ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY
Department of Computer Science
Advanced Image Processing Project

Image Soothing using frequency domain filtering Project

Group Members:	Idno.
1. Arebu Maruf	pgr/40933/18
2. Gezahagn Teshoma	pgr/ 40930/18
3. Dida Gayo	pgr/ 41306/18
4. Biniam Alemayehu	pgr/40941/18
5. Rediet Dereje	pgr/40938/18

Submitted to: Dr Worku J
January 18, 2026

Frequency Domain Filtering

In digital image processing, filtering in the frequency domain is achieved by modifying the Fourier transform of an image. This experiment focuses on Lowpass Filters, which attenuate high-frequency components (such as noise and sharp edges) while preserving low-frequency components (smooth backgrounds). In this paper we implement image smoothing using Frequency domain filtering using different python library such as MATLAB library, NumPy and analysis the Results of the two Lowpass Filters.

Objectives: Demonstrate how noise and sharpness can be reduced in images using Lowpass Filters.

Key Concepts:

- Spatial Domain (Pixels)
- Frequency Domain(Waves)
- Fourier Transform

Background: The Frequency Domain

Spatial Domain

- The standard view where pixels represent light intensity at specific coordinates (x, y).

Frequency Domain

- Represents the **rate of change** in intensity.

- Low Frequencies: Smooth areas (backgrounds, skin, walls).
- High Frequencies: Sharp changes (edges, noise, detailed textures).

The Fourier Transform

Definition: A mathematical tool that decomposes an image into sine and cosine waves.

The Transformation

$$\text{Image}(x, y) \xrightarrow{\text{FFT}} \text{Spectrum}(u, v)$$

Key Insight:

- The Center of the spectrum represents the average brightness (DC component). Moving outward from the center represents higher frequencies.

The Convolution Theorem

Convolution in the Spatial Domain is equivalent to Multiplication in the Frequency Domain.

The Process:

1. Convert image to Frequency Domain (FFT).
2. Multiply by a Filter Mask (H).
3. Convert back to Spatial Domain (Inverse FFT).

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

Types of Lowpass Filters

Goal: To smooth the image by attenuating (reducing) high frequencies.

1. Ideal Lowpass Filter (ILPF) The theoretical "Perfect Cut". Sharp transition.

A 2-D lowpass filter that passes without attenuation all frequencies within a circle of radius D0 from the origin and “cuts off” all frequencies outside this circle is called an ideal lowpass filter (ILPF); it is specified by the function

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \le D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

- It creates a blurred image but introduces ringing artifacts (ripples) around edges due to the sharp discontinuity in the frequency domain.

Filter 1: Ideal Lowpass Filter (ILPF)

It keeps all frequencies inside a radius D_0 and deletes everything outside.

Pros

Simple to implement mathematically.

Cons

Causes **"Ringing"** (ripples/halos) around edges due to the Gibbs phenomenon.

Filter 2: Butterworth Lowpass Filter (BLPF)

- The BLPF of order **n** provides a smooth transition between passed and filtered frequencies.
- Creates a gradual reduction of high frequencies (a slope, not a cliff).

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

- D_0: Cutoff Frequency distance.
- n: Order of the filter (controls slope steepness).

Advantage

No sharp cutoff → **Minimal ringing artifacts.**

Methodology (The Algorithm)

- Read the input image.
- Pre-process: Convert to Grayscale if input image is color(FFT requires 2D input).
- Transform: Compute 2D FFT and shift zero-frequency to center.
- Create Mask: Generate Distance Matrix $D(u, v)$.
  - ✓ Construct an ILPF mask with cutoff frequency D_0.
  - ✓ Construct a BLPF mask with cutoff frequency D_0 and order n.
- Filter: Multiply FFT by Filter Mask ($F \times H$).
- Reconstruct: Inverse FFT to get processed image.

Applying the Filter

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

- $F(u,v)$: The shifted DFT of the image.
- $H(u,v)$: The filter mask (ILPF or BLPF).
- $G(u,v)$: The filtered frequency spectrum.

```
# --- 6. APPLY FILTERS ---
G_ideal_shifted = F_shifted * H_ideal
G_butter_shifted = F_shifted * H_butter
```

Experimental Setup

- Environment: Python (Anaconda Navigator / Jupyter).
- Libraries:
  - ✓ NumPy: For complex mathematics and matrix operations.
  - ✓ Matplotlib: For displaying results.
  - ✓ Scikit-Image: For image I/O.
- Input Data: High-resolution digital photograph.
- Parameters: Cutoff Frequency (D_0) = 50.

Code Implementation snippets (Python)

```python
# --- 1. READ IMAGE FROM DEVICE ---
filename = 'images.jpeg'
try:
    img = io.imread(filename)
    print("Image loaded successfully.")
except FileNotFoundError:
    print(f"ERROR: Could not find the file '{filename}'.")
    print("Please make sure the image is in the correct folder.")
    # We will stop the code here if no image is found
    raise
```

Code Implementation snippets (Python)

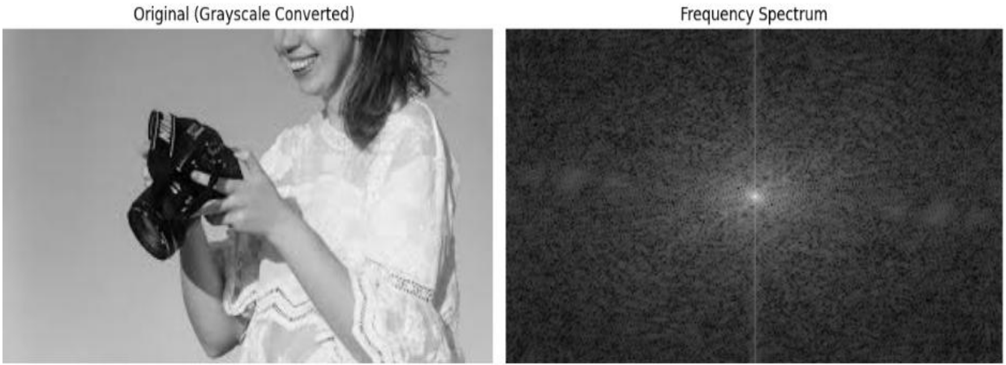
```
# --- 8. DISPLAY RESULTS ---
plt.figure(figsize=(12, 10))

plt.subplot(2, 2, 1)
plt.imshow(img, cmap='gray')
plt.title('Original (Grayscale Converted)')
plt.axis('off')

plt.subplot(2, 2, 2)
# Use Log scale to see the spectrum clearly
plt.imshow(np.log(1 + np.abs(F_shifted)), cmap='gray')
plt.title('Frequency Spectrum')
plt.axis('off')
```

Result 1: Frequency Spectrum

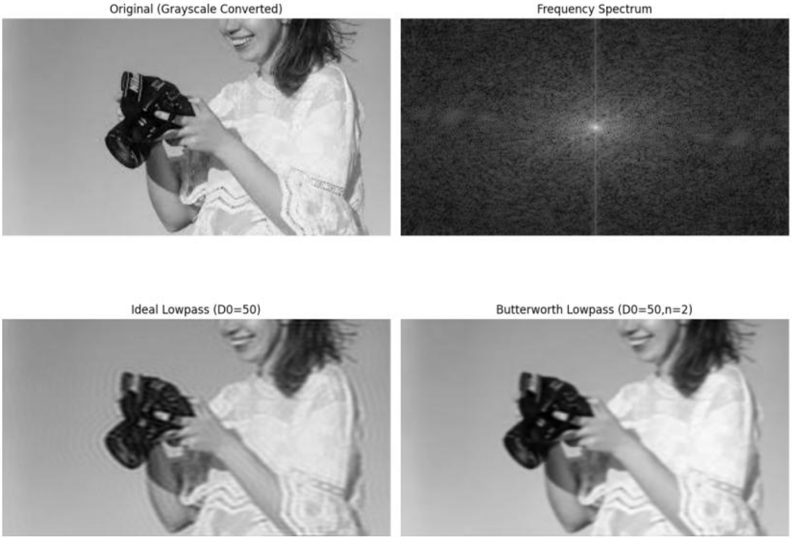
Observation: Center: Bright spot contains low frequencies (most image data). Outer Regions: Darker areas represent high frequencies (details/noise).



Result 2: Ideal Filter Output

Analysis:

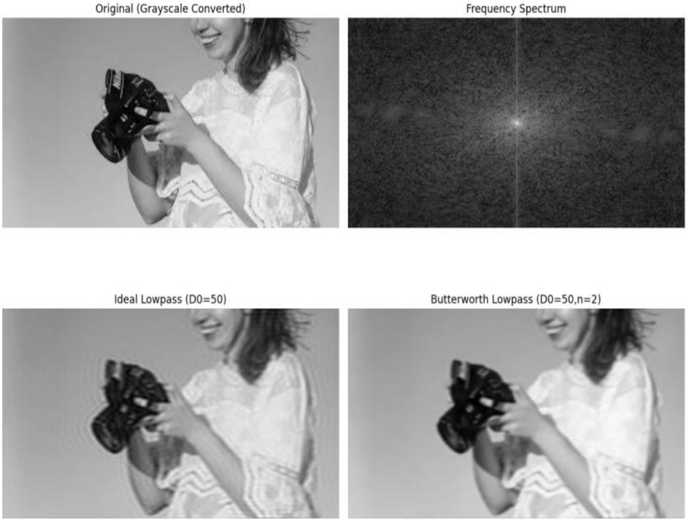
- Image is smoothed.
- Artifacts: noticeable wave-like patterns ("ringing") radiating from strong edges.



Result 3: Butterworth Filter Output

Analysis:

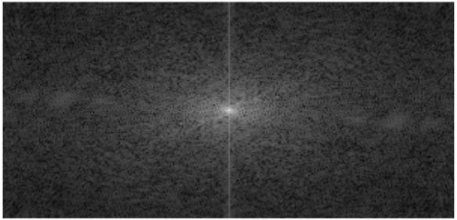
- Image is naturally blurred.
- Quality: Smooth transitions.
- No visible ringing.



Compare Ideal Filter Output and Butterworth Filter Output

Original (Grayscale Converted)

Frequency Spectrum



Ideal Lowpass (D0=40)

Butterworth Lowpass (D0=40,n=2)

Discussion & Comparison

Feature	Ideal Filter	Butterworth Filter
Sharpness	Very Blurry	Softly Blurry
Artifacts	High (Ringing)	Low / None
Transition	Abrupt (Cliff)	Gradual (Slope)
Use Case	Theoretical Study	Practical Editing

Table: Comparison of Lowpass Filters

Conclusion

Summary

- Frequency domain filtering allows us to manipulate global image characteristics efficiently by converting spatial data into waves.

Key Takeaway

- While Ideal filters are mathematically simpler, Butterworth filters offer superior visual results for image smoothing by avoiding ringing artifacts. With small border(n).

Thank You!

Questions?