LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.02**

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)***

|  |  |
| --- | --- |
| Roll No.: C026 | Name: Anirbaan Ghatak |
| Class : B | Batch : B1 |
| Date of Experiment: 02/08/2023 | Date of Submission: 05/08/2023 |
| Grade : |  |

**B.1 Software Code written by student:**

# Aim: To write a program using python to compute different distances within an image and two images

# Name: Anirbaan Ghatak

# Roll No.: C026

import cv2

import numpy as np

#Task 1

image\_path = 'IMG\_2458.jpg'

output\_path = 'IMG\_2458\_grey.jpg'

def show\_image(image, title='image'):

    cv2.imshow(title, image)

    cv2.waitKey(0)

    cv2.destroyAllWindows()

image = cv2.imread(image\_path)

# Convert the image to grayscale

grayscale\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Saving the grayscale image as JPG

cv2.imwrite(output\_path, grayscale\_image)

show\_image(grayscale\_image)

show\_image(image)

image = cv2.resize(grayscale\_image, (1000, 1000))

show\_image(image)

# Modify at 10 pixels randomly in the image

num\_pixels=10

height, width = image.shape[:2]

print(f"Height: {height}, Width: {width}")

mod\_img = image.copy()

for \_ in range(num\_pixels):

    # Generate random coordinates (row, column)

    row = np.random.randint(0, height)

    col = np.random.randint(0, width)

    # Generate a random intensity value (0 to 255)

    intensity = np.random.randint(0, 256)

    # Modify the pixel at the random position with the random intensity

    mod\_img[row, col] = intensity

    print(f"Intensity: {intensity}, Row: {row}, Column: {col}")

show\_image(mod\_img, "Modified Image")

show\_image(image, "Original Image")

block = mod\_img[row-1:row+2, col-1:col+2]

print(block)

# Print the coordinates and intensity values of the modified pixel and its neighbors

print(f"Modified Pixel: ({row}, {col}), Intensity: {intensity}")

print("4-Connectivity Neighbors:")

print("Top:      ", block[0, 1])

print("Bottom:   ", block[2, 1])

print("Left:     ", block[1, 0])

print("Right:    ", block[1, 2])

print("8-Connectivity Neighbors:")

print("Top-Left:     ", block[0, 0])

print("Top-Right:    ", block[0, 2])

print("Bottom-Left:  ", block[2, 0])

print("Bottom-Right: ", block[2, 2])

print("-" \* 20)

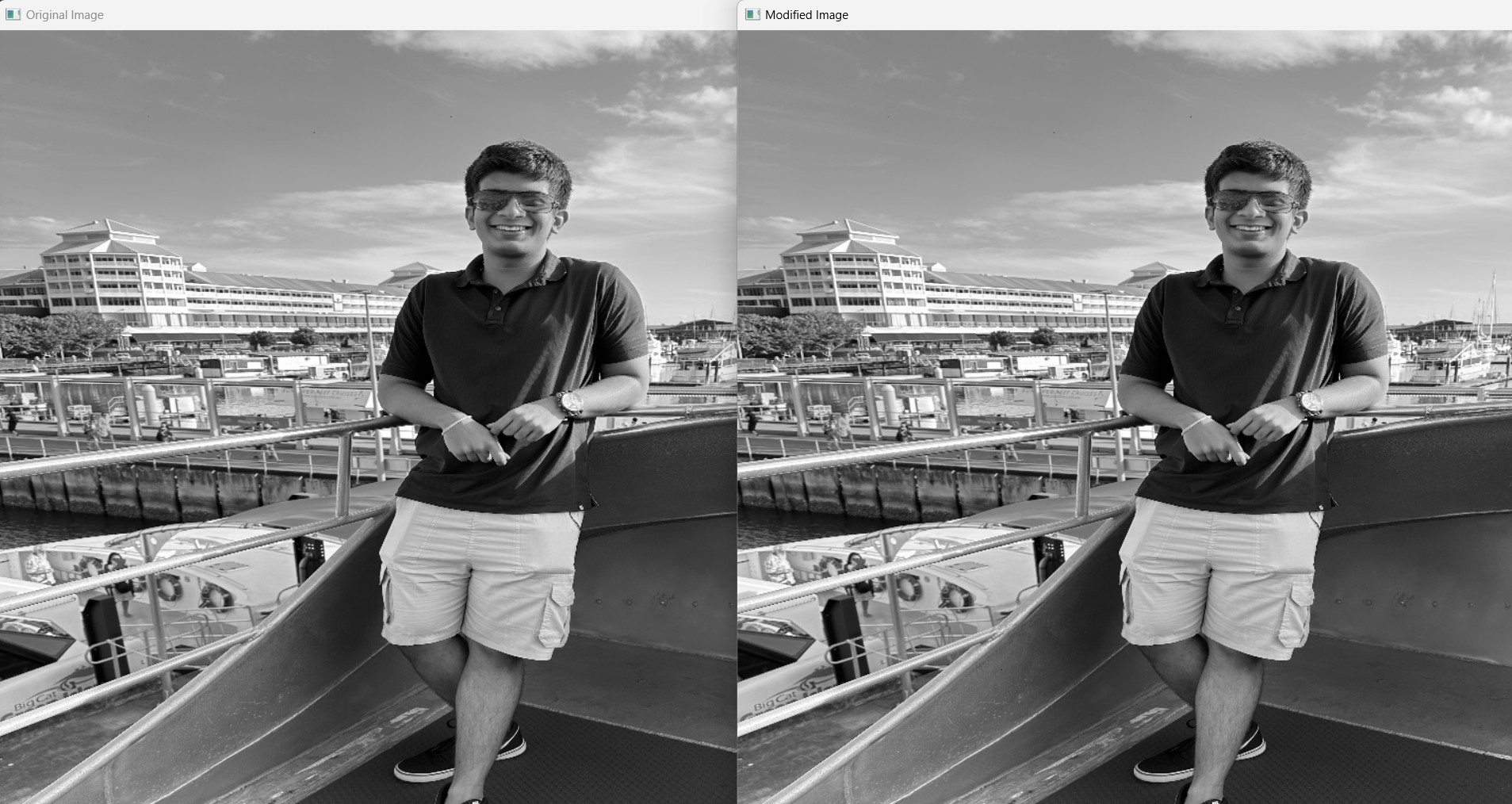
**B.2 Input and Output:**

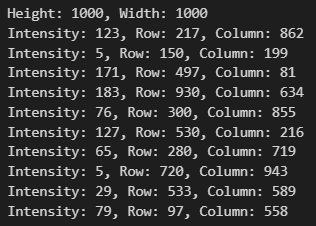
****

**B.2.1 TASK 1**

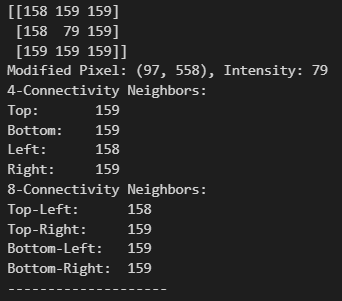
**Input Images:** Your photograph

**Output:**

1. Image with the visible random intensity changes.



1. Distance of each modified pixels in its 3x3 block w.r.t. 4 and 8 connectivity.



1. Percentage of difference w.r.t. the neighbor

50% approx.

1. Comment on closeness of modified pixel with its neighbor.

It all the neighbors are almost equally distant from each other

**B.3 Observations and learning:**

***Understood the importance of grayscale image like faster computation time, less memory use and easier feature extraction due to single-channel information. Learned what are 4 connectivity and 8 connectivity neighbors and implement it on the grayscale image used.***

**B.4 Conclusion:**

***Dived a bit deeper into the OpenCV library and the importance and why grayscale images are used in image processing***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*