APPLIED MATHEMATICS – IV REPORT

IMAGE NOISE REDUCTION USING FOURIER TRANSFORM

TEAM MEMBERS:

Manasi Murarka – J038

Khyaati Nayak – J044

Bharati Panigrahi – J045

PROBLEM STATEMENT:

- Image noise is that by product of an image that's widely seen as undesirable.
- Digital images are inevitably corrupted by noise due to acquisition, transmission, and mathematical computation.
- Noise manifests itself in the digital image in a random uncorrelated manner, making it inescapable to degrade the visual quality of the images besides harshly limiting the precision and accuracy of image interpretation and examination.
- It affects the quality of an image and low image quality is an obstacle for effective feature extraction, analysis, recognition, and quantitative measurements.
- Therefore, it is essential to denoise low-quality optical remote-sensing images.

OUR GOAL:

- Noise reduction has proven to be an indispensable tool for visualization and pre-processing of multidimensional images.
- Our goal is to remove the noise from the image in such a way that the "original" image is discernible. In case if there is much of spotty noise, it has to be muted out.

WHAT IS NOISY IMAGE?

- Image noise is random variation of brightness or colour information in images and is usually an aspect of electronic noise.
- It can be produced by the image sensor and circuitry of a scanner or digital camera. It can also originate in film grain and in the unavoidable shot noise of an ideal photon detector.
- It is an undesirable by-product of image capture that obscures the desired information.
- The original meaning of "noise" was "unwanted signal", which originates from within the camera and has a few root causes. The three main causes are electricity, heat, and sensor illumination levels.

WHY DENOISING OF AN IMAGE IS IMPORTANT?

- Noise can be introduced by transmission errors and compression. Thus, denoising is often a necessary and the first step to be taken before the images data is analysed.
- It is necessary to apply an efficient denoising technique to compensate for such data corruption.
- Image denoising is to remove noise from a noisy image, to restore the true image.
- It plays an important role in a wide range of applications such as image restoration, visual tracking, image registration, image segmentation, and image classification, where obtaining the original image content is crucial for strong performance.

DISCRETE FOURIER TRANSFORM:

- A Fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT). Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa.
- The DFT is obtained by decomposing a sequence of values into components of different frequencies. An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors.
- As a result, it manages to reduce the complexity of computing the DFT from O(N^2), which arises if one simply applies the definition of DFT, to O (N log N), where N is the data size.

Discrete Fourier Transform

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad N = 0, 1, 2, \ldots, N$$

Inverse Discrete Fourier Transform

$$x[n] = DF^{-1}[x[k]] = \frac{1}{N} \sum_{k=0}^{N} x[k] e^{j2\pi kn/N}, \quad N = 0, 1, 2, \ldots, N$$

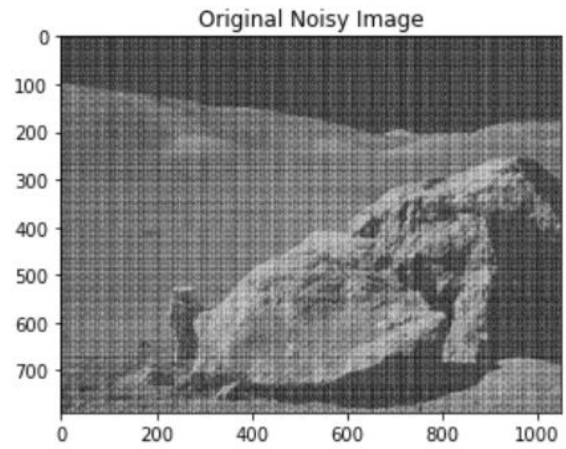
STEPS:

1. Import original noisy image.

```python
image = io.imread('img_noise.png', as_gray=True)
```

```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')
```

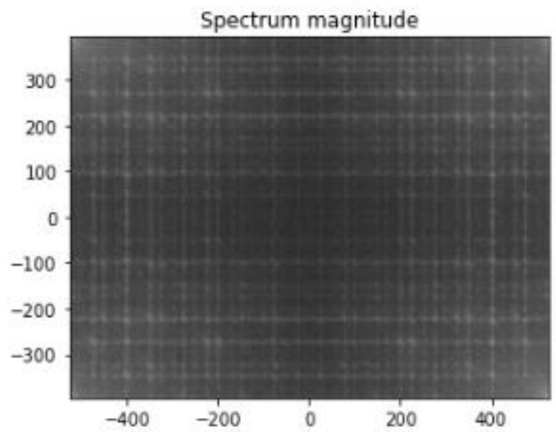


2. Fourier Transform converts the image (time domain function) to a frequency domain function. We take the log of the spectrum to compress the range of values, before displaying.

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

```python
f, ax = plt.subplots(figsize=(5,5))

ax.imshow(np.log(1 + F_magnitude), cmap='gray',
          extent=(-N // 2, N // 2, -M // 2, M // 2))
ax.set_title('Spectrum magnitude');
```

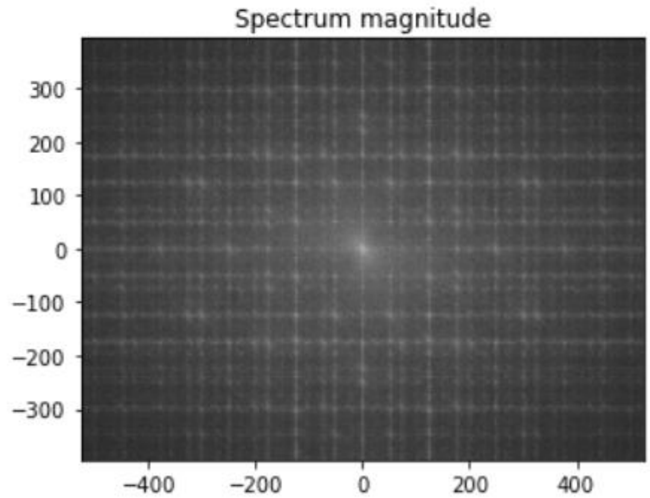


3. Once converted we shift the zero frequency components to the centre.

```python
#shifting 0 frequency component
F_magnitude = fftpack.fftshift(F_magnitude)
```

```python
f, ax = plt.subplots(figsize=(5,5))

ax.imshow(np.log(1 + F_magnitude), cmap='gray',
          extent=(-N // 2, N // 2, -M // 2, M // 2))
ax.set_title('Spectrum magnitude');
```

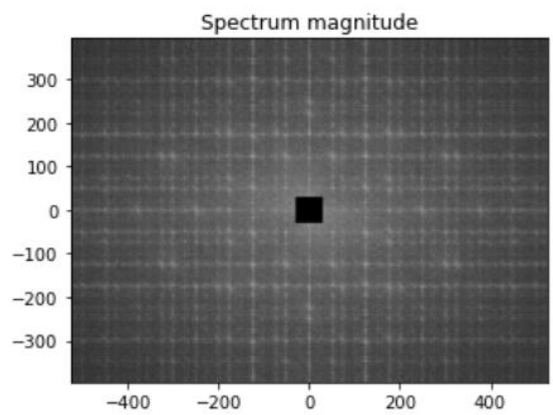


4. We choose a value of "k" such that (2k x 2k) block is formed around the centre to cover maximum cluster of high frequency points.

```python
# Set block around center of spectrum to zero
K = 30
F_magnitude[M // 2 - K: M // 2 + K, N // 2 - K: N // 2 + K] = 0
```

```python
f, ax = plt.subplots(figsize=(5,5))

ax.imshow(np.log(1 + F_magnitude), cmap='gray',
          extent=(-N // 2, N // 2, -M // 2, M // 2))
ax.set_title('Spectrum magnitude');
```



5. For rest of the points, we choose an appropriate percentile of noise to be changed to lower frequencies then we inverse the shift in centre done earlier convert the noisy points to a lower frequency.

```python
# Find all peaks lower than the 98th percentile
peaks = F_magnitude < np.percentile(F_magnitude, 98) #for 98 percentile
peaks1 = F_magnitude < np.percentile(F_magnitude, 100) #for 100 percentile

# Shift the peaks back to align with the original spectrum
peaks = fftpack.ifftshift(peaks) #for 98 percentile
peaks1 = fftpack.ifftshift(peaks1) # for 100 percentile
```

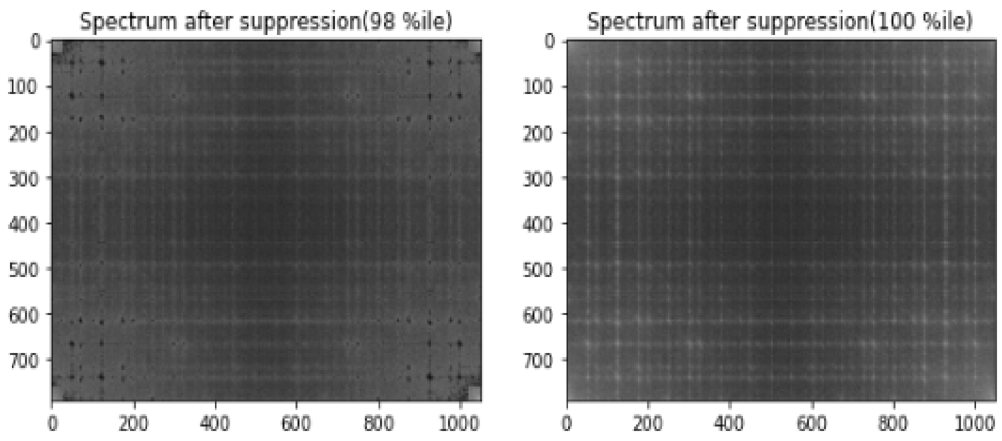
```python
# Make a copy of the original (complex) spectrum
F_dim = F.copy()  # for 98 percentile
F_dim1 = F.copy() #for 100 percentile

# Set those peak coefficients to zero
F_dim = F_dim * peaks.astype(int) # for 98 percentile
F_dim1 = F_dim1 * peaks1.astype(int) #for 100 percentile

f, (ax0, ax2)= plt.subplots(1, 2, figsize=(10,10))
ax0.imshow(np.log10(1 + np.abs(F_dim)), cmap='gray')
ax0.set_title('Spectrum after suppression(98 %ile)')

ax2.imshow(np.log10(1 + np.abs(F_dim1)), cmap='gray')
ax2.set_title('Spectrum after suppression(100 %ile)')
```

```
]: Text(0.5, 1.0, 'Spectrum after suppression(100 %ile)')
```



6. Once the points are changed and use inverse Fourier Transform to get back the image.

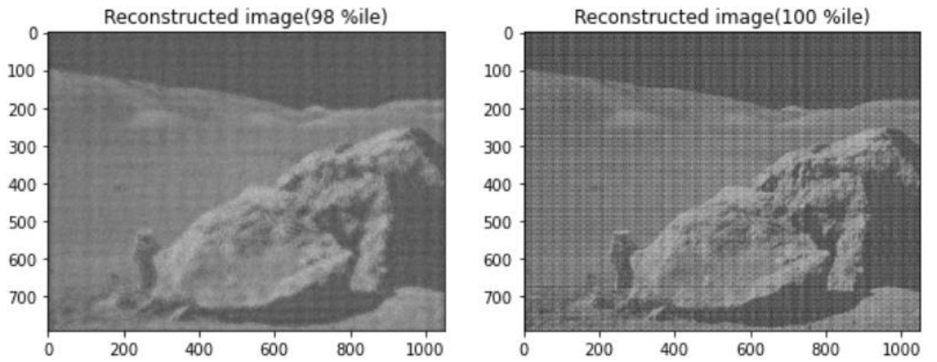
RESULT:

The first image shows our original noisy image. After applying Fourier Transform and Inverse Fourier Transform, we obtain a comparatively clear image i.e., our Reconstructed image.

```
# Do the inverse Fourier transform to get back to an image.
# Since we started with a real image, we only look at the real part of
# the output.
image_filtered = np.real(fftpack.ifft2(F_dim)) #for 98 percentile
image_filtered1 = np.real(fftpack.ifft2(F_dim1)) #for 100 percentile

f, (ax1, ax3) = plt.subplots(1, 2, figsize=(10,10))
ax1.imshow(image_filtered, cmap="gray")
ax1.set_title('Reconstructed image(98 %ile)');

ax3.imshow(image_filtered1, cmap="gray")
ax3.set_title('Reconstructed image(100 %ile)');
```



APPLICATION:

- Image restoration, visual tracking, image registration, image segmentation, and image classification, where obtaining the original image content is crucial for strong performance.
- Artifacts removal from remote sensing data, including ground, drone (UAV), aerial, and space-based measurements.
- The application of image denoising and restoration for classification, land-cover mapping, super-resolution, and sharpening, unmixing, target detection, change detection, multitemporal remote sensing analysis, and data fusion.

CONCLUSION:

- The Fourier transform itself predicts a continuous form of given discrete data, and the transform here performs a nonstationary shift on this continuous function.
- The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent.
- Using this application of Fourier Transform we successfully removed most of the noise present in our image.