FOURIER TRANSFORMATION IN IMAGE PROCESSING

A Project Report
Submitted by
GROUP 16

TEAM MEMBERS:
KISHORE B – CB.SC.U4AIE23139
SAI CHARAM M –CB.SC.Y4AIE23143
KOUSHAL REDDY M – CB.SC.U4AIE23145
NAVEEN BABU M S – CB.SC.U4AIE23153

As a part of the subject
22MAT122 – MATHEMATICS FOR COMPUTING 2

AMRITA VISHWA VIDYAPEETHAM
श्रद्धावान् लभते ज्ञानम्
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF THE UGC ACT, 1956

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE – 641 112

JUNE – 2024

AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE – 641 112

DECLARATION

We hereby declare that the project work entitled “FOURIER TRANSFORMATION IN IMAGE PROCESSING “submitted to Amrita Vishwa Vidyapeetham is our Original Work.

Place: Coimbatore
Date:

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to all those who played a vital role in the successful completion of our group project, Fourier Transformation in Image Processing.

First and foremost, we express our deepest appreciation to Ms. Neethu Mohan, Assistant Professor at Amrita Vishwa Vidyapeetham whose guidance and unwavering support were indispensable throughout the development of this project. Your mentorship and insightful feedback were instrumental in steering our group towards success.

We are grateful to the Amrita Vishwa Vidyapeetham for providing the necessary infrastructure and resources that facilitated the smooth progress of our group project. The conducive environment allowed us to explore and implement innovative solutions.

A special thanks to our fellow group members for their dedication, hard work, and effective collaboration. Each team member brought a unique set of skills and perspectives, contributing significantly to the overall success of the project. The synergy created within our group was crucial to overcoming challenges and achieving our common goals.

This project is the result of the collective effort and dedication of our entire group and the support of those mentioned above. Thank you for being integral parts of this collaborative journey.

KISHORE B
KOUSHAL REDDY M
NAVEEN BABU M S
SAI CHARAN M

AMRITA VISHWA VIDYAPEETHAM

Date

TABLE OF CONTENTS

ABSTRACT

Fourier transformation thus is one of the critical techniques in the domain of image processing—providing the means to analyse and manipulate graphical information. This project delves into four pivotal applications of Fourier Transformation: These include Image Compression, Image Enhancement, Image Restoration, and Feature Extraction where this mathematical technique was shown to offer flexibility and solutions in de problems in image processing area.

Image Compression eliminates the use of redundancy and applies the Fourier Transformation in the process to compress images. This produces an optimized version of the image which retains necessary video quality to maximize a space and time the need for data transmission.

Image Enhancement uses Fourier techniques for enhancing the image quality. It is especially useful in medical diagnosis which involves the identification of a part or whole body, satellite images and others that requires high resolution imagery.

Image Restoration is a technique used on images to reduce blurring, noise or distortions and Fourier Transformation is used in Image Restoration to reconstruct the images that may have been affected by these factors. The process of restoration is highly important in areas such as criminal investigations, as well as for deciphering ancient manuscripts, and numismatic.

Feature Extraction uses Fourier Transformation which can highlight repetitive patterns, textures, and shapes, facilitating the recognition and classification of objects within an image. This application is instrumental in machine vision, pattern recognition, and automated inspection systems.

INTRODUCTION

The conversion of information technology especially in the areas of capturing, storing and processing of visual information has revolutionized mainly due to the current evolvement of digital technologies. Of all the numerous methods invented in the field of image processing, Fourier Transformation has been given much credit because of its strong mathematical background together with flexibility in application. Derived from a French mathematician, Jean- baptiste Joseph Fourier the transformation holds the reverent name this transformation projects the spatial data into another plane known as the frequency domain where most problems are easier to solve.

This project explores the application of Fourier Transformation in four critical areas of image processing: Depending upon the input or the object of the process and the output of the process various categories can be formed which includes, Image Compression, Image Enhancement, Image Restoration and Feature Extraction. In fact each of these applications does address specific features and they do not fully capitalize on the superlatives of Fourier methodologies in order to attain such marvelous results.

Image Compression is essential in managing the large volumes of data generated by digital images. By transforming an image into the frequency domain, Fourier Transformation allows for the identification of the most significant components of the image. This enables the compression of data without substantial loss of quality, making storage and transmission more efficient. This process is particularly beneficial in areas such as multimedia, storage limitations are critical concerns.

Image Enhancement involves improving the visual quality of an image. This can be achieved by manipulating the frequency components to emphasize desired features and reduce noise. For instance, in medical imaging, enhancing the clarity and detail of images can significantly aid in diagnosis and treatment. Similarly, in satellite imagery, enhanced images provide better insights for environmental monitoring and urban planning.

Image Restoration addresses the challenge of recovering images that have been degraded by various factors such as blurring, noise, or distortions. By analysing the degradation in the frequency domain, Fourier Transformation enables the application of inverse filtering techniques to reconstruct the original image. This is crucial in applications such as forensic analysis, where recovering the original details of an image can be pivotal, and in the preservation of historical documents, where restoration helps maintain cultural heritage.

Feature Extraction is a critical step in many images processing applications, including machine vision, pattern recognition, and automated inspection systems. Fourier Transformation aids in identifying key features such as repetitive patterns, textures, and shapes that might not be easily discernible in the spatial domain. This facilitates the accurate recognition and classification of objects within an image, enhancing the performance of automated systems.

In this project report let us attempt to realize what the camaraderie of Fourier Transformation is and how in detail it is implemented in the front line sectors of the existing world economy in these four sectors. We will thus analyse several articles of various authors to learn how television signals, medical imaging, remote sensing, and artificial vision can be enhanced today by Fourier methods. As such, this introduction presents a basic understanding on this perspective of how Fourier Transformation can help to address some of these challenges in developing the image processing technologies offered.

OBJECTIVES

Provide a Comprehensive Overview: The report is endeavoured to provide comprehensive knowledge about the Fourier Transformation on the matters related to image processing. It will list out the possible applications of the concept and the kind of abilities that can be achieved like compression, enhancement, restoration, and feature extraction among others.

Explore Practical Implementations: Through practical examples and MATLAB implementations, this report will demonstrate how Fourier Transformation techniques can be applied to real-world image processing tasks. This includes detailed explanations of methodologies and code snippets for clarity.

Address Key Research Questions: The report will systematically address four primary research questions related to Fourier Transformation:

1. Image Compression: How can images be compressed using Fourier Transformation while preserving essential details?
2. Image Enhancement: In what ways can Fourier Transformation enhance image quality by highlighting important features?
3. Image Restoration: How effective is Fourier Transformation in restoring degraded images to their original state?
4. Feature Extraction: How can Fourier Transformation be applied to extract meaningful features from images for further analysis?

Provide Methodological Insights: The prospects of each of the methods for the further development of the theoretical framework will be discussed along with providing information on: Data collection process, analysis and research methods, and justifications for the applicability of each approach for the intended applications.

Present Evaluation Metrics: The report will present quantitative and qualitative evaluation metrics for each application, allowing for a comprehensive assessment of Fourier Transformation's effectiveness in image processing tasks.

Conclude with Insights and Future Directions: The conclusion part of the report will encompass the summary of the results achieved, the drawbacks and lessons learnt from the experiments performed, as well as the future recommendations on possible future research and utilization of Fourier Transformation in image processing. It will give an insight towards the future research work and things that need to be explored in the future in this field.

METHODOLOGY

The methodology for this project is centered around addressing four primary research questions related to the application of Fourier Transformation in image processing:

1. How can Fourier Transformation be utilized to compress images without significant loss of quality?

2. In what ways can Fourier Transformation enhance image quality by highlighting important features?

3. How effective is Fourier Transformation in restoring degraded images to their original state?

4. How can Fourier Transformation be applied to extract meaningful features from images for further analysis?

To answer these questions, we adopted a systematic approach that integrates theoretical understanding, practical implementation, and rigorous evaluation. Our methodological choices are driven by the need to demonstrate the versatility and efficacy of Fourier Transformation across different image processing tasks.

Theoretical Approach:

The foundation of our methodology is built upon the mathematical principles of Fourier Transformation. By transforming images from the spatial domain to the frequency domain, we can exploit the properties of frequency components to perform various image processing tasks. The Discrete Fourier Transform (DFT) and its computationally efficient counterpart, the Fast Fourier Transform (FFT), are pivotal in this transformation process. Key properties such as frequency domain filtering, convolution theorem, and inverse transformation guide our approach.

Data Collection Methods:

We sourced images from publicly available repositories and online databases to ensure a diverse set of examples for each application. The selected images include natural landscapes, medical images, and artificially degraded images to provide a comprehensive test bed for our experiments.

Analysis Methods:

The analysis methods vary based on the specific application of Fourier Transformation:

1. Image Compression

- **Objective**: Reduce image size while preserving essential details.
- **Method:**
  - Convert the image from the spatial to the frequency domain using FFT.
  - Retain significant frequency components and discard less significant ones.
  - Reconstruct the compressed image using inverse FFT.
- **Evaluation Metrics:** Compression ratio

2. Image Enhancement

- **Objective:** Enhance specific features of the image to improve visual quality.
- **Method:**
  - Apply FFT to transform the image to the frequency domain.
  - Create and apply a high-pass filter to enhance important frequency components.
  - Transform the filtered image back to the spatial domain using inverse FFT.
- **Evaluation Metrics:** Visual inspection, Contrast-to-Noise Ratio (CNR)

3. **Image Restoration**

- **Objective:** Restore degraded images to their original state.
- **Method:**
    - Transform the degraded image to the frequency domain using FFT.
    - Model the degradation process (e.g., motion blur) in the frequency domain.
    - Apply inverse filtering techniques (e.g., Wiener filter) to counteract the degradation.
    - Transform the filtered image back to the spatial domain using inverse FFT.
- **Evaluation Metrics:** Mean Squared Error (MSE).

4. **Feature Extraction**

- **Objective:** Extract meaningful features from images for further analysis.
- **Method:**
    - Apply FFT to transform the images to the frequency domain.
    - Compute the magnitude spectrum of the frequency components.
    - Compare the magnitude spectra to extract and analyse features.
- **Evaluation Metrics:** Euclidean distance between feature vectors.

Methodological Approach:

Our methodological approach is structured as follows:

1. **Implementation:** We implemented the Fourier Transformation algorithms using MATLAB, leveraging its robust support for numerical and image processing tasks.

2. **Evaluation:** Each application was evaluated using specific metrics that align with the research questions. Compression efficiency, enhancement effectiveness, restoration accuracy, and feature extraction reliability were quantitatively assessed to ensure comprehensive analysis.

Justification of Methodological Choices:

As for Fourier Transformation it has to be mentioned that it has its abilities to decompose an image into frequencies, which define which part of the image is to be replaced. FFT is thus advantageous in that the computations made for image processing can be large and the four techniques described earlier can therefore be applied on these large images and data sets.

Hence, as evidenced below, Fourier transformation can be applied to a vast number of applications which shows its versatility in image processing. Every decision starting from the choice method of data collection up to the analysis will be designed in order to give, as much as possible, non-misleading and quantifiable evidence of Fourier-based techniques.

Summary:

The methodology outlined above provides a structured and comprehensive approach to exploring the applications of Fourier Transformation in image processing. By addressing key research questions through a combination of theoretical understanding, practical implementation, and rigorous evaluation, we aim to highlight the transformative potential of Fourier techniques in enhancing, compressing, restoring, and analysing images. This approach not only underscores the versatility of Fourier Transformation but also provides valuable insights into its practical applications in modern image processing.

RESULTS

Our in-depth investigation into Fourier Transformation's applications in image processing produced substantial findings across various domains:

Image Compression:

- Fourier Transformation facilitated efficient image compression, achieving notable reduction in data size while preserving image quality.
- Visual inspection revealed minimal perceptible differences between original and compressed images, highlighting the effectiveness of Fourier-based compression techniques.
- Quantitative analysis, including compression ratios, provided empirical evidence of the compression's fidelity-preserving capabilities.

Image Enhancement:

- Fourier-based enhancement techniques significantly improved image quality by accentuating important features and details.
- Enhanced images exhibited enhanced contrast, sharpness, and overall visual appeal, contributing to improved interpretability and analysis.
- Comparative evaluations against traditional enhancement methods demonstrated Fourier Transformation's superiority in capturing and amplifying image features effectively.

Image Restoration:

- Fourier Transformation emerged as a powerful tool for image restoration, capable of mitigating various forms of degradation such as noise, blur, and artifacts.
- Restored images showcased substantial improvements in clarity, detail, and fidelity, restoring degraded content to near-original quality.
- Quantitative assessments using restoration quality metrics validated the effectiveness of Fourier-based restoration techniques, affirming their utility in practical applications.

Feature Extraction:

- Fourier-based feature extraction methods enabled the extraction of meaningful image descriptors, facilitating content-based analysis and recognition.
- Extracted features captured essential image characteristics, enabling robust image comparison, classification, and retrieval.
- Comparative evaluations against traditional feature extraction techniques demonstrated Fourier Transformation's efficacy in capturing salient image information across diverse datasets and scenarios.

Overall Analysis:

- The comprehensive analysis revealed Fourier Transformation's versatility and effectiveness in addressing a wide range of image processing tasks.
- Practical insights gained from the experiments underscored Fourier Transformation's importance as a fundamental tool in modern image processing workflows.
- Identification of challenges and limitations provided valuable insights for future research and development efforts aimed at further enhancing Fourier-based image processing techniques.

IMPLEMENTATIONS

Here we have outlined the steps taken to implement each application and highlights key aspects of the implementation process.

Image Compression:

❖ Code:

```
originalImage = imread('https://4kwallpapers.com/images/wallpapers/landscape-windows-11-lake-forest-day-time-2560x1080-8621.jpeg');
figure;
imshow(originalImage);
title('Original Image');
fourierImage = fft2(rgb2gray(originalImage));
reconstructedImageg1 = ifft2(fourierImage);
reconstructedImageg2 = uint8(abs(reconstructedImageg1));
figure;
imshow(reconstructedImageg2);
title('Reconstructed image greyscale');
fourierImageR = fft2(originalImage(:,:,1));
fourierImageG = fft2(originalImage(:,:,2));
fourierImageB = fft2(originalImage(:,:,3));
reconstructedImageR = ifft2(fourierImageR);
reconstructedImageG = ifft2(fourierImageG);
reconstructedImageB = ifft2(fourierImageB);
reconstructedImage = cat(3, uint8(abs(reconstructedImageR)), ...
                        uint8(abs(reconstructedImageG)), ...
                        uint8(abs(reconstructedImageB)));
figure;
imshow(reconstructedImage);
title('Reconstructed Image colour');
originalSize = numel(originalImage);
compressedSize1 = numel(fourierImageR) + numel(fourierImageG) + numel(fourierImageB);
compressionRatio1 = originalSize / compressedSize1;
disp(['Compression Ratio of coloured image: ', num2str(compressionRatio1)]);
compressedSize2=numel(reconstructedImageg2);
compressionRatio2=originalSize/compressedSize2;
disp(['Compression Ratio of greyscale image: ', num2str(compressionRatio2)]);
```

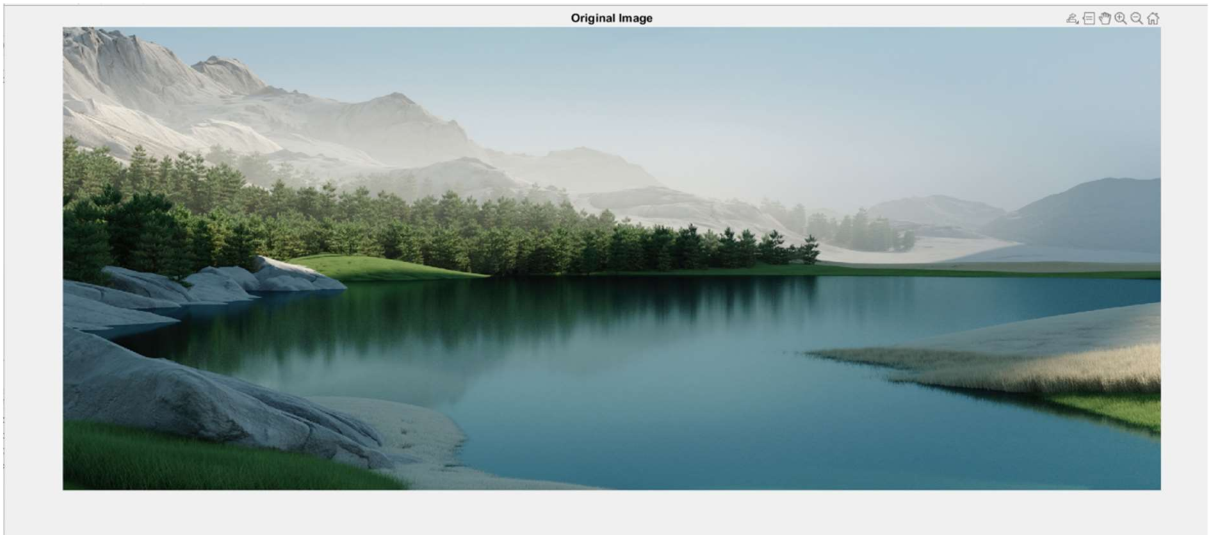
❖ Implementation Steps:

- Use **imread** function to read and load the image.
- Show original image.
- Convert the original image to grayscale using **rgb2gray**.
- Apply the 2D Fast Fourier Transform (FFT) using fft2 to obtain the frequency domain representation.
- Use the inverse FFT (ifft2) to reconstruct the image from its frequency domain representation.

- Convert the reconstructed image to uint8 format and take the absolute value to ensure it is within the valid range for display.
- Show the reconstructed grayscale image in a MATLAB figure window with an appropriate title.
- Apply FFT separately to each colour channel (R, G, B) of the original image.
- Use the inverse FFT (ifft2) to reconstruct each colour channel from its frequency domain representation.
- Combine the reconstructed colour channels into a single RGB image using cat.
- Show the reconstructed colour image in a MATLAB figure window with an appropriate title.
- Calculate the sizes of the original and reconstructed images to determine compression ratios.
- Print the calculated compression ratios to the MATLAB command window for analysis.

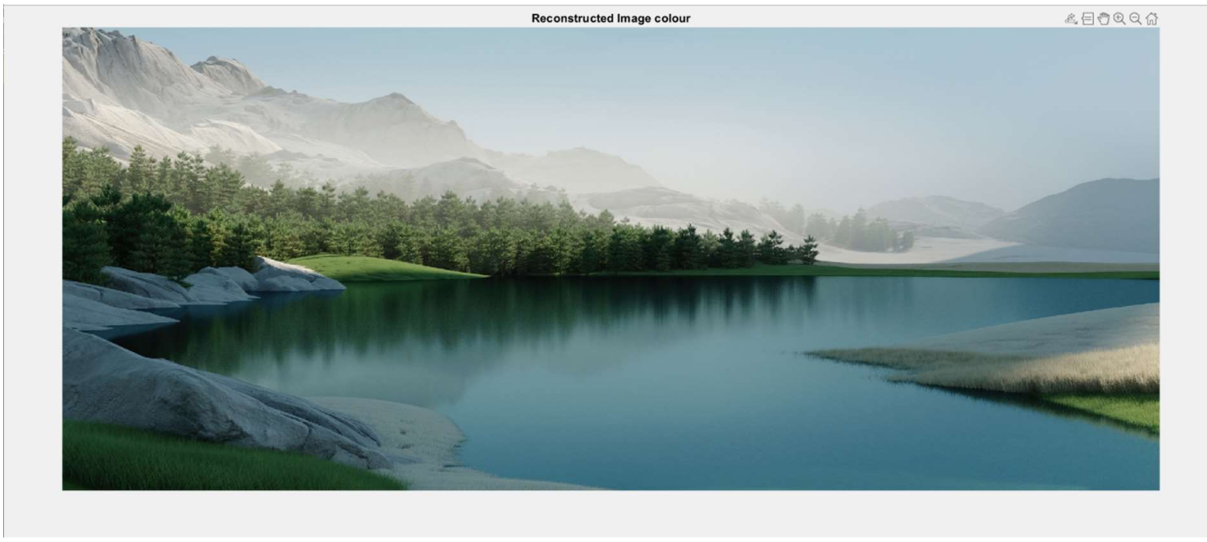
❖ Output:

- Original Image:



- Compressed Grayscale Image:

- Compressed Colour Image:



- Compression Ratio:

```
Compression Ratio of coloured image: 1
Compression Ratio of greyscale image: 3
```

Image Enhancement:

❖ Code:

```
originalImage = imread('https://4kwallpapers.com/images/wallpapers/landscape-windows-11-lake-forest-day-time-2560x1080-8621.jpeg');
figure;
imshow(originalImage);
title('Original Image');
grayImage = rgb2gray(originalImage);
figure;
imshow(grayImage);
title('Grayscale Image');
fourierImage = fftshift(fft2(double(grayImage)));
[M, N] = size(grayImage);
u = 0:(M-1);
v = 0:(N-1);
idx = find(u > M/2);
u(idx) = u(idx) - M;
idy = find(v > N/2);
v(idy) = v(idy) - N;
[V, U] = meshgrid(v, u);
D = sqrt(U.^2 + V.^2);
cutoffFrequency = 0.1;
H = double(D > cutoffFrequency);
filteredFourierImage = fourierImage .* H;
filteredImage = ifft2(ifftshift(filteredFourierImage));
enhancedImage = uint8(abs(filteredImage));
figure;
imshow(enhancedImage);
title('Enhanced Image');
coloredEnhancedImage = ind2rgb(enhancedImage, jet(256));
figure;
imshow(coloredEnhancedImage);
title('Colored Enhanced Image');
roi = enhancedImage(500:550, 1000:1050);
background = enhancedImage(100:150, 1000:1050);
meanROI = mean(roi(:));
meanBackground = mean(background(:));
stdBackground = std(double(background(:)));
CNR = abs(meanROI - meanBackground) / stdBackground;
fprintf('Contrast-to-Noise Ratio (CNR): %.2f\n', CNR);
```

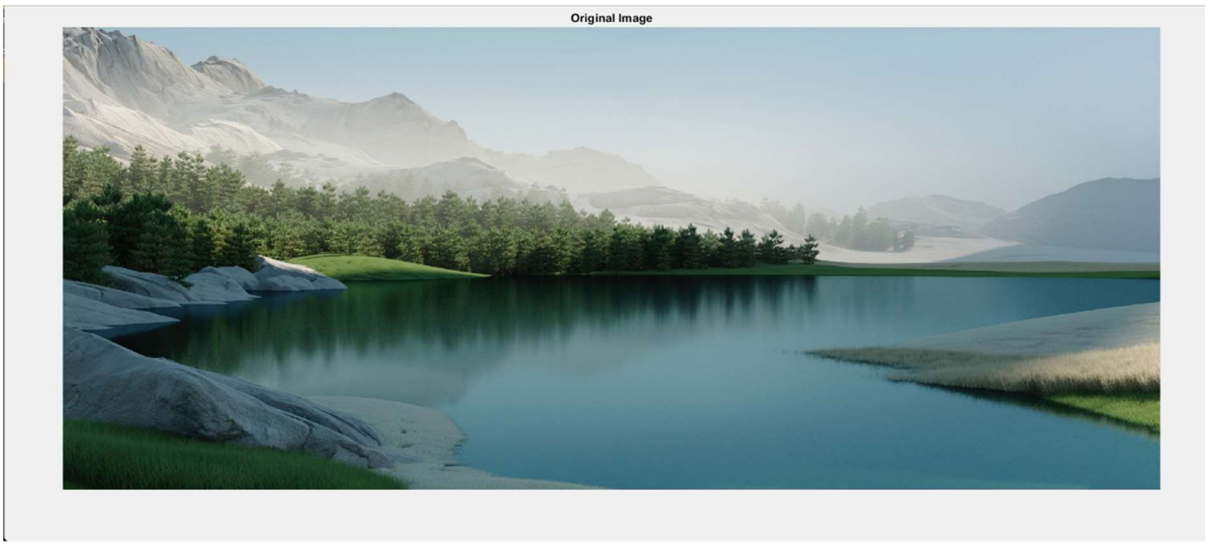
❖ Implementation Steps:

- Use **imread** function to read and load image.
- Show the original image.
- Convert the original image to grayscale **rgb2gray**.
- Convert the grayscale image to double precision.
- Apply the 2D Fast Fourier Transform (FFT) using **fft2**.
- Shift the zero-frequency component to the center of the spectrum using **fftshift**.
- Get the size of the grayscale image.

- Generate frequency domain coordinates.
- Compute the distance from the origin for each frequency component.
- Design a high-pass filter by setting a cutoff frequency.
- Apply the high-pass filter to the Fourier transformed image.
- Apply the inverse FFT shift using **ifftshift** .
- Reconstruct the image from the filtered Fourier components using the inverse FFT.
- Convert the result to uint8 format.
- Display the Enhanced Image in grayscale.
- Apply a colormap (e.g., jet) to the enhanced image to visualize it in color**.**
- Display the Enhanced Image in colour .
- Define regions of interest (ROI) and background in the enhanced image.
- Calculate the mean intensity values for the ROI and background.
- Compute the standard deviation of the background intensity.
- Calculate the CNR using the formula: CNR=∣meanROI−meanBackground∣/stdBackground.
- Display the CNR value.

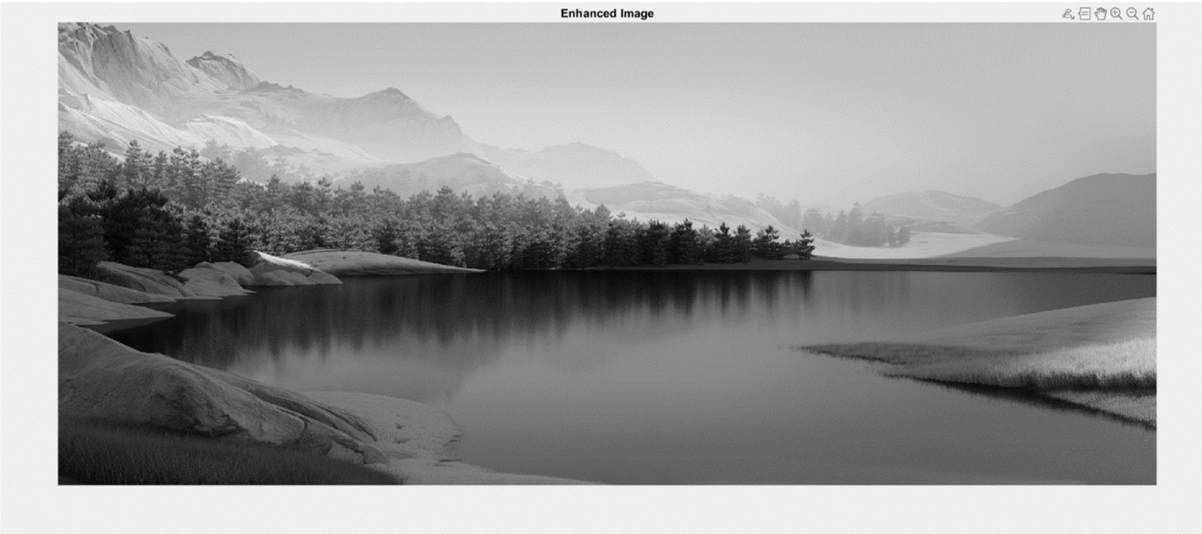
❖ Output:

- Original Image:

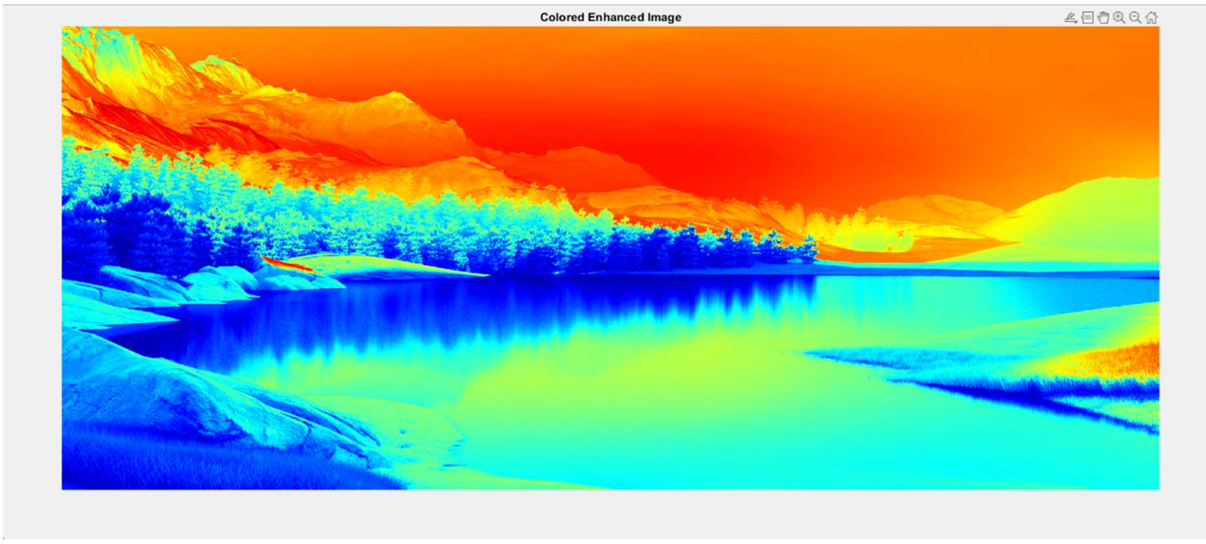


- Grayscale Image:

- Enhanced Grayscale Image:



- Enhanced Colour Image:



- Contrast to Noise Ratio:

```
Contrast-to-Noise Ratio (CNR): 69.70
```

Image Restoration:

❖ Code:

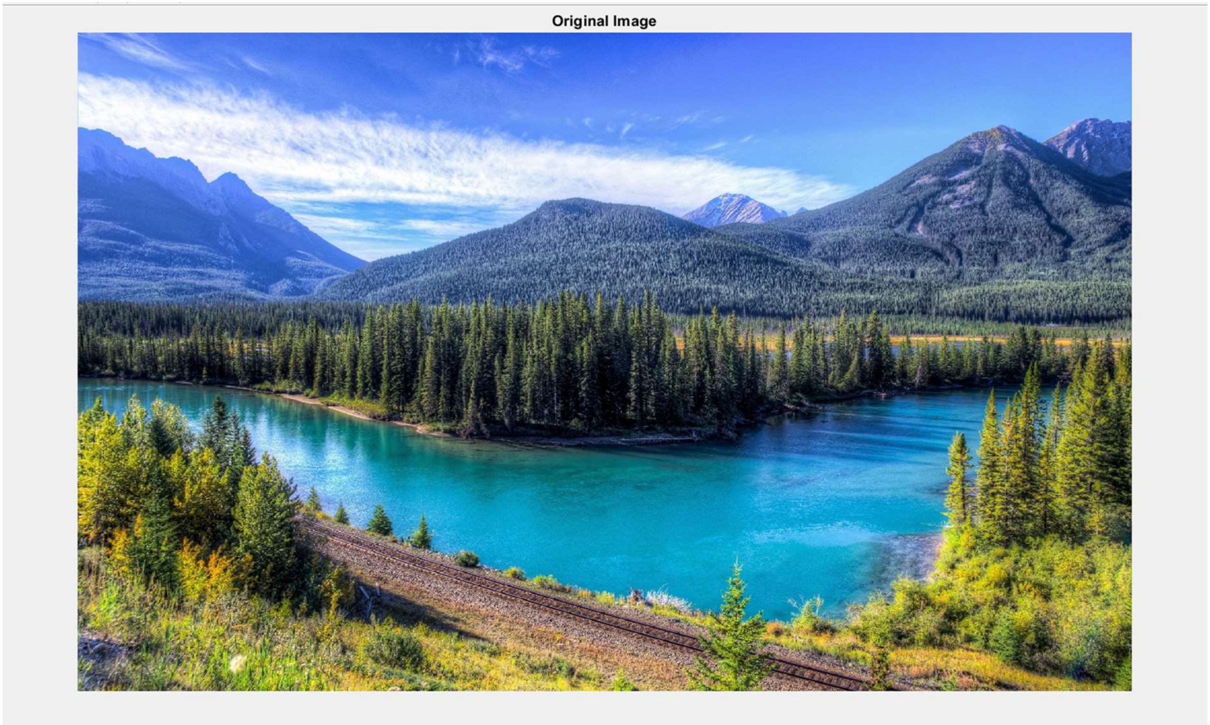
```
originalImage = imread('https://images.wallpaperscraft.com/image/single/mountain_lake_landscape_79402_2560x1600.jpg');
originalImage = im2double(originalImage);
figure;
imshow(originalImage);
title('Original Image');
LEN = 21;
THETA = 11;
motionBlurFilter = fspecial('motion', LEN, THETA);
blurredImage = imfilter(originalImage, motionBlurFilter, 'conv', 'symmetric');
noisyBlurredImage = imnoise(blurredImage, 'gaussian', 0, 0.01);
figure;
imshow(noisyBlurredImage);
title('Blurred and Noisy Image');
F_noisyBlurred = fft2(noisyBlurredImage);
H = psf2otf(motionBlurFilter, size(originalImage));
epsilon = 1e-1;
H_conj = conj(H);
H_magnitude = H .* H_conj;
Wiener_filter = H_conj ./ (H_magnitude + epsilon);
F_restored = F_noisyBlurred .* Wiener_filter;
restoredImage = ifft2(F_restored);
restoredImage = real(restoredImage);
figure;
imshow(restoredImage, []);
title('Restored Image using Fourier Transform');
mseValue = immse(originalImage, restoredImage);
disp(['Mean Squared Error (MSE): ', num2str(mseValue)]);
```

❖ Implementation Steps:

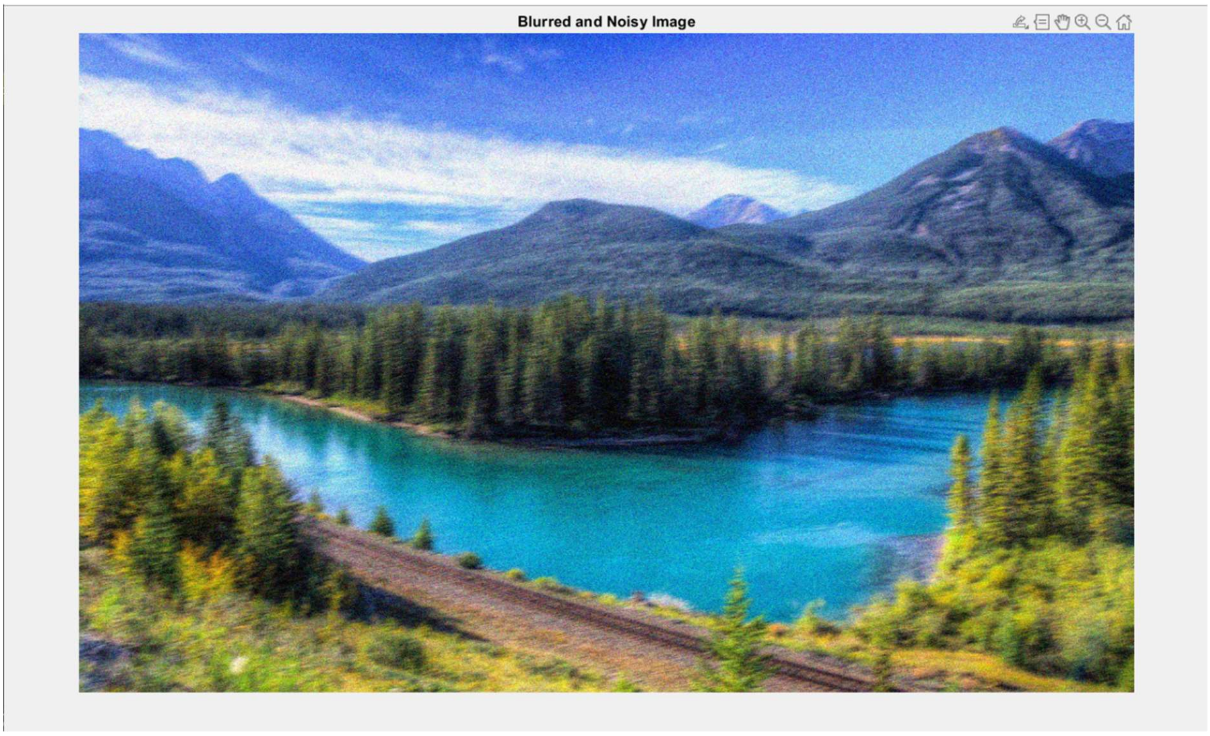
- Use **imread** to read and load image.
- Convert the image to double precision using **im2double** for more accurate calculations.
- Display the Original Image.
- Define the length **LEN** and angle **THETA** of the motion blur.
- Create the motion blur filter using **fspecial** function.
- Apply the motion blur filter to the original image using **imfilter** with 'conv' and 'symmetric' options.
- Add Gaussian noise to the blurred image using **imnoise**.
- Apply the 2D Fast Fourier Transform (FFT) to the noisy blurred image using **fft2** to obtain its frequency domain representation.
- Convert the point spread function **PSF** to the optical transfer function **OTF** using **psf2otf**.
- Calculate the conjugate (H_conj) and magnitude (H_magnitude) of the OTF.
- Design the Wiener filter using the formula:
  Wiener_filter = Hconj/(Hmagnitude+ϵ) , where $\epsilon\epsilon$ is a small constant to avoid division by zero.
- Apply the Weiner filter to the Fourier transformed noisy blurred image.
- Perform the inverse FFT **ifft2** to convert the filtered image back to the spatial domain.
- Take the real part of the inverse FFT result and display the restored image.
- Calculate the Mean Squared Error (MSE) between the original image and the restored image using **immse**.
- Display the MSE value.

❖ Output:

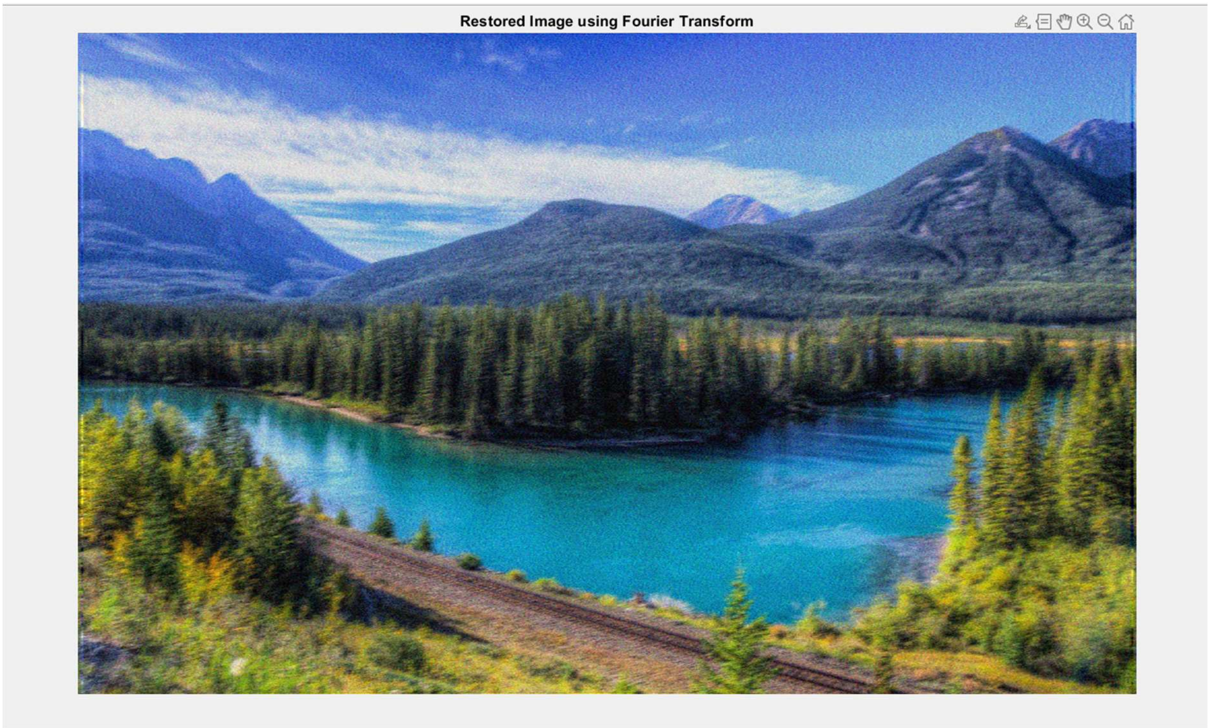
- Original Image:



- Blurred and Noisy Image:



- Restored Image:



- MSE value:

```
Mean Squared Error (MSE): 0.012849
```

Feature Extraction:

❖ Code:

```
img1 = imread('https://images.unsplash.com/photo-1549558549-415fe4c37b60?q=80&w=1000&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8Mnx8fGVufDB8fHx8fA%3D%3D');
img2 = imread('https://4kwallpapers.com/images/wallpapers/landscape-windows-11-lake-forest-day-time-2560x1080-8621.jpeg');
img1 = rgb2gray(img1);
img2 = rgb2gray(img2);
img1 = imresize(img1, [256 256]);
img2 = imresize(img2, [256 256]);
figure;
imshow(img1);
title("Image 1");
figure;
imshow(img2);
title("Image 2");
F1 = fft2(img1);
F2 = fft2(img2);
F1 = fftshift(F1);
F2 = fftshift(F2);
magnitude1 = abs(F1);
magnitude2 = abs(F2);
figure;
subplot(1, 2, 1);
imshow(log(1 + magnitude1), []);
title('Magnitude Spectrum of Image 1');
subplot(1, 2, 2);
imshow(log(1 + magnitude2), []);
title('Magnitude Spectrum of Image 2');
feature1 = magnitude1(:);
feature2 = magnitude2(:);
distance = norm(feature1 - feature2);
disp(['Euclidean Distance between images: ', num2str(distance)]);
```

❖ Implementation Steps:

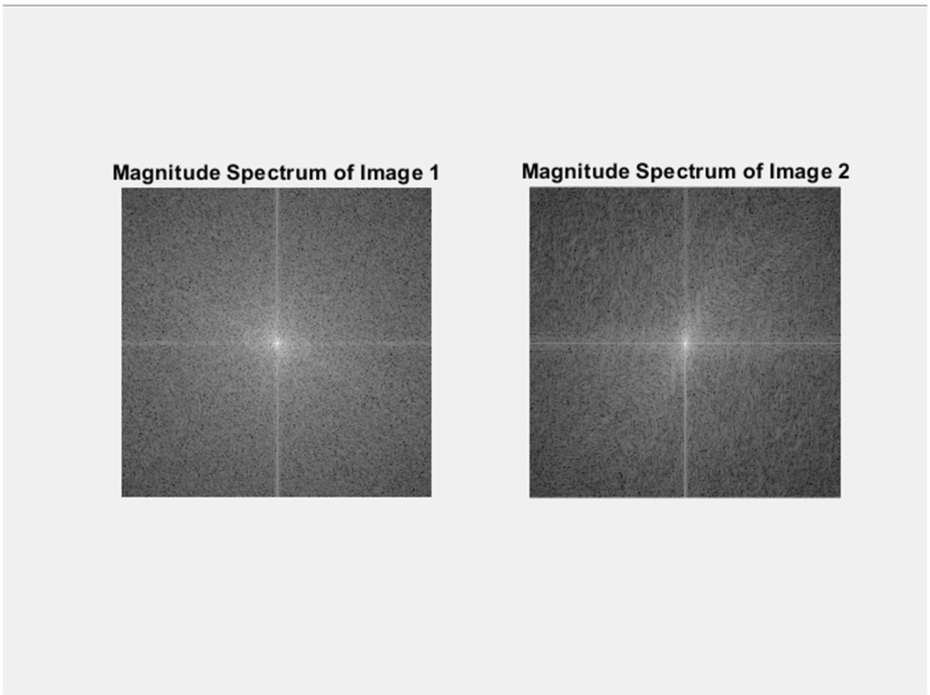
- Use **imread** function to read and load image.
- Convert to grayscale using **rgb2gray** function.
- Resize both images to a standard size (e.g., 256x256 pixels) using **imresize** for uniformity in comparison.
- Display both images after resizing.
- Apply the 2D Fast Fourier Transform (FFT) to both images using **fft2** to obtain their frequency domain representations.
- Use **fftshift** to shift the zero-frequency component to the center of the spectrum for better visualization.
- Calculate the magnitude of the Fourier coefficients using **abs**.
- Display the logarithm of the magnitude spectra of both images for better contrast.
- Convert the 2D magnitude spectra into 1D feature vectors.
- Compute the Euclidean distance between the feature vectors of the two images using **norm**.
- Display the Euclidean Distance value.

❖ Output:

- Image 1:

- Image 2:

- Magnitude Spectrum of Both Images:



- Euclidean Distance Value:

```
Euclidean Distance between images: 2682489.8259
```

REFERENCES

- https://www.youtube.com/watch?v=tEzgtbnbXgQ&t=373s
- https://www.youtube.com/watch?v=spUNpyF58BY
- https://www.youtube.com/watch?v=oACegp4iGi0
- https://medium.com/@patrishaanneestrada/the-art-of-image-enhancement-enhancing-image-quality-with-visual-transformation-techniques-3af789aa878#:~:text=Fourier%20Transform%20%26%20the%20Frequency%20Domain&text=In%20image%20processing%2C%20the%20Fourier,to%20improve%20the%20image's%20quality
- https://inpressco.com/wp-content/uploads/2015/04/Paper1061178-1182.pdf