DIGITAL IMAGE PROCESSING

ASSIGNMENT REPORT

Group Members

- Syed Arsal Rahman
- Areeba Tanveer
- Onkar

Tools Used

- Open CV
- NumPy

Tasks:

- Image Resizing
- RGB to Grayscale Image
- RGB to Binary Image
- Segmentation and object counting

Description Here

OpenCV is an open-source computer vision library that offers tools and algorithms for image and video processing, computer vision tasks, and machine learning integration, widely used in applications ranging from robotics to medical imaging.
NumPy is a Python library for numerical computing, providing support for multi-dimensional arrays and mathematical operations on them

Image Resizing:

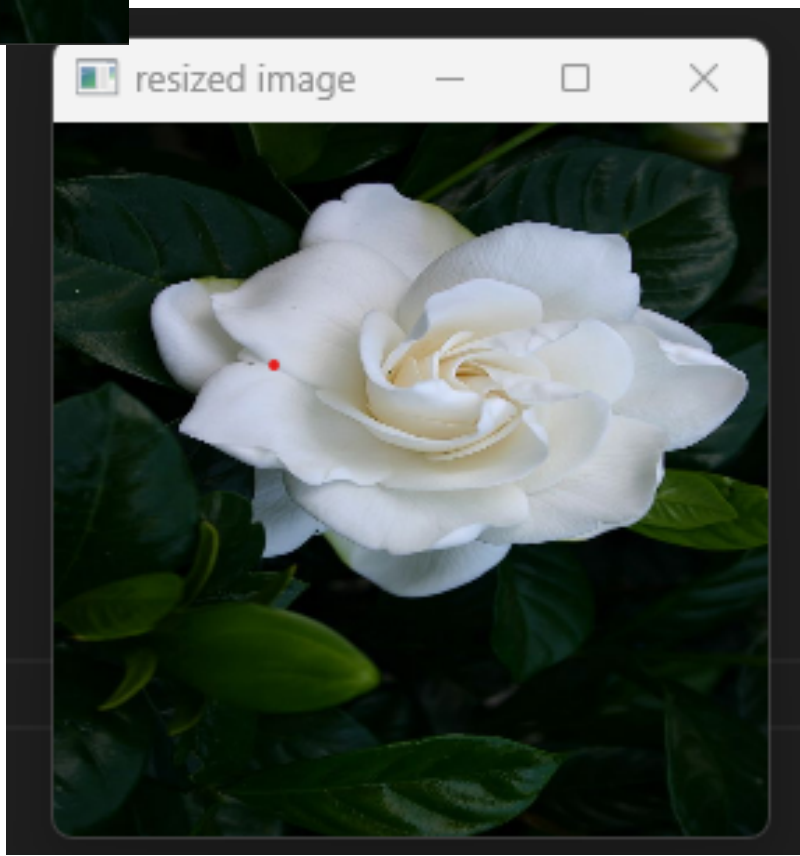
Concept:

We are using the resize function of OpenCv library to resize the image that take Original image, desired size and the type of interpolation as parameters. We used linear interpolation.

resized_img = cv2.resize(img, (256,256) , interpolation = cv2.INTER_LINEAR)

We have displayed the size of the image before and after resizing using the **'img.shape'** that is used to extract the information about image dimensions and channels.

Original vs Resized image



RGB to Grayscale Image

Concept:

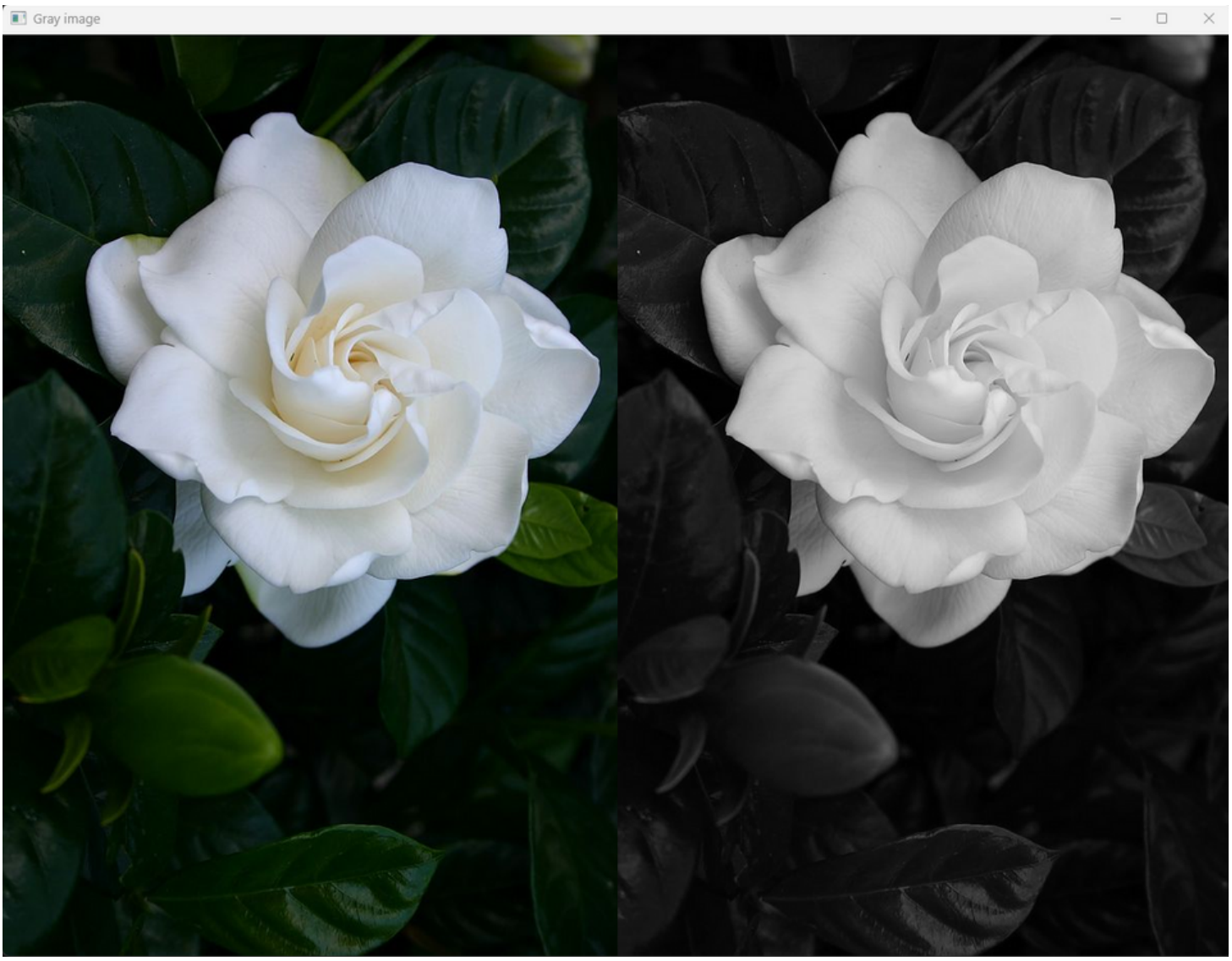
We are using the “**cv2.cvtColor**” function of OpenCv library to convert the Original image to Grayscale image. A grayscale image, often referred to as a "black and white" image, is an image in which each pixel represents a single intensity value, typically ranging from 0 (black) to 255 (white). Line of code is given:

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

The converted image is stored in variable “gray”.

combined_image=np.hstack((img,cv2.cvtColor(gray, cv2.COLOR_GRAY2BGR)))

We are creating a new image called **combined** by horizontally stacking two images, the original one and the one converted to gray scale image



Original vs Grayscaled Image

RGB to Binary Image

Concept:

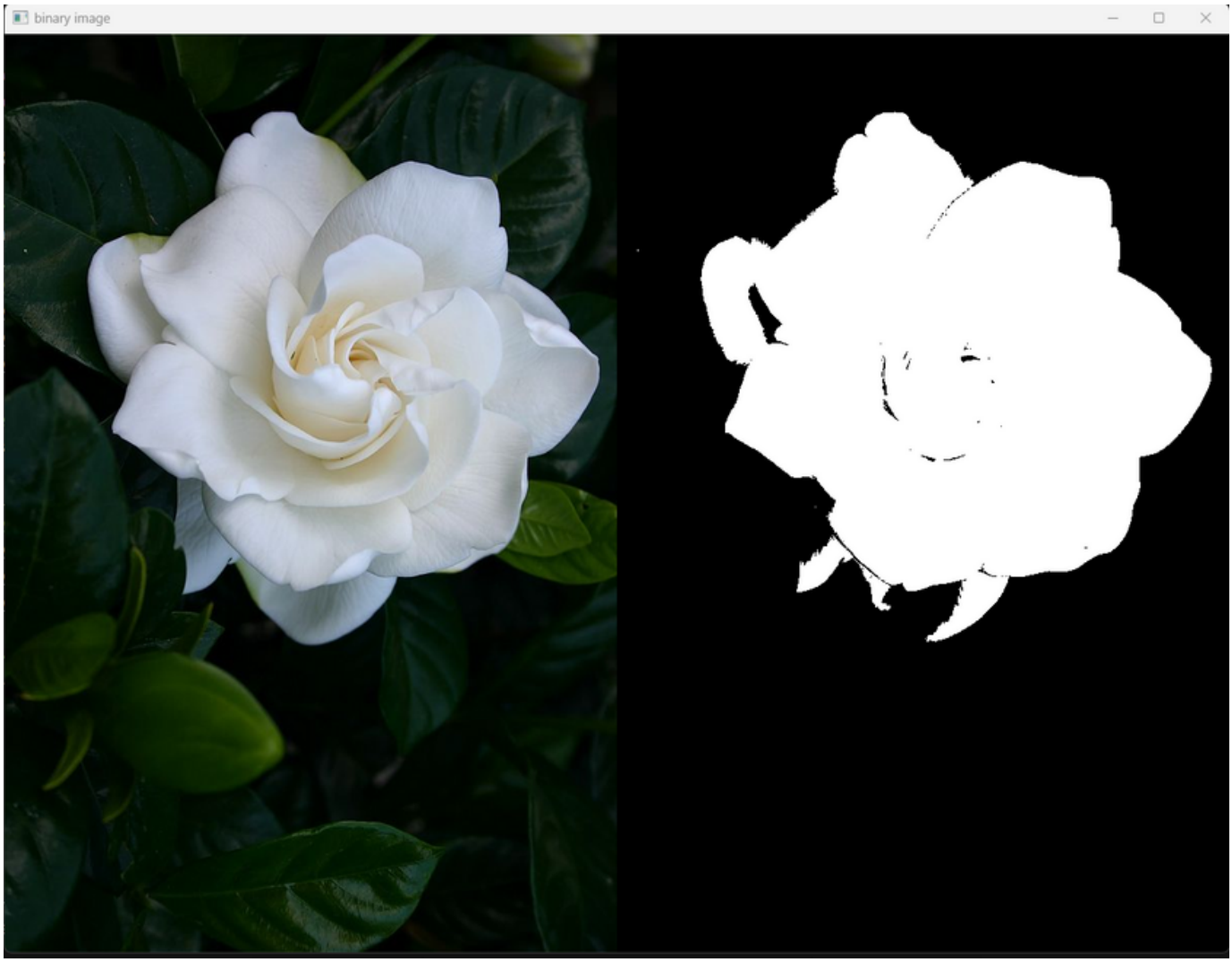
We are using the "**cv2.threshold**" function of OpenCv library to convert the grayscaled image to Binary image.

binary = cv2.threshold(gray,128,255, cv2.THRESH_BINARY)

- The converted image is stored in variable "binary".
- The parameters are the gray image,128 is threshold value that means pixels with intensity equal to or greater than this will be set to maximum value that is 255 and pixels with intensity less than this will be set to 0(black).
- 255: This is the maximum value that the pixels are set to if they pass the threshold.
- **cv2.THRESH_BINARY**: This is the type of thresholding applied. In this case, it's binary thresholding. Pixels are set to either the minimum value (0) or the maximum value (255) based on whether they are below or above the threshold value (128).

combined_image=np.hstack((img,cv2.cvtColor(binary[1], cv2.COLOR_GRAY2BGR)))

A converted version of the binary image (binary[1]) to a 3-channel color image, allowing you to display the binary result alongside the original color image for visual comparison or analysis.



Original vs Binary Image

Segmentation and object count:

Concept

Here are the steps to do so:

Grayscale Conversion:

- gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

Binary thresholding:

- thresh, binary = cv2.threshold(gray,140,255, cv2.THRESH_BINARY)
- Pixels with intensity values greater than or equal to 140 are set to 255 (white), while others are set to 0 (black). This operation separates the objects of interest from the background.

Binary Inversion:

- The binary image is inverted using cv2.bitwise_not to switch the object regions to white and the background to black. This inversion is done to prepare the mask for the next step.

Object Segmentation:

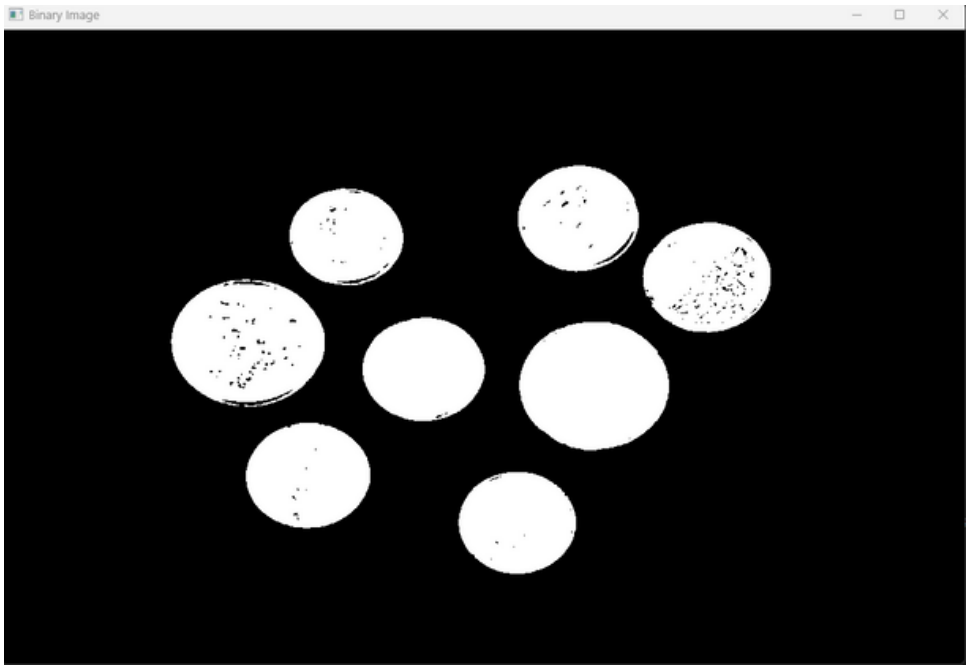
- Object segmentation is performed by applying a bitwise AND operation (**cv2.bitwise_and**) between the original color image and the inverted binary mask. This operation retains only the objects of interest in the original image while making the background black. The result is stored in **result_image**.

Object Counting:

- Connected component analysis is performed using cv2.connectedComponents on the binary image. The total number of objects (excluding the background) is counted and displayed as "Total Objects."

Original

Binary



Segmented