CSE 3113 / CSE 3214

INTRODUCTION TO DIGITAL IMAGE PROCESSING

SPRING 2023

Homework 3 Report

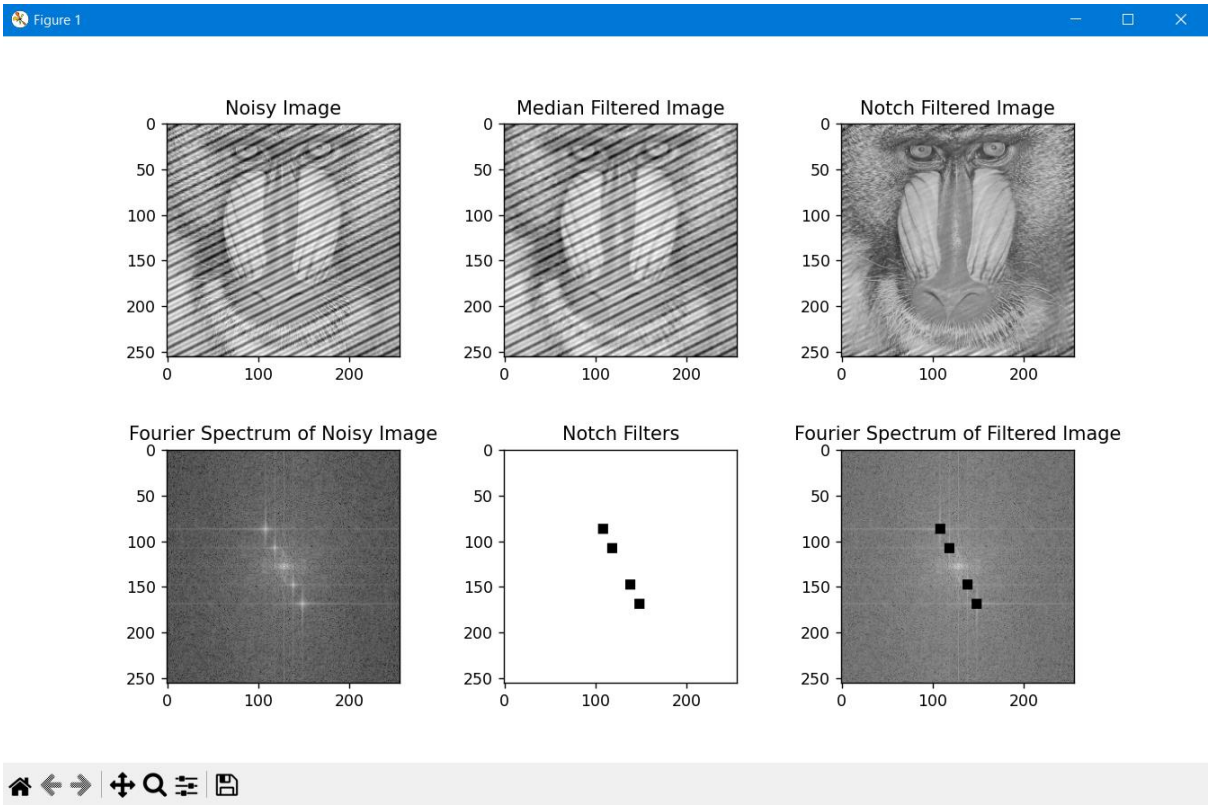
Ferhat Çelik - 190316046

Submission Date: *27 May 2023*

Programming Language	☑ *Python* ☐ *Matlab* ☐ *Octave*
Programming Environment	*I am using Pycharm IDE with Python 3.11.*
Reflections	*It was really hard to write a loop for points.*

Results & Discussion

1. *Paste the output figures that you generated.*



2. *Make a discussion about your results around the following questions.*
   a. *Is the median filter effective at removing this type of noise? Why/why not?*

      It is not because median filter is particularly effective in removing salt-and-pepper noise.

   b. *What is the effect of changing the kernel size of the median filter?*

The kernel size of the median filter determines the size of the neighborhood around each pixel that is considered for computing the median value.

c. *Can you remove this type of noise effectively using some other spatial domain filters?*

The notch filter is the most effective noise filter regarding this type of noise.

Source Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import fftpack
from scipy.ndimage import median_filter

im1 = plt.imread('noisy.tif')
im2 = median_filter(im1, size=3)
im4 = fftpack.fftshift(fftpack.fft2(im1))
noise_points = [(x, y) for x, y in [(87, 108), (108, 118), (148, 138),
(169, 148)]]
im5 = np.ones_like(im4)
for point in noise_points:
    x, y = point
    im5[x-5:x+6, y-5:y+6] = 0

im6 = im4 * im5
notch_filtered_image = np.abs(fftpack.ifft2(fftpack.ifftshift(im6)))

# Create the figure and axes
fig, axes = plt.subplots(2, 3, figsize=(12, 8))

# Plot the first row of the figure
axes[0, 0].imshow(im1, cmap='gray')
axes[0, 0].set_title('Noisy Image')

axes[0, 1].imshow(im2, cmap='gray')
axes[0, 1].set_title('Median Filtered Image')

axes[0, 2].imshow(notch_filtered_image, cmap='gray')
axes[0, 2].set_title('Notch Filtered Image')

axes[1, 0].imshow(np.log(1 + np.abs(im4)), cmap='gray')
axes[1, 0].set_title('Fourier Spectrum of Noisy Image')

axes[1, 1].imshow(np.log(1 + np.abs(im5)), cmap='gray')
axes[1, 1].set_title('Notch Filters')

axes[1, 2].imshow(np.log(1 + np.abs(im6)), cmap='gray')
axes[1, 2].set_title('Fourier Spectrum of Filtered Image')

plt.subplots_adjust(wspace=0.3, hspace=0.4)
plt.show()
```