SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

**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

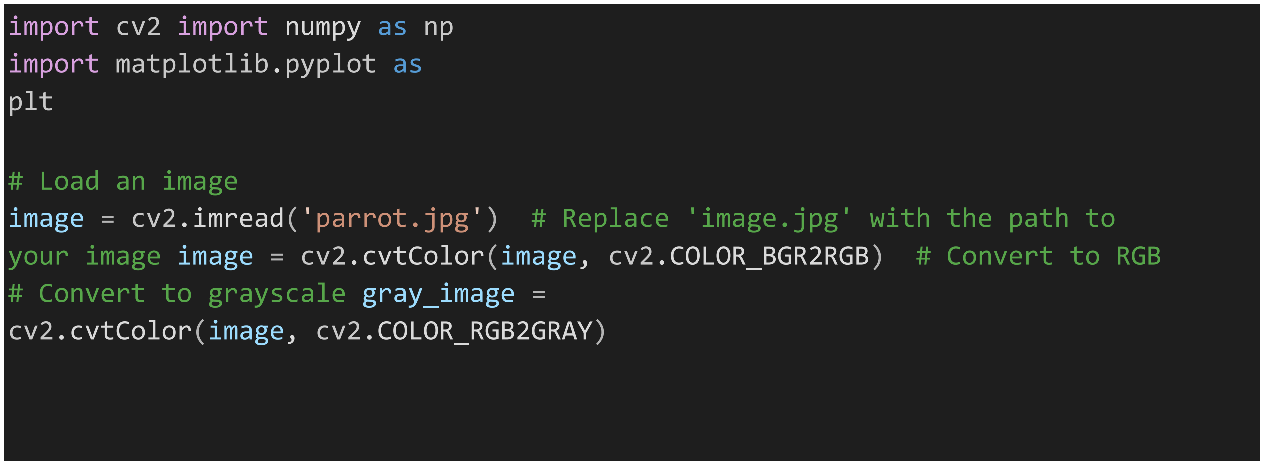
4.Apply Intensity Level Slicing with and without background

5.Apply log transformation

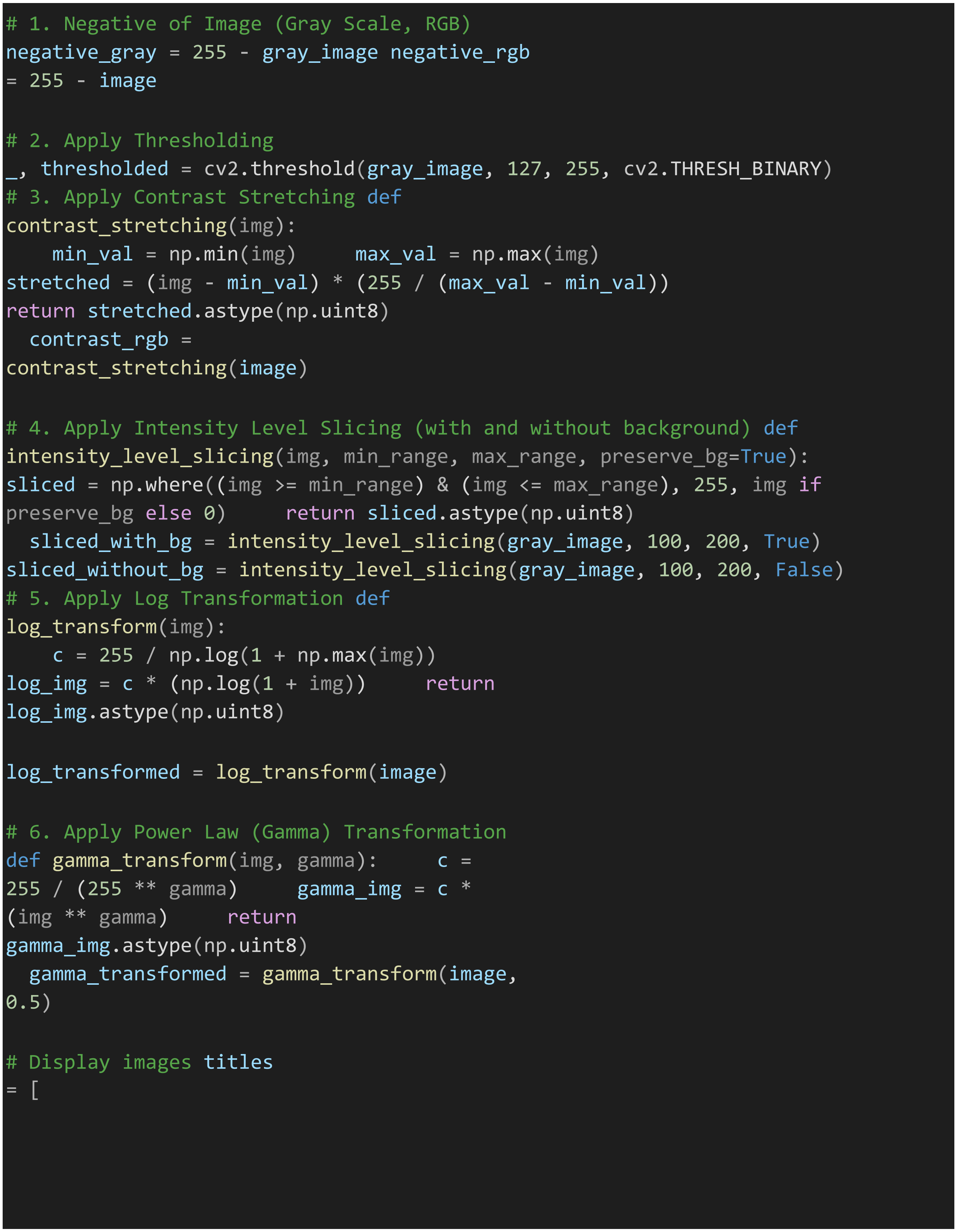
6. Apply power Law Transformation

7. What difference do you observe for the chosen image after applying contrast stretching, log transformation and power law transformation?

**SOURCE CODE:**

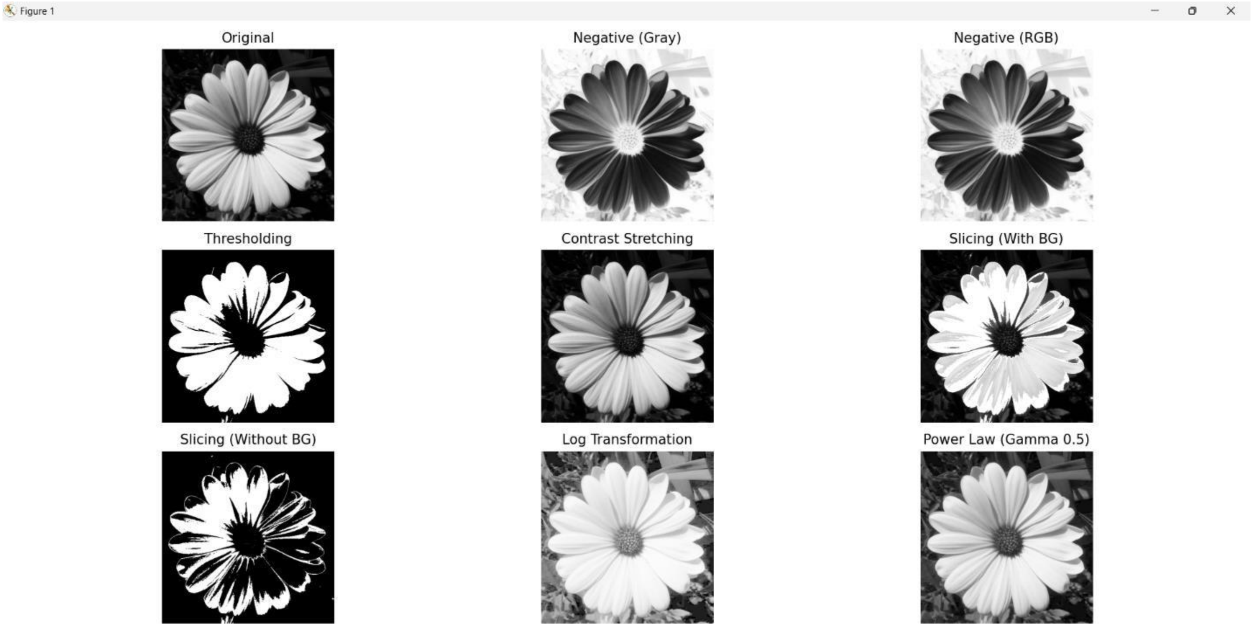


```
import cv2 import numpy as np
import matplotlib.pyplot as
plt

# 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