INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Birigui	**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA** **Campus Birigui** **Bacharelado em Engenharia de Computação**

Disciplina: Processamento Digital de Imagem	**Atividade**
Professor: Prof. Dr. Murilo Varges da Silva	**Data:** 11/09/2023
Nome do Aluno: Henrique Akira Hiraga	**Prontuário:** BI300838X

TRANSFORMADA DE FOURIER

Este relatório analisa um código em Python que realiza a Transformada de Fourier em uma imagem digital, além de várias operações relacionadas ao processamento de imagens e visualização. O código é dividido em várias seções, e cada uma delas é explicada em detalhes abaixo:

1. Importação de Bibliotecas:

O código começa importando três bibliotecas essenciais para a execução das tarefas: NumPy, cv2 (OpenCV) e Matplotlib. Essas bibliotecas são usadas para realizar cálculos numéricos, manipulação de imagens e visualização.

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

2. Função para Calcular a Transformada de Fourier 2D:

A função calcular_transformada_fourier recebe uma imagem como entrada e calcula a Transformada de Fourier 2D dessa imagem. Ela realiza a transformada, centraliza o espectro, calcula a magnitude e a fase e retorna esses valores.

```
def calcular_transformada_fourier(imagem):
    f_transformada = np.fft.fft2(imagem)
    f_transformada_shifted = np.fft.fftshift(f_transformada)
    magnitude = np.abs(f_transformada_shifted)
    fase = np.angle(f_transformada_shifted)
    return magnitude, fase
```

3. Função para Calcular a Transformada Inversa de Fourier:

A função calcular_transformada_inversa_fourier recebe a magnitude e a fase da Transformada de Fourier como entrada e calcula a Transformada Inversa de Fourier. Ela reverte o deslocamento do espectro, realiza a transformada inversa e retorna a imagem

reconstruída.

```
def calcular_transformada_inversa_fourier(magnitude, fase):
    f_transformada_shifted = magnitude * np.exp(1j * fase)
    f_transformada =
np.fft.ifftshift(f_transformada_shifted)
    imagem_reconstruida = np.fft.ifft2(f_transformada)
    return np.abs(imagem_reconstruida)
```

4. Carregamento da Imagem de Exemplo:

O código carrega uma imagem chamada "periodic_noise.png" em escala de cinza usando a biblioteca OpenCV. É importante notar que o arquivo da imagem deve estar localizado no mesmo diretório em que o código está sendo executado.

```
imagem = cv2.imread('periodic_noise.png',
cv2.IMREAD_GRAYSCALE)
```

5. Cálculo da Transformada de Fourier:

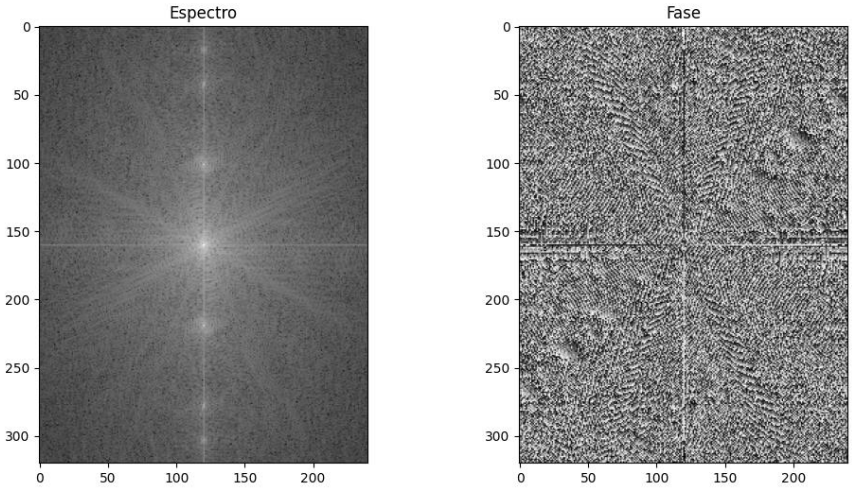
A função calcular_transformada_fourier é usada para calcular a Transformada de Fourier da imagem carregada. Os resultados (magnitude e fase) são armazenados em variáveis para uso posterior.

```
magnitude, fase = calcular_transformada_fourier(imagem)
```

6. Plotagem do Espectro e da Fase da Imagem:

Nesta etapa, o código cria uma figura dividida em duas subtramas usando a biblioteca Matplotlib. A subtrama esquerda mostra o espectro da imagem em escala logarítmica, enquanto a subtrama direita exibe a fase da imagem.

```
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.imshow(np.log1p(magnitude), cmap='gray')
plt.title('Espectro')
plt.subplot(1, 2, 2)
plt.imshow(fase, cmap='gray')
plt.title('Fase')
plt.show()
```



7. Cálculo da Transformada Inversa de Fourier:

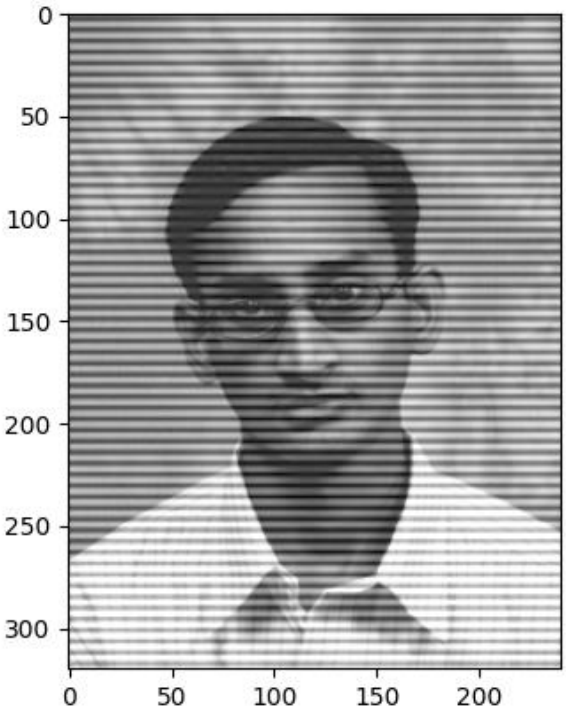
A Transformada Inversa de Fourier é calculada a partir dos valores de magnitude e fase da imagem. A imagem reconstruída é armazenada na variável imagem_reconstruida.

```
imagem_reconstruida = 
calcular_transformada_inversa_fourier(magnitude, fase)
```

8. Plotagem da Imagem Reconstruída:

A imagem reconstruída é plotada usando a biblioteca Matplotlib. A imagem resultante é exibida em escala de cinza.

```
plt.imshow(imagem_reconstruida, cmap='gray')
plt.show()
```



9. Cálculo da Largura e da Altura da Imagem:

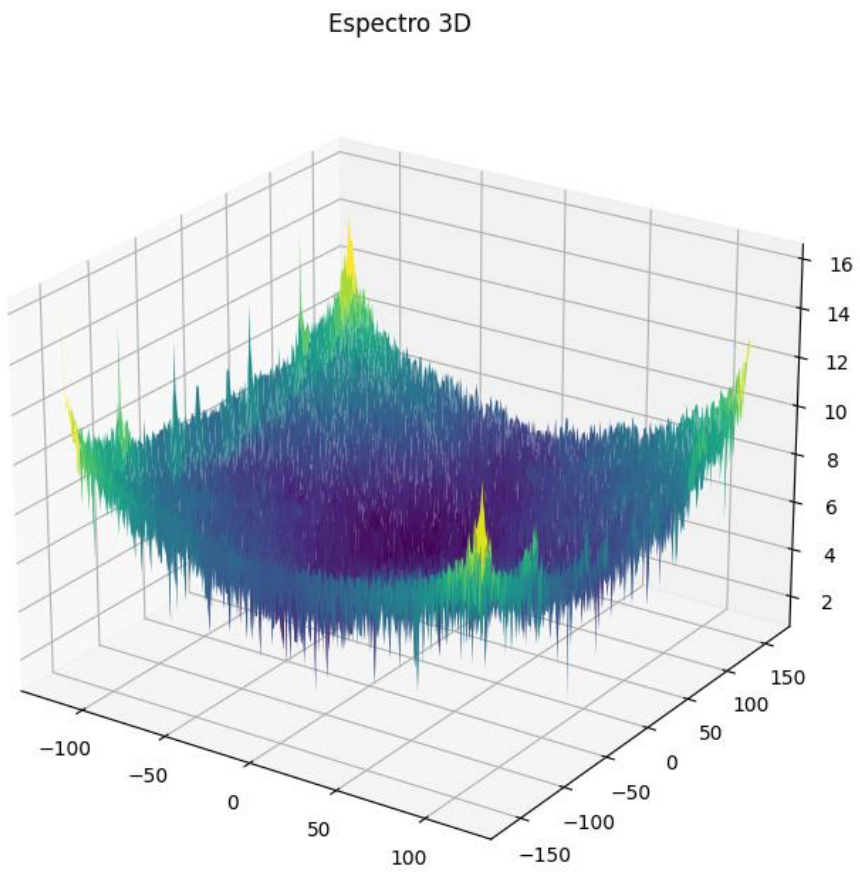
As dimensões (largura e altura) da imagem carregada são calculadas automaticamente para uso posterior.

```
altura, largura = imagem.shape
```

10. Plotagem do Espectro 3D:

Uma figura 3D é criada para visualizar o espectro 3D da imagem. O código gera uma malha de coordenadas para representar o espaço 3D e, em seguida, plota o espectro deslocado em escala logarítmica.

```
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection='3d')
x, y = np.meshgrid(np.arange(-largura // 2, largura // 2),
np.arange(-altura // 2, altura // 2))
ax.plot_surface(x, y, np.fft.fftshift(np.log1p(magnitude)),
cmap='viridis')
ax.set_title('Espectro 3D')
plt.show()
```



11. Criação de uma Imagem SINC Simulada:

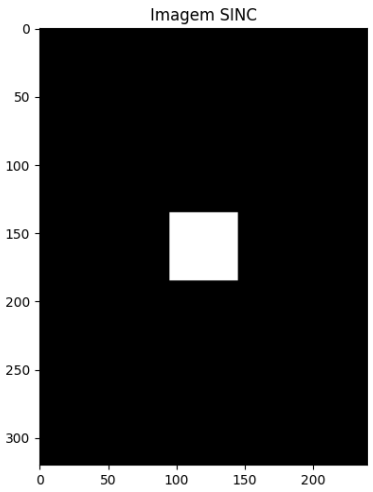
O código cria uma imagem com um fundo branco e um quadrado simulando a função SINC. O tamanho e a posição do quadrado são especificados.

```
altura, largura = imagem.shape
imagem_sinc = np.zeros_like(imagem, dtype=np.float32)
centro_x, centro_y = largura // 2, altura // 2
tamanho_quadrado = 50
imagem_sinc[centro_y - tamanho_quadrado // 2:centro_y +
tamanho_quadrado // 2,
            centro_x - tamanho_quadrado // 2:centro_x +
tamanho_quadrado // 2] = 1.0
```

12. Plotagem da Imagem SINC:

A imagem SINC simulada é plotada usando a biblioteca Matplotlib. A imagem é exibida em escala de cinza.

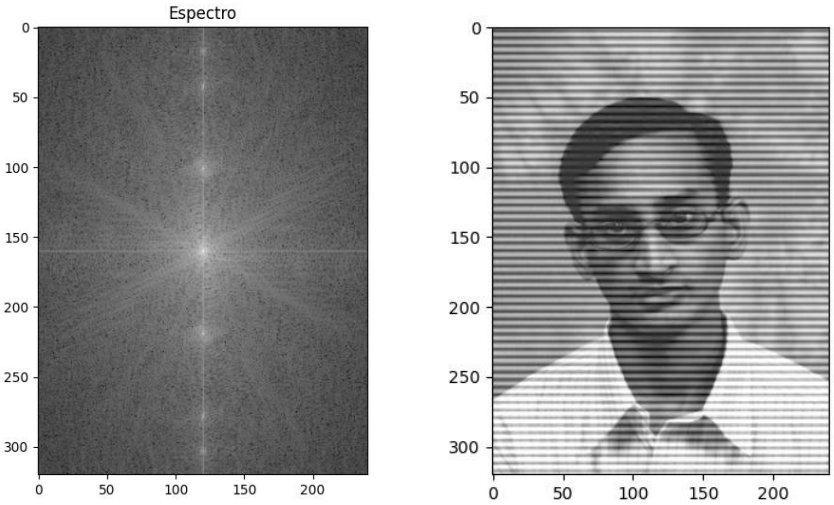
```
plt.figure(figsize=(12, 6))
plt.imshow(imagem_sinc, cmap='gray')
plt.title('Imagem SINC')
plt.show()
```



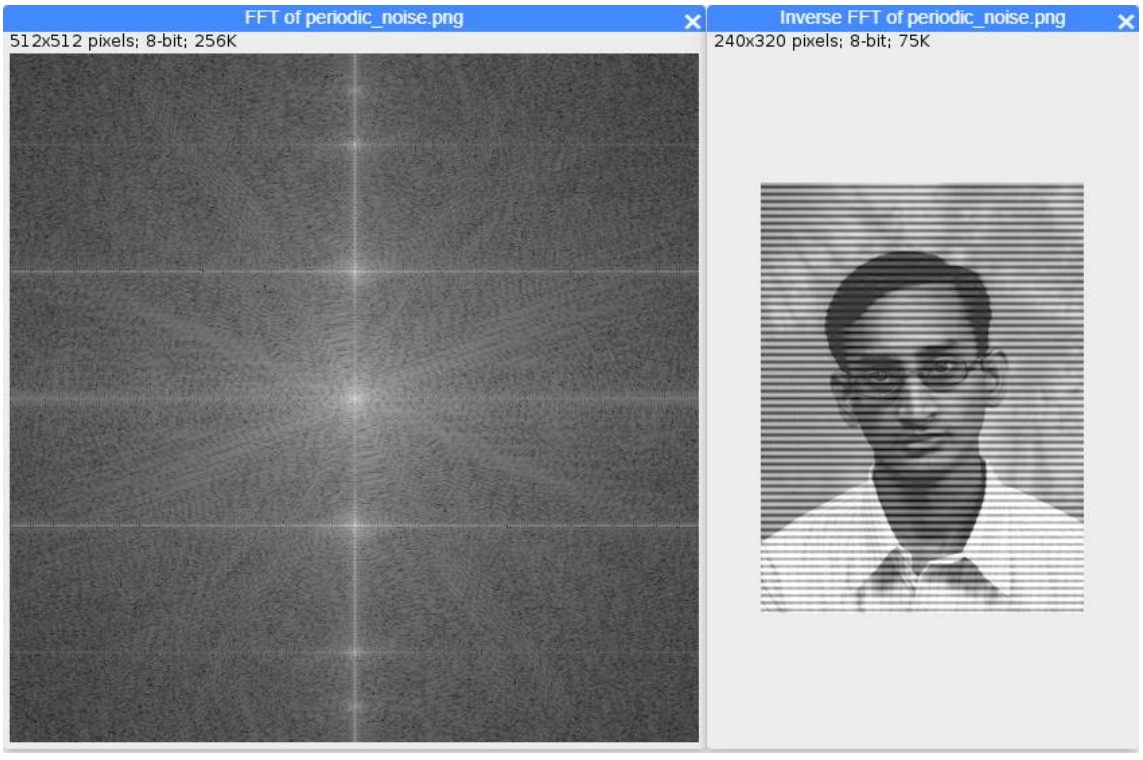
Este código Python demonstra a aplicação da Transformada de Fourier em processamento digital de imagem. Ele realiza cálculos da Transformada de Fourier, visualiza o espectro e a fase da imagem, reconstrói a imagem a partir dos valores da Transformada de Fourier e cria uma imagem simulada para demonstração adicional. Certifique-se de ter as bibliotecas NumPy, OpenCV e Matplotlib instaladas para executar o código com sucesso.

Comparação do código implementado com ImageJ

FFT e FFT inversa pela implementação:



FFT e FFT inversa pelo ImageJ:



Em resumo, as imagens geradas pela Transformada de Fourier (FFT) e pela FFT inversa neste código produziram resultados semelhantes aos obtidos com o software ImageJ. Isso demonstra a eficácia da implementação em Python para análise de frequência e processamento de imagens, com resultados consistentes com ferramentas de imagem estabelecidas.