UNIVERSITY OF TEXAS ARLINGTON

Assignment 1

Student name: *Avinash Shanker, 1001668570*

Due date: *Sep 29, 2019*

Problem 1

- Compare the downsampled images with the original in the same actual size. How do they look?
- Perform upsampling of the image by inserting an empty pixel between every and each pixel. Run it twice to get it back to the original size. How are the upsampled images looking?

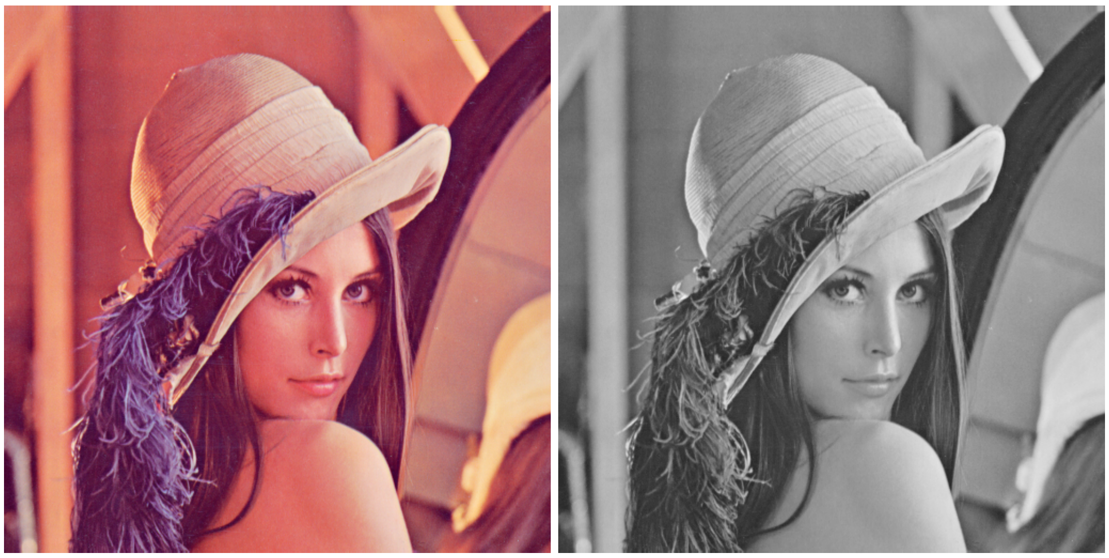
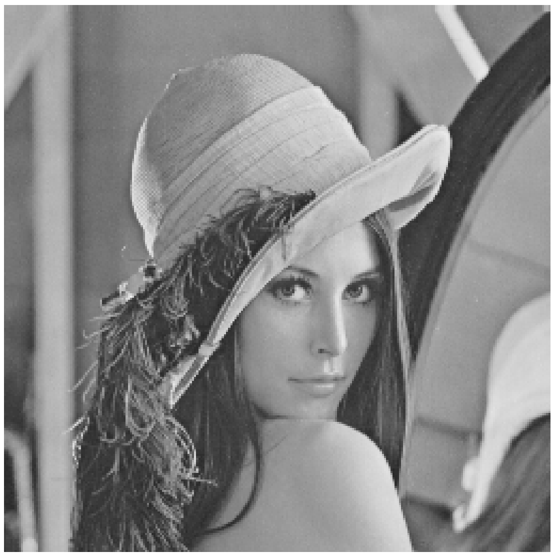

(a) Original (b) Gray Scale

Figure 1: Comparing original vs gray scale with intensity[0,1]


(a) First down sample 256x256

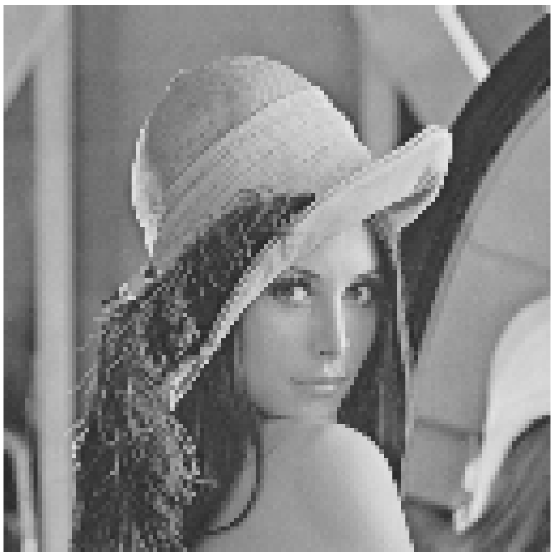
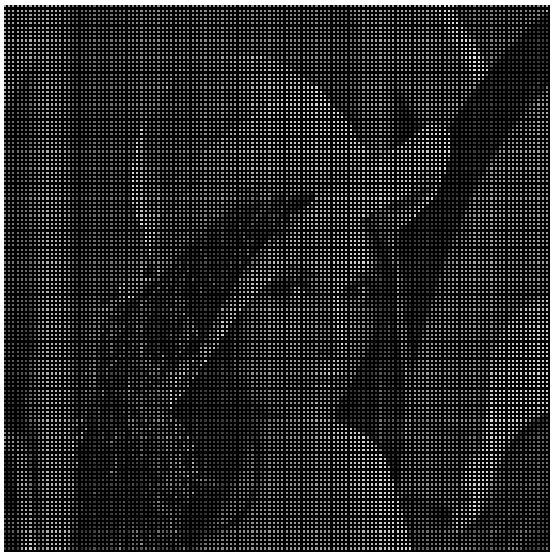

(b) First down sample 128x128

Figure 2: Down sampling the 512x512 pixel image twice


(a) Upsampled to 256x256

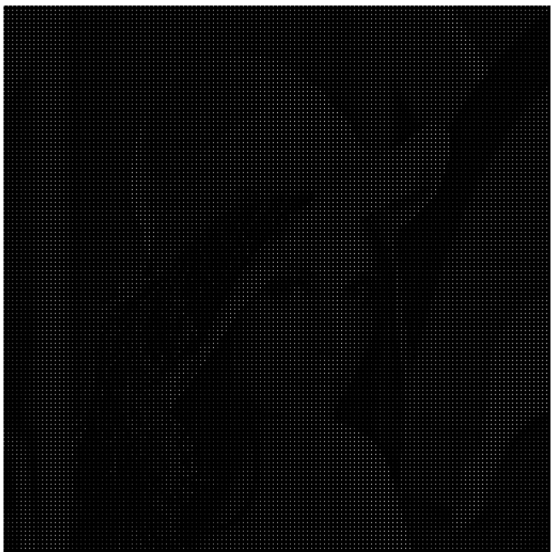

(b) Upsampled to 512x512

Figure 3: Comparing original vs gray scale with intensity[0,1]

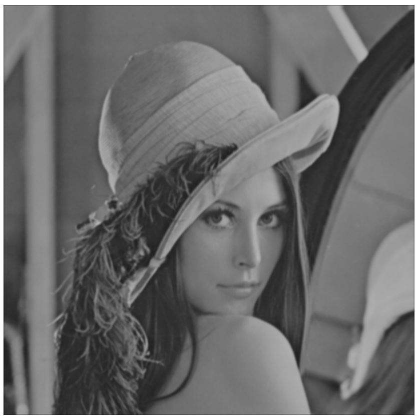
Answer.

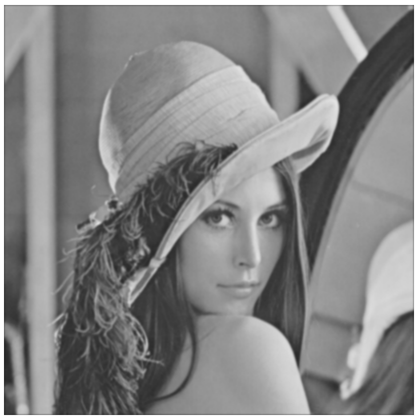
- In Fig 1, we are comparing original **RGB** Lena image with gray scale version. The intensity of gray scale image fig 1.b is in range [0,1].
- When we downsample the image, pixel size for low resolution images is bigger. Throw away every other row and column to create a 1/2 size image. The original image reduces from 512x512pixel to 256x256 as show in Fig 2.a by only keeping every alternative pixel. We can notice that the image quality has gone down.
- In Fig 2.b when comparing second downsample of 128 pixel size with original image. The image has become more blur and the quality has degraded.

- In Fig 3.a after upsampling image from 128 to 256 pixel, by adding empty samples, the image has aquired large pixels.
- In Fig 3.b when upsampling again from 256 to 512pixel image. Image has become very blury, with high amount of empty pixels.

Problem 2

- Implement a function I smooth = myGaussianSmoothing(I, k, s) and test it on the given image lena.png.
- Change kernel size to k = 3, 5, 7, 11, 51 with fixed s = 1. What changes in the result do you see?
- Change the s = 0.1, 1, 2, 3, 5 with fixed kernel size k = 11. How is the result changing?


(a) k=3, s=1


(b) k=5, s=1

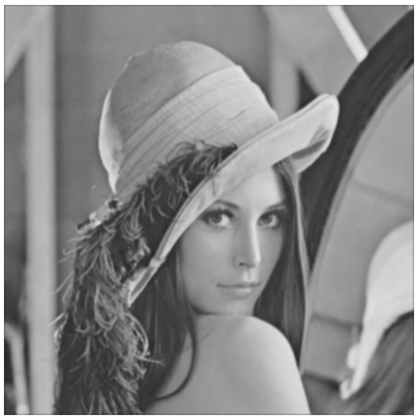
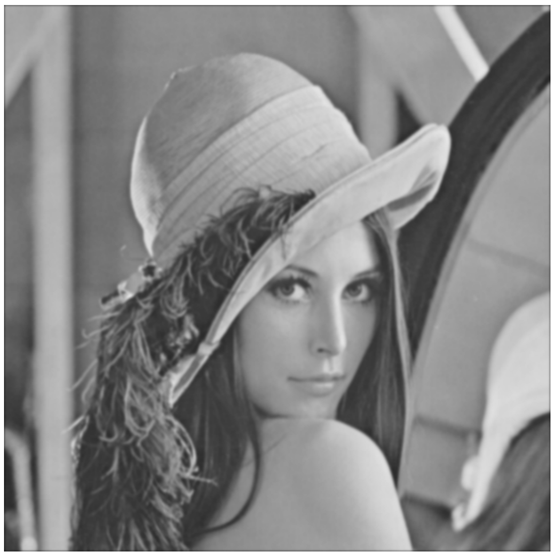

(c) k=7, s=1

Figure 4: Changing kernel size to k = 3, 5, 7 with fixed s = 1


(a) k=11, s=1

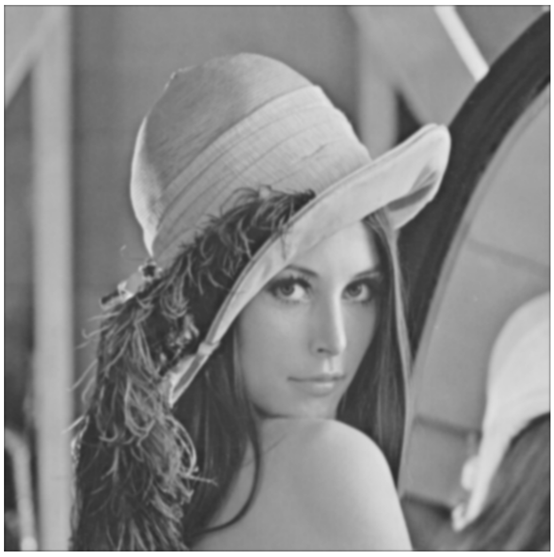

(b) k=51, s=1

Figure 5: Changing kernel size to k = 11, 51 with fixed s = 1

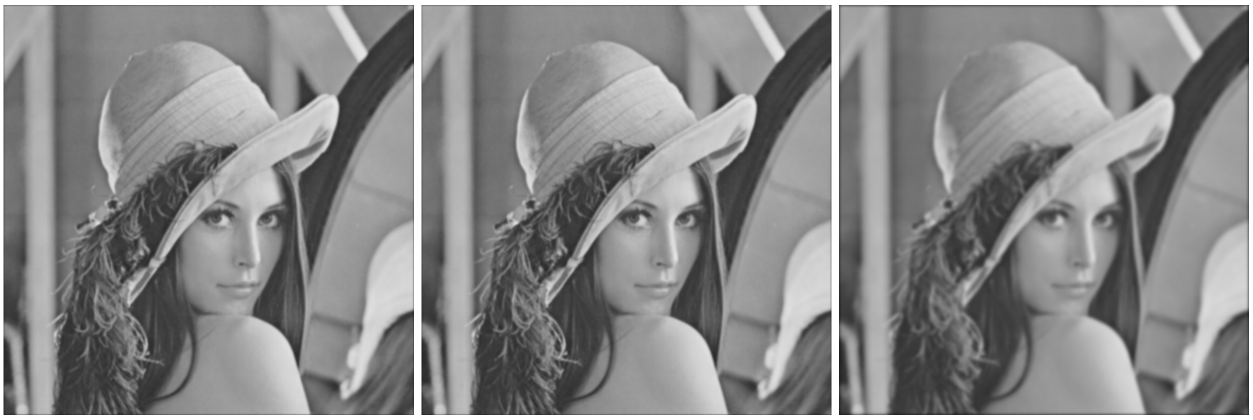

(a) s=0.1, k=11 (b) s=1, k=11 (c) s=2, k=11

Figure 6: Change the s = 0.1, 1, 2 with fixed kernel size k = 11

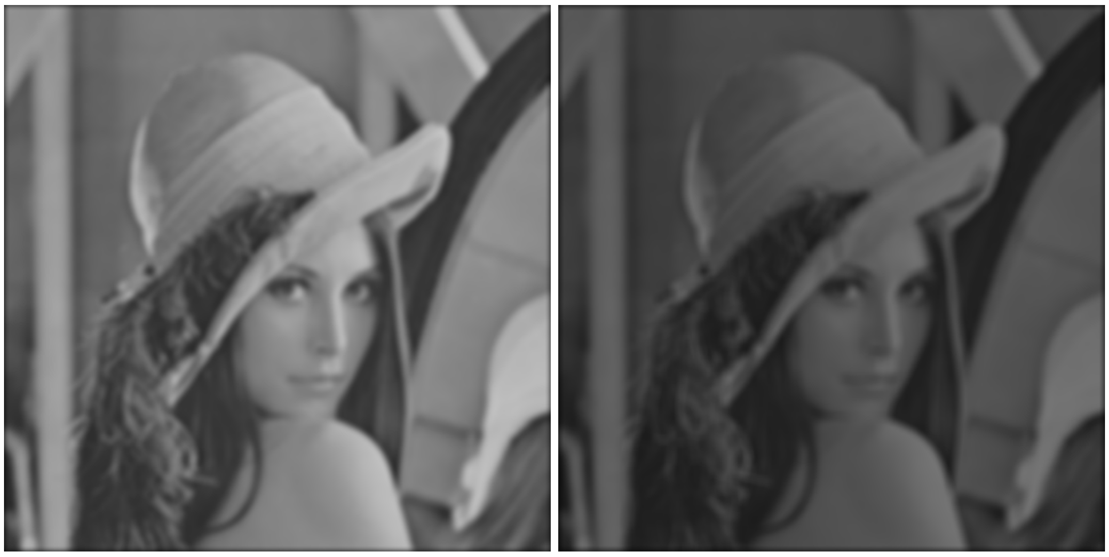

(a) s=3, k=11 (b) s=5, k=11

Figure 7: Change the s = 3, 5 with fixed kernel size k = 11

Answer.

- After creating a function myGaussianSmoothing(I, k, s) where I is image, K is kernel size and S in sigma value. A Gaussian kernel gives less weight to pixels further from the center of the window
- The size of the mask drives the filter amount. A larger size, corresponding to a larger convolution mask, will generally result in a greater degree of filtering.
- Fig 4 and Fig 5 depicts the change of kernel size with constant sigma as 1, shows very less visible change.
- Fig 6 and Fig 7 depicts Sigma change from s = 0.1, 1, 2, 3, 5 with k=11
- We can see that the image gets **more blur** as the value of sigma increases
- **Smoothing with larger standard deviations suppresses noise, but also blurs the image**

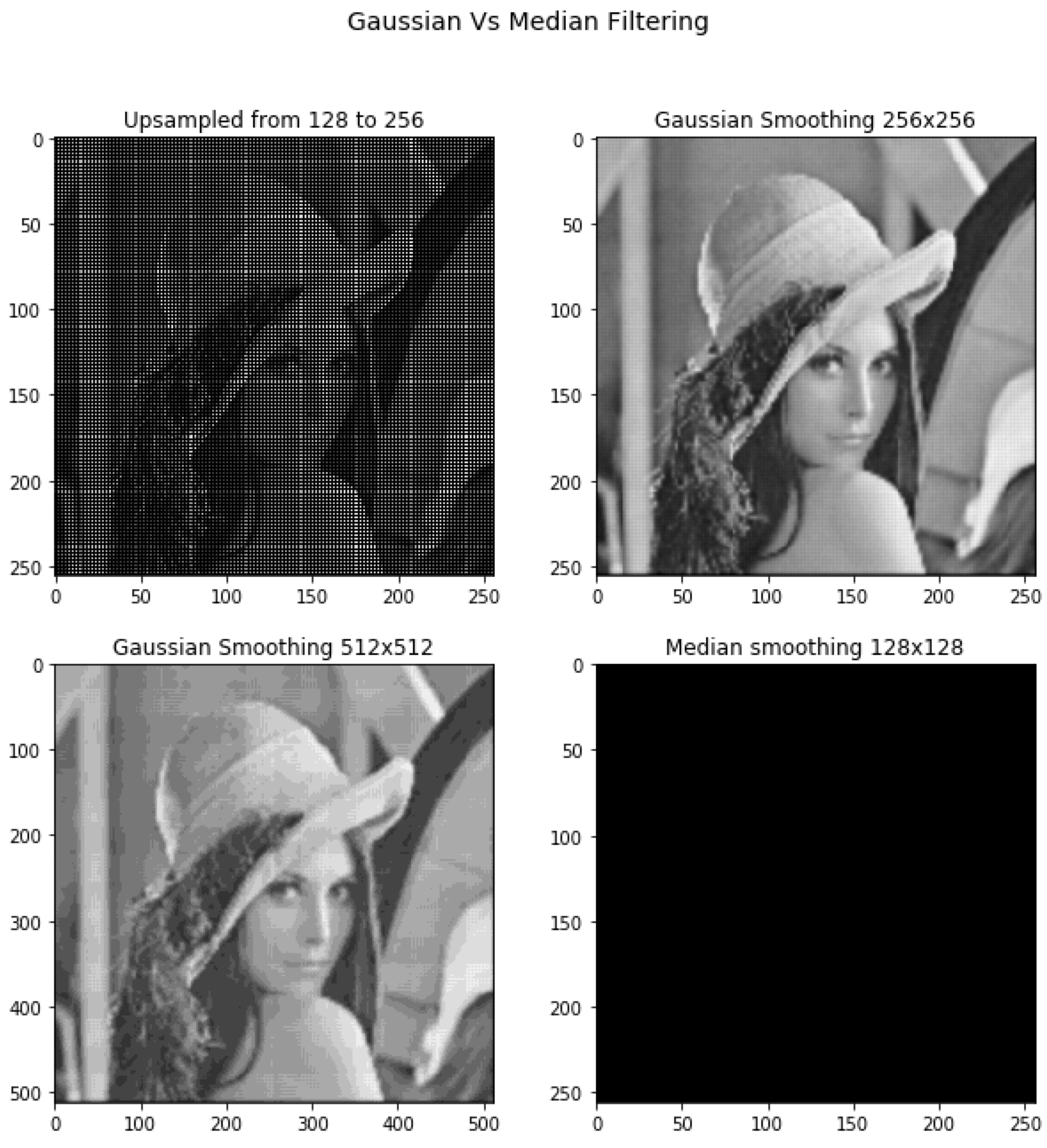
- More smoothing is achieved with higher sigma. But, there is a trade off, higher value means more blurring.

Problem 3

- Everytime after performing upsampling, perform Gaussian smoothing with k=11, s=1. How does it change?
- Implement and see happens when we perform median filtering?
- Which filtering is better?

Answer.

- After downsampling the 512x512 image twice to 128pixel size, we again upsample the image to 256pi by adding empty pixel.
- Fig 8.a Post upsampling the image to 256pi, we perform Gaussian smoothing. In Fig 8.b is the image after gaussian smoothing. Now the empty pixel have been smoothed. A much better image is obtained.
- Fig 8.c we have upscaled the 256pi gaussian smoothed image and again perfomed gaussian smoothing. Though the image is blur, it highly better than Fig 3.b which is upsampled twice by adding empty pixels.
- In Fig 8.d, when we perform median filtering on 256pi upsampled image. We obtain a blank empty image.
- Median smoothing is not working well with image upsampled with empty pixels.
- Conclusion, is that Gaussian filtering performs better when there are empty pixels. Median smoothing performance is poor in this case.


Gaussian Vs Median Filtering
Upsampled from 128 to 256
Gaussian Smoothing 256x256
Gaussian Smoothing 512x512
Median smoothing 128x128

(a) Look at image as a, b, c, d from left to right

Figure 8: Change the s = 3, 5 with fixed kernel size k = 11

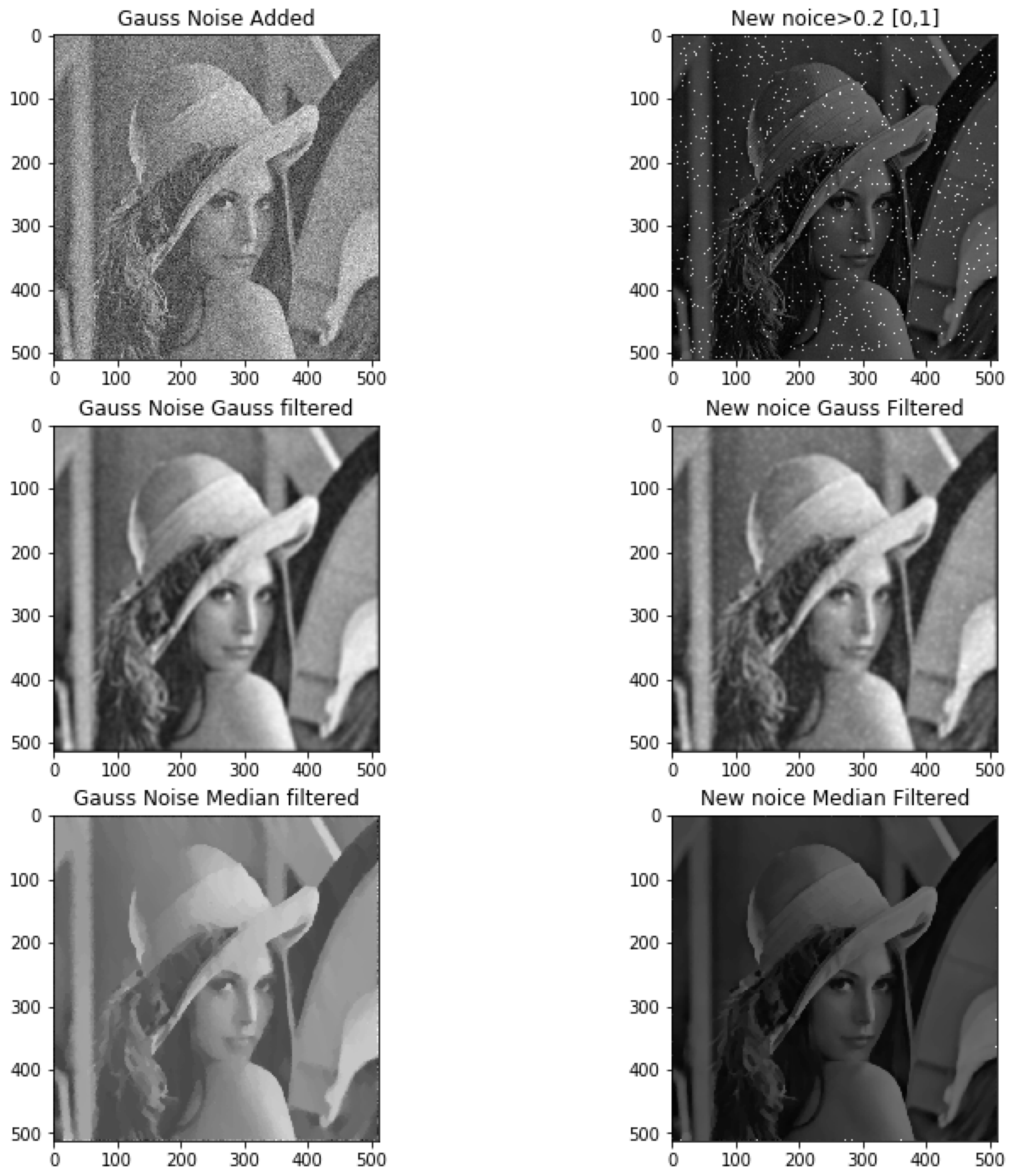
Problem 4

- Add some Gaussian noise N(0, 0.1) at each pixel. How is the image looking?
- Perform Gaussian smoothing on noisy image. How does it look?
- Now, apply median filtering and see how the result shows. How it looks?
- Now change the type of noise to value 1 everywhere where the noise was > 0.2 and 0 otherwise.
- Apply Gaussian and median smoothing on this new noise.
- Compare the results?

Answer.

- In Fig 9 column 1 represents images added with Gaussian noise N(0,0.1). When Gaussian noise is added to an image looks rough with variations in intensity. Gaussian noise was is statistical noise having a probability density function (PDF) equal to that of the normal distribution.
- Fig 9 column 1 2nd image is obtained after applying gaussian filter. It can noticed that gaussian filter works well with gaussian noise. Noise in the image has been well smoothed by the filter. I have used K=11 and Sigma=3 in my code.
- In Gauss filtering Smoothing with larger standard deviations suppresses noise, but also blurs the image
- Fig 9,3rd image in first column is obtained my using median filter on the Gaussian noise image. Median filtering makes the image very blury with loss of details.
- Median filter does not perform well with Gauss noise.

- Fig 9 column 2, contains images with Noise which looks like salt and pepper i.e. noise is in range of [0,1]. And the output of applying gauss and median filter on salt and pepper noise can be seen.
- Gaussian Filter is not performing well with salt and pepper like noise. Output obtained is not very clear.
- When median filtering is performed on salt and pepper noise it results are better
- In conclusion, Gauss filter works wells with gaussian or random noise. And Median filter works well with salt and pepper noise.

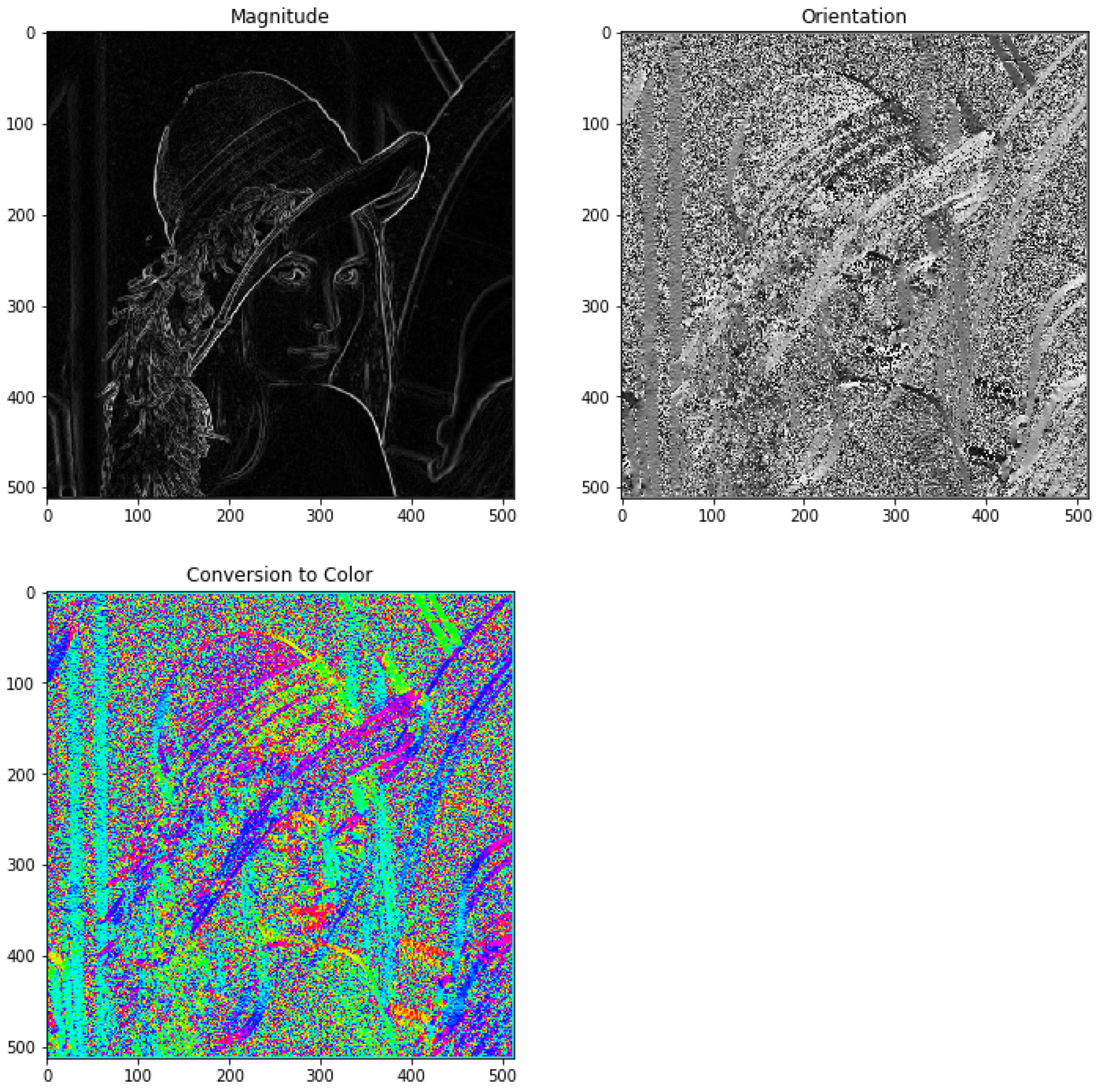

Gauss Noise Added
New noice>0.2 [0,1]
Gauss Noise Gauss filtered
New noice Gauss Filtered
Gauss Noise Median filtered
New noice Median Filtered
0
100
200
300
400
500

(a) Look at image column wise

Figure 9: Apply Gauss and Median filter for 2 types of noise

Problem 5

- Implement a function for sobel filter which should look like [mag, ori] = mySobelFilter(I)
- visualizing the result in color by using the magnitude to specify the saturation and value of an image and the orientation to specify the hue.



(a) Read Image from left to right as a,b,c

Figure 10: Magnitude and Orientation of an image

Answer.

- Sobel filter is particularly used for edge detection which is an approximation of the gradient of the image intensity function
- At each point in the image, the resulting gradient root of gx and gy can be combined to give the gradient magnitude.
- Similarly inv(tan(gy/gx)) is used to compute the orientation information.
- Fig 10 represents magnitude and orientation of the image.
- Fig 10.a Magnitude represents the images edges. Edges are highlighted as white as we normalize the gradients so that all the values lie between [0,1].
- Fig 10.b depicts the Orientation of each pixel with respect to its magnitude.
- Fig 10.c Is the visualization in color using magnitude for saturation and orientation for hue