M.Sc. (Five Year Integrated) in Computer Science (Artificial Intelligence & Data Science)

Fourth Semester

Assignment

21-805-0403: Digital Signal Processing

Submitted in partial fulfillment
of the requirements for the award of degree in
Master of Science (Five Year Integrated)
in Computer Science (Artificial Intelligence & Data Science) of
Cochin University of Science and Technology (CUSAT)
Kochi

Submitted by

ASHFAQ HUSSAIN M S
(80522003)

DEPARTMENT OF COMPUTER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY (CUSAT)
KOCHI-682022

OBJECTIVE

- The main objective of this assignment is to develop a Python code in-order to depict the famous **Parsevel's Theorem**.
- **Parseval's Theorem**, also known as the energy conservation theorem, is a fundamental concept in signal processing and Fourier analysis. It states that the total energy of a signal in the time (spatial) domain is equal to the total energy of its frequency (Fourier) domain representation.
- The following Python code aims to analyze the energy distribution of an image in both the spatial and frequency domains, and to demonstrate Parseval's theorem, which states that the energy of a signal remains the same in both domains.
    1. The grayscale image is loaded and normalized to [0, 1].
    2. Fast Fourier Transform (FFT) is applied to the flattened image to obtain its frequency spectrum.
    3. The energy in both spatial and frequency domains is calculated and compared.
    4. Parseval's theorem is verified by checking if the energies in both domains are close.
    5. The original image and its magnitude spectrum in the Fourier domain are displayed for visual inspection.

FOURIER TRANSFORM

The Fourier Transform is a mathematical technique used to decompose a function (or signal) into its constituent frequencies. It's named after the French mathematician Joseph Fourier, who first introduced the concept. The Fourier Transform is a powerful tool in various fields such as signal processing, image processing, communication systems, physics, and engineering.
Mathematically, the continuous Fourier Transform (for continuous signals) is defined as follows:

$$F(\omega) = \int_{-\infty}^{\infty} f(t).e^{-j\omega t}dt$$

where:

- $F(\omega)$ is the frequency domain representation of the signal f(t).
- ω is the angular frequency.
- t is the time variable.
- j is the imaginary unit ($j^2 = -1$).

For discrete signals, the Discrete Fourier Transform (DFT) is used. The DFT converts a sequence of N equally spaced samples of a function into a corresponding sequence of complex numbers, representing the signal in the frequency domain.

$$X[k] = \sum_{n=0}^{N-1} x[n].e^{-j2\pi(kn/N)}$$

The Fourier Transform provides a way to analyze the frequency content of a signal, revealing information about its periodic components, frequency distribution, and relationships between different frequencies. It is extensively used in applications such as audio and image processing, filtering, spectrum analysis, modulation, and many others.

PROGRAM

```
#Program to Implement Parsevel's energy theorem

import cv2
import numpy as np
import matplotlib.pyplot as plt

# Ensure the image path is correct
image_path = '/content/DSP-5.webp'

# Read the image in grayscale
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

# Check if the image is loaded correctly
if image is None:
    print(f"Failed to load image from {image_path}")
else:
    # Normalize the image to [0, 1]
    image = image / 255.0

    # Flatten the image to a 1D array
    f = np.array(image).flatten()

    # Apply Fast Fourier Transform (FFT)
    F = np.fft.fft(f)

    # Calculate energy in the spatial domain
    space_energy = np.sum(np.abs(f)**2)

    # Calculate energy in the frequency domain
    freq_energy = np.sum(np.abs(F)**2) / f.size

    # Print the energies
    print(f"Energy in the spatial domain: {space_energy}")
    print(f"Energy in the frequency domain: {freq_energy}")

    # Verify Parseval's theorem
```

```
print(f"Parseval's theorem holds: {np.isclose(space_energy,
freq_energy)}")

# Display the original image in the spatial domain
plt.figure(figsize=(10, 10))
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.show()

# Compute the discrete Fourier Transform of the image
fourier = cv2.dft(np.float32(image), flags=cv2.DFT_COMPLEX_OUTPUT)

# Shift the zero-frequency component to the center of the spectrum
fourier_shift = np.fft.fftshift(fourier)

# Calculate the magnitude spectrum of the Fourier Transform
magnitude_spectrum = 20 * np.log(cv2.magnitude(fourier_shift[:,:,0],
fourier_shift[:,:,1]))

# Scale the magnitude spectrum for display
magnitude_spectrum = cv2.normalize(magnitude_spectrum, None, 0, 255,
cv2.NORM_MINMAX, cv2.CV_8UC1)

# Display the magnitude spectrum in the Fourier domain
plt.figure(figsize=(10, 10))
plt.imshow(magnitude_spectrum, cmap='gray')
plt.title('Magnitude Spectrum')
plt.show()
```

OUTPUT

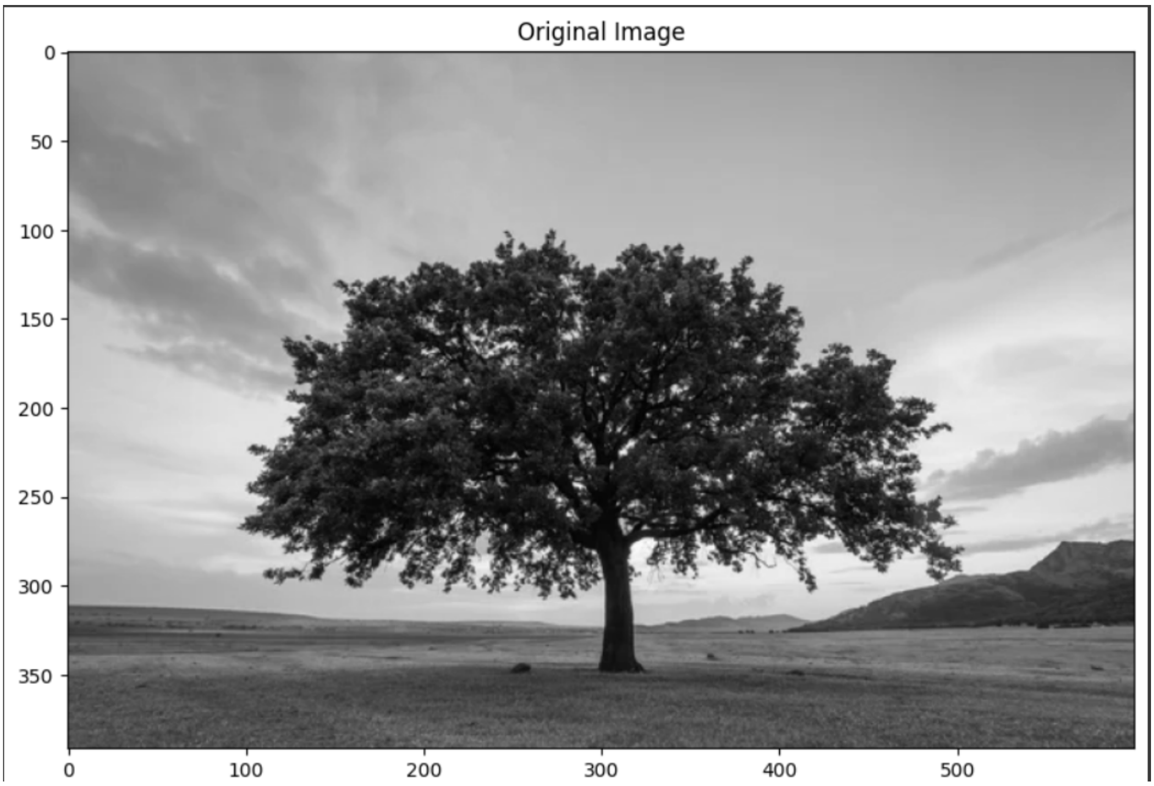
Parsevel's Energy Theorem

```
Energy in the spatial domain: 76606.42778931181

Energy in the frequency domain: 76606.42778931181

Parseval's theorem holds: True
```

Input Image



Transformed Image