**DEPARTMENT OF INFORMATION TECHNOLOGY**

**COURSE CODE: DJS22ITL603 DATE: 20-02-25**

**COURSE NAME: Image Processing and Computer Vision Laboratory CLASS: TY-IT1-1**

**NAME:** Anish Sharma **DIV:** IT1 **ROLL:** I011

**LAB EXPERIMENT NO.3**

**CO/LO:** Apply Image Enhancement Techniques.

**AIM / OBJECTIVE:** To perform Image Enhancement using Pillow/Open CV **EXERCISE:**

Consider an image of your choice. Perform the following operations using methods using PIL/OpenCV:

1.Negative of an Image (Gray Scale, RGB)

2.Apply Thresholding

3.Apply Contrast Stretching

4.Apply Intensity Level Slicing with and without background

5.Apply log transformation

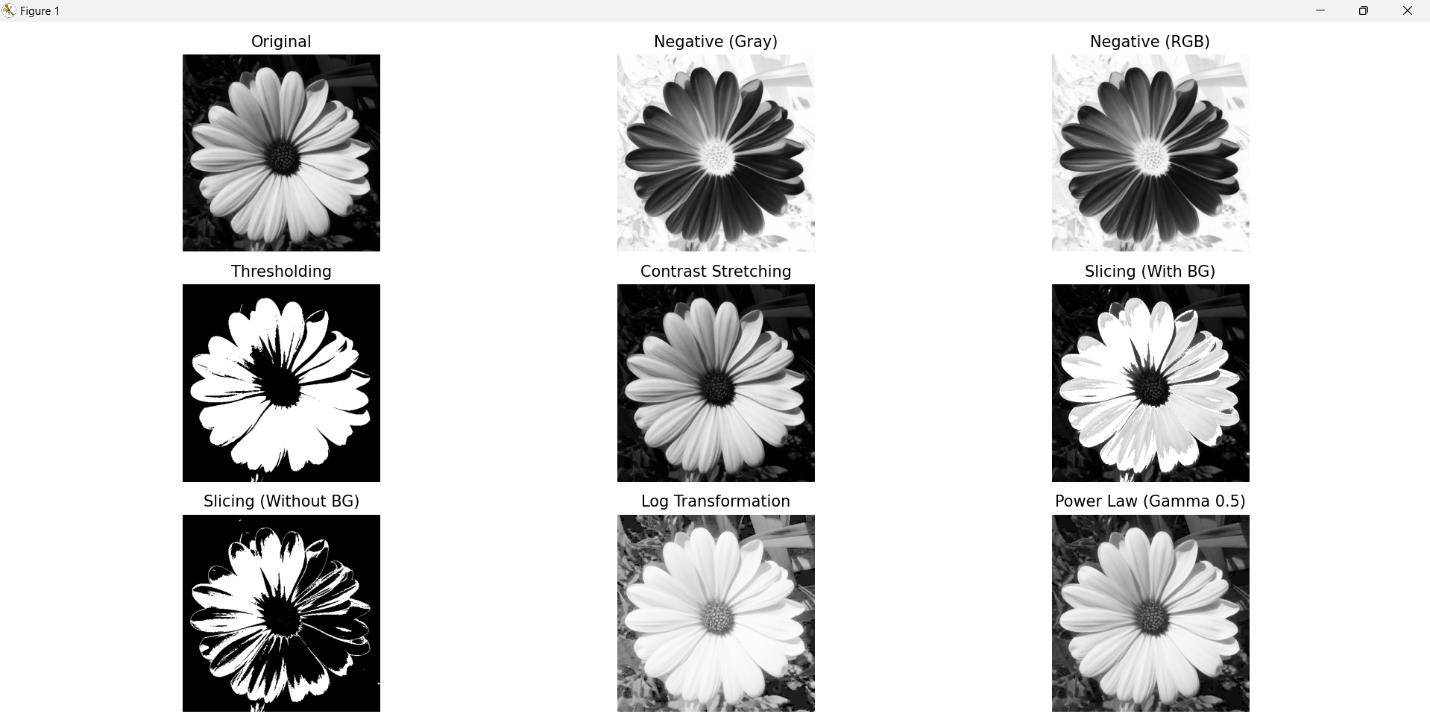
1. Apply power Law Transformation
2. What difference do you observe for the chosen image after applying contrast stretching, log transformation and power law transformation?

**SOURCE CODE:**

|  |
| --- |
| import cv2 import numpy as np import matplotlib.pyplot as plt    # Load an image  image = cv2.imread('parrot.jpg') # Replace 'image.jpg' with the path to your image image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB) # Convert to RGB  # Convert to grayscale gray\_image = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY) |

|  |
| --- |
| # 1. Negative of Image (Gray Scale, RGB) negative\_gray = 255 - gray\_image negative\_rgb = 255 - image    # 2. Apply Thresholding  \_, thresholded = cv2.threshold(gray\_image, 127, 255, cv2.THRESH\_BINARY)  # 3. Apply Contrast Stretching def contrast\_stretching(img):  min\_val = np.min(img) max\_val = np.max(img) stretched = (img - min\_val) \* (255 / (max\_val - min\_val)) return stretched.astype(np.uint8)  contrast\_rgb = contrast\_stretching(image)    # 4. Apply Intensity Level Slicing (with and without background) def intensity\_level\_slicing(img, min\_range, max\_range, preserve\_bg=True): sliced = np.where((img >= min\_range) & (img <= max\_range), 255, img if preserve\_bg else 0) return sliced.astype(np.uint8)  sliced\_with\_bg = intensity\_level\_slicing(gray\_image, 100, 200, True) sliced\_without\_bg = intensity\_level\_slicing(gray\_image, 100, 200, False)  # 5. Apply Log Transformation def log\_transform(img):  c = 255 / np.log(1 + np.max(img)) log\_img = c \* (np.log(1 + img)) return log\_img.astype(np.uint8)    log\_transformed = log\_transform(image)    # 6. Apply Power Law (Gamma) Transformation def gamma\_transform(img, gamma): c = 255 / (255 \*\* gamma) gamma\_img = c \* (img \*\* gamma) return gamma\_img.astype(np.uint8)  gamma\_transformed = gamma\_transform(image, 0.5)    # Display images titles = [ |
| 'Original', 'Negative (Gray)', 'Negative (RGB)',  'Thresholding', 'Contrast Stretching', 'Slicing (With BG)', 'Slicing (Without BG)',  'Log Transformation', 'Power Law (Gamma 0.5)'  ] images = [ image, negative\_gray, negative\_rgb, thresholded, contrast\_rgb, sliced\_with\_bg, sliced\_without\_bg, log\_transformed, gamma\_transformed  ] plt.figure(figsize=(12, 8)) for i in range(len(images)): plt.subplot(3, 3, i + 1) plt.imshow(images[i], cmap='gray' if len(images[i].shape) == 2 else None) plt.title(titles[i]) plt.axis('off')  plt.tight\_layout() plt.show() |

OUTPUT:



CONCLUSION:

The applied image processing techniques resulted in various visual transformations. The negative image inversion (both grayscale and RGB) swapped dark and light areas, while thresholding created a binary image based on pixel intensity. Contrast stretching enhanced image contrast by expanding the pixel range, and intensity level slicing emphasized a specific intensity range, with or without preserving the background. Log transformation brightened darker areas, and the gamma transformation adjusted the image's brightness non-linearly. Each technique altered the image in a unique way, showcasing different methods for enhancing or manipulating visual features.

**REFERENCES:**.

**Website References:**

1. Akyol, G. (2023, January 14). What is Image Enhancement? | Image Processing #3. Medium. Retrieved from https://medium.com/@gokcenazakyol/what-is-imageenhancement-imageprocessing-3-32a813087e0a

2.Javatpoint. (n.d.). Gray Level Transformation. Retrieved February 9, 2025, from https://www.javatpoint.com/dip-gray-level-transformation