

Problem No: 04

Problem Name: Perform Morphological operation (Dilation, Erosion)

Description: Morphological operations are image processing techniques used to analyze and process binary and grayscale images based on their shapes. They apply a structuring element (kernel) to each pixel in the image to modify its shape or structure. These operations are widely used in tasks such as noise removal, object segmentation, boundary detection, and shape analysis. The two most basic morphological operations are Dilation and Erosion.

Dilation

Dilation adds pixels to the boundaries of objects in an image. It causes objects to grow in size and can fill small holes or gaps within the image. Mathematically, it expands the white (foreground) regions by placing the structuring element over them. Dilation is useful for connecting broken parts of objects and enhancing features.

Erosion

Erosion removes pixels from the boundaries of objects, causing them to shrink. It eliminates small white noises and separates connected objects in a binary image. This operation works by eroding the boundaries of foreground regions using the structuring element. Erosion is useful for removing small objects and refining image boundaries.

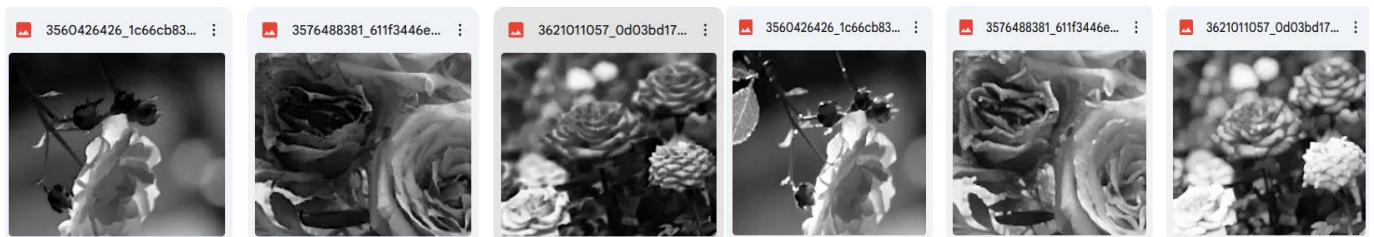


Figure: Images of dilation and erosion folder.

Source code and Output:

```
import os
import cv2
from glob import glob
import numpy as np
# Define the destination directories in Google Drive for dilated and eroded images
output_dir_dilated = '/content/drive/MyDrive/dilated_flower'
output_dir_eroded = '/content/drive/MyDrive/eroded_flower'
# Create the directories if they don't exist
os.makedirs(output_dir_dilated, exist_ok=True)
os.makedirs(output_dir_eroded, exist_ok=True)
# Define a kernel for morphological operations (e.g., a 3x3 kernel)
kernel = np.ones((3,3), np.uint8)
# Process each image in the gray_folder
for img_path in gray_folder:
    # Read the grayscale image
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
```

```

# Apply dilation
dilated_img = cv2.dilate(img, kernel, iterations=1)
# Apply erosion
eroded_img = cv2.erode(img, kernel, iterations=1)
# Get the filename and create the output paths in Google Drive
filename = os.path.basename(img_path)
output_path_dilated = os.path.join(output_dir_dilated, filename)
output_path_eroded = os.path.join(output_dir_eroded, filename)
# Save the dilated and eroded images to Google Drive
cv2.imwrite(output_path_dilated, dilated_img)
cv2.imwrite(output_path_eroded, eroded_img)
print(f"Dilated images saved to {output_dir_dilated}")
print(f"Eroded images saved to {output_dir_eroded}")

```



Figure name: Comparison of Dilated and Eroded image.

Showcase of all operations in one image:



Figure Name: Result of edge detection and morphological operation.