Department of Computer Science and Engineering (Data Science)

Image Processing and Computer Vision I (DJ19DSL603)

Lab 6: Image Enhancement in Spatial Domain using Neighbourhood Processing Techniques

Name: Yash Thakar SAP ID.: 60009210205

Aim: To perform image enhancement in spatial domain using neighbourhood processing techniques: Basic High Pass and High Boost filtering

Theory:

1. **Basic High Pass:**

The principal objective of high pass (sharpening) filter is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition. Uses of image sharpening vary and include applications ranging from electronic printing and medical imaging to industrial inspection and autonomous target detection in military systems.

The shape of the impulse response needed to have a high pass (sharpening) spatial filter indicates that the filter should have positive coefficients in the outer periphery. For a 3 x 3 mask, choosing a positive value in the centre location with negative coefficients in the rest of the mask meets this condition. Thus when the mask is over an area of constant or slowly varying gray level, the output of the mask is zero or very small. This result is consistent with what is expected from the corresponding frequency domain filter.

−1	−1	−1
−1	8	−1
−1	−1	−1

fig 1. A high pass filter mask

2. **High Boost Filter:**

The goal of high boost filtering is to enhance the high frequency information without completely eliminating the background of the image.

We know that:

(High-pass filtered image)=(Original image)-(Low-pass filtered image)

We define:

$$\text{(High boost filtered image)} = A \times \text{(Original image)} - \text{(Low-pass filtered image)}$$

$$\text{(High boost)} = (A-1) \times \text{(Original)} + \text{(Original)} - \text{(Low-pass)}$$

$$\text{(High boost)} = (A-1) \times \text{(Original)} + \text{(High-pass)}$$

Note:

i. when $A>1$, part of the original is added back to the high-pass filtered version of the image in order to partly restore the lowfrequency components that would have been eliminated with standardhighpass filtering. ii. Typical values for A are values slightly higher than 1, as for example 1.15, 1.2, etc.

The resulting image looks similar to the original image with some edge enhancement.

The spatial mask that implements the high boost filtering algorithm is shown below.

$$\frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & w = 9A - 1 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

fig 2. High boost filter mask

The resulting image depends on the choice of A.

$A = 1.15$ $A = 1.2$

Lab Assignments to complete in this session

Department of Computer Science and Engineering (Data Science)

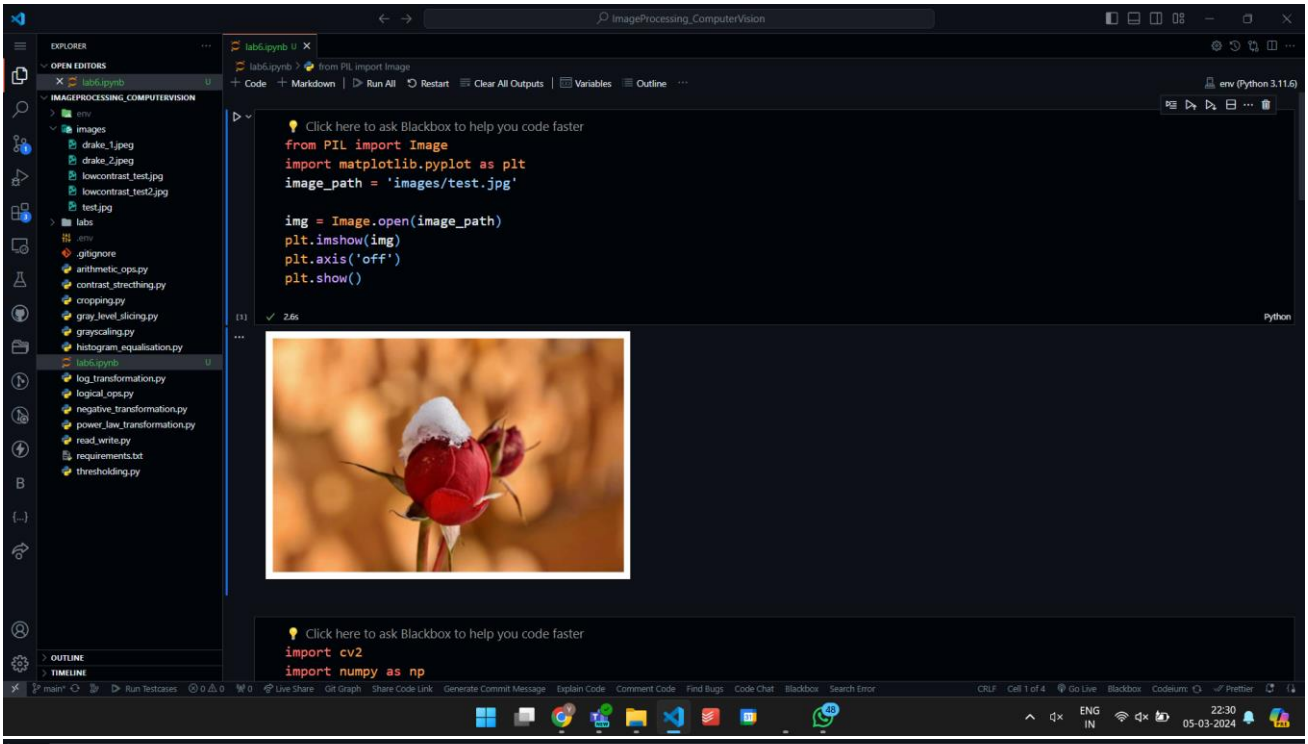
Image Processing and Computer Vision I (DJ19DSL603)

Lab 5: Image Enhancement in Spatial Domain using Neighbourhood Processing Techniques

Problem Statement: Develop a Python program utilizing the OpenCV library to enhace the images in spatial domain using neighbourhood processing with sharpening operators (High pass filtering and High boost filtering). The program should address the following tasks:

1. Read any low contrast image from COVID 19 Image Dataset.
   Dataset Link: [Covid-19 Image Dataset]
2. Display the before & after image(s) used in every task below.
3. Apply basic high pass filter and compare the before and after result. 4. Apply basic high boost filter and compare the before and after result.

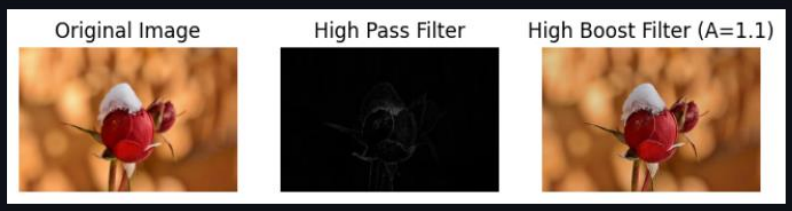
The solution to the operations performed must be produced by scratch coding without the use of built in OpenCV methods.

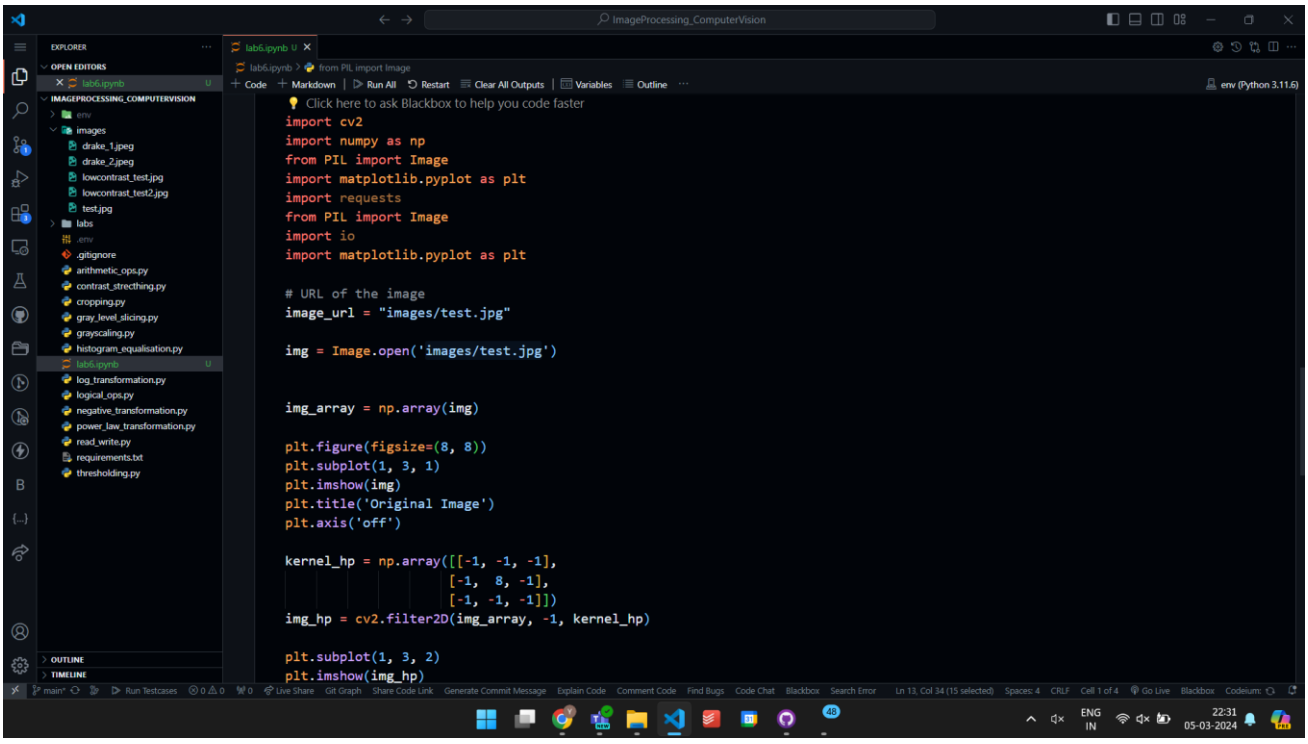


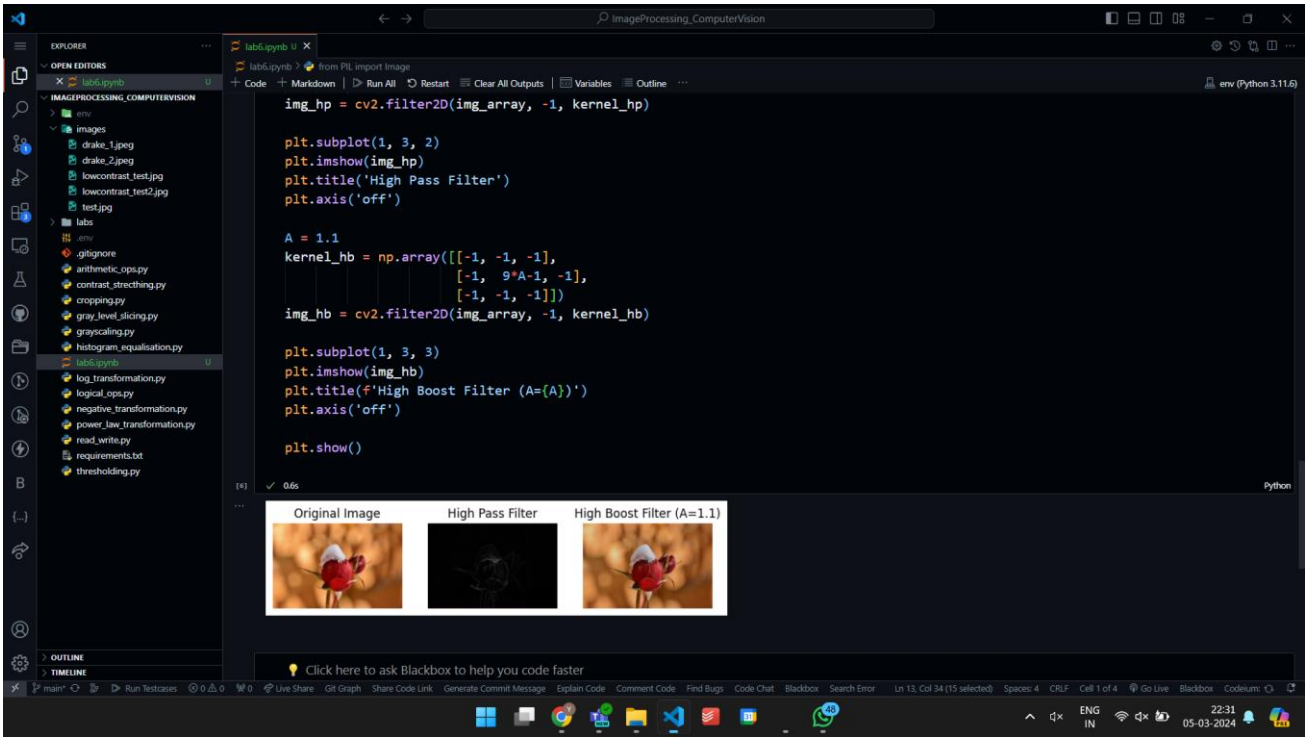

```
Click here to ask Blackbox to help you code faster
import cv2
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

image_path = 'images/test.jpg'
img = Image.open(image_path)

img_array = np.array(img)

plt.figure(figsize=(8, 8))
plt.subplot(1, 3, 1)
plt.imshow(img)
plt.title('Original Image')
plt.axis('off')

kernel_hp = np.array([[-1, -1, -1],
                      [-1,  8, -1],
                      [-1, -1, -1]])
img_hp = cv2.filter2D(img_array, -1, kernel_hp)

plt.subplot(1, 3, 2)
plt.imshow(img_hp)
plt.title('High Pass Filter')
plt.axis('off')

A = 1.1
```

```
A = 1.1
kernel_hb = np.array([[-1, -1, -1],
                      [-1,  9*A-1, -1],
                      [-1, -1, -1]])
img_hb = cv2.filter2D(img_array, -1, kernel_hb)

plt.subplot(1, 3, 3)
plt.imshow(img_hb)
plt.title(f'High Boost Filter (A={A})')
plt.axis('off')

plt.show()
```

[2] ✓ 0.6s






img_hp = cv2.filter2D(img_array, -1, kernel_hp)
plt.subplot(1, 3, 2)
plt.imshow(img_hp)
plt.title('High Pass Filter')
plt.axis('off')
A = 1.1
kernel_hb = np.array([[-1, -1, -1],
[-1, 9*A-1, -1],
[-1, -1, -1]])
img_hb = cv2.filter2D(img_array, -1, kernel_hb)
plt.subplot(1, 3, 3)
plt.imshow(img_hb)
plt.title(f'High Boost Filter (A={A})')
plt.axis('off')
plt.show()
Original Image
High Pass Filter
High Boost Filter (A=1.1)