Assignment 2 – Digital Image Analysis

Sahil Shahare – 2020CS50440

Mojahid Hussain - 2020CH70182

Part1

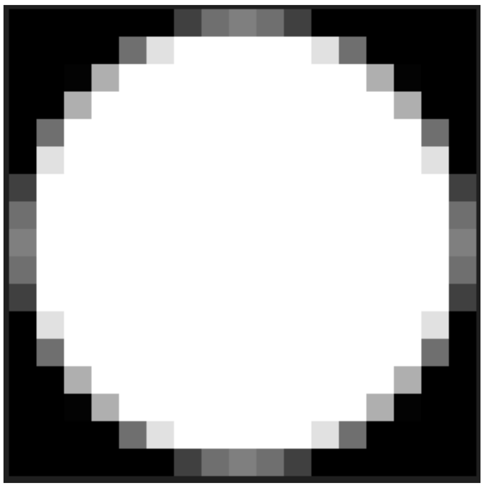
a.

Input Image:

Output Image:

A disk kernel with r = 8 was taken. Hence the size of kernel was 17 x 17. I was convolved with the input image using NumPy library. The image was padded with mirror padding before applying convolution with above kernel.

Kernel:



Kernel with r = 8

b.

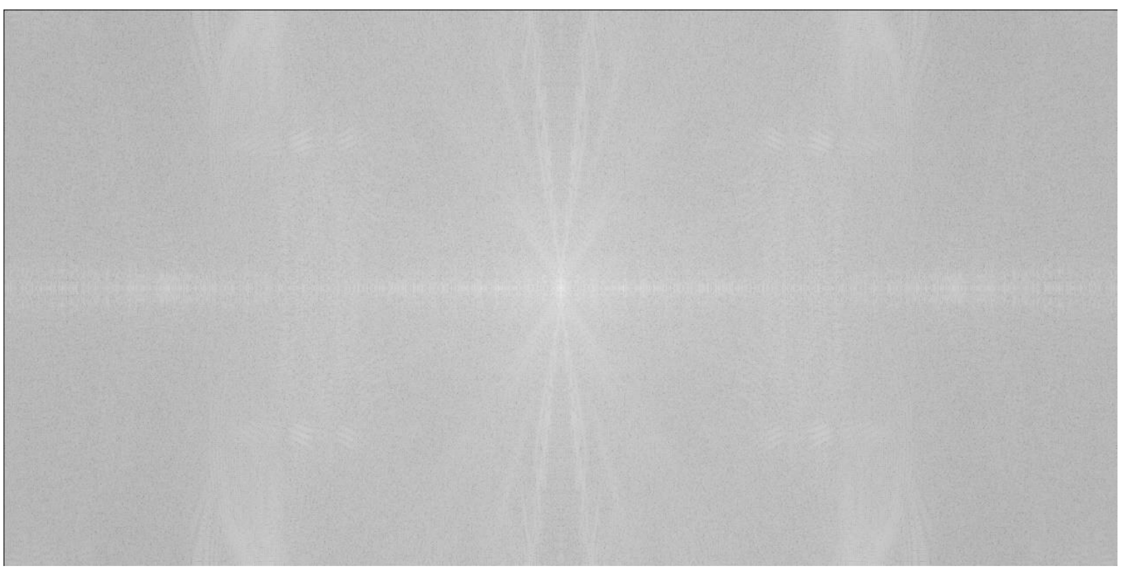
Padded Image:

A 375 x 750 px image was padded by mirror padding to get 750 x 1500 image.

Padded Kernel:

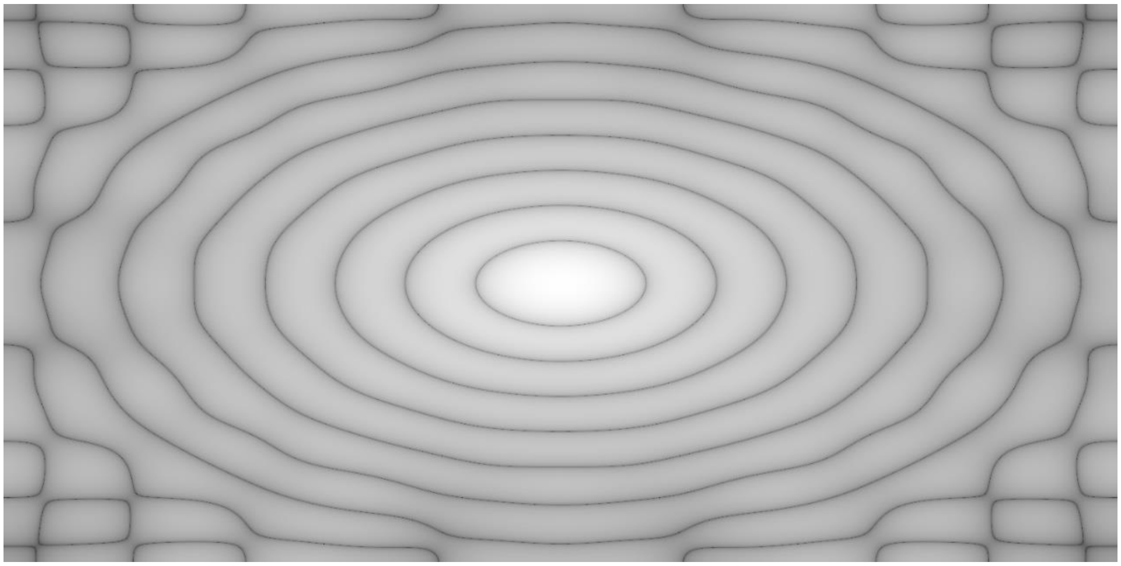
Padded the kernel to make its size 1500 x 750 by padding it with value 0.

DFT of Padded Image (shifted):



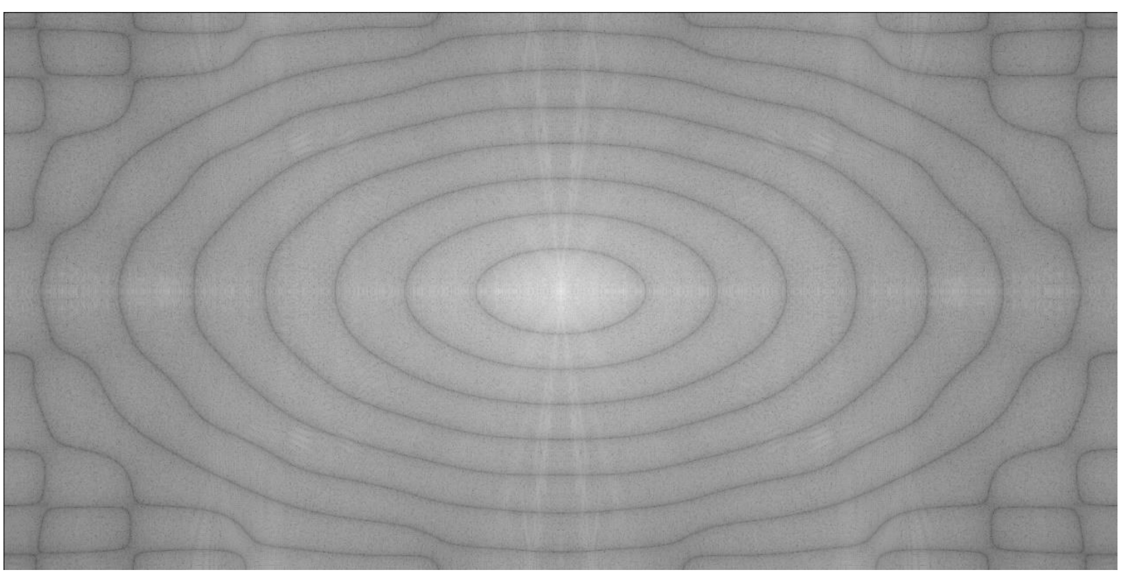
Took DFT of the padded image above using **np.fft.fft2** function. Took log of the DFT result after taking magnitude of complex values and adding a small value of 1e-6 to it. Shifted the values to get zero-indexed values at middle.

DFT of Padded Kernel (Shifted):



Took DFT of the padded kernel above using **np.fft.fft2** function. Took log of the DFT result after taking magnitude of complex values and adding a small value of 1e-6 to it. Shifted the values to get zero-indexed values at middle.

DFT of Multiplication of Image and Kernel in Freq Domain:



Multiplied the DFT of padded image and padded kernel. Took log of the DFT result after taking magnitude of complex values and adding a small value of 1e-6 to it. Shifted the values to get zero-indexed values at middle.

Took inverse DFT of multiplication of DFT of padded image and padded kernel and took its real part only.

Output Image:

Took top-left part of the above image.

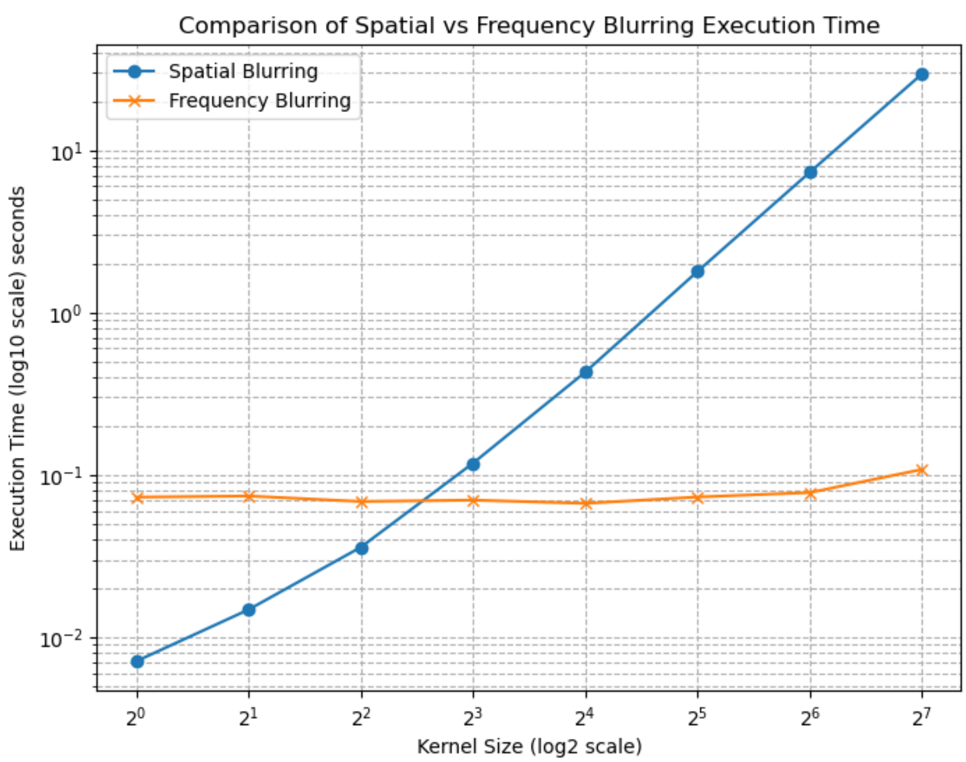
c.

The DFT of the blur kernel h(x,y) appears elliptical, even though h itself is circularly symmetric, because the DFT operates on a discrete grid with different image dimensions M and N.

When M != N (i.e., for a rectangular image), the frequency domain grid is sampled at different rates along the two axes. This leads to a non-uniform scaling of spatial frequencies in the horizontal and vertical directions. As a result, the circular symmetry in the spatial domain is distorted into an elliptical shape in the frequency domain.

Thus, the ellipse arises because the Fourier transform scales differently along the axes due to the differing image dimensions.

d.



Q 2

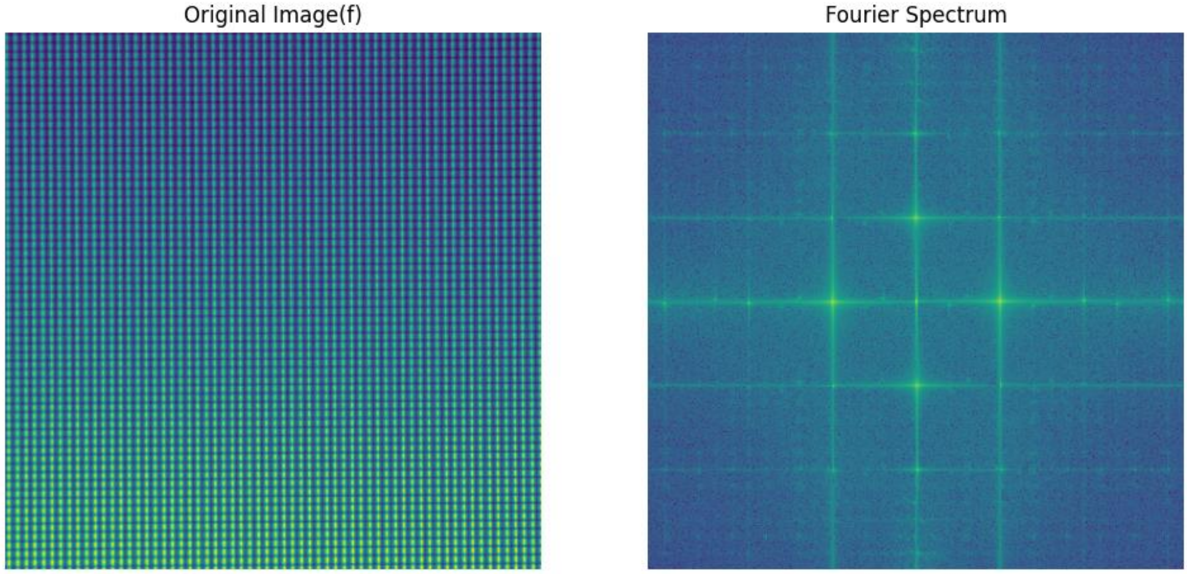


Fig 01. Original image on left side and their Fourier transforms on right side. Clearly, we can see periodicity is visible as spikes.

Part a

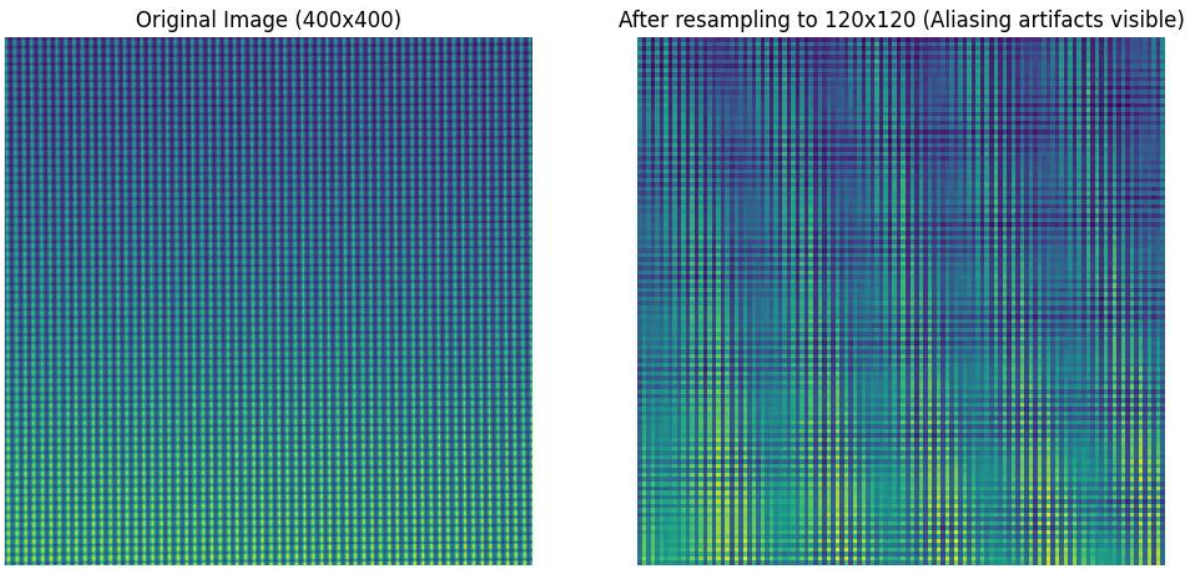


Fig 02. Rescaled the original image by scaling factor of 0.3.

In the right image of Fig 02, the aliasing start visible on scaling down the original image by a factor of 0.3 using bilinear interpolation.

Part b

Let f_x & f_y original sampling frequency in horizontal and vertical direction respectively.

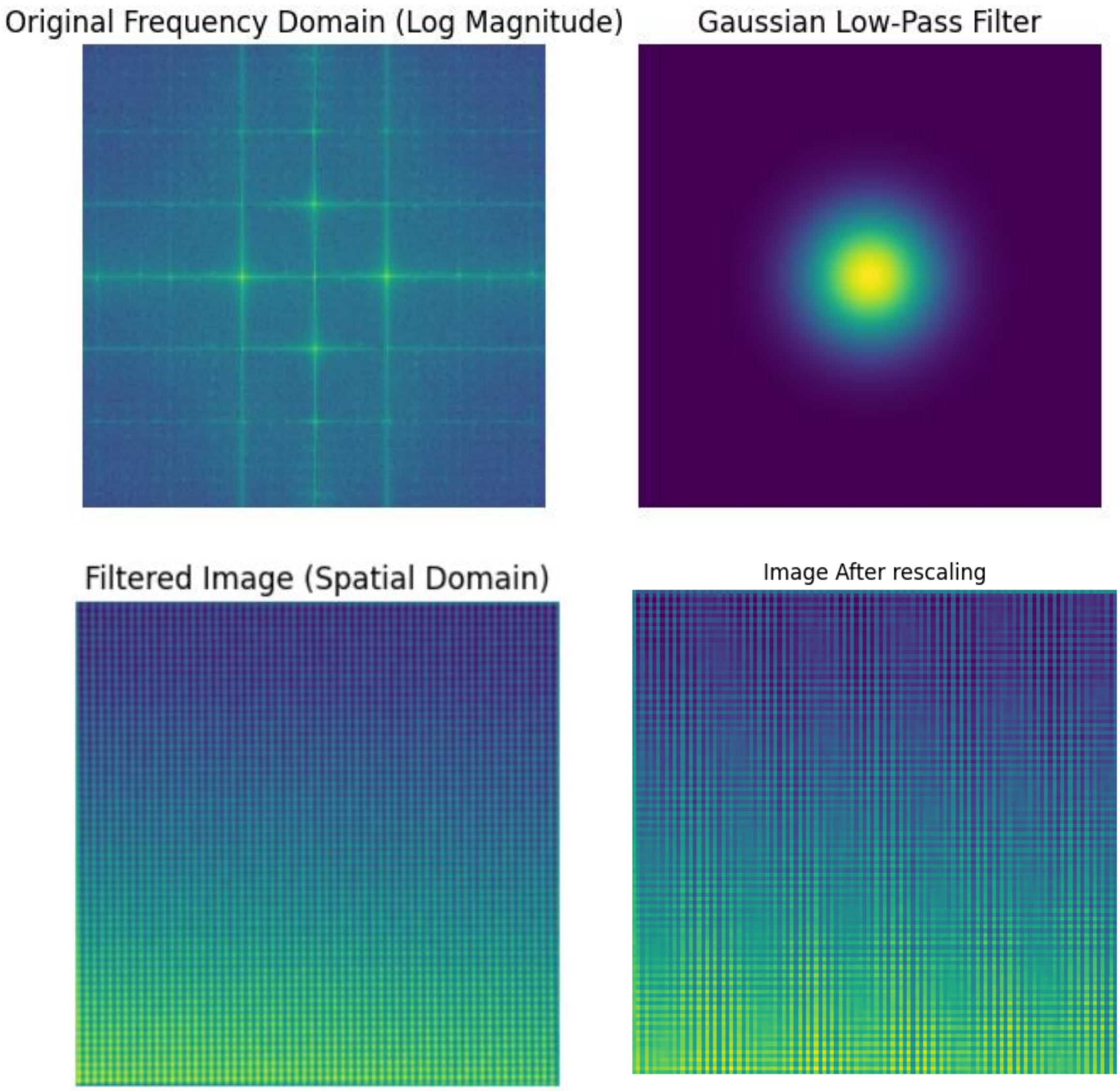
S_x : Scaling factor in horizontal direction

S_y : Scaling factor in vertical direction

In the horizontal direction, the cutoff frequency, $D_{ox} = \frac{f_x}{2S_x}$

In the vertical direction, the cutoff frequency, $D_{oy} = \frac{f_y}{2S_y}$

Applied the Gaussian low pass filter on frequency domain of original image 'f'. The result are shown in fig 03.

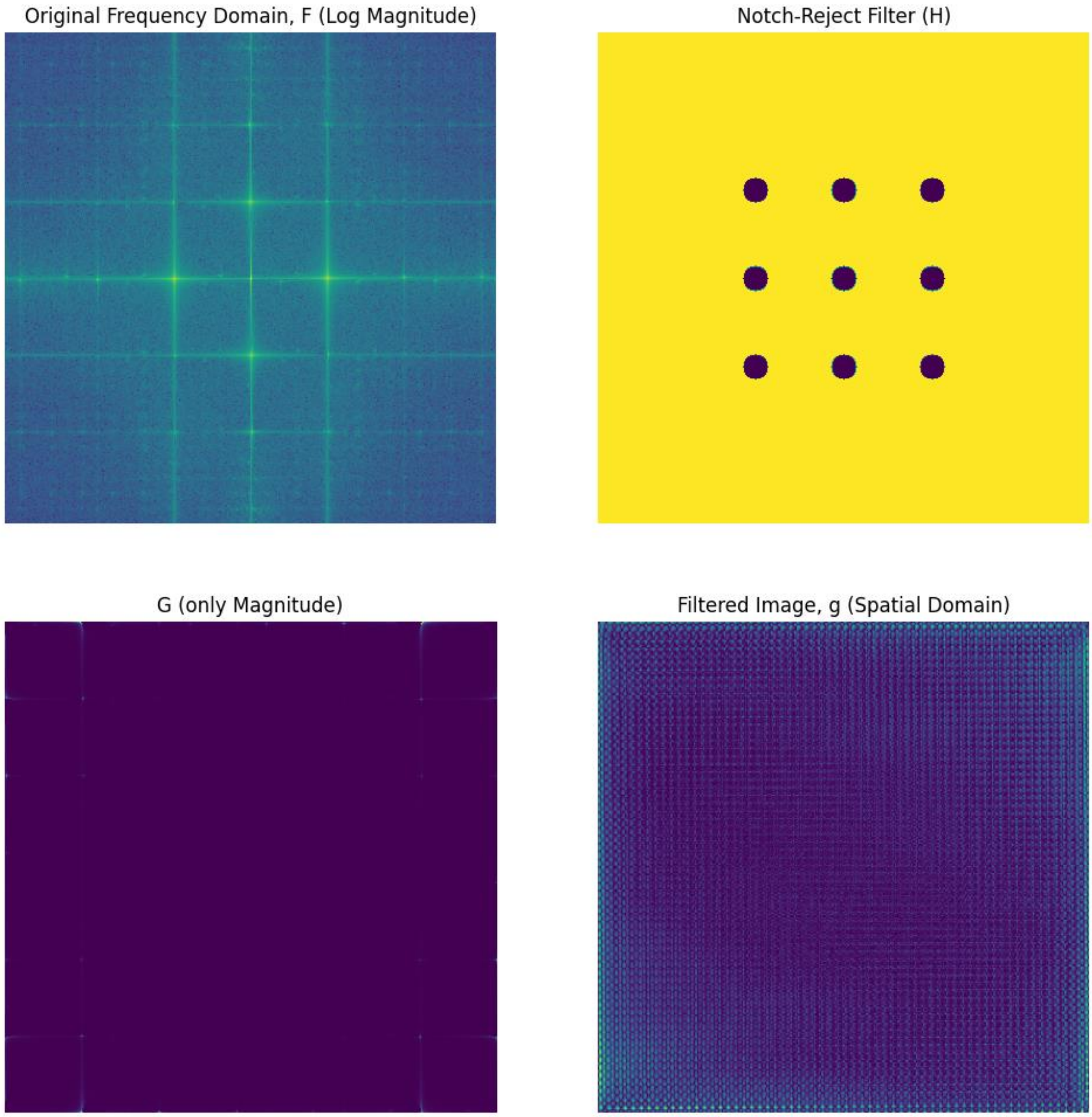


AB
CD

Fig 03. Applied the Gaussian low pass filter on frequency domain of 'f'.

In Fig 03, the Gaussian low pass filter applied on frequency domain of 'f'. The image 'C' is filtered image after bringing it to spatial domain and 'D' are after rescaling it (same thing as in part A). We can see that after applying the low pass filter on 'f', the aliasing artifacts almost gone.

Part C



AB
CD

Fig 04. Applied notch-reject filter 'H' on 'F'.

Defined a notch-reject filter as explained below.

Notch-reject filter

$H(u,v)$: filter in the frequency domain.

$$H(u,v) = 1 \quad \forall (u,v)$$

where $u \in [0, rows]$ & $v \in [0, cols]$

let

centers : list where we have to reject component.

the distance 'D' from each point (u,v) in the frequency domain to the center (u_i, v_i) can be computed as:

$$D(u, v, u_i, v_i) = \sqrt{(u-u_i)^2 + (v-v_i)^2}$$

Final notch filter

$$H(u,v) = \begin{cases} 0, & \text{if } D \le d_0 \text{ for } (u_i, v_i) \\ 1, & \text{otherwise} \end{cases}$$

d_0 : radius of notch.

Part D

Optimum Notch filter

Bunk Pages - The Social Notebook

By using cumulative sums and sliding windows, the mean, variance and covariance of f and η in an $m \times n$ neighborhood around each pixel can be computed in $O(MN(m+n))$ time.

Cumulative sums $S(i,j)$ of 'f' for each pixel (i,j) -

$$S(i,j) = \sum_{i'=0}^{i} \sum_{j'=0}^{j} f(i',j')$$

The sum of f in any rectangular region $(m \times n)$ can be computed in constant time using inclusion-exclusion.

Thus, cumulative sum '$S(i,j)$' for each pixel calculated in $O(1)$.

Total MN pixels.
So, total time $\rightarrow O(MN)$ $\leftarrow$ each pixel takes constant time.

Hence, mean of 'f' can be calculated in $O(MN)$.

$$Var(f) = \underbrace{mean(f^2)}_{\text{can be calculated in } O(MN) \text{ same as } mean(f).} - \underbrace{(mean(f))^2}_{O(MN)}$$

$$cov(f,\eta) = mean(f \cdot \eta) - \underbrace{mean(f) \cdot mean(\eta)}_{O(MN)}$$

Since, cumulative sum for f, f^2 and $f \cdot n$ each take $O(MN)$ time.

And since this need to be done for rows and columns in sliding window.

Hence,

total time $\propto O(MN(m+n))$

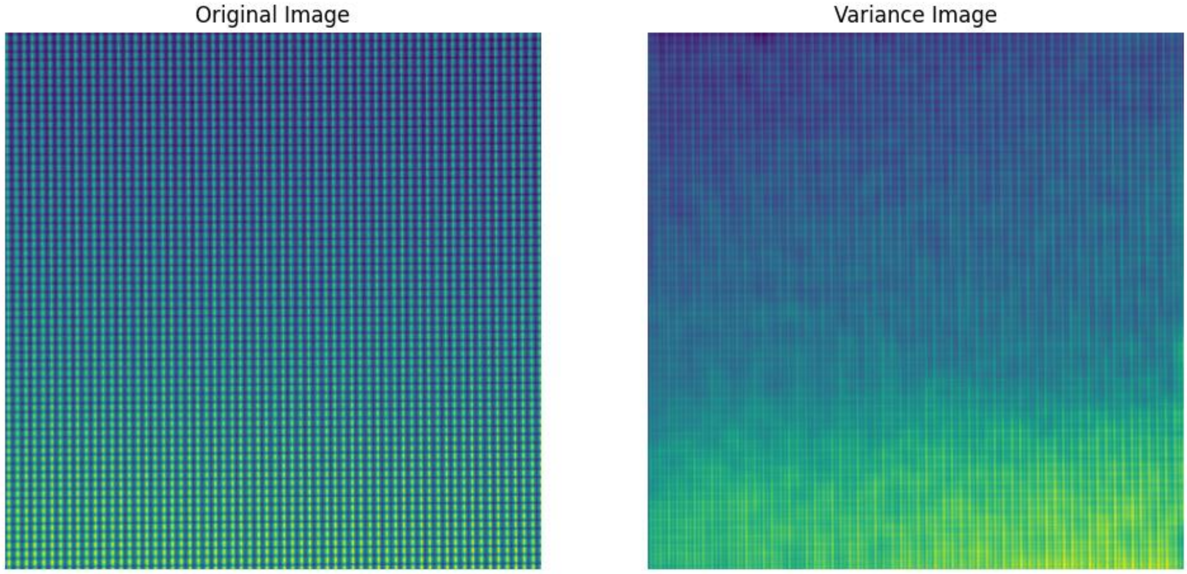


Fig 05. Computed the $Var_{x,y}$(f) using the sliding windows of size 15x15.

Part E

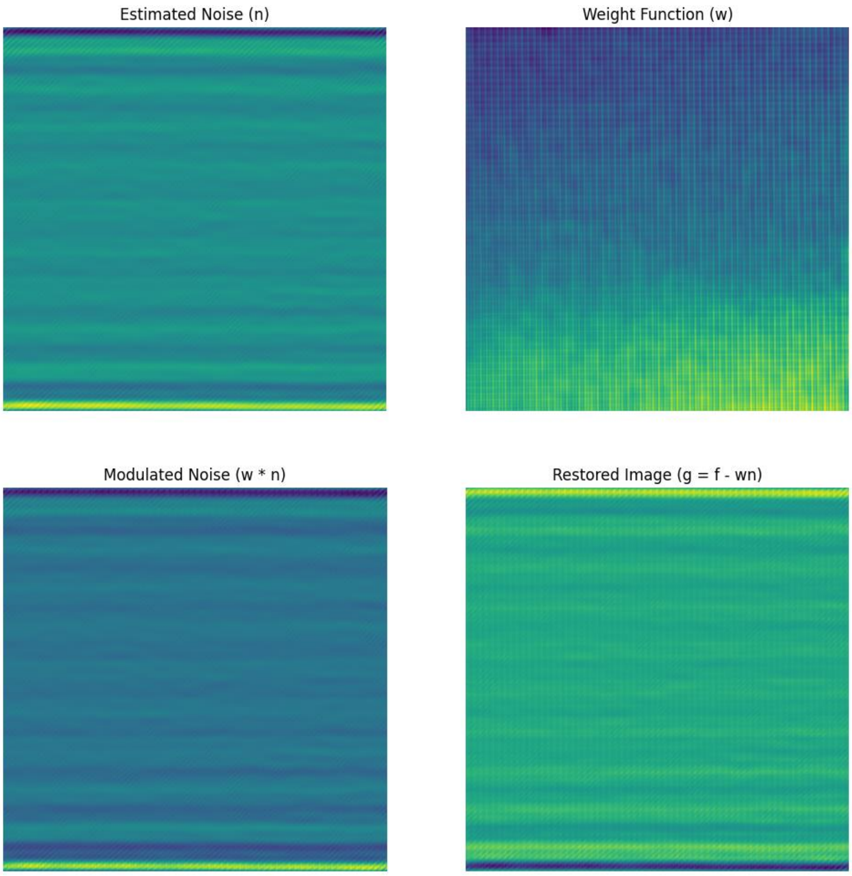
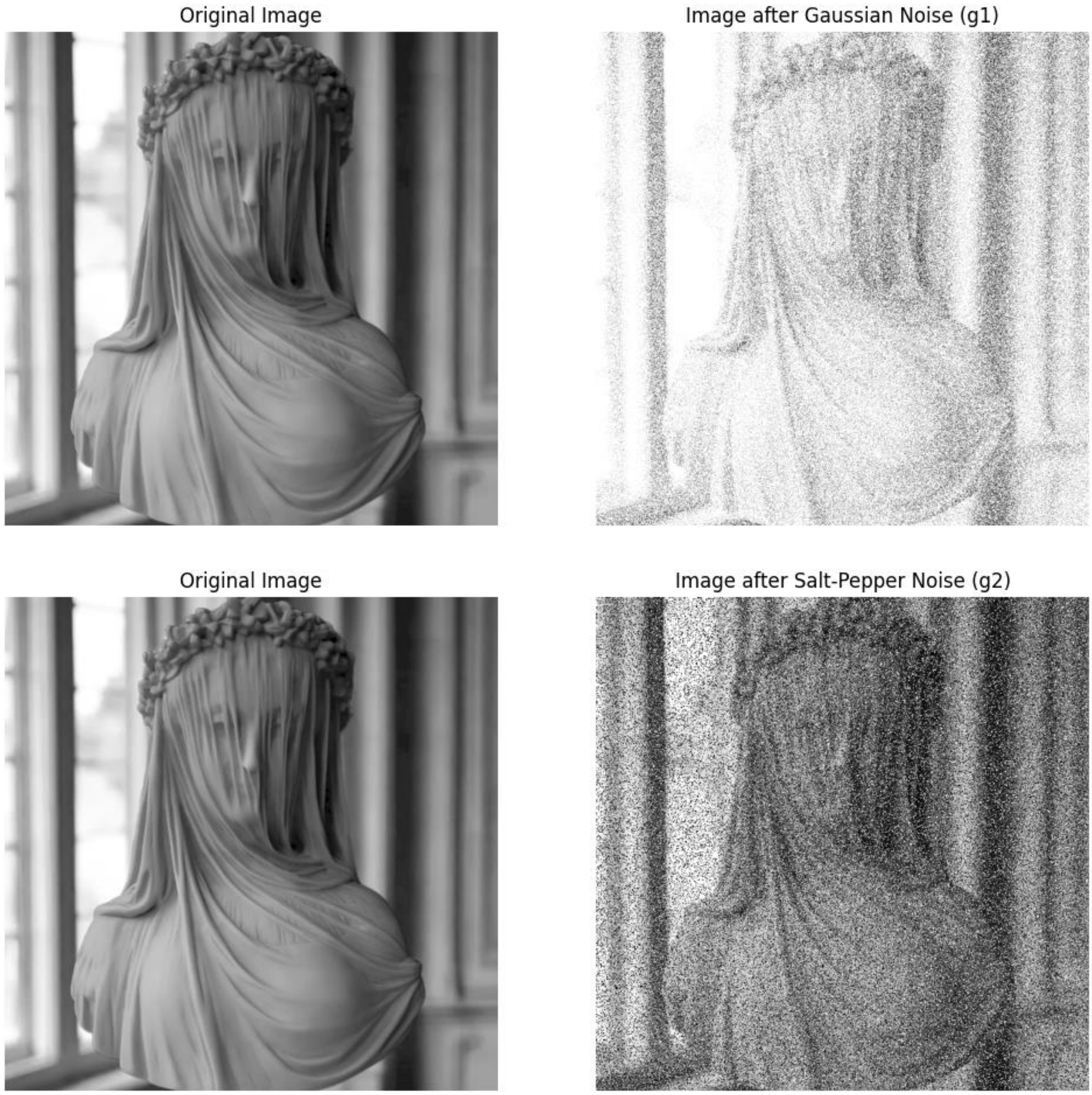


Fig 06. Performed optimum notch filtering

AB
CD

Q 3

Part a



AB
CD

Fig 07. A & C are same. B is created by adding Gaussian noise to A and D is created by adding Salt-Pepper noise to C.

Part B

Mean and Median filter on g1

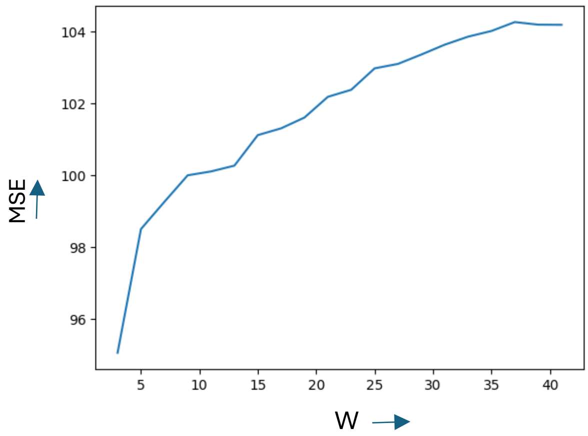


Fig 04. MSE of mean filter on g1

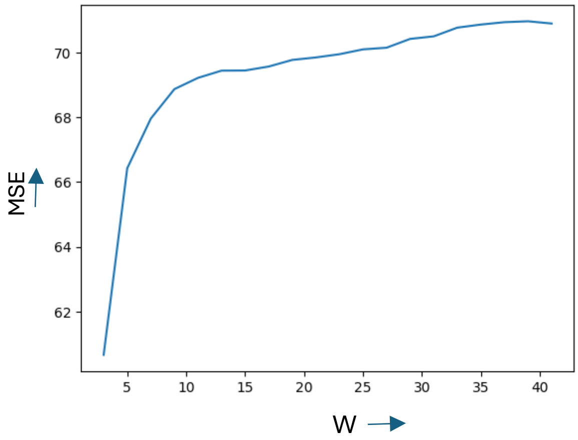


Fig 05. MSE of median filter on g1

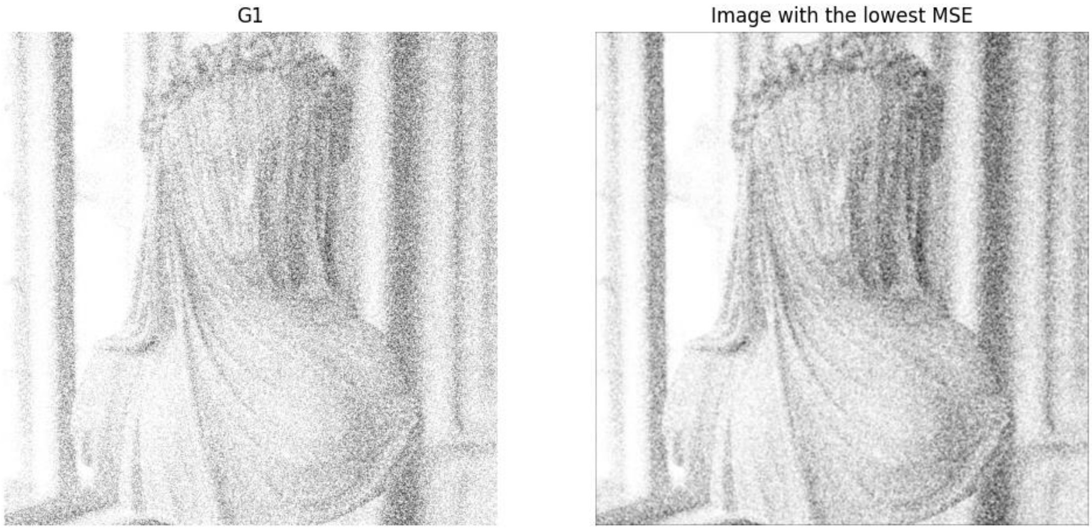


Fig 06. Right image: Mean filter on g1 with minimum MSE (best resulting image)

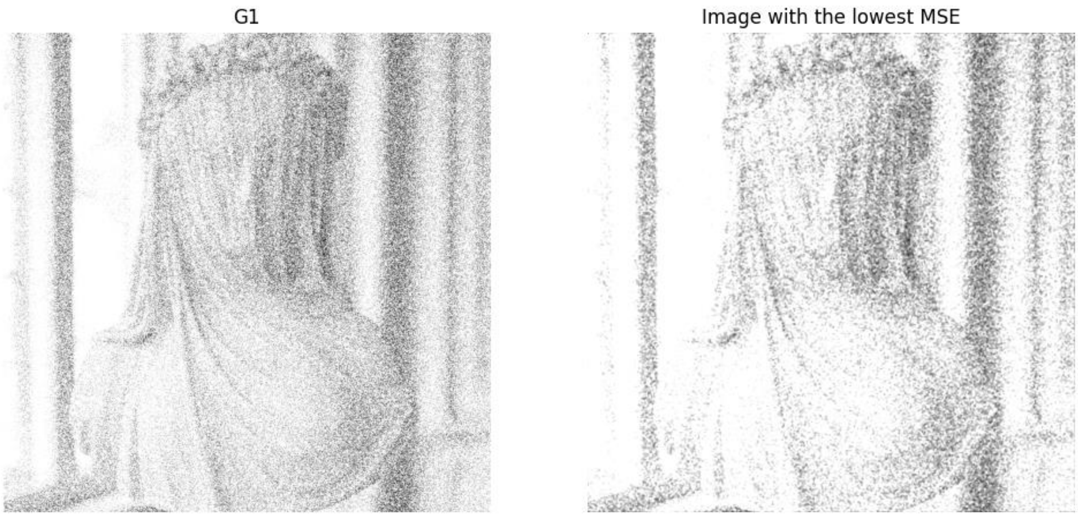


Fig 07. Left image: g1 and Right image: Median filter on g1 with minimum MSE

Mean and Median filter on g2

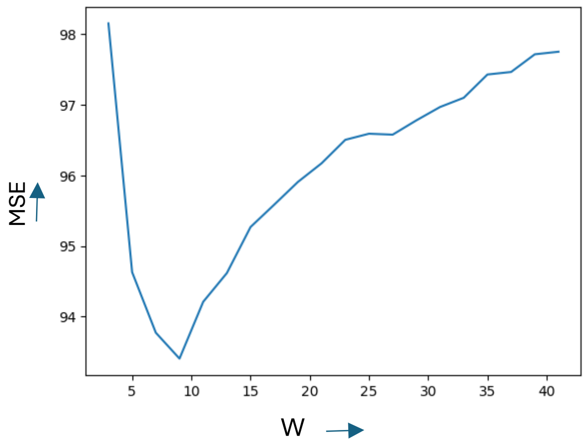


Fig 04. MSE of mean filter on g2

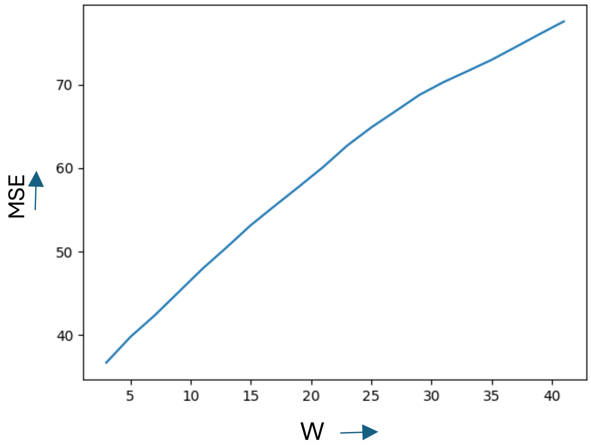


Fig 05. MSE of median filter on g2

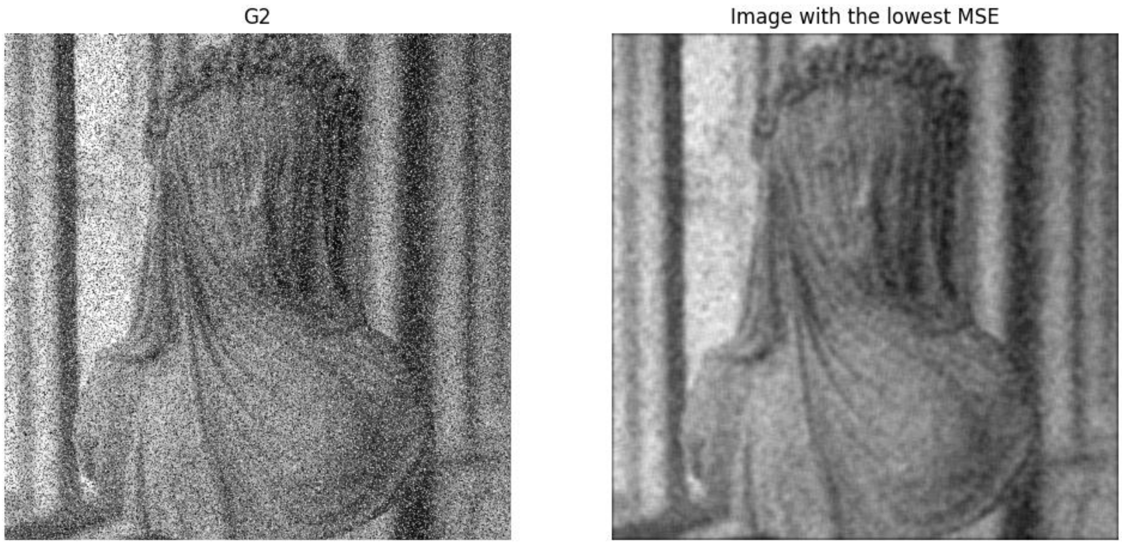


Fig 06. Right image: Mean filter on g2 with minimum MSE (best resulting image)

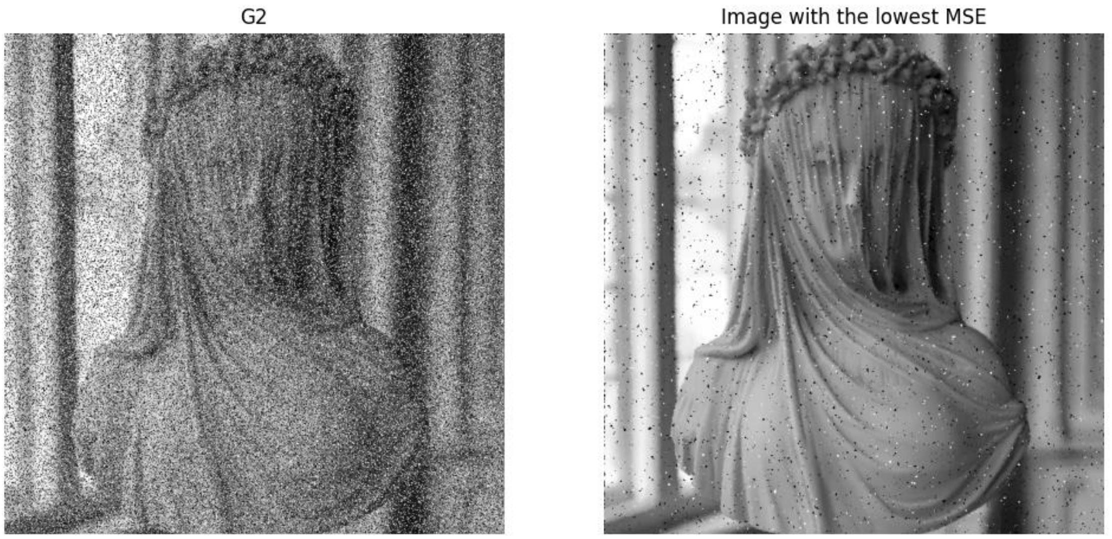


Fig 07. Right image: Median filter on g2 with minimum MSE (best resulting image)

Part C

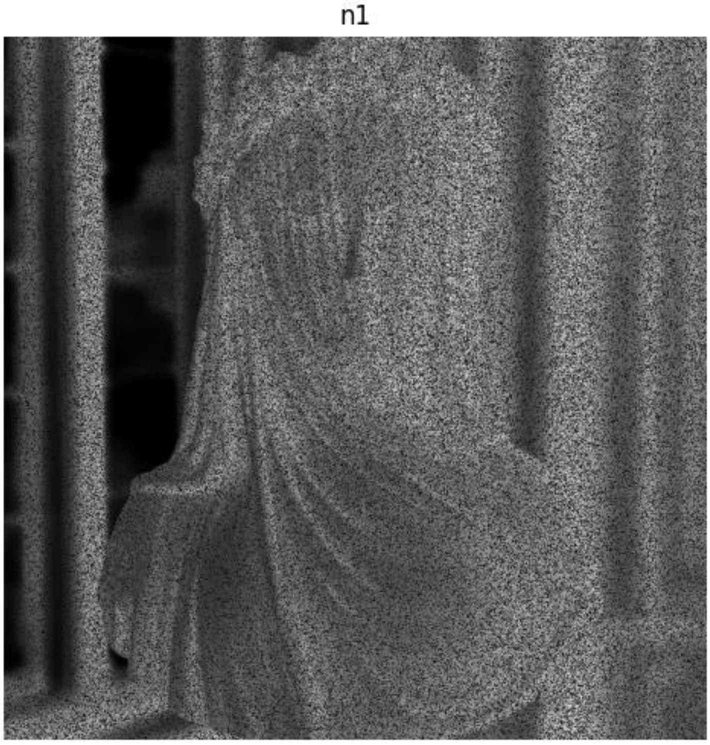


Fig 08. Gaussian noise image, n1= g1-f.

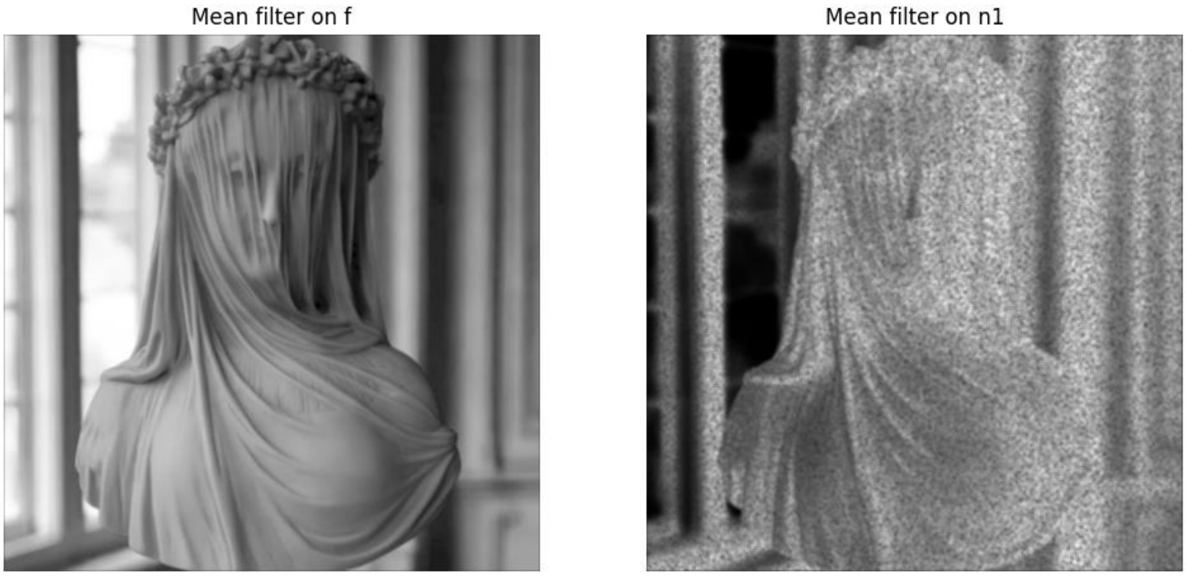


Fig 09. Left Image: Mean filter on f. Right Image: Mean filter on n1. In both image w=5.

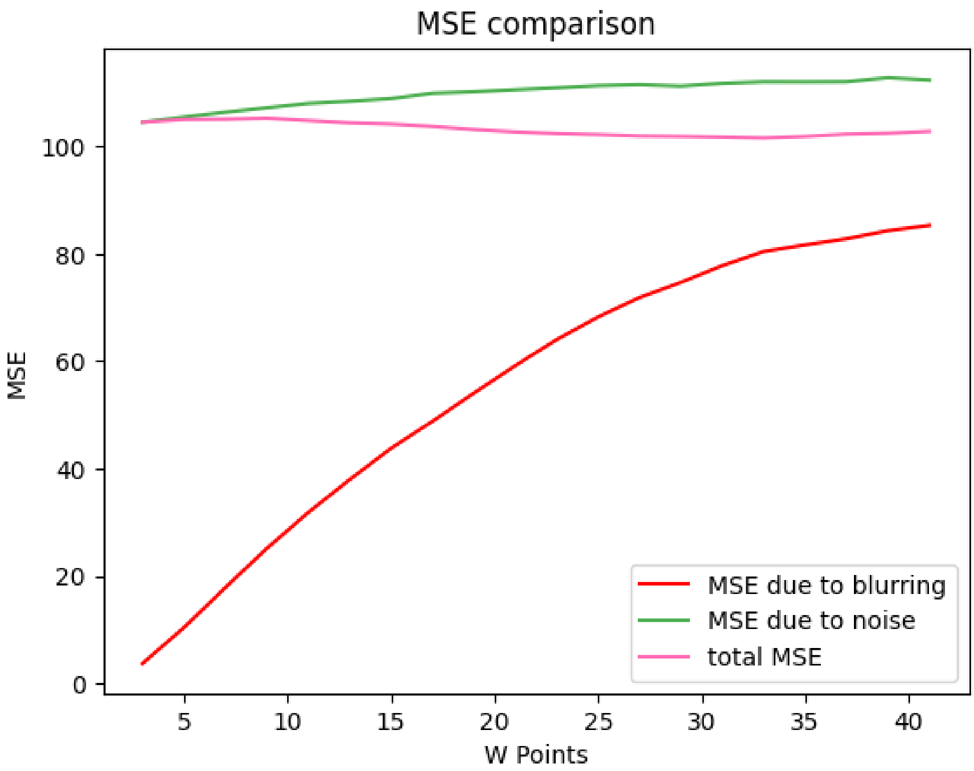


Fig 10. MSE comparison

Observation:

1. MSE due to blurring increasing as the kernel size w increases since larger filters cause more blurring.
2. MSE due to noise gradually increasing with increasing w.
3. Total MSE is first decreased (due to noise reduction) and then increase (due to excessive blurring), explaining the observed behaviour when you vary w.

Part 4

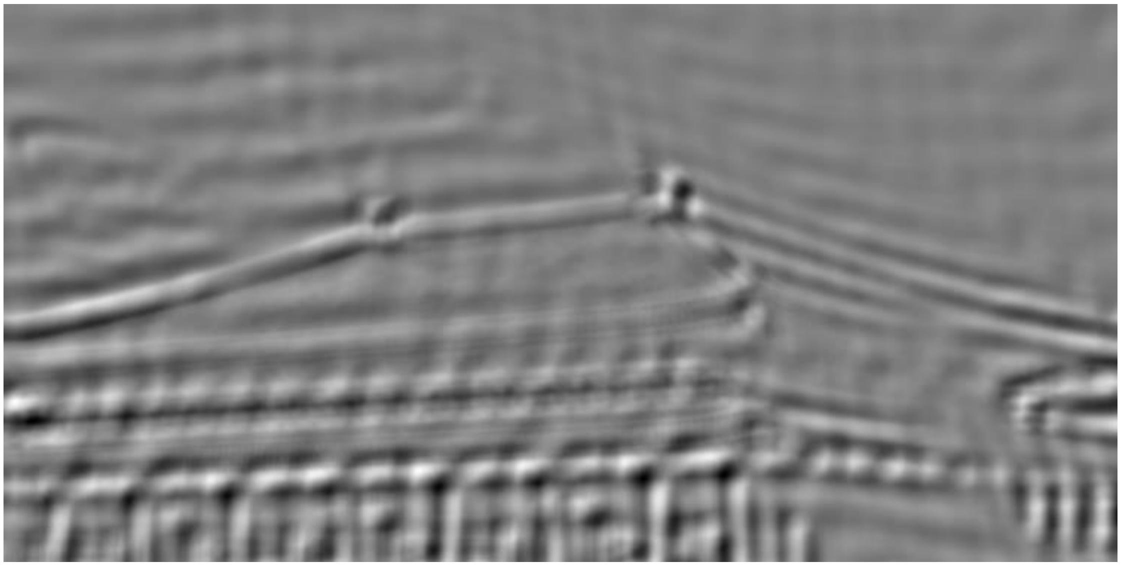
a.

Original Image:

Degraded Image:

Applied Freq Domain Blurring from part1 with kernel with r = 8. Added gaussian noise to the blurred image by taking mean = (0.05 Imax + 0.1 Imax) / 2 and standard deviation = (0.05 Imax)/6 so that 99.7% of noise lie in [0.05 Imax, 0.1 Imax].

Restored Image:



Divided the DFT of above degraded image by DFT of kernel (with r = 8).

b.

c. $S_n(u,v) = \mathbb{E}\{|N(u,v)|^2\}$

Bunk Paper - The Social Notebook

$n(x,y) \sim \mathcal{N}(0,\sigma^2)$ are i.i.d random variable $\forall x,y$.

$$N(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y)\exp\left(-2\pi i\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

$$\mathbb{E}[N(u,v)] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\mathbb{E}[n(x,y)]\exp\left(-2\pi i\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} 0 \times \exp\left(-2\pi i\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

$$= 0$$

$$N(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y)\left[\cos\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right) - \sin\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)\right.$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y)\cos\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right) - \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y)\sin\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

$$N_R(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y)\cos\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

$$N_I(u,v) = -\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y)\sin\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

$$N(u,v) = N_R(u,v) + i\,N_I(u,v).$$

$$\left.\begin{array}{l}\mathbb{E}[N_R(u,v)] = 0\\ \mathbb{E}[N_I(u,v)] = 0\end{array}\right\}\text{ similar to } \mathbb{E}[N(u,v)]$$

$$\operatorname{Var}[N_R(u,v)] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \operatorname{Var}[n(x,y)] \cos^2\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$
$$= \sigma^2 \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \cos^2\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$
$$= \frac{\sigma^2 MN}{2}$$

$$\operatorname{Var}[N_I(u,v)] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \operatorname{Var}[n(x,y)] \sin^2\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$
$$= \sigma^2 \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \sin^2\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$
$$= \frac{\sigma^2 MN}{2}$$

$$\operatorname{Var}[N_R(u,v)] = \mathbb{E}(N_R^2(u,v)) - (\underbrace{\mathbb{E}[N_R(u,v)]}_{0})^2$$
$$\Rightarrow \mathbb{E}(N_R^2(u,v)) = \operatorname{Var}[N_R(u,v)] = \frac{\sigma^2 MN}{2}$$

$$\operatorname{Var}[N_I(u,v)] = \mathbb{E}(N_I^2(u,v)) - (\underbrace{\mathbb{E}[N_I(u,v)]}_{0})^2$$
$$\Rightarrow \mathbb{E}(N_I^2(u,v)) = \operatorname{Var}[N_I(u,v)] = \frac{\sigma^2 MN}{2}$$

$$\therefore \mathbb{E}[|N(u,v)|^2] = \mathbb{E}[N_R^2(u,v) + N_I^2(u,v)]$$
$$= \mathbb{E}[N_R^2(u,v)] + \mathbb{E}[N_I^2(u,v)]$$
$$= \frac{\sigma^2 MN}{2} + \frac{\sigma^2 MN}{2} = \sigma^2 MN$$

$$\boxed{S_n(u,v) = \sigma^2 MN}$$

Created a gaussian noise matrix of size 256 x 256. With standard deviation =1.

Theoretical value of Sn = 65536

Numerical value of Sn obtained = 65819.69873582013 c.

1.

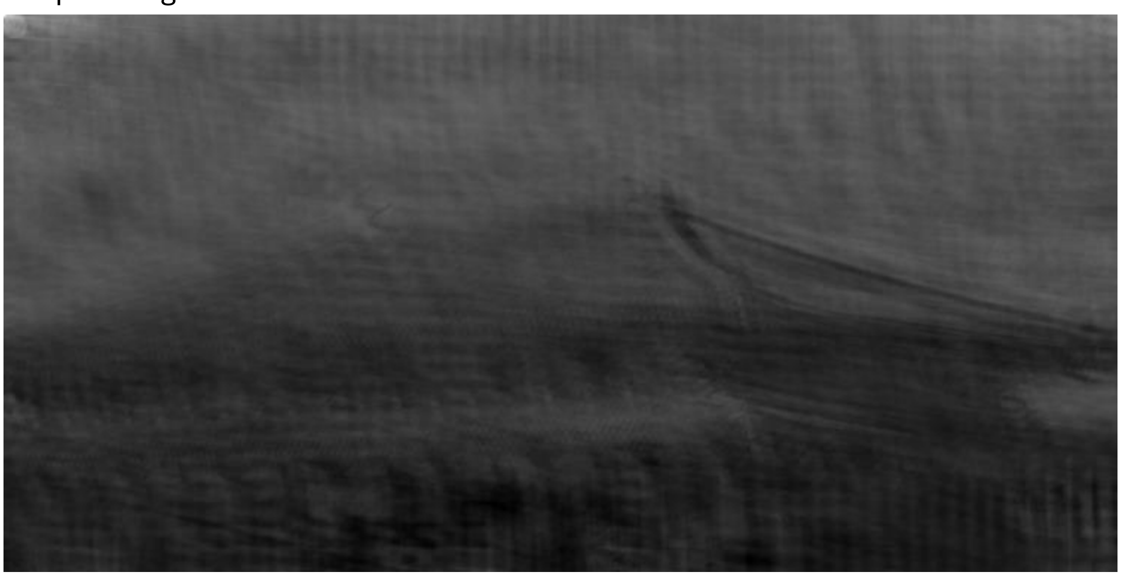
Sf = |F(u,v)|^2

Original Image is same as in part 4.a.

Degraded Image:

Applied Freq Domain Blurring from part1 with kernel with r = 8. Added gaussian noise to the blurred image with mean = 0 and standard deviation =1.

Output Image:



Calculate Weiner Filter with Sn(u,v) = (750x1500) and Sf(u,v) = |F(u.v)|^2.

F_hat(u,v) = G(u,v)* W(u,v)

2.

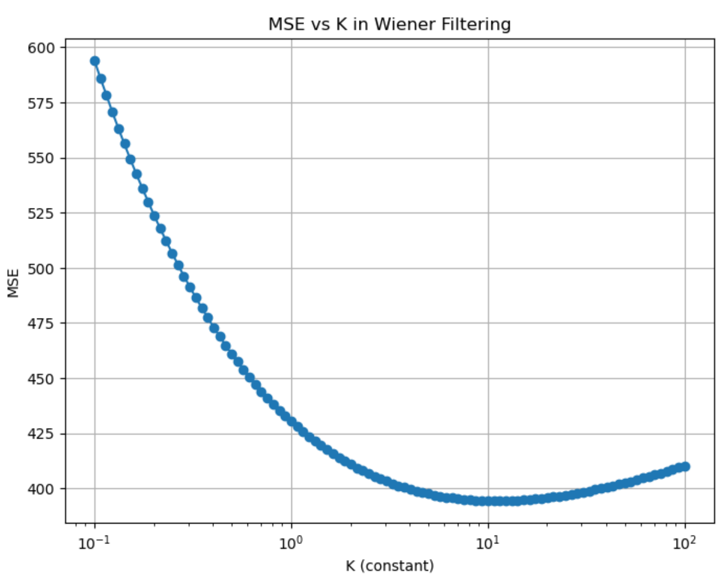
Sf(u,v) / Sn(u,v) = k

Original Image same as part 4.a.

Degraded Image:

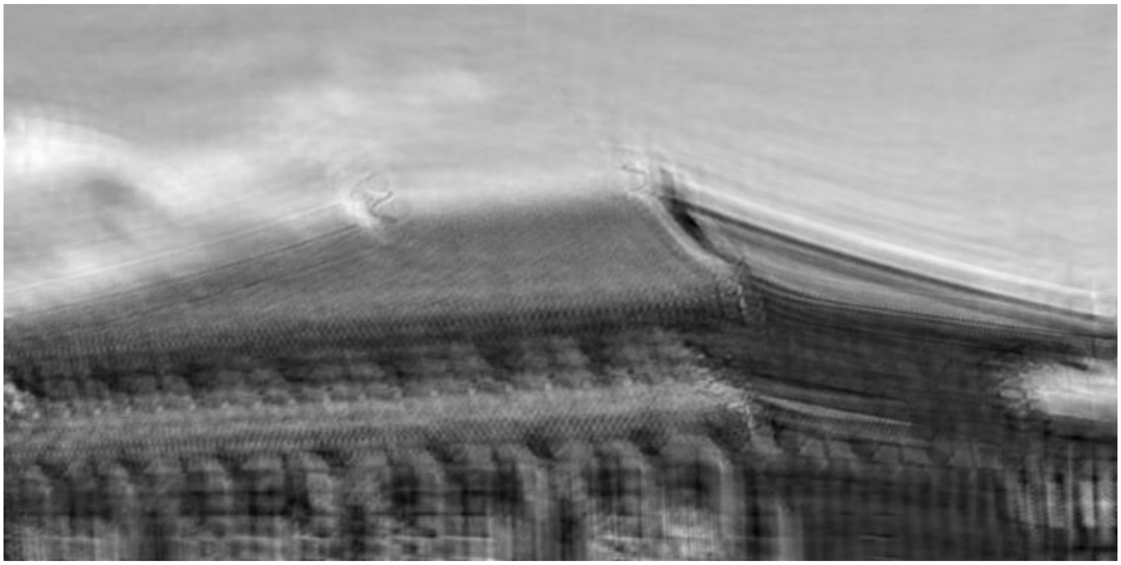
Applied Freq Domain Blurring from part1 with kernel with r = 8. Added gaussian noise to the blurred image with mean = 0 and standard deviation =1.

Graph: MSE v/s k.



Value k = 10.722672220103231 has minimum MSE = 394.41760209330124.

Output Image;



Calculate Weiner Filter with k= Sn(u,v) / Sf(u,v) = 10.722672220103231.
F_hat(u,v) = G(u,v)* W(u,v).

d.

Degraded image with kernel (r= 8)

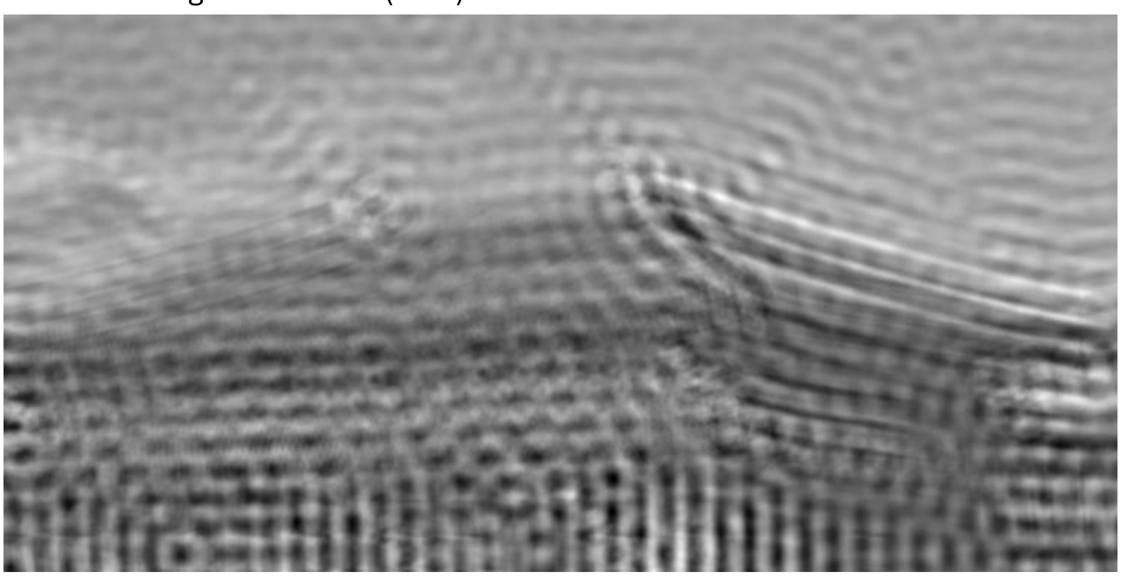
Applied Freq Domain Blurring from part1 with kernel with r = 8. Added gaussian noise to the blurred image with mean = 0 and standard deviation =1.

Restored Image with kernel (r= 6).

Calculate Weiner Filter with k=Sn(u,v) / Sf(u,v) = 100

F_hat(u,v) = G(u,v)* W(u,v).

Restored Image with kernel (r=12)



Calculate Weiner Filter with k=Sn(u,v) / Sf(u,v) = 100

F_hat(u,v) = G(u,v)* W(u,v).