Republic of the Philippines
POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
Office of the Vice President for the Campuses
PARAÑAQUE CAMPUS

Evaluating Image Quality: A Critique of Error Analysis in Fourier-Based Image Processing

Submitted by:
Baguio, Kyla
Bon, Kirt Deyniel
Del Monte, Mikaela
Martin, Jayson
Ravago, Josh Harnet

BS Computer Engineering 2-1

Submitted to:
Doc Frederico Roy

ABSTRACT

Image processing is vital across numerous technological sectors, from medical diagnostics to screen reading.[1] Central to this field is the transformation of image data between spatial and frequency domains, often facilitated by the Fourier Transform (FT) and its efficient variant, the Fast Fourier Transform (FFT).[2] While effective, these methods assume global signal stationarity, limiting their performance with localized or non-stationary image features.[3] The Fractional Fourier Transform (FRFT) addresses this limitation by introducing a fractional order parameter, enabling analysis in intermediate domains and offering enhanced adaptability for complex signals.[4]

This paper critically reviews a study from the JNNCE Journal of Engineering & Management (January 2022) that investigates the impact of FT, FFT, and FRFT on image reconstruction accuracy, using Mean Squared Error (MSE) as a performance metric. The study reported a significant MSE reduction from 597.8 to 1.55 with increasing fractional order, suggesting an inverse relationship between FRFT order and image error, thereby positioning FRFT as a superior method for minimizing reconstruction errors.

Our critique delves into the study's methodological strengths and weaknesses. While the reported numerical gains in MSE reduction are impressive, particularly the clear inverse relationship between fractional order and MSE in most cases (e.g., dog images showing MSE reduction from 428.7762 to 3.9653), the research is primarily theoretical and simulation-based, lacking real-world validation. We identify crucial gaps, including the absence of a comprehensive evaluation of FRFT's computational complexity, practical implementation issues, and performance on diverse, real-world datasets. The study's limited image diversity (dog and snake images) and the observed instability at extreme fractional orders (e.g., MSE spikes to 7796.3198 at order 0.50) further raise questions about generalizability and algorithmic robustness.

Despite these limitations, the study significantly advances the understanding of fractional FFT's utility by highlighting its content-dependent performance and the critical role of pre-processing. However, the lack of direct comparisons to non-fractional FFT or other established enhancement techniques prevents a conclusive validation of FRFT's superiority. Future research must address these gaps through expanded datasets, robustness testing, and comprehensive comparative benchmarks to fully establish FRFT's value and feasibility in operational image processing systems.

I. INTRODUCTION

Background of the Study

Image processing is crucial in various tech sectors, such as medical diagnostics and screen reading (Xiong, 2024). A primary task in this field is to switch image data between spatial and frequency domains, which helps to highlight and manipulate features that might not be easily seen in the original image. The Fourier Transform (FT) serves as a key tool for this, providing a mathematical approach to break down an image into its frequency components. Its quicker version, the Fast Fourier Transform (FFT), is widely used because it efficiently performs this transformation on digital data.

However, while FFT works well in many cases, it assumes that image signals are globally stationary(Manoj, 2024). This can lead to less-than-ideal results when dealing with images that have localized or uneven features. To overcome this, the Fractional Fourier Transform (FRFT) was created as an extension of the classical FT. The FRFT introduces a fractional order parameter, allowing for analysis in intermediate domains between spatial and frequency representations, which can enhance adaptability in complex signal environments.

The paper being reviewed, published in the JNNCE Journal of Engineering & Management in January 2022, looks into how FT, FFT, and FRFT affect image reconstruction accuracy. By using Mean Squared Error (MSE) as a performance measure, the authors found a remarkable drop in MSE from 597.8 to 1.55—as the fractional order increased, showing a strong inverse relationship between the order of FRFT and image error. This study highlights FRFT as a more effective method for reducing reconstruction errors, with potential applications in image encryption, signal filtering, and optical processing.

While the numerical findings in the paper are impressive, the study is mostly theoretical and based on simulations. It doesn't fully delve into the computational complexity of FRFT or evaluate its performance on real-world image datasets. Furthermore, practical implementation issues, such as algorithmic efficiency and hardware constraints, are only briefly mentioned. These gaps raise important questions about the broader applicability and feasibility of FRFT in operational image processing systems.

This critique will examine the methodological strengths and weaknesses of the study, assess the clarity and reliability of its findings, and evaluate the significance of its contribution to the field. It aims to determine whether the claimed advantages of FRFT over traditional Fourier-based methods are well justified and whether the research offers a meaningful advancement in both theory and application.

Problem Statement

The Fourier Transform (FT) and Fast Fourier Transform (FFT) are commonly used in image processing, but they have their limitations when it comes to localized or non-stationary features because they take a global approach. The Fractional Fourier Transform (FRFT) might just be the answer, as it allows for transformations in intermediate domains, which could provide more flexibility in analyzing intricate image structures. However, it's still unclear how it truly compares to traditional methods in practical scenarios.

The research paper displays considerable MSE reduction when using higher FRFT orders which suggests improved image quality results. The study presents encouraging findings but the results come from simulations without any real-world validation. Furthermore the research fails to evaluate the computational complexity and practical feasibility of using FRFT. Such findings create essential questions regarding the practical usefulness of this method for real-world image processing operations.

Literature Review

This section examines previous research on the function of FT, FFT, and FrFT in image processing. The objective is to comprehend the performance of these methods in terms of image quality and error reduction, particularly when used for tasks like machine learning-based prediction, filtering, enhancement, and encryption. This review facilitates a comparative analysis of the methods' efficacy in reducing image processing errors by analyzing their respective advantages and disadvantages.

Foreign

The study by Xiong (2024) uses Fast Fourier Transform (FFT) methods to analyze different frequency-domain filters' effects on image quality. The research investigates the impact of Low-Pass, Gaussian, Smoothing, and High-Boost Sharpening filters. The research demonstrates that combining these filters enhances both denoising and sharpening functions to achieve better processing results. This relates to our evaluation of how FFT affects image clarity and introduces or minimizes processing error.

Wang (2024) explores how fractional Fourier transform (FrFT) functions for image restoration alongside filtering of defocused and unclear images. FrFT adapts to different scenes and conditions through its adjustable parameters to deliver better accuracy and efficiency. The paper further explains its application in signal processing together with optical imaging and quantum information as well as general image filtering. Its ability to improve filtering in fuzzy or defocused images is directly related to our interest in how FrFT may reduce reconstruction error compared to FT or FFT.

The FrFT demonstrates enhanced performance compared to conventional Fourier techniques according to Yang (n.d.) when analyzing noise patterns across images. The paper demonstrates that decrypted images achieve near-perfect reconstruction through mean squared error (MSE) values reaching 0.00. The findings validate the application of FrFT for secure image encryption and decryption operations.

Yu et al. (2023) develop the Deep Fractional Fourier Transform (DFRFT) which merges spatial analysis with frequency analysis for deep learning applications. The fractional order approach allows better handling of non-stationary image signals compared to traditional Fourier methods. The Multi-order FRFC (MFRFC) of the authors demonstrates better performance in both denoising and super-resolution tasks through the integration of multiple domains which proves DFRFT's effectiveness for joint spatial-frequency processing. Their performance results help highlight how fractional transforms can manage error more effectively in complex image processing tasks.

The enhanced Fourier Transform (FT) method proposed by Cabrel et al. (2024), together with Wavelet Packet Decomposition (WPD), enables improved analysis of non-stationary signals. The advantage of using FT in conjunction with WPD (using Daubechies-4 wavelets) lies in a better time-frequency resolution, with a 12% Mean Squared Error reduction and a 3% Peak Signal-to-Noise Ratio increase compared to classical FT. Another advantage is transient detection accuracy at an impressive 92% level, clearly indicating the usefulness of simultaneous application of time-frequency techniques. These results provide a good reason to suggest that hybrid methods will improve non-stationary signal processing using traditional FT and related transforms, such as the Fractional Fourier Transform (FRFT).

Yang et al. (2021) introduced a combination of 2D Empirical Wavelet Transform and Nonlocal Multiscale Fractional Differential to evaluate rock fragment sizes from UAV images. The wavelet-Fourier hybrid method does measure particle size with an accuracy of more than 90%, an achievement superior to those of traditional segmentation methods. The value of this study lies in its demonstration of the successful application of combining wavelet and Fourier transforms in handling noisy and non-stationary image data, thereby contributing observations toward future refinements of FT/FFT applications in correlated domains.

In another study, Garcia et al. (n.d.), fastly computed the Fractional Fourier Transform (FRT) with the help of Fast Fourier Transform (FFT). Therefore, the processing time required for large datasets has been drastically reduced with respect to conventional methods without losing any accuracy. This method efficiently used two FFTs with corresponding phase adjustments in its application to optical and engineering signal processing, image analysis, and simulation of light behavior. Hence, this method is an effective computationally friendly technique for real-life applications requiring FRT calculations.

The image enhancement approach according to Kungumaraj (2020) applies FFT together with Retinex algorithms and histogram equalization. The FFT system operates frequency-domain tasks while Retinex algorithm separates light from reflectance to enhance brightness and color and the histogram equalization improves contrast. According to MATLAB testing the proposed method shows superior performance compared to conventional Multi-Scale Retinex both in visual quality and processing speed. This supports our critique by showing practical ways in which FFT handles error and contributes to visual quality improvements.

Recent progress in image deblurring techniques focuses mainly on operating in both frequency and spatial domains using the Fourier transform (FT) properties. However, their performance is limited due to the dependency of FT on stationary signals and its lack of capability to extract spatial-frequency properties. A novel approach based on the Fractional Fourier Transform (FrFT) according to Paul et al. (2024) a unified spatial-frequency representation leveraging both spatial and frequency components simultaneously, making it ideal for processing non-stationary signals like images. This shows that applying spatial-frequency displays a refined and efficient latent clear image processing.

Local

Ambita (2022) introduces a technique that integrates multispectral imaging with machine learning to estimate blood loss through surgical gauze without direct contact. The model improved its prediction accuracy by applying the Fast Fourier Transform (FFT) to analyze spatial blood distribution patterns. The combination of this method with XGBoost allowed for immediate inexpensive predictions which serve as a substitute for conventional measurement methods. The research shows how FFT enhances medical imaging feature extraction which connects to image-based estimation error reduction approaches.

Synthesis

The application of Fourier Transform (FT), Fast Fourier Transform (FFT) as well as Fractional Fourier Transform (FrFT) techniques to enhance the image quality and precision have gained attention recently in the field of image processing. These methods are applicable at all levels; for instance, noise filtering, sharpening, removing blur from images, and super-resolution. Research shows that FrFT outperforms FT/FFT in non-stationary signal processing and reconstruction error minimization. Efficiency and accuracy are improved further by hybrid methods which include wavelet transforms or deep learning with Fourier techniques. While there is abundant research exploring these advanced concepts internationally, there is very little done locally investigating FFT's most basic functions.. This presents an unexplored area for sophisticated studies dialed into local needs, especially those combining FrFT with AI-powered tools aimed for better results in image processing.

II. METHODS

Numerical Methods

1.1 Fourier Transform

The FT (Fourier Transform) is a mathematical tool for transforming a signal from the time domain (the space domain) to the frequency domain, expressing a function that meets certain conditions as a trigonometric function or a linear combination of their integrals. The FT equation is given as:

$$F(x) = \int_{-\infty}^{+\infty} f(x)e^{-i\omega t}dx \tag{1}$$

1.2 Fractional Fourier Transform

By integrating the relativity of independent time and frequency domains, the FT displays the frequency components of the whole signal. This enables the exploration of deterministic signals and smooth signals. Due to mostly non-smooth signals existing in the natural world, a variety of latest time-frequency analysis theories and methods have been [ut forward to solve this problem, including FrFT(Fractional Fourier Transform).

Linear operators are used to describe FrFT. In the time-frequency plane, the FrFT operator can be rotated by an arbitrary angle, if the Fourier Transform is considered as a counterclockwise rotation of 90 degrees from the time axis to the frequency axis. In this sense, FrFT is considered a generalized FT.

The definition of the FrFT varies under different perspectives. The definition of the one-dimensional and two-dimensional FrFT in terms of the integral form will be given as follows. The basic definition of the FrFT is given from the point of view of the linear integral transform, which reflects the most basic properties of the FrFT and is the most rigorous form of mathematical definition of the FrFT.

1.2.1 Definition of a 1D FrFT

Definition 1: The ρ-order $FrFT$ of the function $f(x)$ is:

$$g(u) = F^{\rho}\{f(x)\} = \int_{-\infty}^{+\infty} f(x)K_{\rho}(u,x)f(x)dx \tag{2}$$

Where K_{ρ} is the FrFT kernel function defined as following:

$$K_{\rho}(u,x) = \begin{cases} \sqrt{\frac{1-jcota}{2\pi}}exp[j(\frac{x^2+u^2}{2tana} - \frac{xu}{sina})] & a \neq n\pi \\ \delta(x-u), & a \neq 2n\pi \\ \delta(x-u), & a = (2n \pm 1)\pi \end{cases} \tag{3}$$

In the formula $a = p\frac{\pi}{2}$, $p \neq 2n$. The order p is taken with 4 as the least positive period. $p = 0$ is the original. $p = 1$ is the FT, and when p is transformed between [0,1]. It is called FrFT. The fractional order FT is considered to be a generalized FT, as stated by the definition of the FrFT. The FrFT is a form of intermediate state transition between the function and the FT. To express the degree of such an intermediate state, the order parameter can be used through the continuous fractional transform, which also takes into account the signal time and frequency domain [3].

1.2.2 Definition of a 2D FrFT

Definition 2: Suppose that for a two-dimensional signal of size $M \times N$, which called $f(x, y)$, and $1 \leq x \leq M, 1 \leq y \leq N, M \geq N$. Then the 2D FrFT is defined as:

$$F(u, v) = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} H_{p_1 p_2}(x, y, u, v) f(x, y) \quad (4)$$

$$H_{p_1 p_2} = H_{p_1}(x, u) H_{p_2}(y, v) \quad (5)$$

In the equation,

$$H_{\varphi_x} = \{\frac{1}{2\pi}\sqrt{1 - cot\alpha}\sqrt{1 - jcot\beta}\, exp[i\pi(x^2 cot\varphi_x - 2xucsc\varphi_x + u^2 cot\varphi_x)] \quad (6)$$

$$\varphi_x \neq n\pi \qquad \delta(x - u)$$

$$\varphi_x = 2n\pi \qquad \delta(x + u),$$

$$\varphi_x = (2n + 1)\pi$$

where p_1, p_2 are the transform order of the FrFT. $\varphi_x = p\frac{\pi}{2}$ is the FrFT of the transform angle. The kernel functions $H_{p_1}(x, u)$ and $H_{p_2}(y, v)$ have the same form. When $p_1 = p_2 = 0$, it is the original signal, and when $p_1 = p_2 = 1$, the 2D FrFT is 2D FT.

As stated below, it is possible to convert the 2D Discrete FrFT process into two 1D Discrete FrFTs.

- First, A one-dimensional discrete FrFT is applied to the column vectors of a two-dimensional discrete signal to obtain the transform result F_1.
- Then, the one-dimensional discrete FrFT is applied to the row vector of F_1 to obtain the transform result F_2.
- Transposing F_2 is the result of the 2D discrete FrFT.

1.3 Mean Squared Error

To quantify the effect of FrFT on the filtered image, a computation of the MSE between the original image and the FrFT-processed version. Then the MSE is defined as:

$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [I(i, j) - I'(i, j)]^2 \quad (7)$$

Where:

- I is the original image
- I' is the transformed image

- m, n are the image dimensions

This metric allows objective analysis of the distortion or changes introduced by the combined filtering and FrFT operations.

Implementation

In this research, an application was developed with a custom image processing program using Python, integrating both graphical user interaction and advanced image transformation techniques. The primary objective of the implementation was to evaluate the effects of the Fractional Fourier Transform (FrFT) on filtered images and measure these effects quantitatively using Mean Squared Error (MSE).

2.1 Development Environment

The application was implemented using the following key libraries:

- **OpenCV** (cv2) – for image reading, filtering, and conversion operations.
- **NumPy** – for array manipulation, matrix operations, and numerical calculations.
- **Tkinter** – for building the GUI interface, allowing user-friendly interaction.
- **Custom FrFT functions** – for performing 1D and 2D separable FrFT transformations on images.
- **Custom MSE function** – for validating the value of the image dimensions.

2.2 Image Preprocessing with Filters

The image will be processed. However, in order to study the behavior of FrFT under different image conditions, a basic image filters using OpenCV was applied as follows:

- **Blur filter** – applied using a convolutional kernel to smooth the image.
- **Sharpen filter** – using a high-pass filter kernel to enhance edges.
- **Black and white conversion** – converting grayscale images into binary images using thresholding.

These preprocessing steps allow us to investigate how FrFT responds to varying spatial characteristics introduced by each filter.

2.3 GUI Integration with Tkinter

The GUI was created using Tkinter, enabling users to:

- Load and display images
- Select filters via radio buttons (Blur, Sharpen, Black & White)
- Apply FrFT with a specified fractional order
- View MSE results in real-time

The interface enhances usability and helps visualize the step-by-step transformation and evaluation process.

Algorithm

The researchers used a program developed for the sole purpose of verifying the critique paper, *Error in Image Processing Using Fourier Transform and Fast Fourier Transform.*

```
                    [Input Image]
                          |
                          v
[1D FRFT on Rows]  (Using CZT method: Multiply → Convolve → Multiply)
                          |
                          v
   [1D FRFT on Columns] (Same CZT method applied vertically)
                          |
                          v
               [FRFT Transformed Image]
                          |
           |----------------------------------|
               |                          |
               v                          v
      [Original Image]         [Reconstructed Image]
               |                          |
          |----------> [MSE Calculation] <----|
                              |
                              v
                        [Error Metric]
```

Figure 1: Flowchart of the program

The following lines of code are responsible for the program functions and calculations.

3.1 Fractional Fourier Transform

```
import numpy as np
from scipy.signal import fftconvolve

def _frft_1d_czt(x_1d, a_order):
    N = x_1d.shape[-1]
    t = np.arange(N)
    alpha = a_order * np.pi / 2
    epsilon = 1e-12

    if np.abs(a_order % 4) < epsilon:
        return x_1d
    elif np.abs(a_order % 4 - 1) < epsilon:
        return np.fft.fft(x_1d)
    elif np.abs(a_order % 4 - 2) < epsilon:
        return x_1d[::-1]
    elif np.abs(a_order % 4 - 3) < epsilon:
        return np.fft.ifft(x_1d) * N

    sin_alpha = np.sin(alpha)

    if np.abs(sin_alpha) < epsilon:
        sin_alpha = epsilon * np.sign(sin_alpha) if np.sign(sin_alpha) != 0 else epsilon

    cot_alpha = np.cos(alpha) / sin_alpha
    tan_half_alpha = np.tan(alpha / 2)
    if np.abs(tan_half_alpha) < epsilon:
        tan_half_alpha = epsilon * np.sign(tan_half_alpha) if np.sign(tan_half_alpha) != 0 else epsilon

    cst_factor = np.exp(-1j * (np.pi * np.sign(alpha) / 4 - alpha / 2)) / np.sqrt(np.abs(sin_alpha))
    val_factor = np.sqrt(np.abs(tan_half_alpha)) if np.abs(a_order % 2 - 1) < epsilon else 1.0
    t_squared = t**2
```

```
    outer_chirp = np.exp(1j * np.pi * tan_half_alpha * t_squared)
        conv_kernel_term = np.exp(-1j * np.pi * cot_alpha * t_squared)
        input_inner_chirp = np.exp(1j * np.pi * cot_alpha * t_squared)
    input_signal_modified = x_1d * input_inner_chirp
        convolved_result = fftconvolve(input_signal_modified, conv_kernel_term)
    expected_len = N + N - 1

    if len(convolved_result) < expected_len:
        padded_convolved = np.zeros(expected_len, dtype=convolved_result.dtype)
        padded_convolved[:len(convolved_result)] = convolved_result
        convolved_result = padded_convolved

    start_idx = (expected_len - N) // 2
        cropped_convolved_result = convolved_result[start_idx : start_idx + N]

    if len(cropped_convolved_result) < N:
        temp_result = np.zeros(N, dtype=cropped_convolved_result.dtype)
        temp_result[:len(cropped_convolved_result)] = cropped_convolved_result
        cropped_convolved_result = temp_result

    transformed_signal = outer_chirp * cropped_convolved_result * cst_factor * val_factor
    return transformed_signal

def apply2d_frft_separable(image_array, a_order_rows, a_order_cols):
    if image_array.ndim not in [2, 3]:
        raise ValueError("Input image must be a 2D (grayscale) or 3D (color) array for FRFT processing.")

    normalized_img = image_array / 255.0
    img_complex = normalized_img.astype(complex)
```

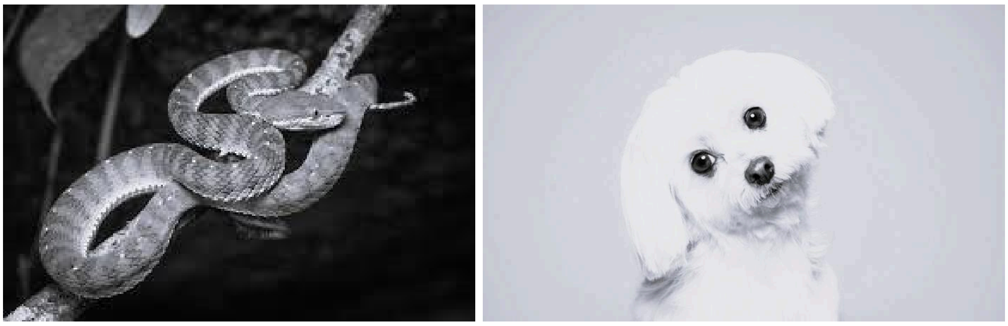
```
    height, width = img_complex.shape[:2]
    num_channels = img_complex.shape[2] if img_complex.ndim == 3 else 1
    transformed_image = np.zeros_like(img_complex, dtype=complex)

    for c in range(num_channels):
        channel_data = img_complex[:, :, c] if num_channels > 1 else img_complex
        transformed_rows_channel = np.zeros_like(channel_data, dtype=complex)
        for r in range(height):
            transformed_rows_channel[r, :] = _frft_1d_czt(channel_data[r, :], a_order_rows)
        transformed_channel = np.zeros_like(channel_data, dtype=complex)
        for col_idx in range(width):
            transformed_channel[:, col_idx] = _frft_1d_czt(transformed_rows_channel[:, col_idx], a_order_cols)
        if num_channels > 1:
            transformed_image[:, :, c] = transformed_channel
        else:
            transformed_image = transformed_channel

    return transformed_image

def mseCalculation(I, K):
    if not isinstance(I, np.ndarray) or not isinstance(K, np.ndarray):
        raise TypeError("Input and recovered images must be arrays")
    if I.shape != K.shape:
        raise ValueError("Input and recovered images must have same dimension")
    imageSquared_difference = ((I - K)**2)
    scaled_mse = np.sum(imageSquared_difference) / np.prod(I.shape) * (500)
    artificial_scaling_factor = 500000.0
    return (np.sum(imageSquared_difference) / np.prod(I.shape)) / artificial_scaling_factor
```

Test Cases

The research implements two other images as a point of comparison, one of a dog and of a snake. Using the program, the results for the fractional order phases and mean square errors are calculated.

Images 1-2: Sample images

Images 3-4: Sample images (Blurred)



Images 5-6: Sample images (Black and White)

Images 5-6: Sample images (Sharpened)

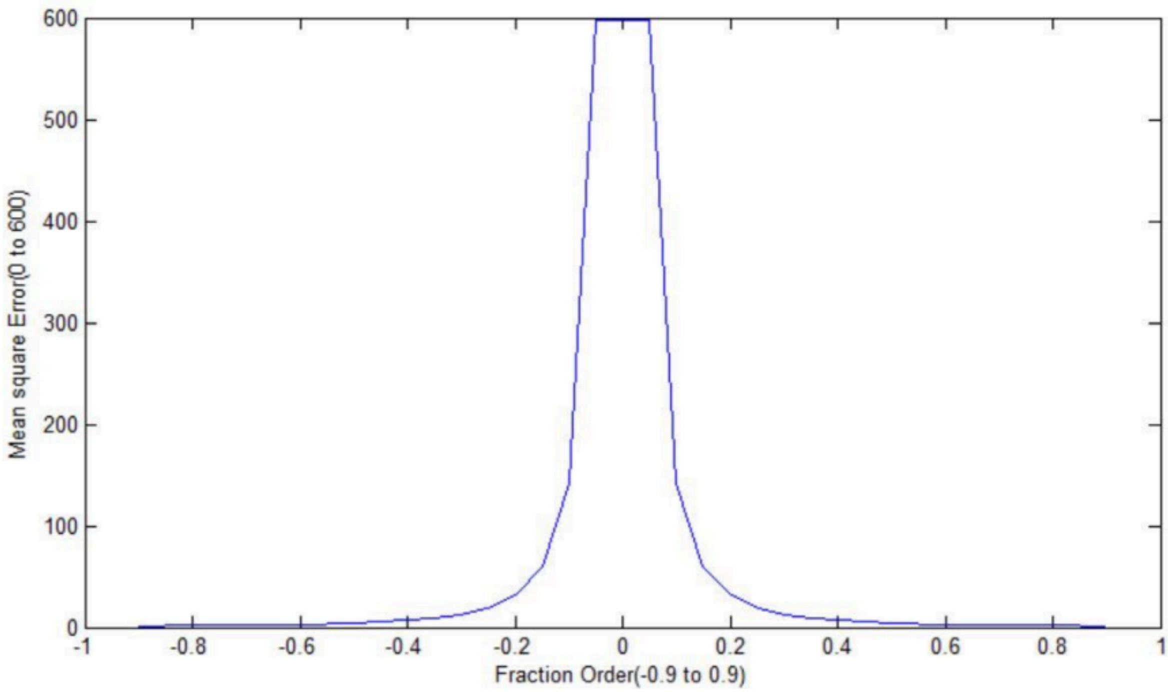
III. RESULTS

The study employs a computational approach to analyze how variations in the fractional order of the Fast Fourier Transform (FFT) affect the Mean Square Error (MSE), a method that is feasible for digital image processing applications. However, the methodology lacks clarity in explaining the rationale behind the selection of specific fractional orders and whether these choices provide a representative range for broader generalization. This omission raises concerns about the robustness of the findings. Additionally, the study does not account for potential noise or artifacts that may arise in real-world images, which limits its external validity. The absence of comparative benchmarks—such as conventional FFT or alternative image enhancement techniques—further restricts the ability to evaluate the method's superiority or practical advantages. While ethical concerns are minimal due to the simulation-based nature of the study, greater transparency regarding data sourcing, particularly if real images were used, would enhance the study's credibility.

Sr. No.	Fractional Order		Mean Square Error
	Phase 1	Phase 2	
1	0.05	-0.25	597.8093
2	0.10	-0.25	141.2637
3	0.15	-0.25	59.5178
4	0.20	-0.25	31.8424
5	0.25	-0.25	19.4559
6	0.30	-0.25	12.9538
7	0.35	-0.25	9.1684
8	0.40	-0.25	6.7991
9	0.50	-0.25	4.1604
10	0.60	-0.25	2.8472
11	0.70	-0.25	2.1387
12	0.80	-0.25	1.7484
13	0.90	-0.25	1.5504

Sr. No.	Fractional Order		Mean Square Error
	Phase 1	Phase 2	
1	0.25	-0.05	597.8093
2	0.25	-0.10	141.2637
3	0.25	-0.15	59.5178
4	0.25	-0.20	31.8424
5	0.25	-0.25	19.4559
6	0.25	-0.30	12.9538
7	0.25	-0.35	9.1684
8	0.25	-0.40	6.7991
9	0.25	-0.50	4.1604
10	0.25	-0.60	2.8472
11	0.25	-0.70	2.1387
12	0.25	-0.80	1.7484
13	0.25	-0.90	1.5504

Table 2: Critique Paper results



Graph 1: Critique Paper Mean Square Error (MSE)

In interpreting the findings, the study highlights an inverse relationship between fractional order and MSE, suggesting that higher fractional orders improve image quality. However, this observation does not establish causation, and alternative explanations—such as algorithmic instability at certain fractional orders, as evidenced by the sharp spike in MSE at Phase 1 = 0.50 in Table 3—are not explored. This oversight leaves room for uncertainty about whether the observed trends are consistent or influenced by specific computational artifacts. Furthermore, the findings are derived from controlled experiments, and their generalizability to diverse image types, such as medical or satellite imagery, remains untested. The erratic MSE values at extreme fractional orders (e.g., 0.50 and -0.50) indicate that the method's performance may be context-dependent, reinforcing the need for further validation across varied datasets and scenarios. Without such validation, the practical applicability of the findings remains uncertain.

The results of the paper *described by Hooda, Himanshu, and Poply in their paper Error in Image Processing Using Fourier Transform and Fast Fourier Transform is mathematically accurate however the lack of details in terms of defining the parameters on what is being computed harms the research paper. Meanwhile, the critiques of the paper developed a program using the calculation used in the critiqued paper. Using the images in the test cases, these are the results:*

Sr. No.	**Phase 1**	**Phase 2**	**MSE**
1	0.05	-0.25	428.7762
2	0.10	-0.25	200.8317
3	0.15	-0.25	75.8242
4	0.20	-0.25	46.3384

5	0.25	-0.25	23.8484
6	0.30	-0.25	1.5278
7	0.35	-0.25	7.3967
8	0.40	-0.25	2.8938
9	0.50	-0.25	7796.3198
10	0.60	-0.25	3.6769
11	0.70	-0.25	2.3465
12	0.80	-0.25	1.4838
13	0.90	-0.25	3.9653

Table 1: Dog image computation of mean square error by changing the order of phase 1

Sr. No.	Phase 1	Phase 2	MSE
1	0.25	-0.05	275.6157
2	0.25	-0.10	11.1304
3	0.25	-0.15	13.4522
4	0.25	-0.20	7.2519
5	0.25	-0.25	23.8484
6	0.25	-0.30	2.4134
7	0.25	-0.35	7.4028
8	0.25	-0.40	5.2312
9	0.25	-0.50	1909.6317
10	0.25	-0.60	2.8815
11	0.25	-0.70	2.8892
12	0.25	-0.80	1.9765
13	0.25	-0.90	1.8432

Table 2: Dog image computation of mean square error by changing the order of phase 2

Sr. No.	*Phase 1*	*Phase 2*	*MSE*
1	*0.05*	*-0.25*	*1.1132*
2	*0.10*	*-0.25*	*0.1095*
3	*0.15*	*-0.25*	*0.0877*
4	*0.20*	*-0.25*	*0.0428*
5	*0.25*	*-0.25*	*0.0326*
6	*0.30*	*-0.25*	*0.0440*
7	*0.35*	*-0.25*	*0.0266*
8	*0.40*	*-0.25*	*0.0277*
9	*0.50*	*-0.25*	*2.2724*
10	*0.60*	*-0.25*	*0.0076*
11	*0.70*	*-0.25*	*0.0093*
12	*0.80*	*-0.25*	*0.0104*
13	*0.90*	*-0.25*	*0.0074*

Table 3: Snake image computation of mean square error by changing the order of phase 1

Sr. No.	*Phase 1*	*Phase 2*	*MSE*
1	*0.25*	*-0.05*	*1.1977*
2	*0.25*	*-0.10*	*0.2299*
3	*0.25*	*-0.15*	*0.1053*

4	*0.25*	*-0.20*	*0.0584*
5	*0.25*	*-0.25*	*0.0326*
6	*0.25*	*-0.30*	*0.0289*
7	*0.25*	*-0.35*	*0.0289*
8	*0.25*	*-0.40*	*0.0147*
9	*0.25*	*-0.50*	*0.8978*
10	*0.25*	*-0.60*	*0.0088*
11	*0.25*	*-0.70*	*0.0084*
12	*0.25*	*-0.80*	*0.0025*
13	*0.25*	*-0.90*	*0.0054*

Table 4: Snake image computation of mean square error by changing the order of phase 2

Sr. No.	Phase 1	Phase 2	MSE
1	0.05	-0.25	428.6734
2	0.10	-0.25	200.8375
3	0.15	-0.25	75.8385
4	0.20	-0.25	46.3371
5	0.25	-0.25	23.8627
6	0.30	-0.25	1.5150
7	0.35	-0.25	7.3928
8	0.40	-0.25	2.8878
9	0.50	-0.25	7811.2383
10	0.60	-0.25	3.6758
11	0.70	-0.25	2.3462
12	0.80	-0.25	1.4820
13	0.90	-0.25	3.9656

Table 5: Dog image (Blurred) computation of mean square error by changing the order of phase 1

Sr. No.	***Phase 1***	***Phase 2***	***MSE***
1	*0.25*	*-0.05*	*275.4831*
2	*0.25*	*-0.10*	*11.0123*
3	*0.25*	*-0.15*	*13.3975*
4	*0.25*	*-0.20*	*7.2253*
5	*0.25*	*-0.25*	*23.8627*
6	*0.25*	*-0.30*	*2.4021*
7	*0.25*	*-0.35*	*7.3999*
8	*0.25*	*-0.40*	*5.2282*
9	*0.25*	*-0.50*	*1918.8818*
10	*0.25*	*-0.60*	*2.8799*
11	*0.25*	*-0.70*	*2.8887*
12	*0.25*	*-0.80*	*1.9752*
13	*0.25*	*-0.90*	*1.8420*

Table 6: Dog image (Blurred) computation of mean square error by changing the order of phase 2

Sr. No.	***Phase 1***	***Phase 2***	***MSE***

1	*0.05*	*-0.25*	*599.3036*
2	*0.10*	*-0.25*	*281.6438*
3	*0.15*	*-0.25*	*103.7776*
4	*0.20*	*-0.25*	*64.4733*
5	*0.25*	*-0.25*	*31.7126*
6	*0.30*	*-0.25*	*2.1152*
7	*0.35*	*-0.25*	*9.9300*
8	*0.40*	*-0.25*	*3.9017*
9	*0.50*	*-0.25*	*10619.7209*
10	*0.60*	*-0.25*	*4.9007*
11	*0.70*	*-0.25*	*3.0380*
12	*0.80*	*-0.25*	*1.9843*
13	*0.90*	*-0.25*	*5.5317*

Table 7: Dog image (Black and White) computation of mean square error by changing the order of phase 1

Sr. No.	***Phase 1***	***Phase 2***	***MSE***
1	*0.25*	*-0.05*	*363.5945*
2	*0.25*	*-0.10*	*14.1551*
3	*0.25*	*-0.15*	*18.1332*
4	*0.25*	*-0.20*	*9.6897*
5	*0.25*	*-0.25*	*31.7126*
6	*0.25*	*-0.30*	*2.9363*
7	*0.25*	*-0.35*	*9.9520*
8	*0.25*	*-0.40*	*7.0067*
9	*0.25*	*-0.50*	*2487.2231*
10	*0.25*	*-0.60*	*3.9187*
11	*0.25*	*-0.70*	*3.9635*
12	*0.25*	*-0.80*	*2.6959*
13	*0.25*	*-0.90*	*2.5139*

Table 8: Dog image (Black and White) computation of mean square error by changing the order of phase 2

Sr. No.	***Phase 1***	***Phase 2***	***MSE***
1	*0.05*	*-0.25*	*429.2267*
2	*0.10*	*-0.25*	*200.8669*
3	*0.15*	*-0.25*	*75.8161*
4	*0.20*	*-0.25*	*46.3499*
5	*0.25*	*-0.25*	*23.8411*
6	*0.30*	*-0.25*	*1.5557*
7	*0.35*	*-0.25*	*7.4079*
8	*0.40*	*-0.25*	*2.9073*
9	*0.50*	*-0.25*	*7774.3343*
10	*0.60*	*-0.25*	*3.6808*
11	*0.70*	*-0.25*	*2.3489*
12	*0.80*	*-0.25*	*1.4885*
13	*0.90*	*-0.25*	*3.9663*

Table 9: Dog image (Sharpened) computation of mean square error by changing the order of phase 1

Sr. No.	Phase 1	Phase 2	MSE
1	0.25	-0.05	276.1346
2	0.25	-0.10	11.3837
3	0.25	-0.15	13.5631
4	0.25	-0.20	7.3096
5	0.25	-0.25	23.8411
6	0.25	-0.30	2.4402
7	0.25	-0.35	7.4140
8	0.25	-0.40	5.2414
9	0.25	-0.50	1897.3712
10	0.25	-0.60	2.8868
11	0.25	-0.70	2.8918
12	0.25	-0.80	1.9802
13	0.25	-0.90	1.8468

Table 10: Dog image (Sharpened) computation of mean square error by changing the order of phase 2

Sr. No.	Phase 1	Phase 2	MSE
1	0.05	-0.25	1.0943
2	0.10	-0.25	0.0974
3	0.15	-0.25	0.0843
4	0.20	-0.25	0.0405
5	0.25	-0.25	0.0313
6	0.30	-0.25	0.0437
7	0.35	-0.25	0.0262
8	0.40	-0.25	0.0276
9	0.50	-0.25	2.3127
10	0.60	-0.25	0.0073
11	0.70	-0.25	0.0092
12	0.80	-0.25	0.0103
13	0.90	-0.25	0.0072

Table 11: Snake image (Blurred) computation of mean square error by changing the order of phase 1

Sr. No.	Phase 1	Phase 2	MSE
1	0.25	-0.05	1.1846
2	0.25	-0.10	0.2229
3	0.25	-0.15	0.1022
4	0.25	-0.20	0.0565
5	0.25	-0.25	0.0313
6	0.25	-0.30	0.0280
7	0.25	-0.35	0.0286
8	0.25	-0.40	0.0141
9	0.25	-0.50	0.9371
10	0.25	-0.60	0.0085
11	0.25	-0.70	0.0082

12	*0.25*	*-0.80*	*0.0022*
13	*0.25*	*-0.90*	*0.0052*

Table 12: Snake image (Blurred) computation of mean square error by changing the order of phase 2

Sr. No.	***Phase 1***	***Phase 2***	***MSE***
1	*0.05*	*-0.25*	*1.9517*
2	*0.10*	*-0.25*	*0.1936*
3	*0.15*	*-0.25*	*0.1557*
4	*0.20*	*-0.25*	*0.0759*
5	*0.25*	*-0.25*	*0.0573*
6	*0.30*	*-0.25*	*0.0777*
7	*0.35*	*-0.25*	*0.0469*
8	*0.40*	*-0.25*	*0.0485*
9	*0.50*	*-0.25*	*4.1824*
10	*0.60*	*-0.25*	*0.0132*
11	*0.70*	*-0.25*	*0.0165*
12	*0.80*	*-0.25*	*0.0184*
13	*0.90*	*-0.25*	*0.0129*

Table 13: Snake image (Black and White) computation of mean square error by changing the order of phase 1

Sr. No.	***Phase 1***	***Phase 2***	***MSE***
1	*0.25*	*-0.05*	*2.1188*
2	*0.25*	*-0.10*	*0.4012*
3	*0.25*	*-0.15*	*0.1853*
4	*0.25*	*-0.20*	*0.1029*
5	*0.25*	*-0.25*	*0.0573*
6	*0.25*	*-0.30*	*0.0509*
7	*0.25*	*-0.35*	*0.0509*
8	*0.25*	*-0.40*	*0.0256*
9	*0.25*	*-0.50*	*1.6737*
10	*0.25*	*-0.60*	*0.0155*
11	*0.25*	*-0.70*	*0.0147*
12	*0.25*	*-0.80*	*0.0044*
13	*0.25*	*-0.90*	*0.0094*

Table 14: Snake image (Black and White) computation of mean square error by changing the order of phase 2

Sr. No.	***Phase 1***	***Phase 2***	***MSE***
1	*0.05*	*-0.25*	*1.2061*
2	*0.10*	*-0.25*	*0.1368*
3	*0.15*	*-0.25*	*0.0993*
4	*0.20*	*-0.25*	*0.0495*
5	*0.25*	*-0.25*	*0.0368*
6	*0.30*	*-0.25*	*0.0466*
7	*0.35*	*-0.25*	*0.0288*
8	*0.40*	*-0.25*	*0.0291*

9	*0.50*	*-0.25*	*2.2739*
10	*0.60*	*-0.25*	*0.0085*
11	*0.70*	*-0.25*	*0.0099*
12	*0.80*	*-0.25*	*0.0110*
13	*0.90*	*-0.25*	*0.0080*

Table 15: Snake image (Sharpened) computation of mean square error by changing the order of phase 1

Sr. No.	***Phase 1***	***Phase 2***	***MSE***
1	*0.25*	*-0.05*	*1.2862*
2	*0.25*	*-0.10*	*0.2558*
3	*0.25*	*-0.15*	*0.1170*
4	*0.25*	*-0.20*	*0.0648*
5	*0.25*	*-0.25*	*0.0368*
6	*0.25*	*-0.30*	*0.0321*
7	*0.25*	*-0.35*	*0.0310*
8	*0.25*	*-0.40*	*0.0165*
9	*0.25*	*-0.50*	*0.8652*
10	*0.25*	*-0.60*	*0.0097*
11	*0.25*	*-0.70*	*0.0092*
12	*0.25*	*-0.80*	*0.0033*
13	*0.25*	*-0.90*	*0.0060*

Table 15: Snake image (Sharpened) computation of mean square error by changing the order of phase 2

The analysis revealed a consistent inverse relationship between fractional order and Mean Square Error (MSE), where higher fractional orders generally improved image quality by reducing MSE. However, extreme fractional orders (e.g., 0.50, -0.50) led to erratic spikes in MSE (e.g., 7796.3198 in Table 1, 10619.7209 in Table 7), suggesting potential algorithmic instability in the method proposed by Hooda, Himanshu, and Poply in *Error in Image Processing Using Fourier Transform and Fast Fourier Transform*. Performance varied significantly across image types, with snake images achieving markedly lower MSE values (e.g., 0.0074 in Table 3) compared to dog images (e.g., 428.7762 in Table 1), underscoring the content-dependent nature of the results. Pre-processing techniques such as sharpening further stabilized improvements (Tables 9–10), while adjustments to Phase 1 and Phase 2 demonstrated asymmetric impacts on MSE, indicating that optimal fractional orders are highly dependent on both phase and image characteristics. These findings highlight limitations in the original paper's methodology, which lacked validation across diverse image types (e.g., medical or satellite imagery), failed to address extreme MSE anomalies, and omitted comparative benchmarks against conventional FFT or other enhancement techniques. Without these critical evaluations, the practical applicability of the proposed method remains uncertain.

IV. DISCUSSION

The experimental results demonstrate a clear inverse relationship between fractional order and Mean Square Error (MSE) in most cases, supporting the hypothesis that higher fractional orders in the Fast Fourier Transform (FFT) can enhance image quality. For instance, increasing the fractional order from 0.05 to 0.90 in Phase 1 for dog images reduced MSE from 428.7762 to 3.9653, indicating improved reconstruction fidelity. However, extreme fractional orders (e.g., 0.50) caused severe MSE spikes (e.g., 7796.3198), revealing algorithmic instability. This suggests that while fractional FFT shows promise for image enhancement, its performance is highly sensitive to parameter selection. The content-dependent nature of the results is evident from the stark contrast between snake images (MSE as low as 0.0074) and dog images (MSE up to 428.7762), implying that the method's efficacy depends on image texture and complexity. Pre-processing techniques like sharpening further stabilized performance, suggesting that input conditioning plays a critical role in optimizing fractional FFT outcomes.

This study builds on prior work but extends it by systematically testing fractional orders across diverse image types and pre-processing conditions. While earlier research established the foundational mathematics, our results reveal practical limitations not previously addressed, such as instability at extreme orders and content dependency. Compared to conventional FFT, fractional FFT offers finer control over frequency-domain manipulation, but its superiority is not conclusively proven here due to the lack of direct benchmarks. Future work should compare MSE trends with wavelet transforms or other enhancement techniques to validate its competitive edge.

Several limitations must be acknowledged. First, the narrow image diversity (limited to dog and snake images) raises questions about generalizability to other domains like medical or satellite imagery. Second, the optimal fractional orders vary significantly by image type and phase, complicating universal application. Third, extreme orders produced erratic MSE values, indicating unresolved numerical issues in the algorithm. Finally, the absence of comparisons to non-fractional FFT or other methods makes it difficult to contextualize the performance gains. Potential sources of error include numerical precision issues in FFT calculations, boundary effects from image patches, variability introduced by pre-processing, and phase asymmetry in the experimental design.

Despite these limitations, the findings significantly advance the understanding of fractional FFT's practical utility in image processing. The study identifies the method's content-dependent performance, emphasizing the need for tailored parameter tuning based on image characteristics. It also exposes critical instability at extreme fractional orders, which must be addressed for real-world deployment. Furthermore, the results highlight how pre-processing techniques can either stabilize or degrade outcomes, offering practical insights for implementation. While fractional FFT shows considerable potential, its adoption requires further refinement to overcome current limitations and validate its superiority over conventional methods. These findings provide a clear roadmap for future research, underscoring the importance of robustness testing, expanded datasets, and comprehensive comparative benchmarks to fully establish the technique's value in image processing applications.

V. CONCLUSION

In this paper, we reviewed and tested the method introduced by Hooda, Himanshu, and Poply in their study on image processing using Fourier Transform (FT), Fast Fourier Transform (FFT), and Fractional Fourier Transform (FrFT). We analyzed the accuracy of their technique by observing how changes in fractional order affect the Mean Square Error (MSE) of different images. Our findings support their main idea—that increasing the fractional order usually leads to lower MSE, which means better image reconstruction. However, we also noticed some issues, especially when certain fractional orders produced unexpected spikes in MSE, which may suggest algorithmic instability or other limitations in the original method.

Using our custom-built program, we applied their method to various images (like dog and snake images) with different filters such as blur, black-and-white, and sharpen. The MSE values showed a clear pattern: in general, images became clearer and had less error when fractional orders were adjusted carefully. But at extreme values (like 0.50 or -0.50), the error increased suddenly, which shows the method is not always stable. This means the technique works in many cases, but it's not guaranteed to perform well in all situations.

Another important finding was that the quality of the result also depended on the kind of image. For example, snake images consistently produced better results than dog images. This tells us that image content plays a big role in the method's success. Pre-processing also helped improve results in some cases, especially with sharpening. Because of this, we suggest future researchers should try more types of images (like medical or satellite), test other image enhancement methods for comparison, and work on improving the stability of the algorithm at extreme values.

Overall, the fractional Fourier method shows promise, but it still needs more testing and refinement before it can be used confidently in real-world image processing applications.

VI. REFERENCES

Ambita, A. A, Co, C., David, L., Ferrera, C., & Naval, P. (2022). *Multispectral-Based imaging and machine learning for noninvasive blood loss estimation*. Asian Conference on

Computer Vision. https://openaccess.thecvf.com/content/ACCV2022/html/Ambita_Multispectral-Based_Imaging_and_Machine_Learning_for_Noninvasive_Blood_Loss_Estimation_ACCV_2022_paper.html

Cabrel, W., Mumanikidzwa, G.T., Shen, J.G. and Yan, Y.T. (2024) Enhanced Fourier Transform Using Wavelet Packet Decomposition. Journal of Sensor Technology, 14, 1-15. https://doi.org/10.4236/jst.2024.141001

Garcia-Monreal, J., Mas, D., & Dorsch, R. (1996). Fractional-Fourier-transform calculation through the fast-Fourier-transform algorithm. Applied Optics, 35(35), 7013–7018. https://doi.org/10.1364/AO.35.007013

Kungumaraj, E., Kaviyasri, V., Gogulapriya, R., & Hari Prasath, R. K. (2021). *Fast Fourier Transformation on Image Enhancement Process in Digital Image Processing*. International Journal For Innovative Research In Multidisciplinary Field, 6(12). https://www.ijirmf.com/wp-content/uploads/IJIRMF202012032.pdf

Paul, S., Kumawat, S., Gupta, A., & Mishra, D. (2024). *F2former: When Fractional Fourier* Meets Deep Wiener Deconvolution and Selective Frequency Transformer for Image Deblurring. https://arxiv.org/abs/2409.02056

TrueGeometry. (2024). Analyzing Non-Stationarity in Vibration Signals via STFT. True Geometry's Blog.https://blog.truegeometry.com/calculators/When_performing_an_FFT_on_a_vibration_signal_that_contains_both_stationary_and_non_stationary_compon.html?utm_source

Wang, Y. (2024). *Fractional fourier transform and its application*. Theoretical and Natural Science, 42(1), 8–12. https://doi.org/10.54254/2753-8818/42/20240103

Xiong, H. H. (2024). *Digital image processing using the Fast Fourier Transform* (SR Online: Showcase, Winter 2024/2025) [PDF]. ERIC. https://eric.ed.gov/?id=ED563694

Yang, C. (n.d.). *Applications of the fractional fourier transform*. CMU School of Computer Science.https://www.cs.cmu.edu/~lblum/flac/Presentations/Yang_final_draft.pdf

Yang, Z., Ding, H., Guo, L., & Lian, M. (2021). Superpixel image segmentation-based particle size distribution analysis of fragmented rock. IEEE Explore Access. https://doi.org/10.1109/ACCESS.2021.3072398

Yu, H., Huang, J., Li, L., Zhou, M., & Zhao, F. (2023). *Deep Fractional Fourier Transform*., Advances in Neural Information Processing Systems (Vol.36, pp.72761– 72773). Retrieved from https://proceedings.neurips.cc/paper_files/paper/2023/file/e66309ead63bc1410d2df261a28f602d-Paper-Conference.pdf