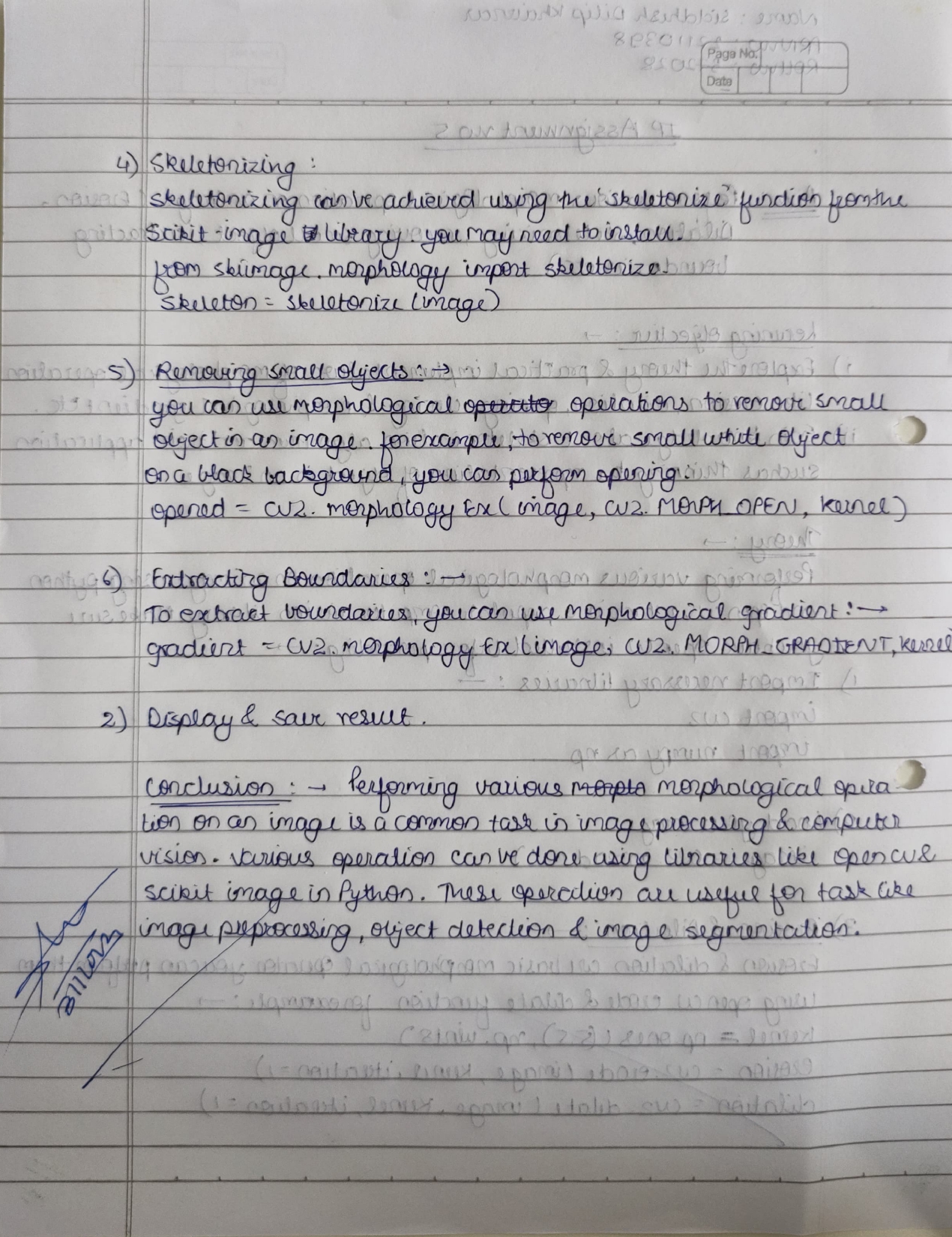
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Bansilal Ramnath Agarwal Charitable Trust's  Vishwakarma Institute of Information  Technology    **Department of**  **Artificial Intelligence and Data Science** | | | |
| Name: Siddhesh Dilip Khairnar | | | | |
| Class: TY | Division: B | | | Roll No: 372028 |
| Semester: V | | Academic Year: 2023-2024 | | |
| Subject Name & Code: Image Processing: ADUA31205(B) | | | | |
| Title of Assignment: Perform various morphological operations on an image. (Erosion, Dilation, Skeletonizing, removing small objects, Extracting boundaries etc.) | | | | |
| Date of Performance: 29-08-2023 | | | Date of Submission: 15-09-2023 | |

**ASSIGNMENT NO. 5**

A piece of paper with writing on it

Description automatically generated



Program Code:

import cv2

import numpy as np

# Reading the input image

img = cv2.imread("C:/Users/asus/Downloads/EvXjoAkUYAE5K7O.jpg", 0)

# Taking a matrix of size 5 as the kernel

kernel = np.ones((5, 5), np.uint8)

# The first parameter is the original image,

# the kernel is the matrix with which the image is

# convolved, and the third parameter is the number

# of iterations, which will determine how much

# you want to erode/dilate a given image.

img\_erosion = cv2.erode(img, kernel, iterations=1)

img\_dilation = cv2.dilate(img, kernel, iterations=1)

cv2.imshow('Input', img)

cv2.imshow('Erosion', img\_erosion)

cv2.imshow('Dilation', img\_dilation)

cv2.waitKey(0)

# Threshold the image

ret, img = cv2.threshold(img, 127, 255, 0)

# Step 1: Create an empty skeleton

size = np.size(img)

skel = np.zeros(img.shape, np.uint8)

# Get a Cross Shaped Kernel

element = cv2.getStructuringElement(cv2.MORPH\_CROSS, (3, 3))

# Repeat steps 2-4

while True:

    # Step 2: Open the image

    opening = cv2.morphologyEx(img, cv2.MORPH\_OPEN, element)

    # Step 3: Subtract open from the original image

    temp = cv2.subtract(img, opening)

    # Step 4: Erode the original image and refine the skeleton

    eroded = cv2.erode(img, element)

    skel = cv2.bitwise\_or(skel, temp)

    img = eroded.copy()

    # Step 5: If there are no white pixels left

    # i.e., the image has been completely eroded, quit the loop

    if cv2.countNonZero(img) == 0:

        break

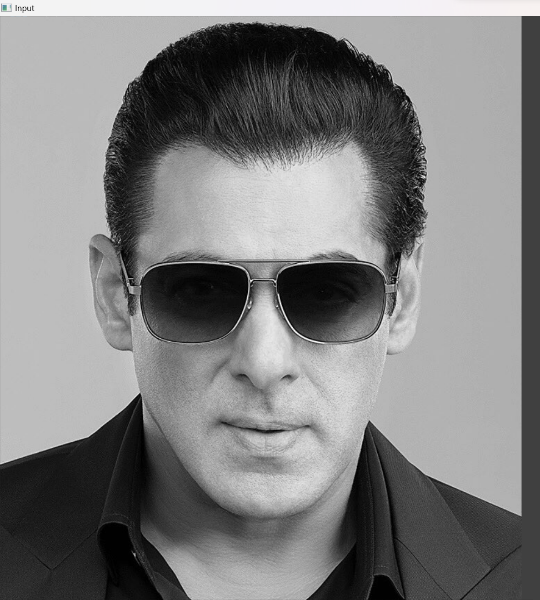
# Displaying the final skeleton

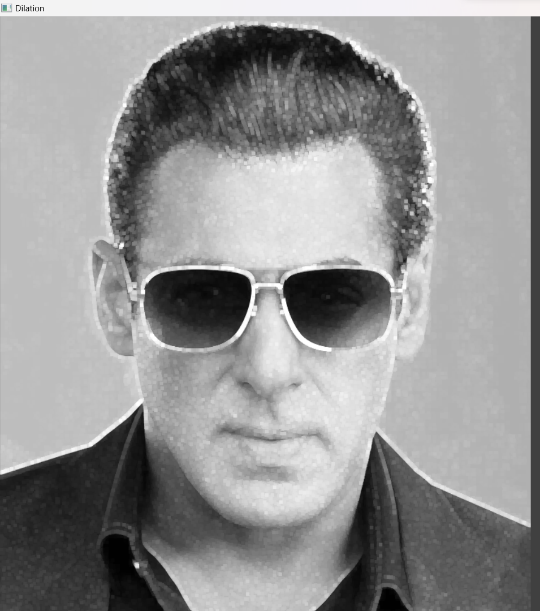
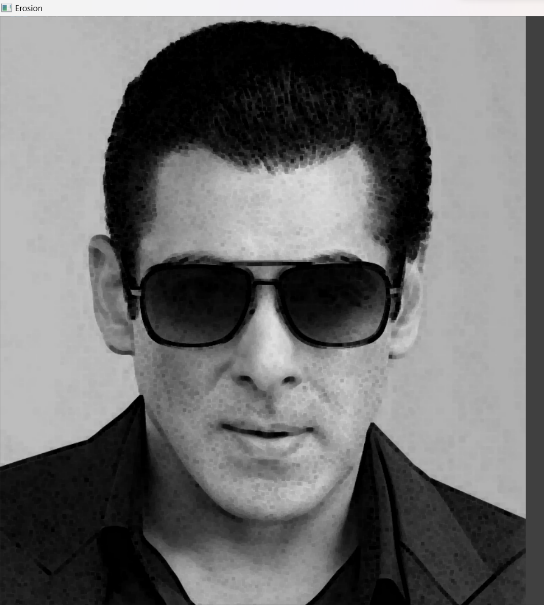
cv2.imshow("Skeleton", skel)

cv2.waitKey(0)

cv2.destroyAllWindows()

Output:



 A screenshot of a phone

Description automatically generated