Shri Vile Parle 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 Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)

Lab 2: Image Negative Transformation, Thresholding, Gray Level Slicing with without background

Name:Suyash S Konduskar **sap id: 60009220109**

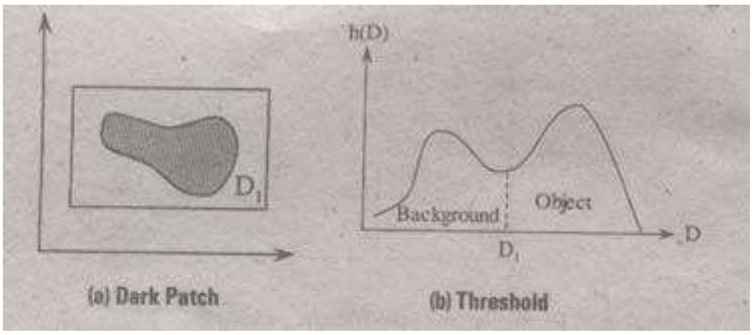
Aim: To Perform Gray Level slicing with and without background

Theory: Thresholding is a simple but effective image processing technique that is widely used in computervision and image analysis applications. The basic idea behind thresholding is to segment an image into foreground and background regions based on the intensity values of the pixels in the image

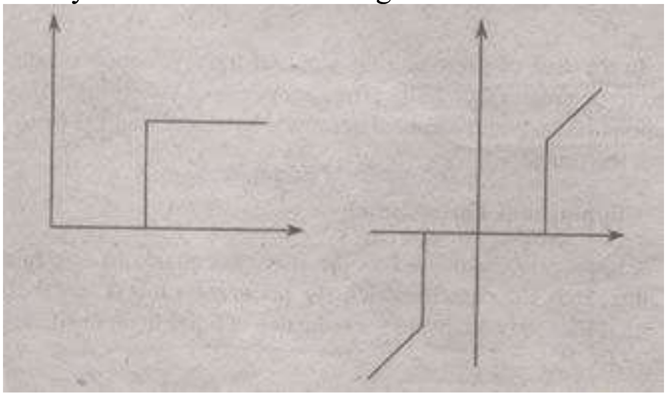
Clipping and Thresholding:

Clipping is considered as the special scenario of contrast stretching. It is the case in which the parameters are α = γ = 0. Clipping is more advantageous for reduction of noise in input signals of range [a, b].

Threshold of an image is selected by means of its histogram. Let us take the image shown in the following figure.



(b) consists of two peaks i.e., background and object. At the abscissa of histogram minimum (D1) the threshold is selected. This selected threshold (D1) can separate background and object to convert the image into its respective binary form. The thresholding transformations are shown in figure

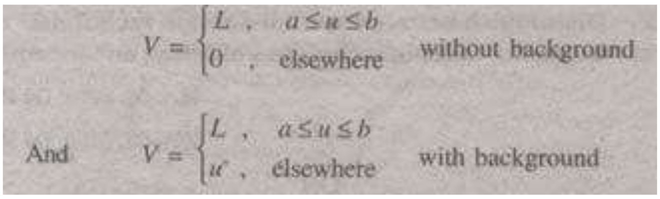


Intensity Level Slicing:

The images which consist of grey levels in between intensity at background and other objects require to reduce the intensity of the object. This process of changing intensity level is done with the help of intensity level slicing. They are expressed as

Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)

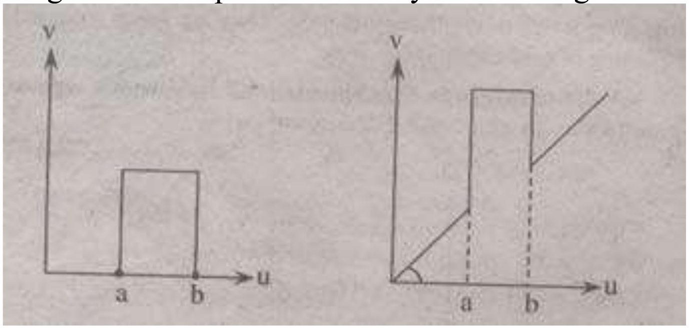
Lab 2: Image Negative Transformation, Thresholding, Gray Level Slicing with without background



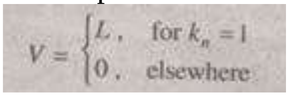
$$V = \begin{cases} L, & a \le u \le b \\ 0, & \text{elsewhere} \end{cases} \quad \text{without background}$$

$$\text{And} \quad V = \begin{cases} L, & a \le u \le b \\ u, & \text{elsewhere} \end{cases} \quad \text{with background}$$

The histogram of input image and its respective intensity level slicing is shown in the figure



When an image is uniformly quantized then, the nth most significant bit can be extracted and displayed. Let, u = k1 2^B-1 + k2 2^B-2 +……………..+ k^B-1 2 + kB Then, the output is expressed as



$$V = \begin{cases} L, & \text{for } k_n = 1 \\ 0, & \text{elsewhere} \end{cases}$$

Grey Level Slicing without background:

Grey level slicing is an image processing technique used to enhance the contrast of an image by selectively enhancing a range of pixel intensities. The basic idea behind grey level slicing is to setall pixel values within a specific intensity range to a maximum value, while leaving all other pixelvalues unchanged

.

Lab Assignments to complete in this session

Problem Statement: Develop a Python program utilizing the OpenCV library to manipulate images from the Fashion MNIST digits dataset. The program should address the following tasks:

1. Read random image(s) from the MNIST fashion dataset.
2. **Dataset Link:** Fashion MNIST Github
3. Display the before & after image(s) used in the task below.
4. Perform image negative transformation.
5. Perform image thresholding operation at various threshold level and write your observation.
6. Perform gray level slicing with and without background intensity slicing and write your observation.

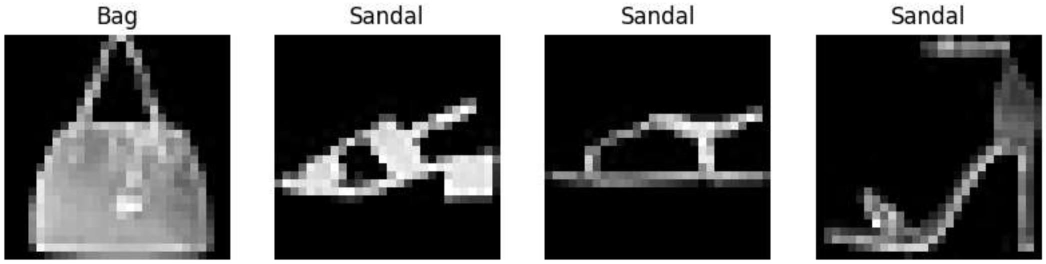
The solution to the operations performed must be produced by scratch coding without the use of built in OpenCV methods.

Shri Vile Parle 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 Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)
Lab 2: Image Negative Transformation, Thresholding, Gray Level Slicing with without background

```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import fashion_mnist

(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()

class_names = ["Sandal", "Sneaker", "Bag"]
labels_to_display = [5, 7, 8]
filtered_indices = [i for i, label in enumerate(y_train) if label in labels_to_display]
x_filtered = x_train[filtered_indices]
y_filtered = y_train[filtered_indices]

random_indices = np.random.choice(len(x_filtered), 4, replace=False)
selected_images = [(x_filtered[idx], class_names[labels_to_display.index(y_filtered[idx])]) for idx in random_indices]

plt.figure(figsize=(10, 4))
for i, (image, label) in enumerate(selected_images):
    plt.subplot(1, 4, i + 1)
    plt.imshow(image, cmap='gray')
    plt.title(label)
    plt.axis('off')

plt.show()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 ━━━━━━━━━━━━━━━━━━━━ 1s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 ━━━━━━━━━━━━━━━━━━━━ 0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```



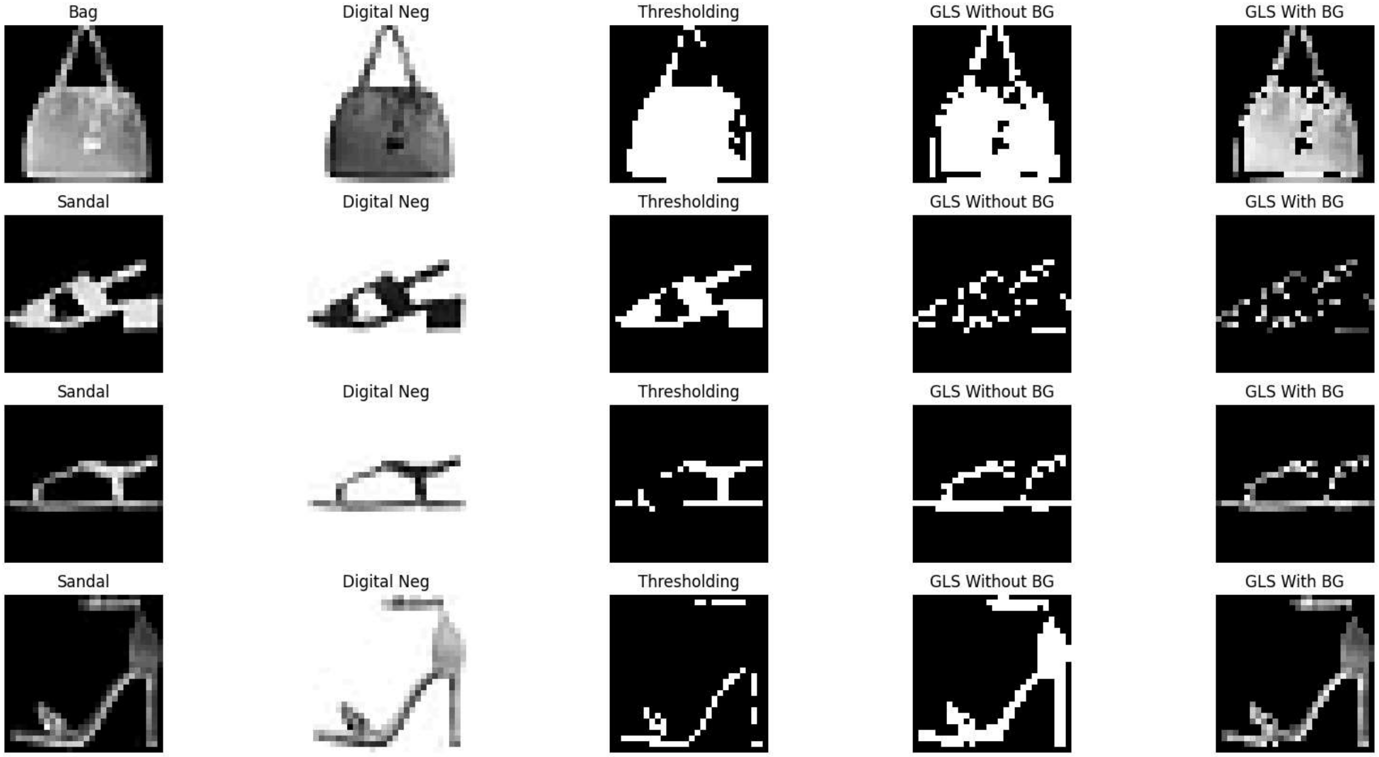
```
import numpy as np

def digital_neg(img):
    return 255 - img
```

```
import numpy as np

def thresholding(img, threshold=127):
    return np.where(img > threshold, 255, 0)
```

```
import numpy as np

def gls_without_bg(img, a, b):
    return np.where((img >= a) & (img <= b), 255, 0)
```

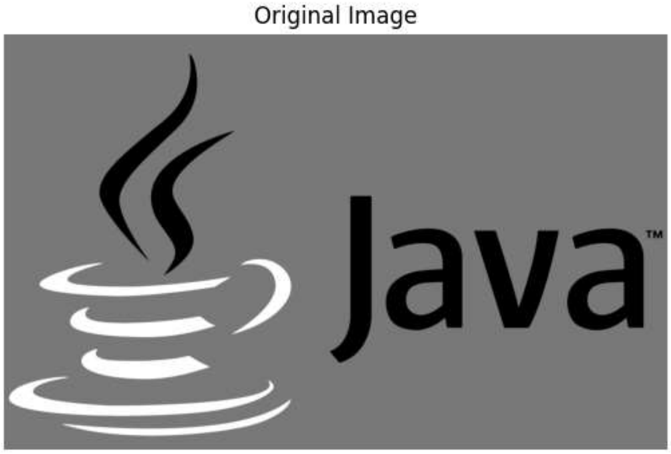
```
import numpy as np

def gls_with_bg(img, a, b):
    return np.where((img >= a) & (img <= b), img, 0)
```

```
plt.figure(figsize=(20, 10))

for i, (image, label) in enumerate(selected_images):
    # Original image
    plt.subplot(4, 5, i * 5 + 1)
    plt.imshow(image, cmap='gray')
    plt.title(label)
    plt.axis('off')

    # Digital negative
    plt.subplot(4, 5, i * 5 + 2)
    plt.imshow(digital_neg(image), cmap='gray')
    plt.title('Digital Neg')
    plt.axis('off')

    # Thresholding
    plt.subplot(4, 5, i * 5 + 3)
    plt.imshow(thresholding(image), cmap='gray')
    plt.title('Thresholding')
    plt.axis('off')

    # GLS without background
    plt.subplot(4, 5, i * 5 + 4)
    plt.imshow(gls_without_bg(image, 50, 200), cmap='gray')
    plt.title('GLS Without BG')
    plt.axis('off')

    # GLS with background
    plt.subplot(4, 5, i * 5 + 5)
    plt.imshow(gls_with_bg(image, 50, 200), cmap='gray')
    plt.title('GLS With BG')
    plt.axis('off')

plt.show()
```

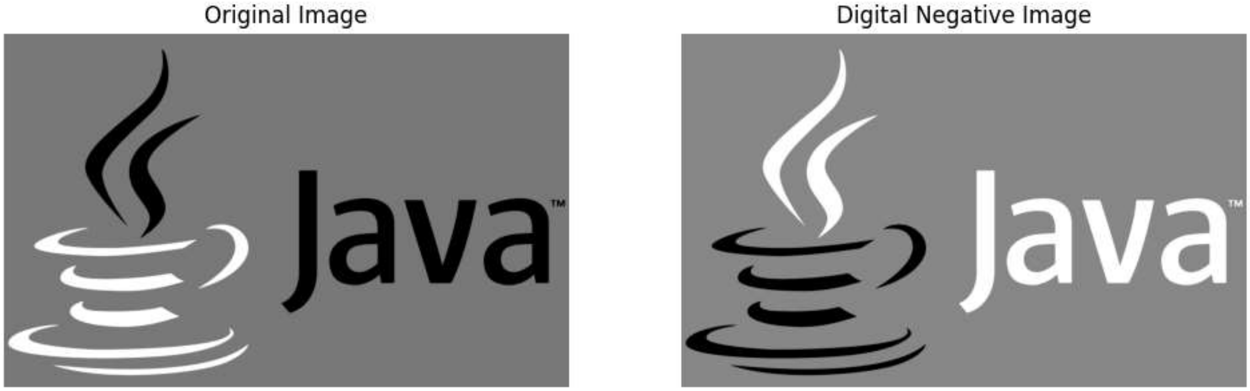


```
import cv2

img = cv2.imread('/content/1704733456854.png')
img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')
```

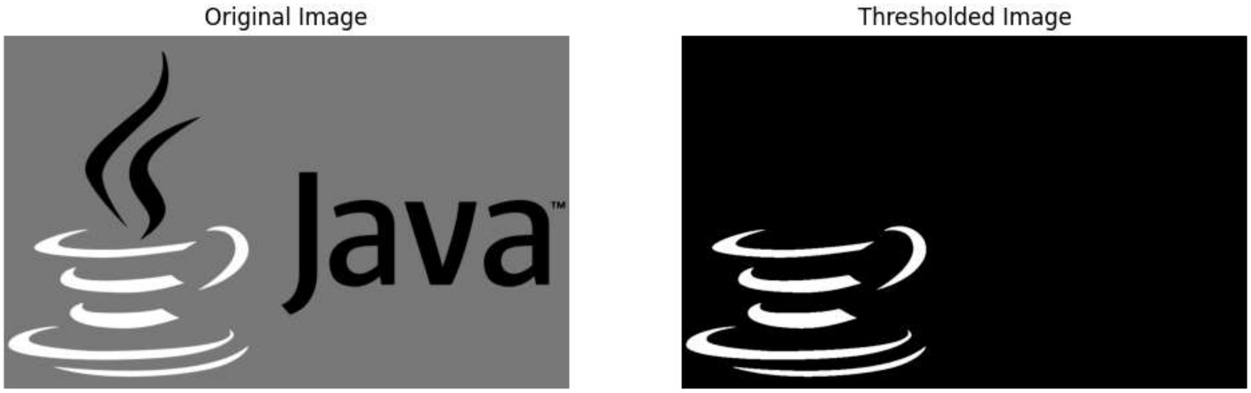
```
(-0.5, 1151.5, 719.5, -0.5)
```



```
import matplotlib.pyplot as plt

imgNeg = digital_neg(img)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('Digital Negative Image')
plt.imshow(imgNeg, cmap='gray')
plt.axis('off')

plt.show()
```

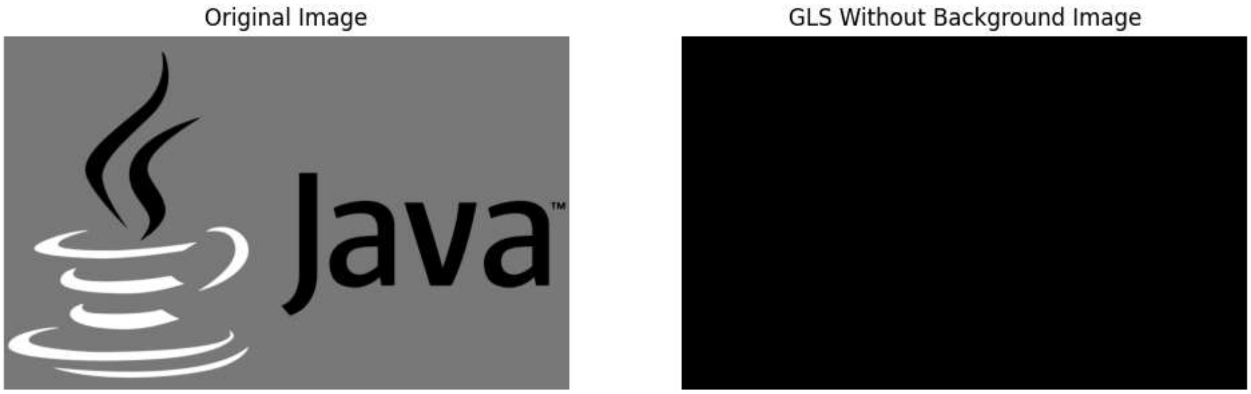


```
import matplotlib.pyplot as plt

imgThres = thresholding(img)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')
plt.subplot(1, 2, 2)
plt.title('Thresholded Image')
plt.imshow(imgThres, cmap='gray')
plt.axis('off')

plt.show()
```


Original Image

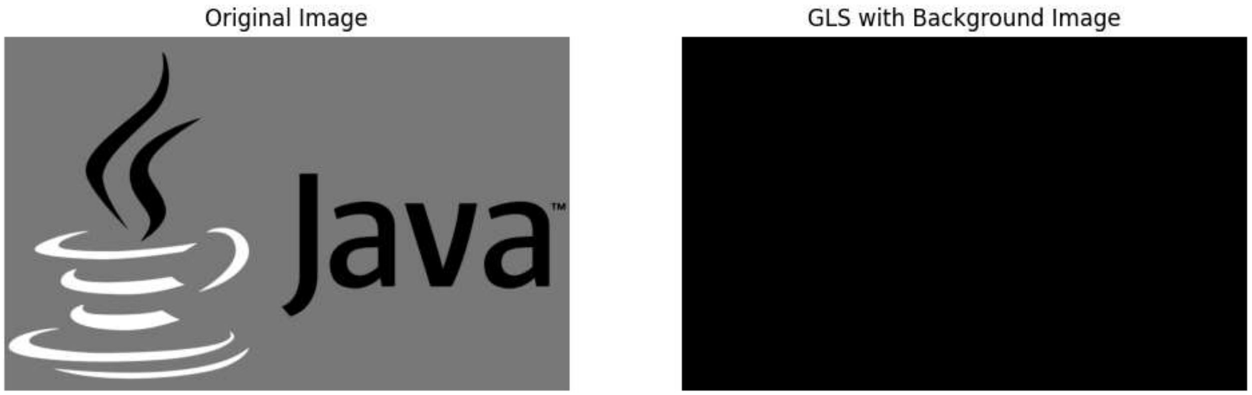
Thresholded Image

```
import matplotlib.pyplot as plt

imgGlsWithout = gls_without_bg(img, 200, 100)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('GLS Without Background Image')
plt.imshow(imgGlsWithout, cmap='gray')
plt.axis('off')

plt.show()
```


Original Image

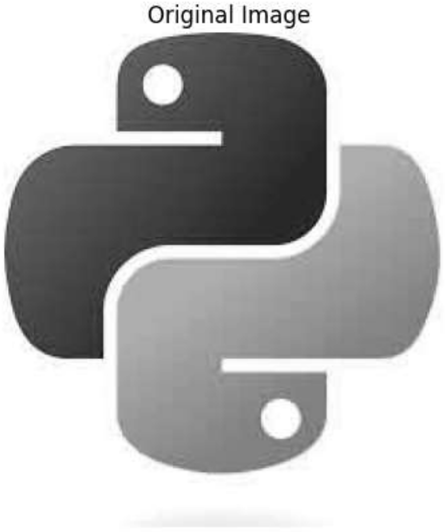
GLS Without Background Image

```
import matplotlib.pyplot as plt

imgWithGls = gls_with_bg(img, 200, 100)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('GLS with Background Image')
plt.imshow(imgWithGls, cmap='gray')
plt.axis('off')
plt.axis('off')

plt.show()
```

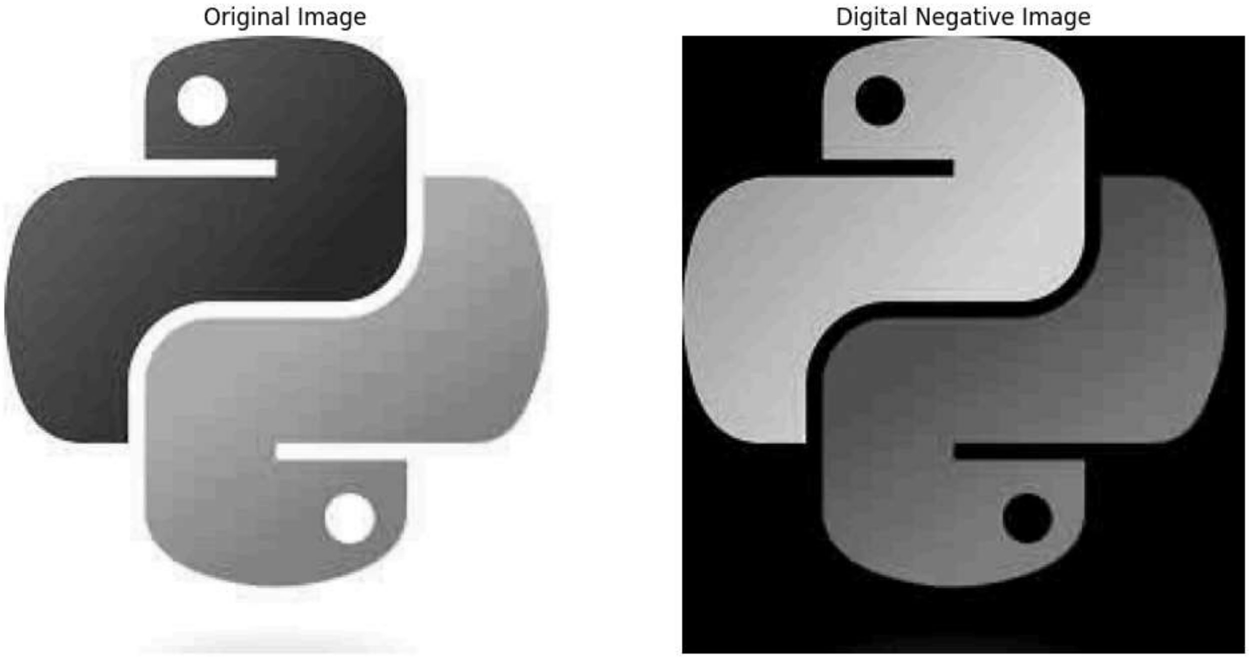


```
import cv2

img = cv2.imread('/content/python.jpeg')
img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')
```

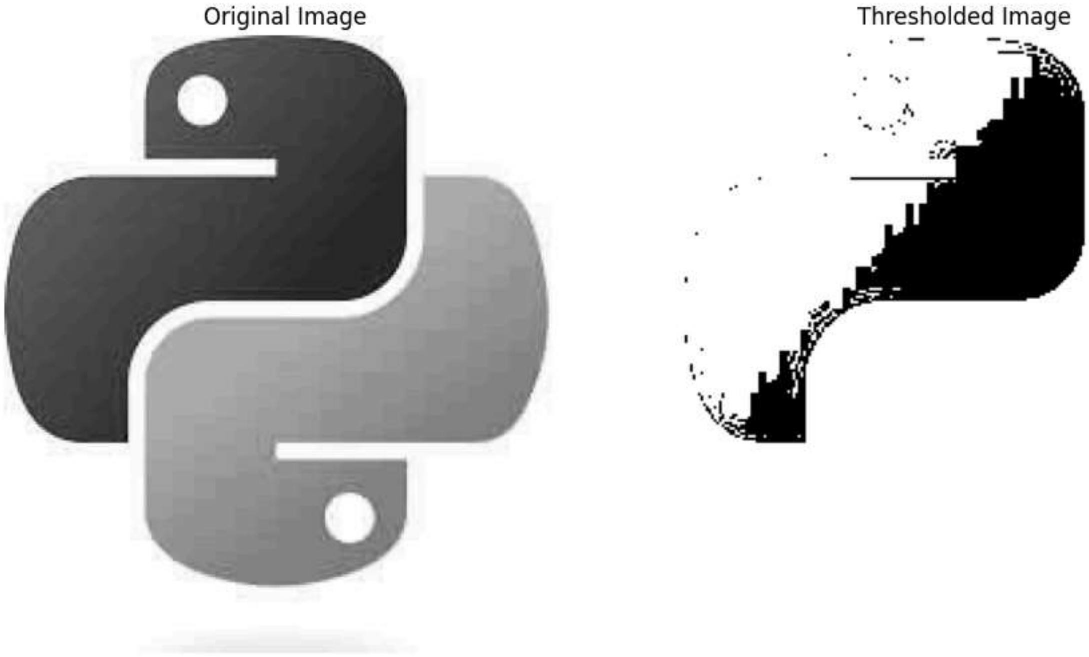
```
(-0.5, 213.5, 234.5, -0.5)
```



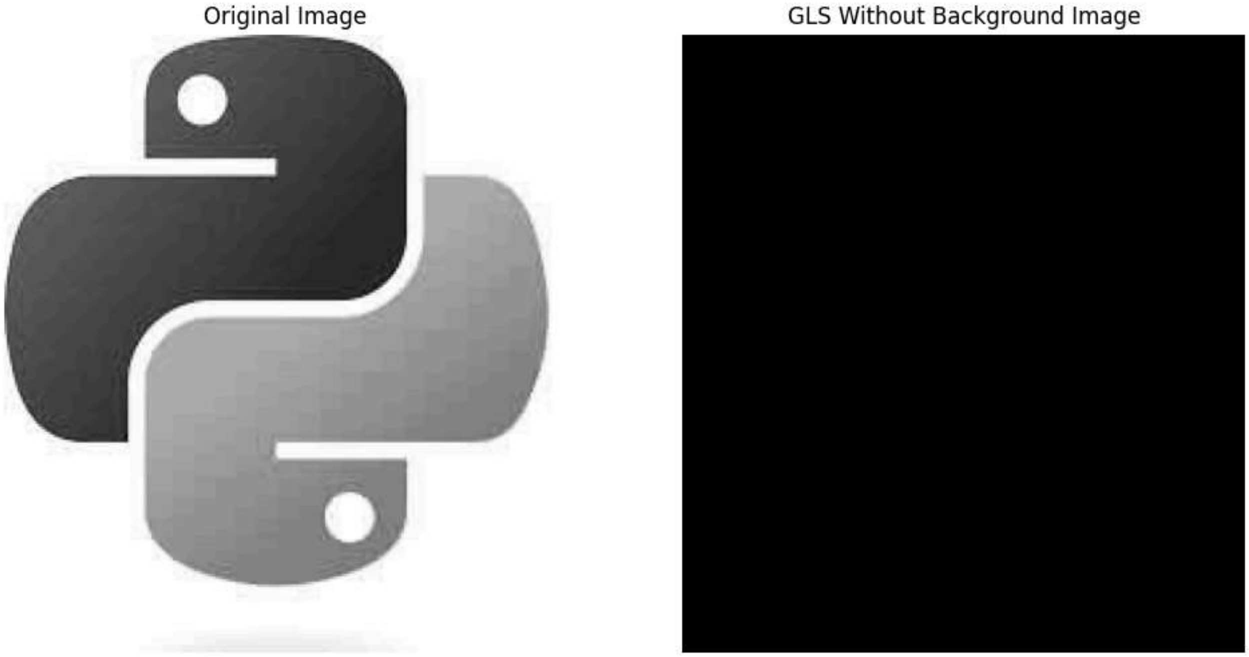
```
import matplotlib.pyplot as plt

imgNeg = digital_neg(img)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('Digital Negative Image')
plt.imshow(imgNeg, cmap='gray')
plt.axis('off')

plt.show()
```



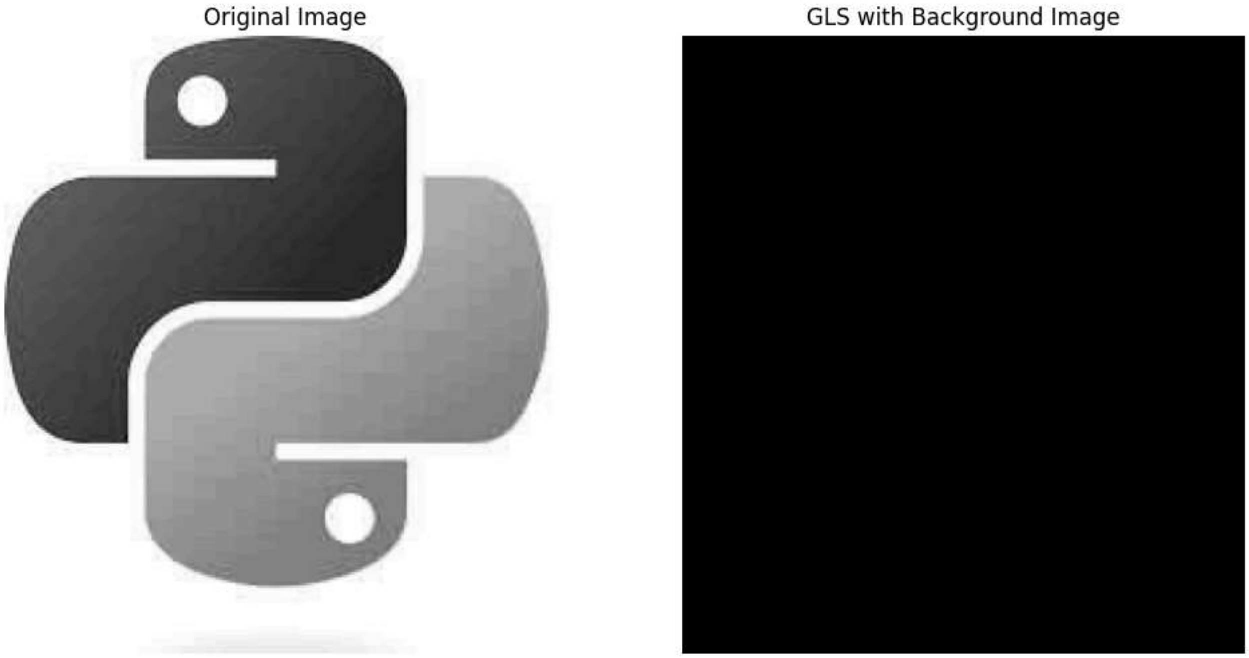
```
import matplotlib.pyplot as plt

imgThres = thresholding(img)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('Thresholded Image')
plt.imshow(imgThres, cmap='gray')
plt.axis('off')

plt.show()
```



```
import matplotlib.pyplot as plt

imgGlsWithout = gls_without_bg(img, 200, 100)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('GLS Without Background Image')
plt.imshow(imgGlsWithout, cmap='gray')
plt.axis('off')

plt.show()
```



```
import matplotlib.pyplot as plt

imgWithGls = gls_with_bg(img, 200, 100)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('GLS with Background Image')
plt.imshow(imgWithGls, cmap='gray')
plt.axis('off')

plt.show()
```



In this experiment, we explored several image processing techniques on the Fashion MNIST dataset using NumPy and OpenCV. By creating digital negatives, applying thresholding, and manipulating background elements, we were able to visualize and isolate different aspects of the fashion items. These transformations not only provided insights into the dataset but also demonstrated how preprocessing can enhance the