Assignment for Chapter 01: Introduction to Image Processing

Objective:

To develop a fundamental understanding of image processing, its applications, and the mathematical representation of digital images.

Assignment Tasks

Part 1: Theory

1. Define **Image Processing** and explain its significance in various fields.
2. List and describe at least five real-world applications of image processing.
3. Differentiate between:
   - Grayscale and RGB images.
   - Raster and Vector images.
4. Explain the concept of a **digital image** with a mathematical representation. Include:
   - Grayscale image formula:
     $f(x, y)$
   - RGB image formula:
     $f(x, y) = [R(x, y), G(x, y), B(x, y)]$
5. Describe the following terms with examples:
   - **Pixel**
   - **Resolution**
   - **Bit Depth**

Part 2: Practical Tasks

1. **Read and Display an Image**
   - Use OpenCV to load an image of your choice.
   - Display the image in grayscale and RGB format using Matplotlib.
2. **Pixel Intensity Analysis**
   - Write a program to extract and print the intensity value of a specific pixel (e.g., at coordinates (50, 50)) for both grayscale and RGB images.
3. **Image Conversion**

- Convert an image from RGB to grayscale using:
  - OpenCV functions.
  - The formula:
    $I = 0.2989R + 0.5870G + 0.1140B$
- Compare and display both grayscale images.

4. **Image Histogram**
   - Generate and plot the histogram of a grayscale image using Matplotlib.
   - Explain the significance of the histogram in analyzing image properties.
5. **Image Dimensions and Data Type**
   - Write a program to display the following properties of an image:
     - Width and Height (Resolution).
     - Number of Channels.
     - Data Type (e.g., `uint8`).

Sample Solutions

Part 1: Theory Solutions

1. **Define Image Processing:**
   Image processing involves techniques to manipulate or analyze images for various applications like enhancing image quality, object detection, and recognition. It is significant in fields like medical imaging, autonomous vehicles, and digital photography.
2. **Real-World Applications:**
   - **Medical Imaging:** MRI, CT scans.
   - **Surveillance:** Real-time object detection.
   - **Satellite Imaging:** Weather forecasting.
   - **Augmented Reality:** Overlaying virtual elements.
   - **Digital Forensics:** Enhancing evidence images.
3. **Differences:**
   - **Grayscale vs RGB:** Grayscale contains intensity information, while RGB represents color.
   - **Raster vs Vector:** Raster uses pixels, Vector uses mathematical formulas for scalability.
4. **Digital Image Representation:**

- Grayscale:
  $f(x, y)$
- RGB:
  $f(x, y) = [R(x, y), G(x, y), B(x, y)]$

5. **Key Terms:**
   - **Pixel:** Smallest unit of an image.
   - **Resolution:** Number of pixels in an image.
   - **Bit Depth:** Number of bits per pixel.

Part 2: Practical Solutions

1. **Read and Display an Image**

```
import cv2
import matplotlib.pyplot as plt

image_path = "/path/to/image.jpg"
image = cv2.imread(image_path)

# Display RGB image
rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.figure(figsize=(8, 6))
plt.imshow(rgb_image)
plt.title("RGB Image")
plt.axis("off")
plt.show()

# Display Grayscale image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.figure(figsize=(8, 6))
plt.imshow(gray_image, cmap="gray")
plt.title("Grayscale Image")
plt.axis("off")
plt.show()
```

2. **Pixel Intensity Analysis**

```
x, y = 50, 50
print(f"Grayscale Intensity at ({x},{y}): {gray_image[y, x]}")
print(f"RGB Intensity at ({x},{y}): {rgb_image[y, x]}")
```

3. **Image Conversion**

```
# OpenCV grayscale conversion
gray_image_opencv = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Manual grayscale conversion
gray_manual = 0.2989 * rgb_image[:, :, 0] + 0.5870 *
rgb_image[:, :, 1] + 0.1140 * rgb_image[:, :, 2]

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.imshow(gray_image_opencv, cmap="gray")
plt.title("Grayscale (OpenCV)")
plt.axis("off")

plt.subplot(1, 2, 2)
plt.imshow(gray_manual, cmap="gray")
plt.title("Grayscale (Manual)")
plt.axis("off")
plt.show()
```

4. **Image Histogram**

```
plt.hist(gray_image.ravel(), bins=256, range=[0, 256],
color="blue", alpha=0.7)
plt.title("Image Histogram")
plt.xlabel("Pixel Intensity")
plt.ylabel("Frequency")
plt.grid(True)
plt.show()
```

5. **Image Dimensions and Data Type**

```
print(f"Resolution: {image.shape[1]}x{image.shape[0]}")
print(f"Number of Channels: {image.shape[2]}")
print(f"Data Type: {image.dtype}")
```

Submission Instructions

1. Submit your Python code as a `.py` or Jupyter notebook (`.ipynb`).
2. Include a PDF report with:
   - Answers to theory questions.
   - Screenshots of outputs from practical tasks.
   - Explanations and observations.
3. Zip all files into a single folder named `Chapter01_Assignment_YourName.zip`.