



CSE 3113 / CSE 3214

INTRODUCTION TO DIGITAL IMAGE PROCESSING

SPRING 2023

Homework 2 Report

Ferhat Çelik - 190316046

Submission Date: 2 May 2023

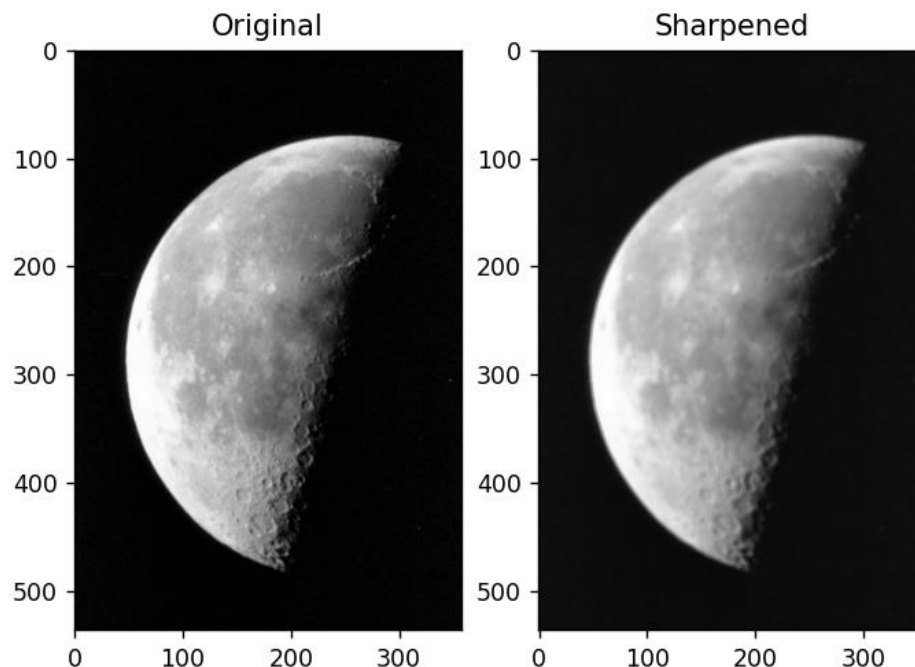
Programming Language	<input checked="" type="checkbox"/> Python <input type="checkbox"/> Matlab <input type="checkbox"/> Octave
Programming Environment	I used Python version 3.11. The libraries I used were numpy, scipy, matplotlib, and pillow.
Your filter (3x3)	<p>Write the values of your filter for size 3x3</p> $h = \begin{bmatrix} -0.11111111 & -0.11111111 & -0.11111111 \\ -0.11111111 & 1.88888889 & -0.11111111 \\ -0.11111111 & -0.11111111 & -0.11111111 \end{bmatrix}$
Reflections	It was really hard to calculate h. Thanks to the information given at the end of video 7, I was able to do it.

Discussion

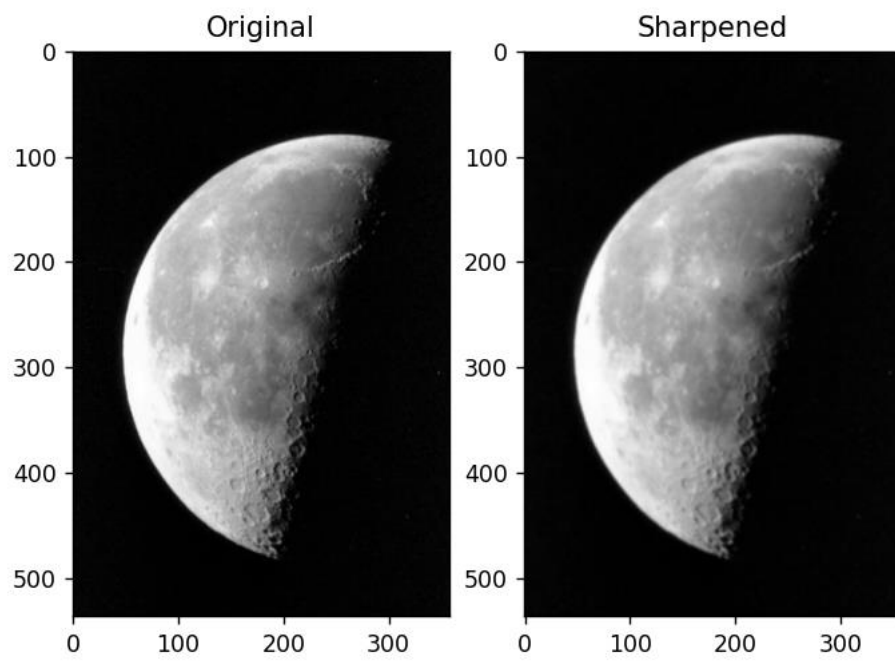
1. Discuss the behavior of your filter for different k values (especially $k < -1$, $k = -1$, $k = 0$, $k = 1$, $k > 1$), with screenshots.

For that purpose I will define s as 3 in every image.

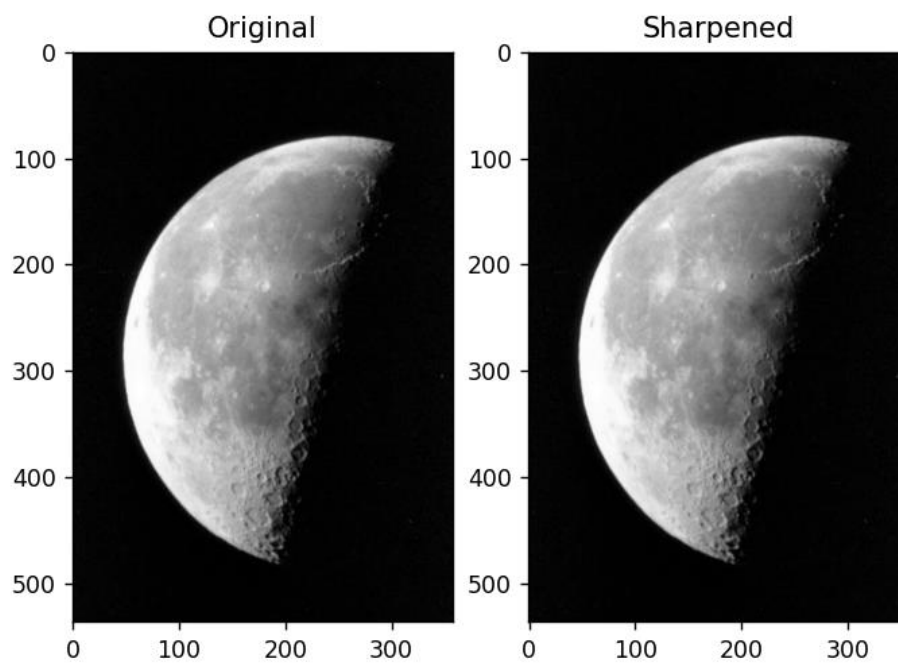
- a. If $k < -1$:



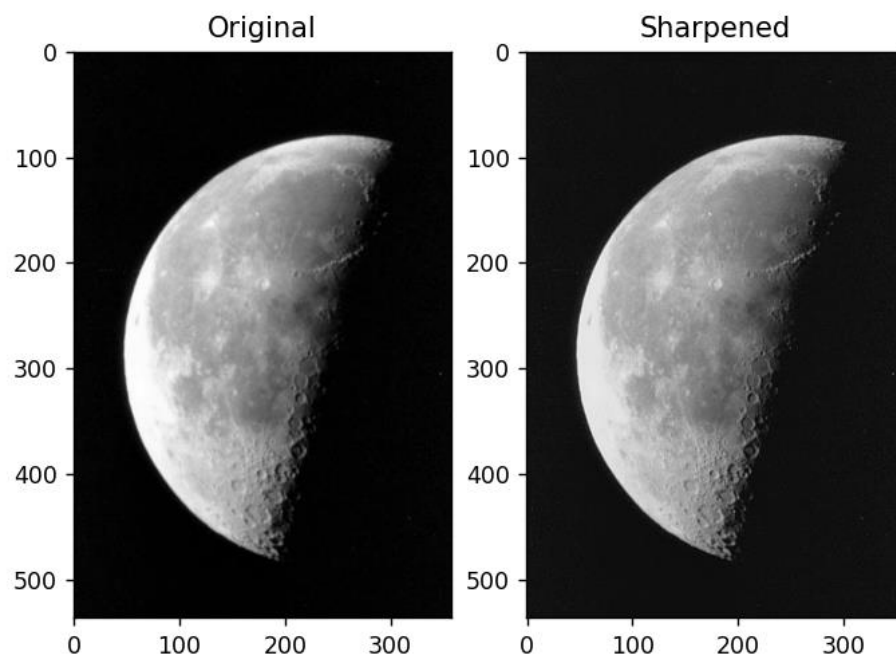
b. If $k = -1$:



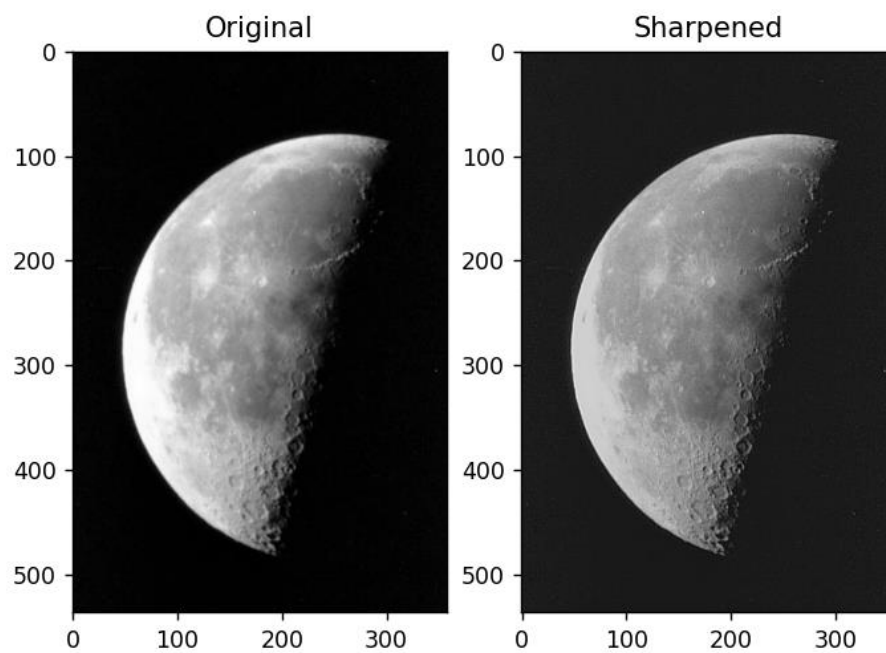
c. If $k = 0$:



d. If $k = 1$:



e. If $k > 1$:



2. What is the effect of filter size s ?

When user increase s the render will look grainier than before.

Source Code

```
import numpy as np
from scipy.signal import convolve2d
import matplotlib.pyplot as plt
from PIL import Image

def calculate_h(s, k):
    identity_filter = np.zeros((s, s))
    identity_filter[s // 2, s // 2] = 1
    box_filter = np.ones((s, s)) / (s ** 2)
    h = (k + 1) * identity_filter - k * box_filter
    return h

s = int(input("Enter s: "))
k = int(input("Enter k: "))
image = Image.open('sample images/moon.tif')

h = calculate_h(s, k)
print("h =")
print(h)

sharpened_image = convolve2d(np.array(image.convert('L')), h, mode="same")

plt.subplot(1, 2, 1)
plt.imshow(image, cmap="gray")
plt.title("Original")
plt.subplot(1, 2, 2)
plt.imshow(sharpened_image, cmap="gray")
plt.title("Sharpened")
plt.show()
```