Marwadi University
Faculty of Engineering and Technology
Department of Information and Communication Technology

Experiment-14

Name: Milan Jani
Enroll No.: 92301733041

AIM: Simulate Boundary Extraction on images.

Theory: Boundary extraction is a morphological operation used to extract the boundary or contour of objects in an image. It highlights the boundaries between object regions and the background, providing important information about the shape and structure of objects.

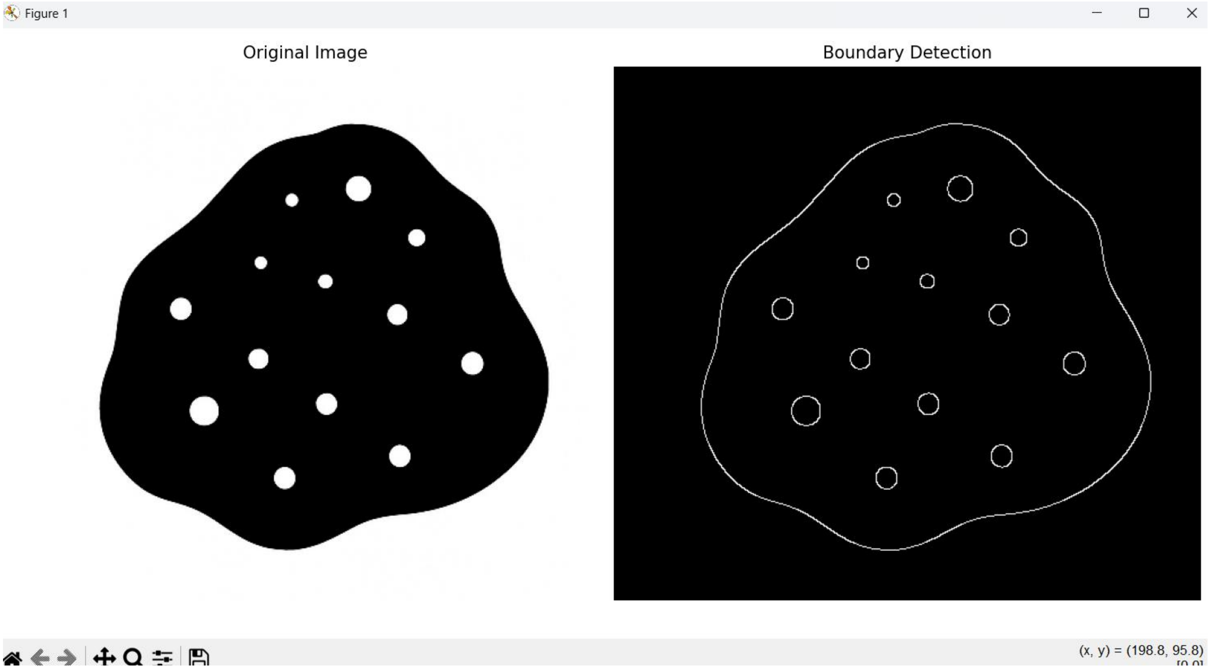
The boundary extraction operation can be achieved by subtracting the input image from its morphological dilation. This highlights the pixels that are on the object boundaries, as the dilation operation expands the object while maintaining its shape.

Program:

```
import cv2

import numpy as np

from google.colab.patches import cv2_imshow

# Load the image

image = cv2.imread('/content/Certificate 1698299913.jpg', cv2.IMREAD_GRAYSCALE)


# Apply Gaussian blur to reduce noise

image_blurred = cv2.GaussianBlur(image, (5, 5), 0)


# Apply Canny edge detection

edges = cv2.Canny(image_blurred, threshold1=30, threshold2=100)


# Create a black image with the same size as the original image

boundary_image = np.zeros_like(image)


# Copy the detected edges to the boundary image

boundary_image[edges > 0] = 255


# Display the original image and the boundary image

cv2_imshow(image)

cv2_imshow(boundary_image)
```

```
# Save the boundary image

cv2.imwrite('boundary_image.jpg', boundary_image)


# Wait for a key press and then close the windows

cv2.waitKey(0)

cv2.destroyAllWindows()
```

Output



Conclusion

- The Canny edge detection algorithm successfully identified the boundaries in the image.
- Gaussian blur preprocessing helped reduce noise for cleaner edge detection.
- The detected edges provide a clear outline of objects in the original image.