CS 663 - Assignment 2

Shaan Ul Haque
180070053

Samarth Singh
180050090

Niraj Mahajan
180050069

September 29, 2020

Question 1

Image Sharpening using Unsharp Masking

(1) Sharpening on lionCrop.mat

Parameters:

- Gaussian Size = 4
- Gaussian Sigma = 2
- Scale = 10

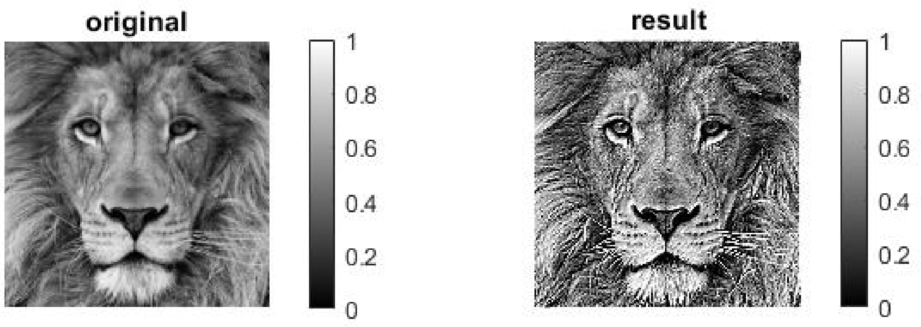


Figure 1.1: Sharpening on lionCrop.mat

(2) Sharpeing on superMoonCrop.mat

<u>Parameters</u>:

- Gaussian Size = 4
- Gaussian Sigma = 2
- Scale = 10

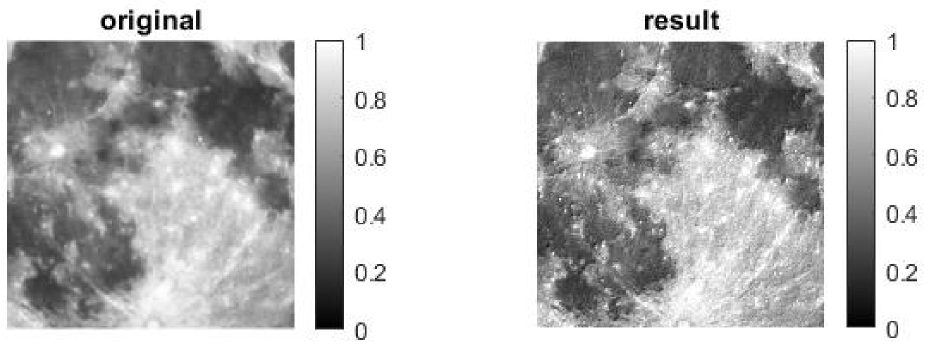


Figure 1.2: Sharpening on superMoonCrop.mat

Shaan Ul Haque(180070053)
Samarth Singh (180050090)
Niraj Mahajan (180050069)

Question 2

Barbara Image

We use a window of size 7*7. Standard deviation for spatial filter(σ_s) is 1.5 while Standard deviation for intensity filter(σ_i) is 9.75. RMSD obtained was 3.2921.

$$RMSD(0.9 * \sigma_s, \sigma_i) = 3.2950$$

$$RMSD(1.1 * \sigma_s, \sigma_i) = 3.2949$$

$$RMSD(\sigma_s, 0.9 * \sigma_i) = 3.3192$$

$$RMSD(\sigma_s, 1.1 * \sigma_i) = 3.3029$$

We observe that the parameters obtained produce the least RMSD. Original, Noisy and Filtered image are shown below along with the spatial Gaussian filter. We used color jet to represent it to get better visualization of the filter.

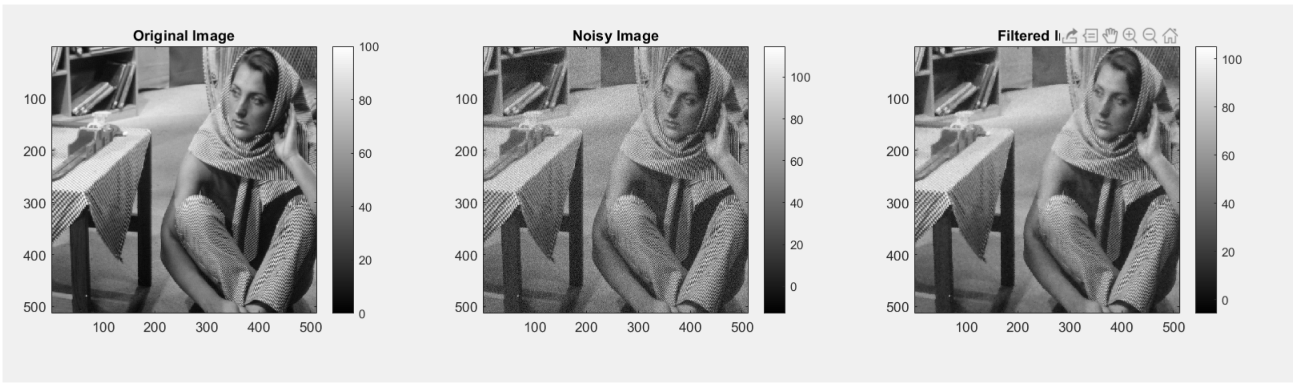


Figure 1: Barbara Images

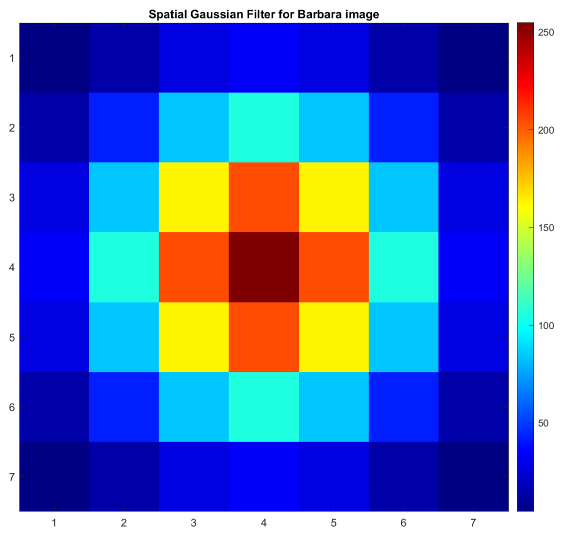


Figure 2: Spatial Gaussian Filter

Shaan Ul Haque(180070053)
Samarth Singh (180050090)
Niraj Mahajan (180050069)

Grass Image

We use a window of size 3*3. Standard deviation for spatial filter(σ_s) is 0.95 while Standard deviation for intensity filter(σ_i) is 42. RMSD obtained was 7.5410.

$$RMSD(0.9 * \sigma_s, \sigma_i) = 7.5593$$

$$RMSD(1.1 * \sigma_s, \sigma_i) = 7.5499$$

$$RMSD(\sigma_s, 0.9 * \sigma_i) = 7.5641$$

$$RMSD(\sigma_s, 1.1 * \sigma_i) = 7.5573$$

We observe that the parameters obtained produce the least RMSD. Original, Noisy and Filtered image are shown below along with the spatial Gaussian filter. We used color jet to represent it to get better visualization of the filter.

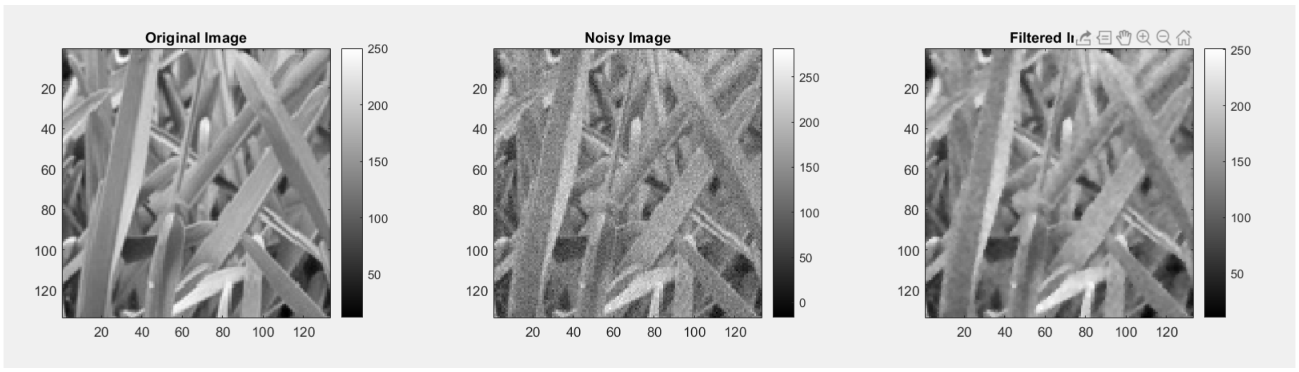


Figure 3: Grass Images

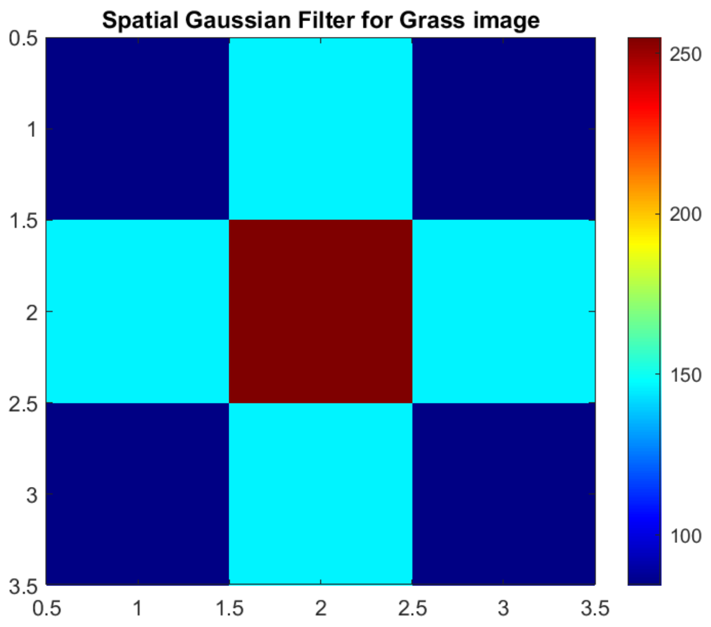


Figure 4: Spatial Gaussian Filter

Honey Comb Image

We use a window of size 3*3. Standard deviation for spatial filter(σ_s) is 1.25 while Standard deviation for intensity filter(σ_i) is 40. RMSD obtained was 7.3021.

$$RMSD(0.9 * \sigma_s, \sigma_i) = 7.3077$$

$$RMSD(1.1 * \sigma_s, \sigma_i) = 7.3047$$

$$RMSD(\sigma_s, 0.9 * \sigma_i) = 7.3246$$

$$RMSD(\sigma_s, 1.1 * \sigma_i) = 7.3299$$

We observe that the parameters obtained produce the least RMSD. Original, Noisy and Filtered image are shown below along with the spatial Gaussian filter. We used color jet to represent it to get better visualization of the filter.

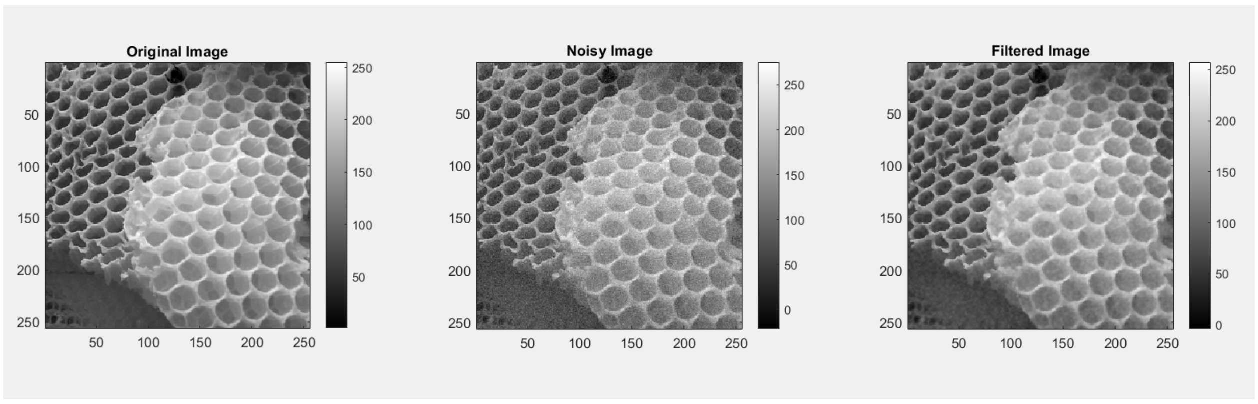


Figure 5: Grass Images

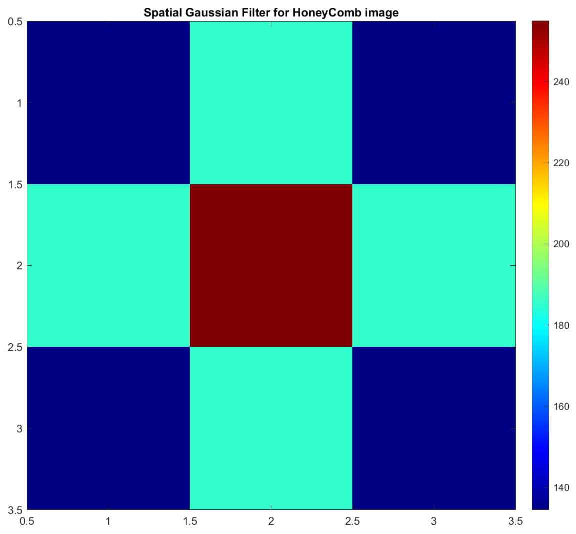


Figure 6: Spatial Gaussian Filter

Code Usage

The main file myMainScript.m is divided into 4 parts.

- Part A (Barbara Image)- This part processes the filtered image for the noisy Barbara Image. We have used a seed to have a constant RMSD every time we run the code. To generate the filtered image the code uses the function myBilateralFiltering which takes the pixel p, window around that pixel, σ_s(spatial standard deviation) and σ_i(intensity standard deviation). It returns the filtered intensity at that point.
- Part B (Grass Image)- This part processes the filtered image for the noisy Grass Image. We have used a seed to have a constant RMSD every time we run the code. To generate the filtered image the code uses the function myBilateralFiltering which takes the pixel p, window around that pixel, σ_s(spatial standard deviation) and σ_i(intensity standard deviation). It returns the filtered intensity at that point.
- Part C (Honey Comb Image)- This part processes the filtered image for the noisy Honey Comb Image. We have used a seed to have a constant RMSD every time we run the code. To generate the filtered image the code uses the function myBilateralFiltering which takes the pixel p, window around that pixel, σ_s(spatial standard deviation) and σ_i(intensity standard deviation). It returns the filtered intensity at that point.
- Part D (Spatial Filters)- This part generates the spatial Gaussian filters as an image for visualization. We use colorjet to have better visualization.

Question 3

3.1 Isotropic Gaussian Mask

We used a gaussian mask of $\sigma = 1.5$ to make the patches isotropic.

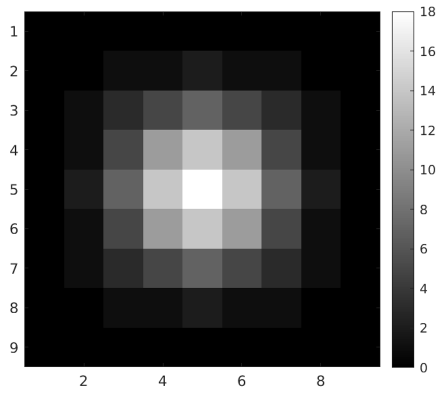
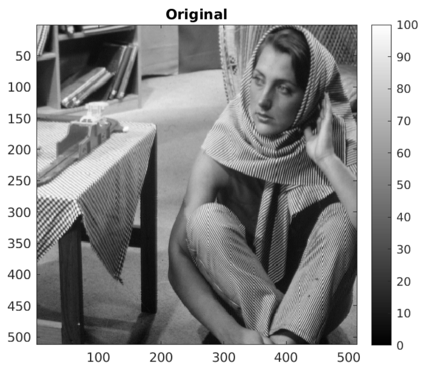


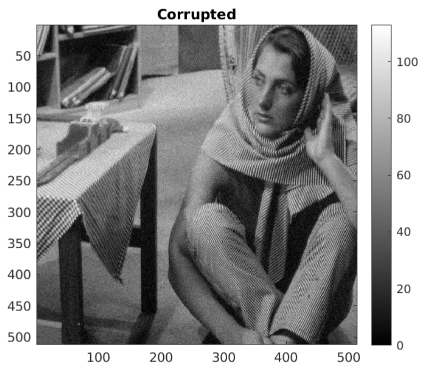
Figure 3.1: Gaussian Mask

3.2 Patch Based Filtering on barbara.png

The optimal filtering for barbara.png was attained at $\sigma_{barbara} = 0.8424$.
The RMSD values are:

- $\text{RMSD}_{\sigma} = 2.614193$
- $\text{RMSD}_{0.9\sigma} = 2.636064$
- $\text{RMSD}_{1.1\sigma} = 2.669242$





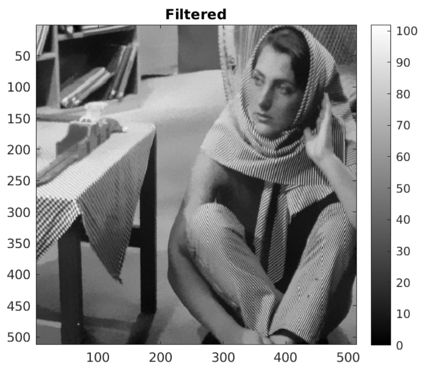


Figure 3.2(a): Original **Figure 3.2(b):** Corrupted **Figure 3.2(c):** Filtered

Shaan Ul Haque(180070053)
Samarth Singh (180050090)
Niraj Mahajan (180050069)

3.3 Patch Based Filtering on grass.png

The optimal filtering for barbara.png was attained at $\sigma_{grass} = 1.8527$.
The RMSD values are:

- $\text{RMSD}_{\sigma} = 7.336307$
- $\text{RMSD}_{0.9\sigma} = 7.471906$
- $\text{RMSD}_{1.1\sigma} = 7.474340$

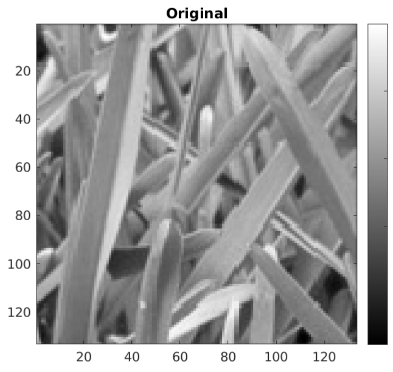


Figure 3.3(a): Original

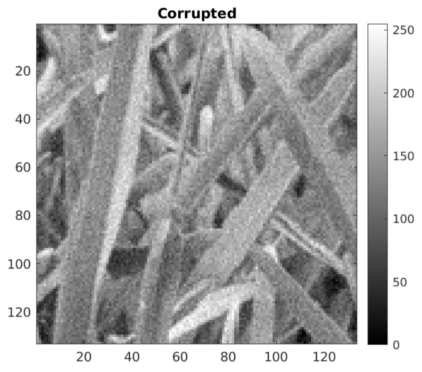


Figure 3.3(b): Corrupted

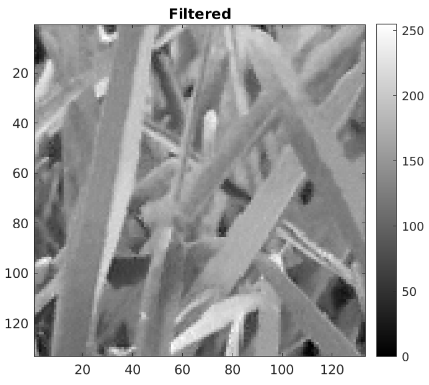


Figure 3.3(c): Filtered

3.4 Patch Based Filtering on beehive.png

The optimal filtering for barbara.png was attained at $\sigma_{beehive} = 2.0842$.
The RMSD values are:

- $\text{RMSD}_{\sigma} = 7.424229$
- $\text{RMSD}_{0.9\sigma} = 7.545112$
- $\text{RMSD}_{1.1\sigma} = 7.557518$

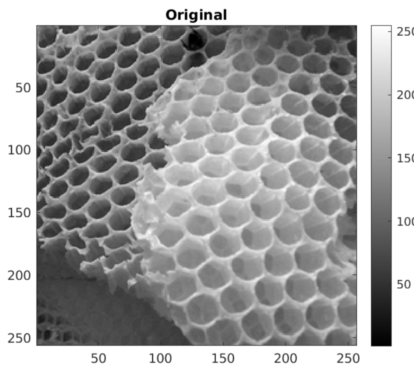


Figure 3.4(a): Original

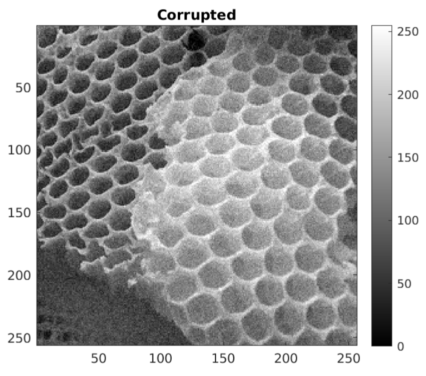


Figure 3.4(b): Corrupted

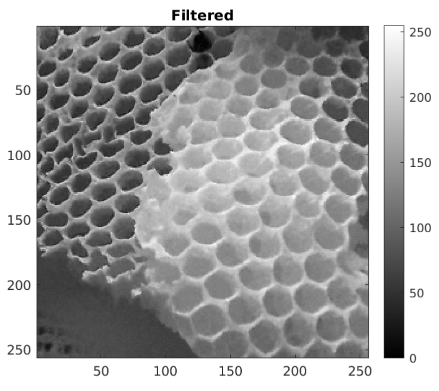


Figure 3.4(c): Filtered

3.4 Usage of code

- Running the **myMainScript.m** file will plot the original, corrupted and filtered images for all three test cases. This takes less than 5 minutes and hence wait bar is not shown.

- By default, the 0.9σ and 1.1σ RMSD values will not be printed. In order to print these values, uncomment line 33 to 38 in **workOnImage.m**. Note that this will take 3 times the time required to run just for the optimal values and will exceed 5 minutes.

- I have not scaled each image to [0,1] but simply converted the original pixel values to double and applied patch based filtering. This method is equivalent to scaling the image to [0,1], applying patch based filtering, and then rescaling to plot it. The quality of the filtered image will be the same. Just the σ values and RMSD values will be scaled appropriately.
  TL;DR: The RMSD values obtained by us will appear higher in magnitude as compared to those who have scaled the image to [0,1] but on scaling, the RMSD values are in the same range [let's hope better :)], and the filtered image quality is also the same [again, let's hope better :)]