Praktikum 9 Transformasi_Fourier_Deonising

April 18, 2022

```python
from skimage import io
from skimage.color import rgb2gray
%matplotlib inline
import matplotlib.pyplot as plt
from scipy import fftpack
import numpy as np
import cv2
```

```python
image = io.imread('Data/india.jpg', as_gray=True)
```

```python
image.shape, image.dtype
```
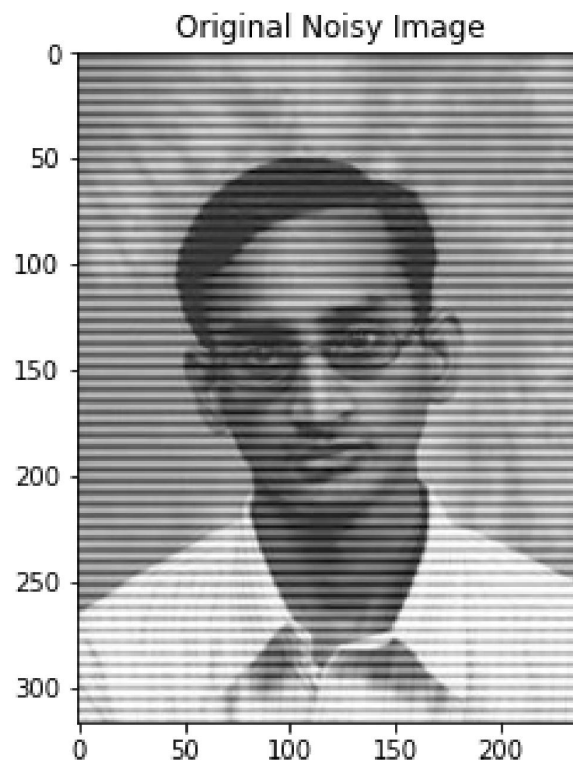
```
((317, 239), dtype('float64'))
```

```python
M,N = image.shape
```

```python
M,N
```

```
(317, 239)
```

```python
f, ax = plt.subplots(figsize=(5,5))
ax.imshow(image, cmap = "gray")
ax.set_title('Original Noisy Image')
```

```
Text(0.5, 1.0, 'Original Noisy Image')
```

Original Noisy Image

```python
#Return multidimensional discrete Fourier transform.
F = fftpack.fft2(image)
F_magnitude = np.abs(F)    #absolute value of F
```

```python
# Show the results

def plot_spectrum(F):
    from matplotlib.colors import LogNorm
    plt.imshow(np.abs(F), norm=LogNorm(vmin=5))
    plt.colorbar()
```
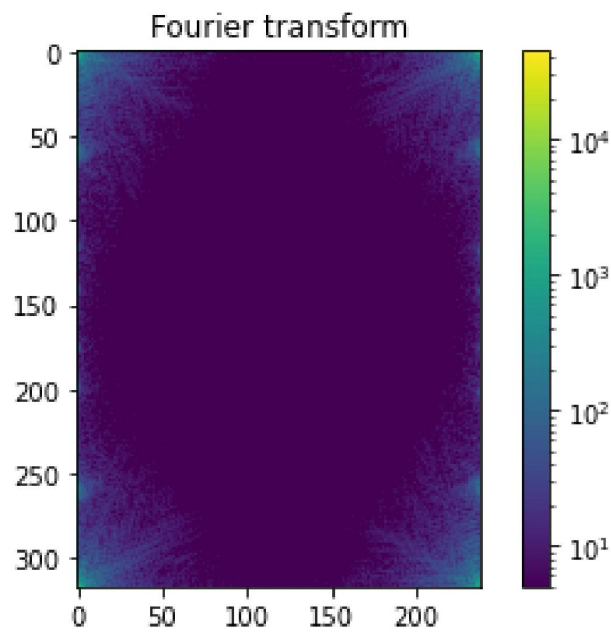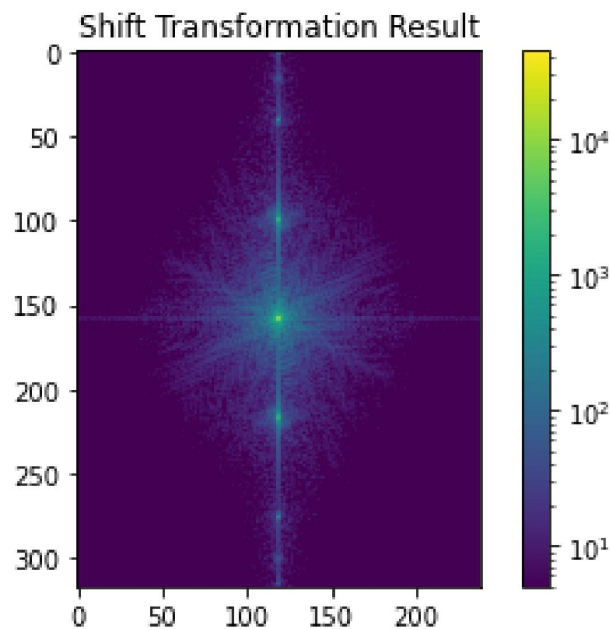
```python
# A logarithmic Colormap

plt.figure()
plot_spectrum(F)
plt.title('Fourier transform')
```

```
Text(0.5, 1.0, 'Fourier transform')
```

Fourier transform

```
[ ]: # Using shift to centering high frequency

     Fs = fftpack.fftshift(F)
```

```
[ ]: plt.figure()
     plot_spectrum(Fs)
     plt.title('Shift Transformation Result')
```

```
[ ]: Text(0.5, 1.0, 'Shift Transformation Result')
```

## Shift Transformation Result


```
for j in range(115, 126):
    for i in range(12, 22):
        Fs[i,j] = 0
    for i in range(37, 47):
        Fs[i,j] = 0
    for i in range(96, 106):
        Fs[i,j] = 0
    for i in range(216, 226):
        Fs[i,j] = 0
    for i in range(274, 284):
        Fs[i,j] = 0
    for i in range(298, 308):
        Fs[i,j] = 0
```

Fs[96:106, 115:125] = 0 Fs[216:226, 115:125] = 0 Fs[274:284, 115:125] = 0 Fs[298:308, 115:125] = 0
Fs[12:22, 115:125] = 0 Fs[37:47, 115:125] = 0

```
plt.figure(figsize=(8,8))
plt.imshow(np.log(abs(Fs)), cmap='gray');
plt.title('Modified Fourier Transform');
```
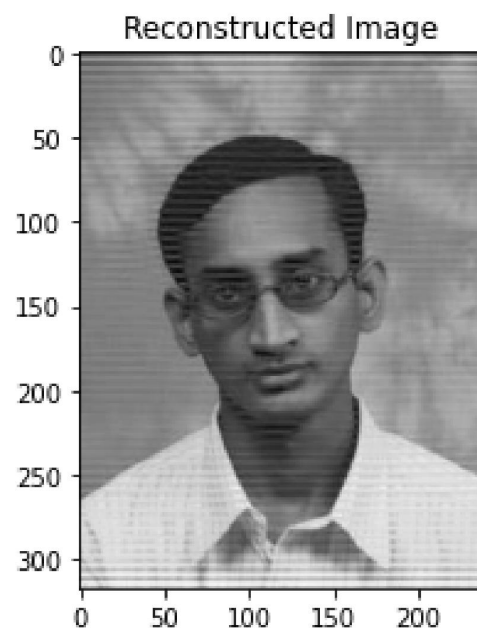
C:\Users\TIK\AppData\Local\Temp/ipykernel_2820/1420656708.py:2: RuntimeWarning:
divide by zero encountered in log
  plt.imshow(np.log(abs(Fs)), cmap='gray');

4

Modified Fourier Transform
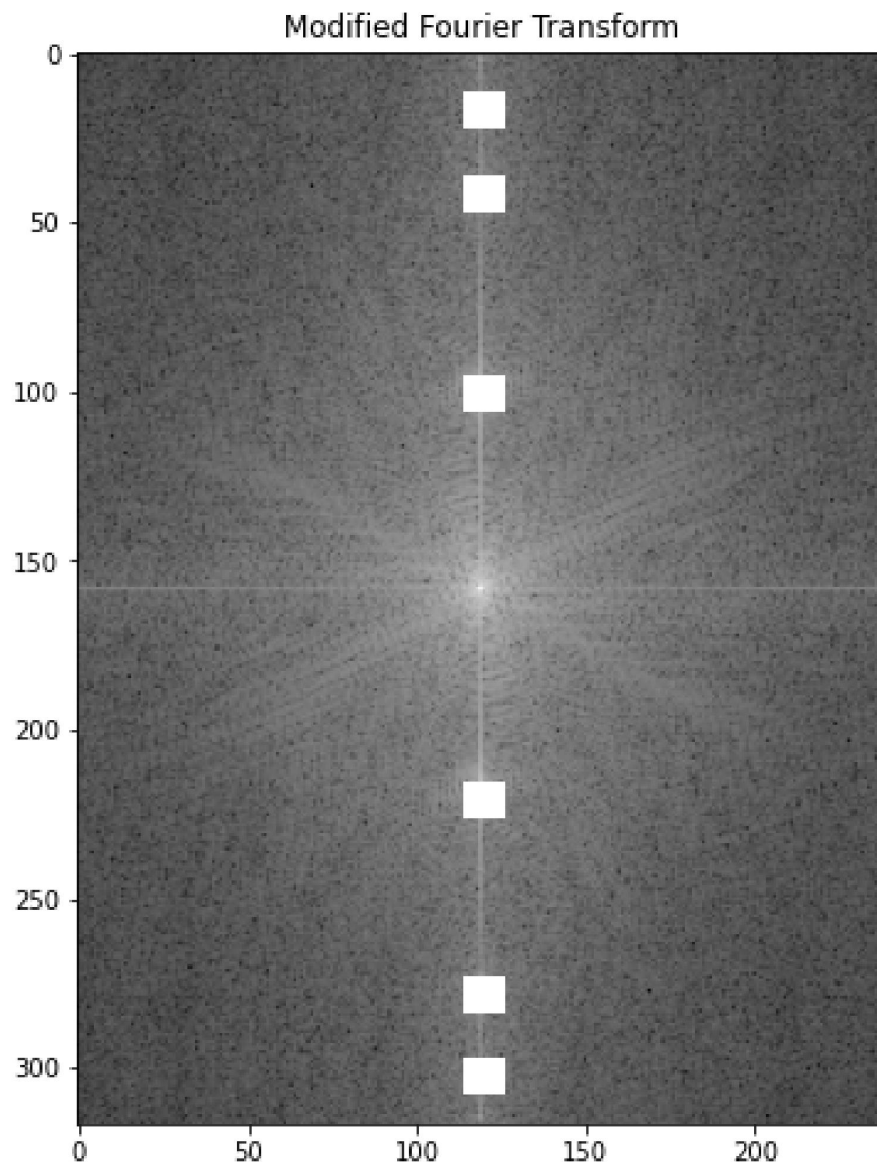
```
[ ]: # Unshift

     Fus = fftpack.ifftshift(Fs)

[ ]: # Inverse FFT

     im_new = fftpack.ifft2(Fus).real
```

```
plt.figure()
plt.imshow(im_new, plt.cm.gray)
plt.title('Reconstructed Image')
```

Text(0.5, 1.0, 'Reconstructed Image')