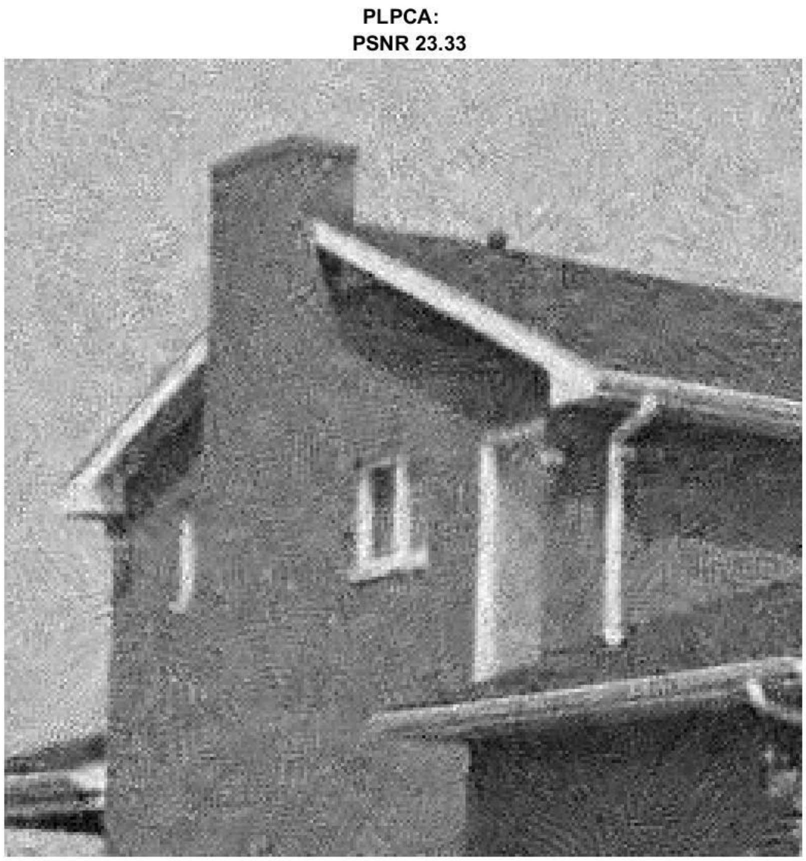
b) Principal Component Analysis (PCA):

- **Approach and Procedures:**

Reference: http://josephsalmon.eu/code/index_codes.php?page=GPPCA

PCA is a dimensionality reduction algorithm. This can be used for denoising as it can be applied to noisy image and then inverse PCA is applied to retrieve the image. The components are the orthogonal basis learnt from the noisy images. The patches are extracted from the noisy image and the patches are clustered. The local features are defined by these clusters and help in retrieving denoised image.

- **Results:** The PCA result for House_noisy image with sigma = 25 is as shown below.

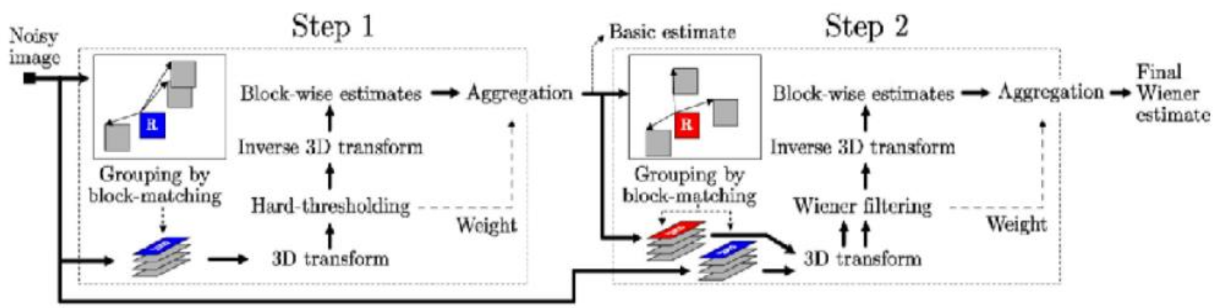


Sigma	Delta/ HW	WP	Threshold	PLPCA PSNR (db)	Noisy Image PSNR (dB)
5	8	7	12.5	39.48	34.17
10	10	7	25	35.78	28.15
20	11	7	55	32.37	22.13
40	13	12	110	28.42	16.11
25	11	7	55	23.33	22.13

- **Discussion:** From the above results, we can observe that as the Gaussian noise is added, the PSNR value also decreases.

c) Block Matching and 3-D transform filter:

- **Approach and Procedures:**
  Reference: http://www.cs.tut.fi/~foi/GCF-BM3D/



BM3D Algorithm (Source: Discussion slides)

BM3D is a state of the art method for image denoising. This is achieved by stacking similar blocks or blocks that have similar neighbors. The stacking of these 2D blocks results in 3D arrays called groups. Spatial and frequency filtering is applied on these 3D blocks. The noise is reduced and this collaborative filtering helps to identify the finest details shared by the blocks and preserves its own features. These blocks are later returned to their locations in the original image, where aggregation is performed using Weiner filter. This procedure avoids the redundant features and preserves the important ones.

- **Results:** The denoised image for Noisy house image (sigma = 25) is as shown below. The results are tabulated for BM3D algorithm with and without weiner filter or step 2. Different transforms for Weiner are used and the results are tabulated. Comparison between PCA and BM3D is also provided below.

BM3D Denoised Image

Weiner filter	PSNR Noisy image(dB), sigma = 25	PSNR Denoised image(dB)	Transform used for Weiner Filter
With Weiner Filter (Step 2)	20.145	32.865	DCT
Without Weiner Filter (Step 2)	20.145	32.271	DCT

PSNR Noisy image(dB), sigma = 25	PSNR Denoised image(dB)	Transform used for Weiner Filter
20.145	32.865	DCT
20.145	32.733	DST
20.145	32.818	Hadamard

PSNR Noisy image(dB), sigma = 25	PSNR Denoised image(dB)
20.145	32.865
22.13	23.33

- **Discussion:** BM3D is a collaborative filtering method, ie, it is both spatial domain and frequency domain filter. The filters share the details between the grouped blocks and also preserve the distinct features of each and every individual block.
  The above results show the performance for different transforms on Weiner filter. The Discrete Cosine Transform is proved to be performing better than the Discrete Sine Transform and the Hadamard Transformations. The BM3D also proves to perform better than PCA, thus declaring itself as the state of the art algorithm.