IMAGE PROCESSING AND INTERPRETATION

CS474.674.1001

PROGRAMMING ASSIGNMENT NO: 4

A Report

By

PRATIK WALUNJ

JALPA KAILA

Division of Work Statement

Pratik Walunj and Jalpa Kaila divided the work for this assignment as follows:

Coding:

- Pratik handled coding for Questions 1, 2, 3.

- Jalpa handled coding for Questions 1, 2, 3.

Report Write-up:

- Pratik contributed to Section 2.1, 2.2, 3.1 and 3.2.

- Jalpa contributed to Section 2.2, 2.3, 3.2 and 3.3.

We collaborated on coding and provided mutual input for report sections to ensure balanced contribution in both aspects.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1. MOTIVATION

This assignment is like a puzzle where we get to explore something called the Discrete Fourier Transform (DFT) and a quick way to do it using computer code. We got the chance to implement what we learned in class and this assignment lets us play with images in frequency domain. It's like peeking into the hidden details of images. We will be using a special technique called Fast Fourier Transform (FFT) in python. This is like a hands-on journey to understand how computers process images in frequency domain. It's not just theory; we will be doing real experiments, and that's what makes it so interesting for us!

CHAPTER 2. THEORY

2.1 Frequency Domain Filter

Frequency Domain Filtering works on Two cases based on H(u,v) specified in frequency domain or spatial domain. As you can see in figure 1, Filter mask and result after Fourier transform of input image multiply and we get the Filtered spectrum. After that by taking the inverse FFT we get the Filtered image.

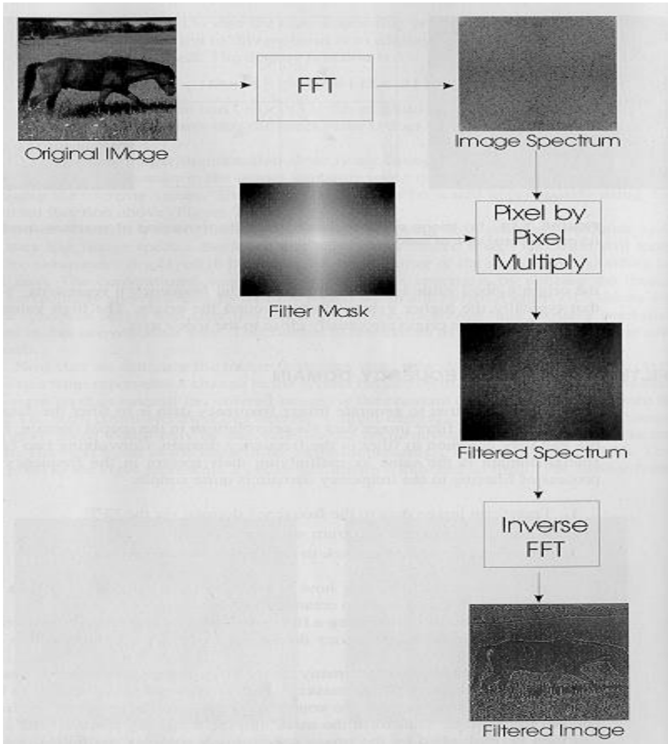
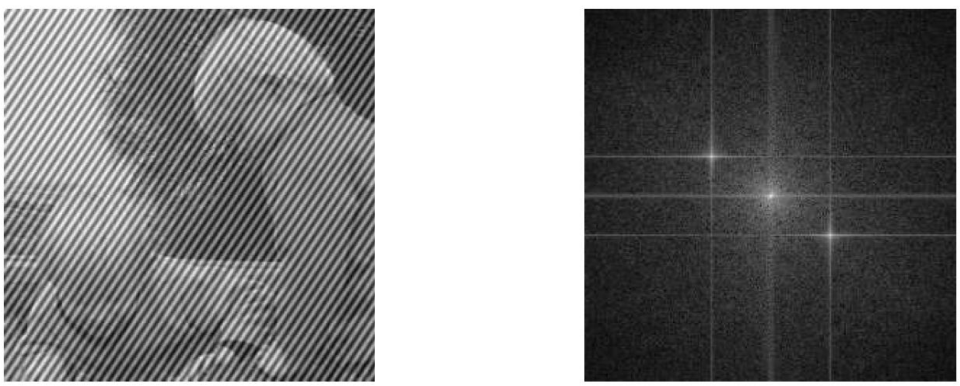


Figure 1: Represents the Process of Frequency Domain filter.

2.1.1 Band-reject Filter

Band-reject Filter is very effective for removing periodic noise which can arise from interferences during image acquisition. Periodic noise shows up as spikes away from the origin of the spectrum. Band-reject filters could be used for removing periodic noise. Their main weakness is that they also remove additional frequencies, not necessarily related to noise. Figure 2 (a) shows the noisy image and (b) shows the spectrum of that image. Band reject filter use to remove the periodic noise shows up as spikes away from the origin of the spectrum.



(a) **(b)**

Figure 2: Represents the (a) Noisy image and (b) spectrum of noisy image.

There are three types of Band reject filters:

1. Ideal
2. Butterworth
3. Gaussian

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

Figure 3: Represents the Ideal, Butterworth and Gaussian filters.

2.1.2 Notch filter

A Notch filter rejects frequencies in a pre-defined area (neighborhood) about the center of the frequency domain. Notch filters are symmetric about the origin, so the notches must occur in symmetric pairs (i.e., at (u_0,v_0) and at ($-u_0$, $-v_0$)). Figure 4 shows the input image and spectrum and when we apply notch pass filter to that, the noise pattern extract from input noisy image. Noth pass filter can be obtained as follows:

$$H_{NP}(u,v) = 1 - H_{NR}(u,v)$$

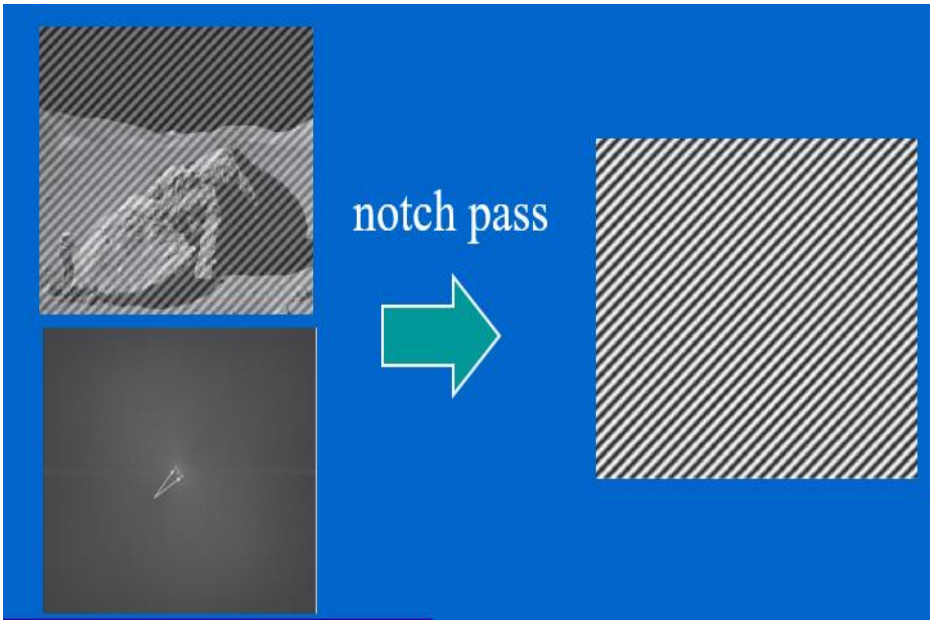


Figure 4: Represents the Noise pattern extract from input image.

2.2 Convolution in Frequency Domain

Frequency Domain Filtering works on Two cases based on H(u,v) specified in frequency domain or spatial domain.

$$f(x,y) * h(x,y) = g(x,y)$$

$$F(u,v)\, H(u,v) = G(u,v)$$

Case 1: H(u,v) is specified in the frequency domain.

Case 2: H(u,v) is specified in the spatial domain by specifying h(x,y) and taking the DFT of h(x,y).

Steps for Case 1:

1. Given an input image f(x,y) of size M x N, set the padding sizes P and Q, e.g., P=2M and Q=2N (note: P and Q must also be a power of 2 to use FFT)
2. Form a padded image $f_p(x,y)$ of size P x Q
3. Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center its spectrum.
4. Compute F(u,v) from Step 3.

$$F(u,v) = R(u,v) + jI(u,v)$$

5. Generate H(u,v) of size P x Q with center at (P/2, Q/2). This is typically done by specifying and sampling the desired filter function.
6. Compute G(u,v)=H(u,v)F(u,v) using element-wise "complex" multiplication.
7. Obtain the filtered image $g_p(x,y)$ by computing the inverse FFT of G(u,v):

$$g_p(x+y) = (real[£^{-1}\{G(u,v)\}])(-1)^{x+y}$$

   Here, we explicitly disregard the imaginary part and perform undo operations on centering transformation.

8. Obtain the final filtered result, g(x,y), of size M x N, by extracting the M x N region from the top, left quadrant of $g_p(x,y)$ (i.e., "undo" padding).

Steps for Case 2:

H(u,v) is indirectly specified by specifying h(x,y). Same steps as in Case 1 except that H(u,v) must be generated from h(x,y). Steps for that is as follows:

1. Form $h_p(x,y)$ by padding h(x,y) with zeroes.
2. Multiply $h_p(x,y)$ by $(-1)^{x+y}$ to center its spectrum.
3. Compute its FFT to obtain H(u,v).

2.3 Homomorphic Filtering

The main idea behind homomorphic Filtering is, remove shading effects due to uneven illumination. It Enhance high frequencies. Attenuate low frequencies but preserve fine detail.

$$\boldsymbol{f(x,y) = i(x,y)\, r(x,y)}$$

$$F(u,v) = I(u,v) * R(u,v)$$

Where, i(x,y): illumination component and r(x,y): reflectance component. Illumination i(x,y) varies slowly and affects low frequencies mostly. Reflectance r(x,y) varies faster and affects high frequencies mostly.

Low frequencies from i(x,y) and high frequencies from r(x,y) are intermixed together. When applying filtering, it is impossible to manipulate them separately. To separate them we apply ln(f(x,y)) function.

$$F(u,v)H(u,v) = [I(u,v) * R(u,v)]H(u,v)$$

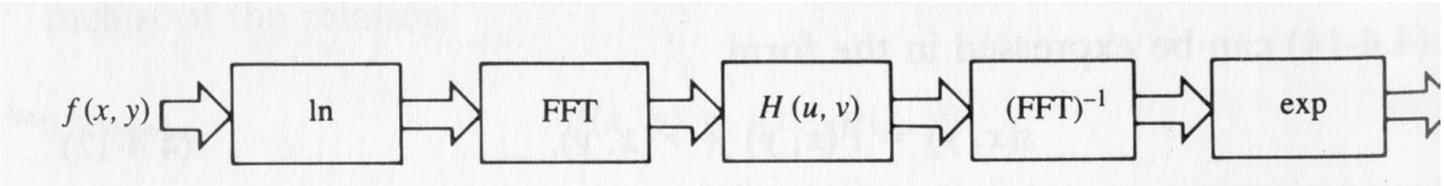


Figure 5: Represents the Sequence of Homomorphic filter.

Steps for Homomorphic Filtering:

1. Take $ln(f(x,y) = ln\big(i(x,y)\big) + ln(r(x,y))$
2. Apply FT: $F(ln(f(x,y)) = F(ln\big(i(x,y)\big)) + F(ln\big(r(x,y)\big))$

   Or $Z(u,v) = Illum(u,v) + Refl(u,v)$

   Low and high frequencies have now been separated.

3. Apply H(u,v):

$$Z(u,v)H(u,v) = Illum(u,v)H(u,v) + Refl(u,v)H(u,v)$$

4. Take Inverse FT:

$$F^{-1}(Z(u,v)H(u,v)) = F^{-1}(Illum(u,v)H(u,v)) + F^{-1}(Refl(u,v)H(u,v))$$

   Or $s(x,y) = i'(x,y) + r'(x,y)$

5. Take Exp():

$$e^{s(x,y)} = e^{i\prime(x,y)}e^{r\prime(x,y)}$$

Or $g(x,y) = i_0(x,y) + r_0(x,y)$

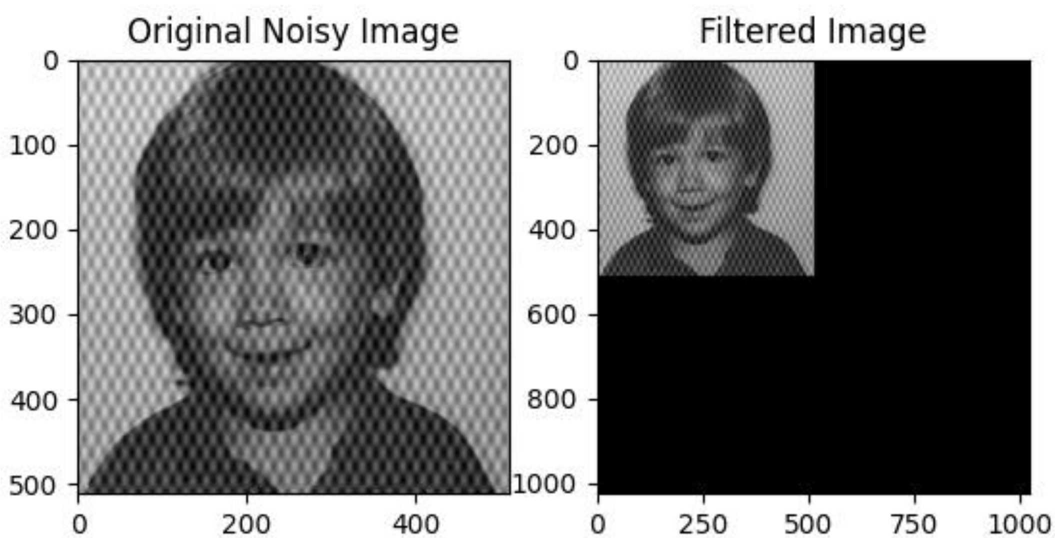
CHAPTER 3. RESULTS AND DISCUSSION

3.1 **Frequency domain filtering**

3.1.1 Band Reject Filter and Notch Filter

Band Reject Filter:

This section 3.1.1 and Figure 6 shows the result of frequency domain filtering using Ideal band-reject filter. We implement this equation shows in section 2.1 figure 3 for Ideal band reject filter on input noisy boy image. Here we perform filtering in frequency domain and resulting image we get by applying filtering in frequency domain.



(a) **(b)**

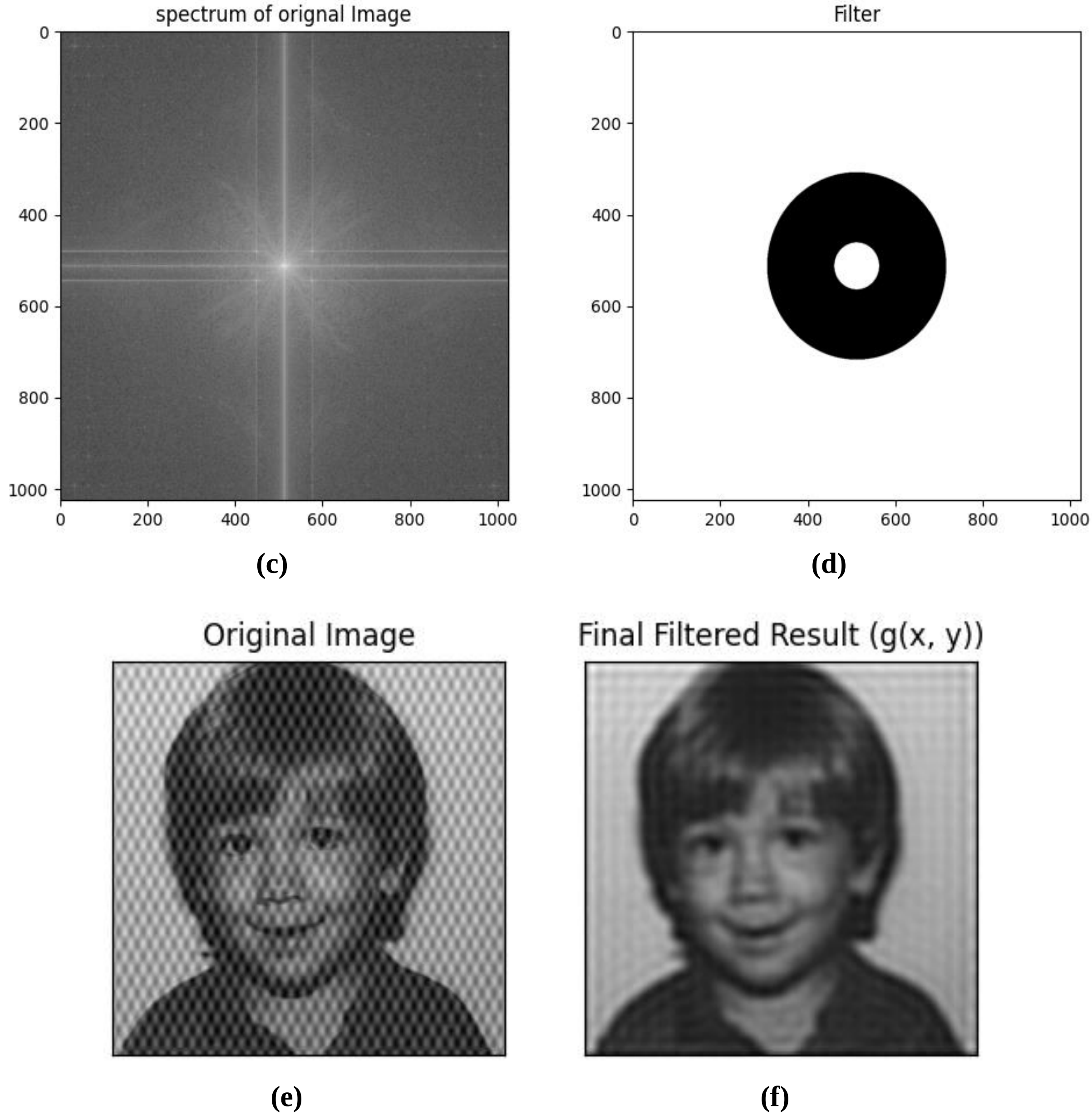


Figure 6: Represents the filtered image result using ideal band reject filter.

(a) input noisy image (b) cropped image (c) centered spectrum of input image (d) Ideal band reject filter (e) original noise image (f) filtered image.

Figure 7 shows the result of filtering using Gaussian filter. For comparison purposes, we implement this using a gaussian filter in spatial domain. Figure 7 shows the gaussian filter (7 x 7 and 15 x 15 filters). Here you can see that gaussian filtering using spatial domain, it smooths the image but cannot remove the noise as frequency domain can.

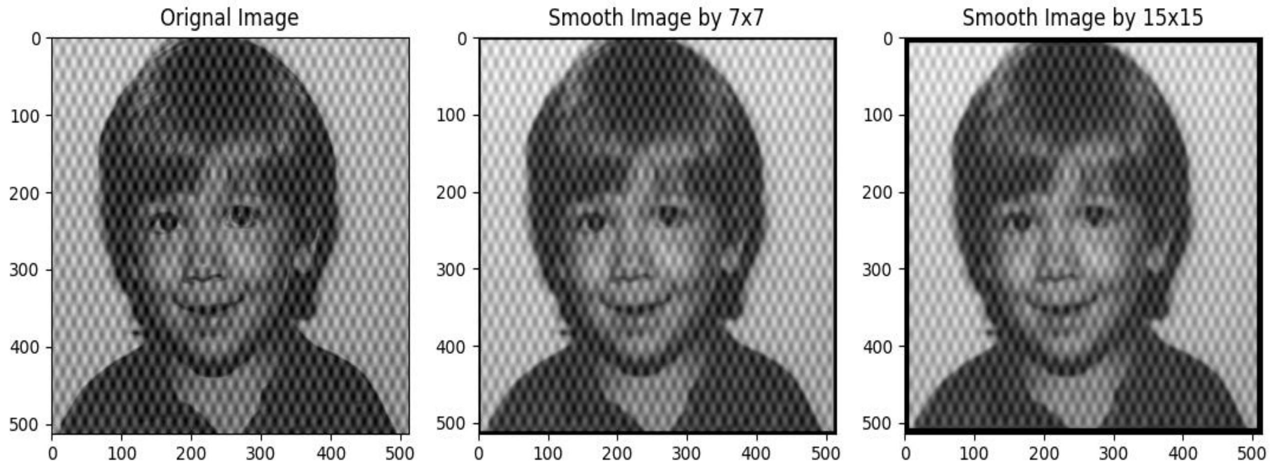


Figure 7: Represents the result of gaussian filter in spatial domain.

Notch Filter:

As mentioned in section 2.1, the main weakness of band-reject filter is they also remove the additional frequencies, not necessarily related to the noise. So, for that notch filter is used. A notch filter rejects frequencies in a pre-defined area (neighborhood) about the center of the frequency domain.

Notch reject filters can be constructed as products of high pass filters whose centers have been translated to the notch locations. A notch pass filter can be obtained by subtracting the result of notch reject filter from 1. From the notch pass filter, we will get the noise pattern of the image. By comparing figures 6 and 8, we can see the difference between filtered images. Notch filter gives better results than ideal band reject filter.

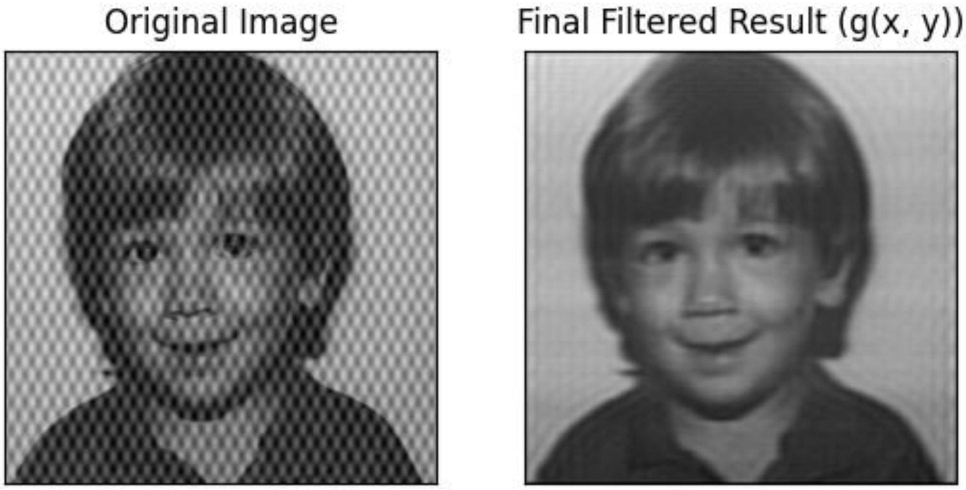


Figure 8: Represents the result of notch filter in frequency domain.

3.1.2 Noise pattern

Section 3.1.2 and figure 9 show the result of the noisy pattern extracted from the original input noisy boy image. The resulting noise pattern we get by subtracting the ideal filter from 1. Here we are subtracting the detailed image from 1 so we got the noise pattern.

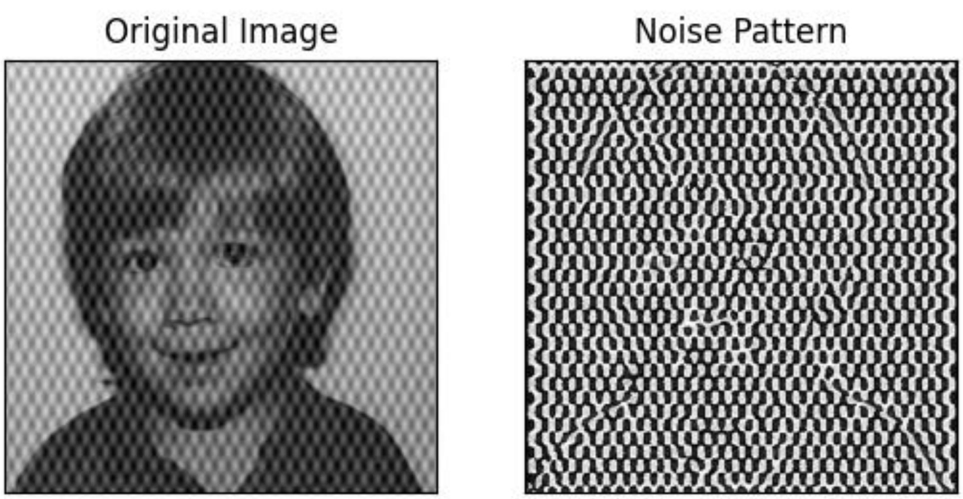


Figure 9: Represents the extracted noisy pattern.

3.1.3 Face verification and Face recognition

Face verification is concerned with validating a claimed identity based on the image of a face (one-to-one matching) [1]. Face recognition is concerned with identifying a person based on the image of a face (one-to-many matching) [1]. The Following are some possible implications on face verification and recognition performance of failing to remove/reduce noise in face images or introducing artifacts due to algorithmic errors.

Safety implications:

Unauthorized access: Due to noise and some artifacts, it is possible that some unauthorized individuals gain access to some sensitive areas such as government facilities, airports, and server rooms. There is also the chance of misuse of personal information and banking information loss and fraud.

Criminal investigations: due to noise and some artifacts in image, it may result in misidentification and leads to wrong investigations and identification of individual involved crime.

Economic Implications:

- Legal issues: Due to poor quality of image leads to incorrect identification and may result in legal consequences for lawsuit or well-known organization.
- Increase financial investment: To solve the impact of noise and artifacts, organizations may need to invest more money to develop strong and advanced algorithms which increases the financial burden of the organization.

3.2 Convolution in Frequency Domain

This section 3.2.1 shows the result of convolution in frequency domain. The convolution is performed between input Lenna image and Sobel mask. Here we performed Filtering in frequency and spatial domain. For filtering in frequency domain, we follow the steps mentioned in section 2.2 of chapter 2. We also need to preserve a odd symmetry of Sobel mask and for that padding is done on leading rows and columns. After that mask pad with zeros such that the centers of the two arrays coincide. Then centered that spectrum and compute the Fourier transform of that. We set the real part to zero as the Sobel mask is real and odd symmetrical, so the output is from imaginary part. Then undo the centered spectrum and as you can see in Figure 10, the results of filtering in both frequency and spatial domain. As you can see in the figure both the results are identical.

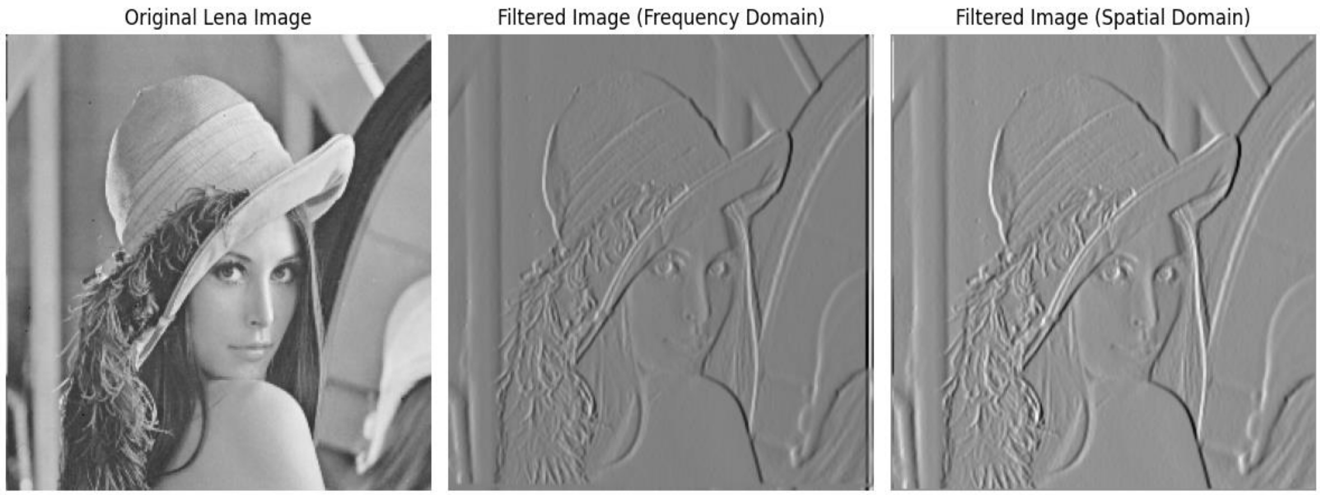


Figure 10: (a) Input Lenna image. (b) Filtered image in frequency domain (c) Filtered image in spatial domain.

3.3 Homomorphic Filtering

Implementing the Homomorphic filter, we need to apply the appropriate high-pass filter. This high pass filter emphasizes the high frequencies and attenuates the lower ones, preserves the fine details. The High pass filter used here is High-frequency emphasis filter shown as below:

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c\left[\frac{u^2+v^2}{D_0^2}\right]}\right] + \gamma_L$$

where D_0 is the cutoff frequency of the filter and γ_L, γ_H are the gains for the low and high frequencies correspondingly.

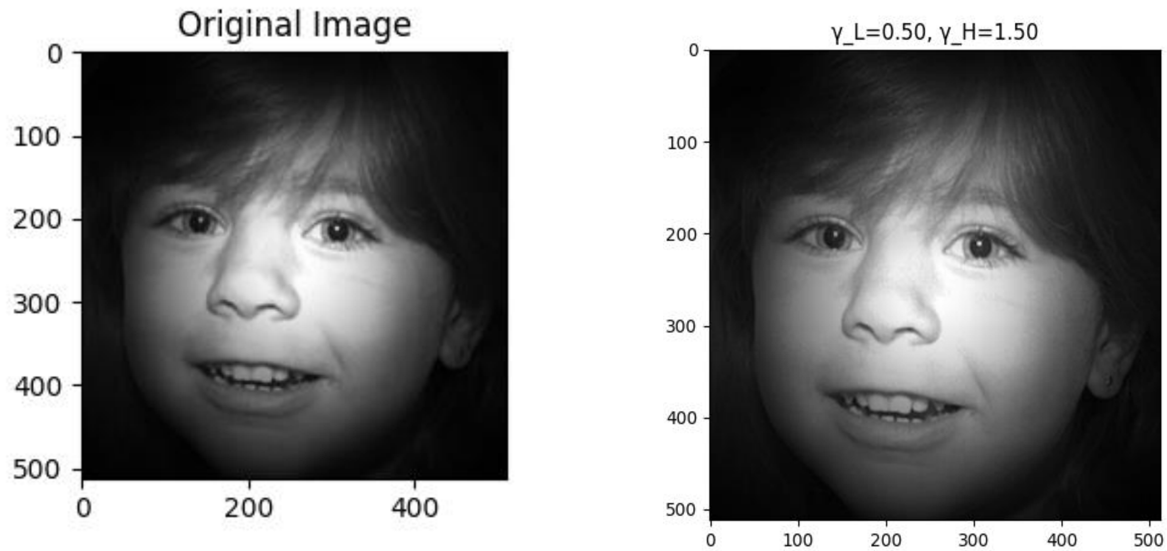


Figure 11: Represents the original image and result using D_0=1.8, c=1, γ_L=0.5 and γ_H=1.5.

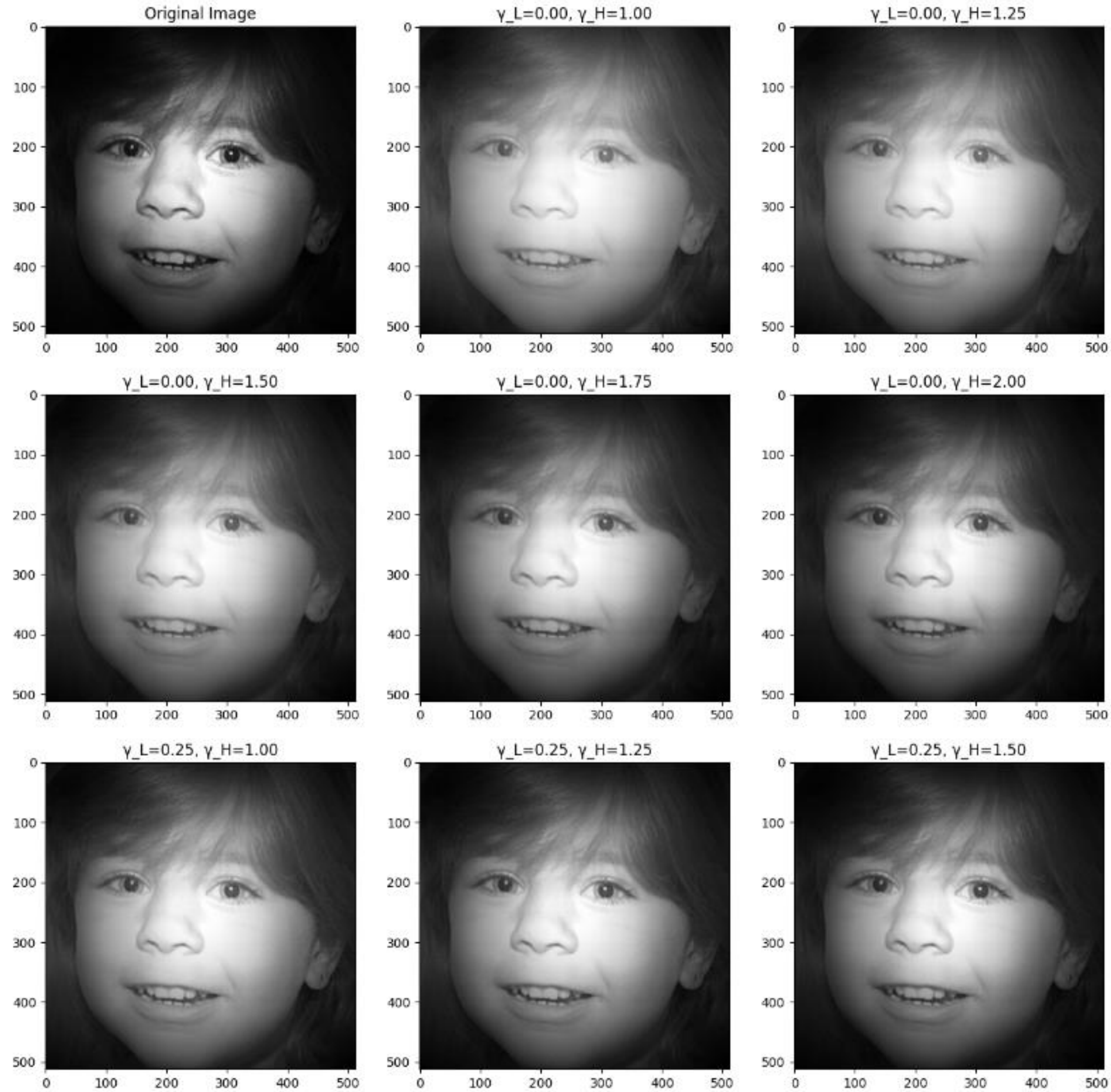


Figure 12: Represents the different combination of γ_L [0.0-1.0] and γ_H [1.0-2.0].

In 3.3 section, the given input image suffers from shading problem due to uneven illumination. The Homomorphic filter we implemented is separating the illumination and reflectance component by applying the logarithm function on image as mentioned in section 2.3. Then we apply a high pass filter as given here is high-frequency emphasis filter.

We perform an experiment with different parameter values. As a starting point, we choose D_0=1.8, c=1, γ_L=0.5 and γ_H =1.5. The result you can see in figure 11 which represents the input image and result image with low frequency 0.5 and high frequency 1.5. Then, keep the cutoff frequency the same and increase/decrease γ_L and γ_H. For example, we took different combinations of γ_L and γ_H, with γ_L taking values from [0.0-1.0] and γ_H taking values from [1.0-2.0]. The result you can see in figure 12 which represents the input image and different combinations of γ_L and γ_H. As per our experiment with these different values represents the different output and best result, we can see with γ_L=0.25 and γ_H=1.

References:

1) https://www.cse.unr.edu/~bebis/CS474/

2) R. Gonzalez and R. Woods Digital Image Processing, 4th edition, Pearson, 2018.