Introduction

An image is a matrix of pixel values, where each pixel represents a color or intensity. In digital image processing, images can be represented in various color spaces, such as RGB (Red, Green, Blue), BGR (Blue, Green, Red, used by OpenCV), or grayscale (single intensity channel). Image processing involves manipulating these pixel values to achieve tasks like enhancement, analysis, or transformation.

Key Operations:

1.Loading Images: OpenCV's cv2.imread() loads images in BGR format by default.

2.Displaying Images: Matplotlib's plt.imshow() is used to display images, but OpenCV's BGR format must be converted to RGB for correct color rendering.

3.Color Conversions: Images can be converted between color spaces (e.g., BGR to RGB, BGR to grayscale) using cv2.cvtColor().

4.Pixel Extraction: Pixel values can be accessed as a NumPy array, allowing inspection of intensity values (e.g., for grayscale images).

Install and Import Libraries

```
# !pip install opencv-python
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Load and Display Image

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

# Load the image
image_bgr = cv2.imread('/content/drive/MyDrive/Colab Notebooks/imageprocessingimages/everest
if image_bgr is None:
    raise Exception("Image not found. Please upload 'everest.jpg' to Colab.")

# Resize image for consistent processing
image_bgr = cv2.resize(image_bgr, (250, 250))

# Convert BGR to RGB
```

```
image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)

# Convert BGR to Grayscale
image_gray = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)

# Display images
plt.figure(figsize=(15, 5))

# Original BGR Image
plt.subplot(1, 3, 1)
plt.imshow(image_bgr[:, :, ::-1])  # Convert BGR to RGB for display
plt.title('BGR Image')
plt.axis('off')

# RGB Image
plt.subplot(1, 3, 2)
plt.imshow(image_rgb)
plt.title('RGB Image')
plt.axis('off')

# Grayscale Image
plt.subplot(1, 3, 3)
plt.imshow(image_gray, cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')

plt.show()

# Extract and print pixel values for grayscale image
print("Grayscale Image Pixel Values (first few rows):")
print(image_gray[:5, :10])  # Print first 5 rows and 10 columns
print("Shape of Grayscale Image:", image_gray.shape)
```

```
Grayscale Image Pixel Values (first few rows):
[[109 108 113 112 111 113 110 111 111 110]
 [109 109 111 111 112 111 111 110 112 111]
 [111 110 111 109 112 112 112 111 112 109]
 [113 112 110 112 110 112 112 112 111 112]
 [113 112 110 112 111 113 111 112 111 113]]
Shape of Grayscale Image: (250, 250)
```