**Aim:** Perform Gray Level Operations Images.

**Theory:-**

* Gray-level operations involve manipulating the pixel values of an image to enhance or modify its appearance. These operations are commonly used in image processing tasks such as contrast adjustment, brightness correction, and image thresholding.

**Programm:-**

import cv2

def perform\_gray\_level\_operation(image, operation):

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

if operation == 'contrast':

contrast\_image = cv2.equalizeHist(gray\_image)

processed\_image = cv2.cvtColor(contrast\_image, cv2.COLOR\_GRAY2BGR)

elif operation == 'brightness':

alpha = 1.5 # brightness factor

processed\_image = cv2.convertScaleAbs(gray\_image, alpha=alpha)

processed\_image = cv2.cvtColor(processed\_image, cv2.COLOR\_GRAY2BGR)

elif operation == 'thresholding':

\_, threshold\_image = cv2.threshold(gray\_image, 127, 255, cv2.THRESH\_BINARY)

processed\_image = cv2.cvtColor(threshold\_image, cv2.COLOR\_GRAY2BGR)

else:

print("Invalid operation. Available operations: 'contrast', 'brightness', 'thresholding'")

return None

return processed\_image

# Load the input image

image\_path = './Images.jpg'

input\_image = cv2.imread(image\_path)

# Perform gray level operation

operation\_type = 'contrast' # Change this to the desired operation: 'contrast', 'brightness', 'thresholding'

output\_image = perform\_gray\_level\_operation(input\_image, operation\_type)

if output\_image is not None:

# Display the processed image

cv2.imshow('Processed Image', output\_image)

cv2.waitKey(0)

# Save the processed image (optional)

output\_path = 'output\_image.jpg'

cv2.imwrite(output\_path, output\_image)

print(f"Processed image saved at: {output\_path}")

**Output:-**

|  |  |
| --- | --- |
| **Original Image** | **Processed Image** |
|  |  |

**Conclusion :-**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_