**Practical No.7**

**Aim**: Write a program in python for Point Processing in image.

**Softwares Used**: Jupyter Notebook, Opencv

**Environment**: python 3.11.2

**Theory**:

Python is a high-level programming language that is designed to be easy to read and write. Python comes with a large standard library that includes modules for various tasks, such as working with regular expressions, networking, and file I/O. This means that you don't have to write code from scratch for every task, as there's likely a pre-existing module you can use. python is dynamically typed, which means that variables don't need to be declared before they are used, and their types can change at runtime.

OpenCV (Open Source Computer Vision) is an open-source computer vision and machine learning software library. It provides a set of tools and algorithms for image and video processing, machine learning, and computer vision tasks. OpenCV is written in C++, but it also has interfaces for Python, Java, and MATLAB.

**Program Code**:

import cv2

import numpy as np

# Image negative

img = cv2.imread('dolphin.jpg',0)

# To ascertain total numbers of

# rows and columns of the image,

# size of the image

m,n = img.shape

# To find the maximum grey level

# value in the image

L = img.max()

# Maximum grey level value minus

# the original image gives the # negative image

img\_neg = L-img

# convert the np array img\_neg to

# a png image

cv2.imwrite('Dolphin\_Negative7.png', img\_neg)

# Thresholding without background

# Let threshold =T

# Let pixel value in the original be denoted by r

# Let pixel value in the new image be denoted by s

# If r<T, s= 0

# If r>T, s=255

T = 150

# create an array of zeros

img\_thresh = np.zeros((m, n), dtype=int)

for i in range(m):

for j in range(n):

if img[i, j] < T:

img\_thresh[i, j] = 0

else:

img\_thresh[i, j] = 255

# Convert array to png image

cv2.imwrite('Dolphin\_Thresh7.png', img\_thresh)

# the lower threshold value

T1 = 100

# the upper threshold value

T2 = 180

# create an array of zeros

img\_thresh\_back = np.zeros((m, n), dtype=int)

for i in range(m):

for j in range(n):

if T1 < img[i, j] < T2:

img\_thresh\_back[i, j] = 255

else:

img\_thresh\_back[i, j] = img[i, j]

# Convert array to png image

cv2.imwrite('Dolphin\_Thresh\_Back7.png', img\_thresh\_back)

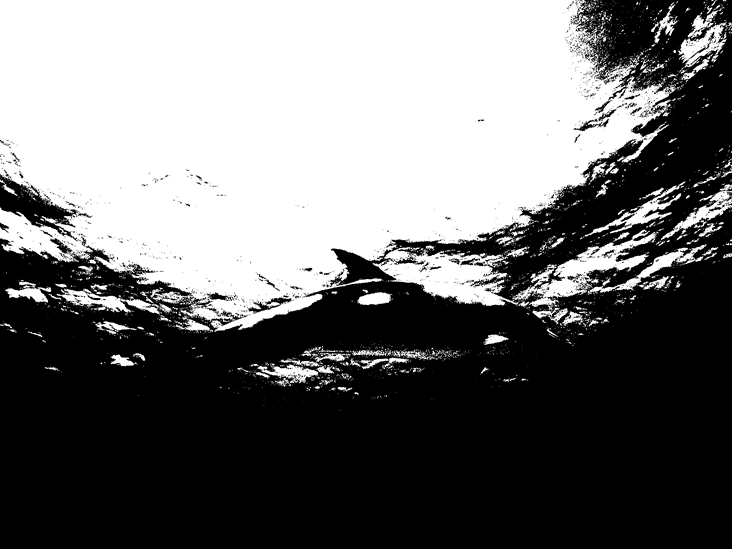
**Output**:

1]Dolphin\_Negative

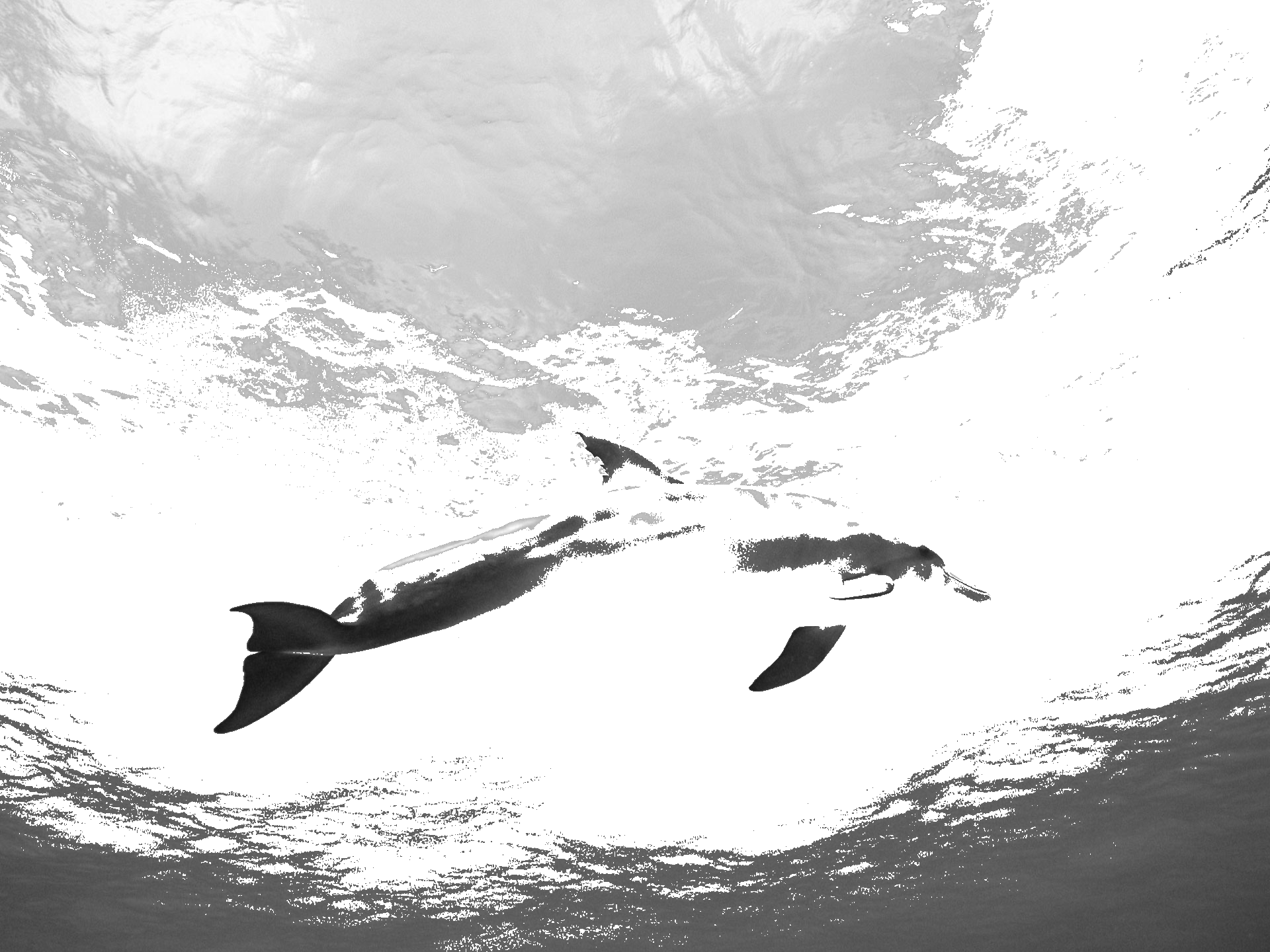


4.7.0

True

2]Dolphin\_Thresh

True

3] Dolphin\_Thresh\_Back

True

**Conclusion**: In this practical , we successfully study and implement about point processing an image using Python.