Submitted By:

Md. Aukerul Moin Shuvo

Roll: 1603061

CSE, RUET.

Mounting Google Drive

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Importing Python Libraries

In [ ]:
```
from tensorflow.keras.preprocessing.image import load_img
from PIL import Image, ImageOps
import matplotlib.pyplot as plt
import numpy as np
from skimage.morphology import *
```

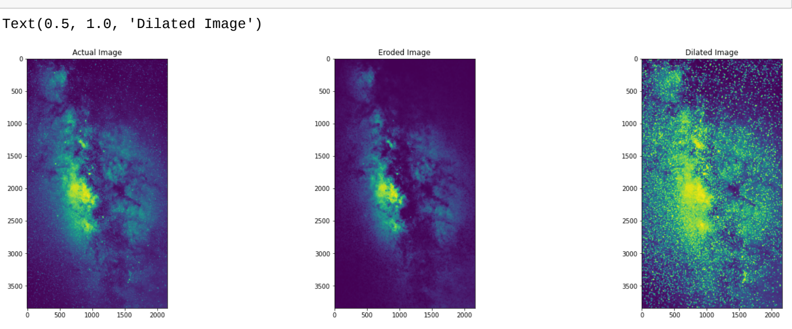
Loading the image from google drive and preprocessing it to a Gray Scale Image

In [ ]:
```
img = Image.open(r"/content/drive/My Drive/DIP/space.jpg")
img = ImageOps.grayscale(img)
```

Erosion & Dilation Using Disk shaped Structuring Element

In [ ]:
```
selem = disk(6)
eroded_disk = erosion(img, selem)

dilated_disk = dilation(img, selem)

plt.figure(figsize = (25,25))
plt.subplot(3,3,1)
plt.imshow(img)
plt.title('Actual Image')

plt.subplot(3,3,2)
plt.imshow(eroded_disk)
plt.title('Eroded Image')

plt.subplot(3,3,3)
plt.imshow(dilated_disk)
plt.title('Dilated Image')
```

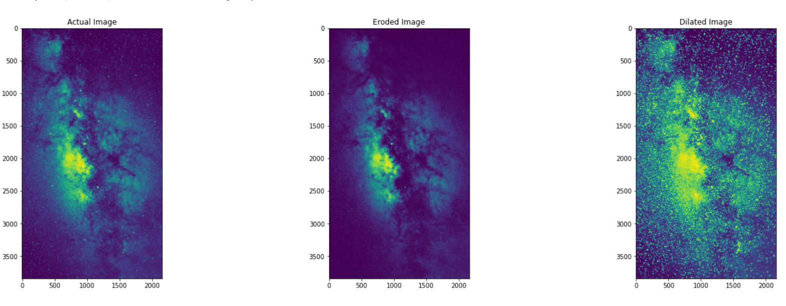
Out[ ]: `Text(0.5, 1.0, 'Dilated Image')`



Erosion & Dilation Using Diamond shaped Structuring Element

In [ ]:
```
selem = diamond(6)
eroded_diamond = erosion(img, selem)

dilated_diamond = dilation(img, selem)

plt.figure(figsize = (25,25))
plt.subplot(3,3,1)
plt.imshow(img)
plt.title('Actual Image')

plt.subplot(3,3,2)
plt.imshow(eroded_diamond)
plt.title('Eroded Image')

plt.subplot(3,3,3)
plt.imshow(dilated_diamond)
plt.title('Dilated Image')
```

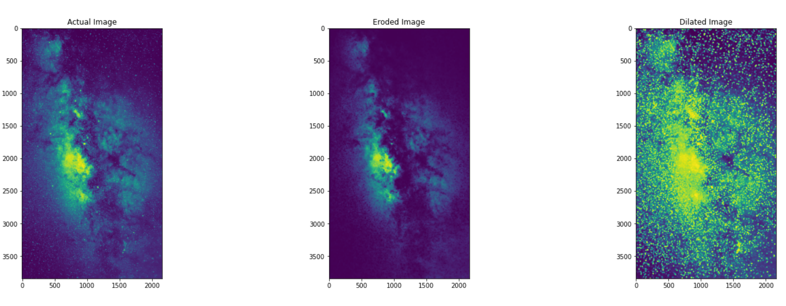
Out[ ]: `Text(0.5, 1.0, 'Dilated Image')`



Erosion & Dilation Using Star shaped Structuring Element

In [ ]:
```
selem = star(6)
eroded_star = erosion(img, selem)

dilated_star = dilation(img, selem)

plt.figure(figsize = (25,25))
plt.subplot(3,3,1)
plt.imshow(img)
plt.title('Actual Image')

plt.subplot(3,3,2)
plt.imshow(eroded_star)
plt.title('Eroded Image')

plt.subplot(3,3,3)
plt.imshow(dilated_star)
plt.title('Dilated Image')
```

Out[ ]: `Text(0.5, 1.0, 'Dilated Image')`

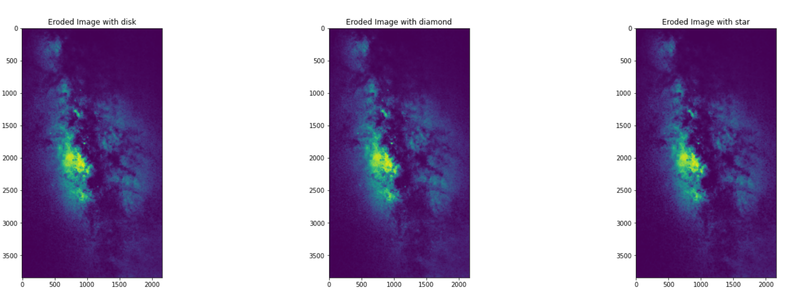


Comparisons

Eroded Images

In [ ]:
```
plt.figure(figsize = (25,25))
plt.subplot(3,3,1)
plt.imshow(eroded_disk)
plt.title('Eroded Image with disk')

plt.subplot(3,3,2)
plt.imshow(eroded_diamond)
plt.title('Eroded Image with diamond')

plt.subplot(3,3,3)
plt.imshow(eroded_star)
plt.title('Eroded Image with star')
```

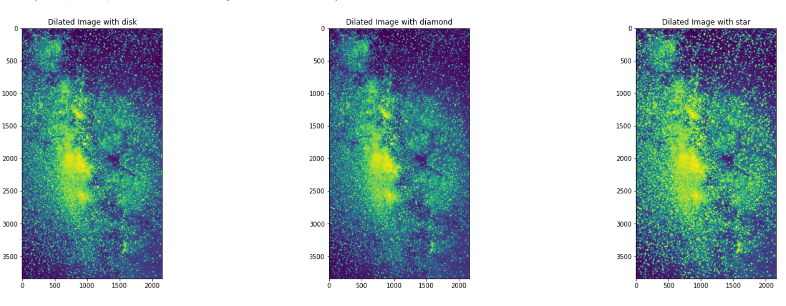
Out[ ]: `Text(0.5, 1.0, 'Eroded Image with star')`



Dilated Images

In [ ]:
```
plt.figure(figsize = (25,25))
plt.subplot(3,3,1)
plt.imshow(dilated_disk)
plt.title('Dilated Image with disk')

plt.subplot(3,3,2)
plt.imshow(dilated_diamond)
plt.title('Dilated Image with diamond')

plt.subplot(3,3,3)
plt.imshow(dilated_star)
plt.title('Dilated Image with star')
```

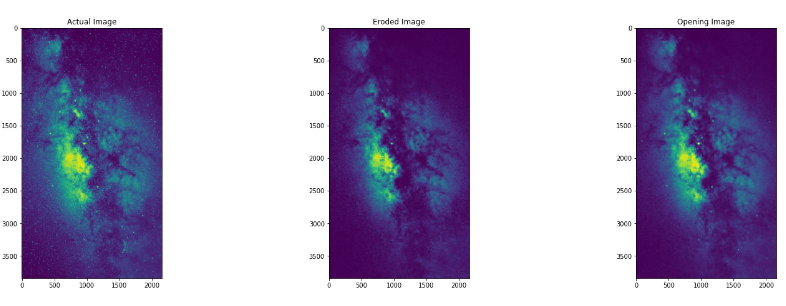
Out[ ]: `Text(0.5, 1.0, 'Dilated Image with star')`



Opening

In [ ]:
```
selem = disk(6)
eroded_disk = erosion(img, selem)

opening_disk = dilation(eroded_disk, selem)

plt.figure(figsize = (25,25))
plt.subplot(3,3,1)
plt.imshow(img)
plt.title('Actual Image')

plt.subplot(3,3,2)
plt.imshow(eroded_disk)
plt.title('Eroded Image')

plt.subplot(3,3,3)
plt.imshow(opening_disk)
plt.title('Opening Image')
```

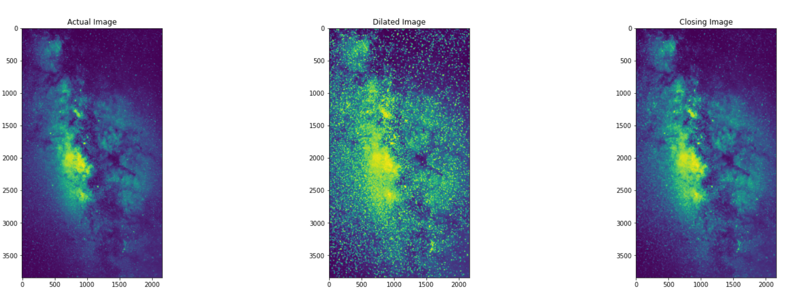
Out[ ]: `Text(0.5, 1.0, 'Opening Image')`



Closing

In [ ]:
```
selem = disk(6)
dilated_disk = dilation(img, selem)

closing_disk = erosion(dilated_disk, selem)

plt.figure(figsize = (25,25))
plt.subplot(3,3,1)
plt.imshow(img)
plt.title('Actual Image')

plt.subplot(3,3,2)
plt.imshow(dilated_disk)
plt.title('Dilated Image')

plt.subplot(3,3,3)
plt.imshow(closing_disk)
plt.title('Closing Image')
```

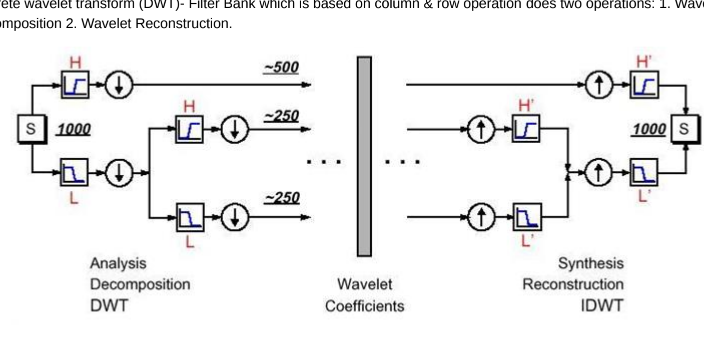
Out[ ]: `Text(0.5, 1.0, 'Closing Image')`



Wavelet Transformation of Image

A discrete wavelet transform (DWT) is a transform that decomposes a given signal into a number of sets, where each set is a time series of coefficients describing the time evolution of the signal in the corresponding frequency band. Wavelets allow both time and frequency analysis of signals simultaneously because of the fact that the energy of wavelets is concentrated in time and still possesses the wave-like (periodic) characteristics. As a result, wavelet representation provides a versatile mathematical tool to analyze transient, time-variant (non-stationary) signals that are not statistically predictable especially at the region of discontinuities – a feature that is typical of images having discontinuities at the edges.

Discrete wavelet transform (DWT)- Filter Bank which is based on column & row operation does two operations: 1. Wavelet Decomposition 2. Wavelet Reconstruction.



Wavelet Decomposition: A wavelet filter bank is an array of wavelet filters used to decompose a signal into sub-bands over different regions of the frequency spectrum, without losing the time domain characterization as performed by the Fourier transform. Wavelet decomposition operation return some coefficients, like:

cA= Approximation Coefficient

cV= Vertical Detailed Coefficient

cH= Horizontal Detailed Coefficient

cD= Diagonal Detailed Coefficient.

For downsampling in this decomposition process, the image shape is reduced by half.

Wavelet Reconstruction: This is the opposite of decomposition process. Same coefficients are also output of this process. Here for upsampling, the reconstructed image becomes the same size of the input image.

Importing Libraries

In [ ]:
```
import numpy as np
import pywt
import pywt.data
import matplotlib.pyplot as plt
```

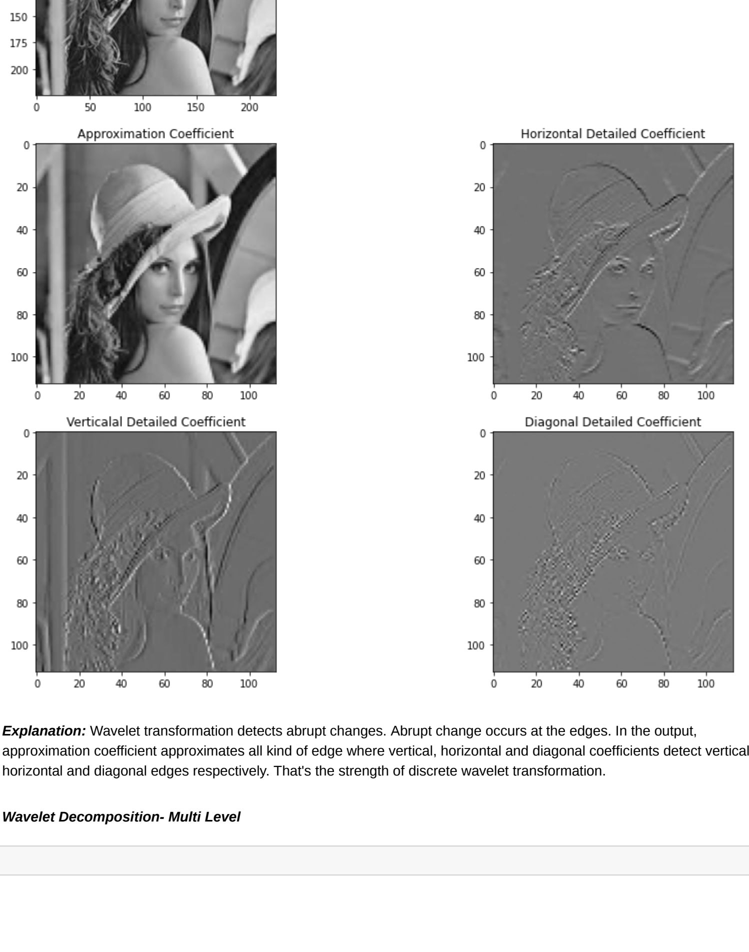
Loading the image from google drive and preprocessing it to a Gray Scale Image

In [ ]:
```
img = Image.open(r"/content/drive/My Drive/DIP/lina.jpg")
img = ImageOps.grayscale(img)
```

Wavelet Decomposition Level-1

In [ ]:
```
#Image decomposition using Discrete Wavelet Transformation
coeffs2 = pywt.dwt2(img, 'haar', mode='periodization')
c = pywt.cwt
cA, (cH, cV, cD) = coeffs2

#Image reconstruction using Inverse Discrete Wavelet Transformation
re_img = pywt.idwt2(coeffs2, 'haar', mode='periodization')


#Image Plotting
plt.figure(figsize = (15,15))
plt.subplot(3,2,1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')

plt.subplot(3,2,3)
plt.title('Approximation Coefficient')
plt.imshow(cA, cmap='gray')

plt.subplot(3,2,4)
plt.title('Horizontal Detailed Coefficient')
plt.imshow(cH, cmap='gray')

plt.subplot(3,2,5)
plt.title('Verticalal Detailed Coefficient')
plt.imshow(cV, cmap='gray')

plt.subplot(3,2,6)
plt.title('Diagonal Detailed Coefficient')
plt.imshow(cD, cmap='gray')
```

Out[ ]: `<matplotlib.image.AxesImage at 0x7f4f001270d0>`



Explanation: Wavelet transformation detects abrupt changes. Abrupt change occurs at the edges. In the output, approximation coefficient approximates all kind of edge where vertical, horizontal and diagonal coefficients detect vertical, horizontal and diagonal edges respectively. That's the strength of discrete wavelet transformation.

Wavelet Decomposition- Multi Level

In [ ]: