

DSIP

Image Operations

Code:

```
import cv2
import numpy as np

orig = cv2.imread("gray_test.jpg", 0)
cv2.imshow("Original", orig)

# Image Negative
img_not = cv2.bitwise_not(orig)
cv2.imshow("Invert", img_not)

# Gray level slicing
rows, columns = orig.shape
sliced_image = np.zeros((rows, columns), dtype = 'uint8')
min_range = 20
max_range = 50
for i in range(rows):
    for j in range(columns):
        if orig[i, j] >= min_range and orig[i, j] <= max_range:
            sliced_image[i, j] = 0
        else:
            sliced_image[i, j] = 255
cv2.imshow("Gray Level slicing", sliced_image)

# Thresholding
_, thresh1 = cv2.threshold(orig, 127, 255, cv2.THRESH_BINARY)
cv2.imshow("Binary Thresholding", thresh1)
_, thresh2 = cv2.threshold(orig, 127, 255, cv2.THRESH_BINARY_INV)
cv2.imshow("Inverted Binary Thresholding", thresh2)
thresh3 = cv2.adaptiveThreshold(orig, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
cv2.THRESH_BINARY, 3, 2)
cv2.imshow("Adaptive Mean Thresholding", thresh3)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:

