### **Assignment Number 01**

### **Name: Mihir Unmesh Patil**

**Roll No: TYCOC213**

**Batch: C/C-3**

### 

### **Title:**

**Image Processing Techniques and their Applications using OpenCV**

### **Aim:**

The aim of this project is to demonstrate various image processing techniques such as image negation, thresholding, inverse thresholding, brightening, darkening, and bit-plane slicing. These techniques are implemented using Python and OpenCV library to enhance, manipulate, and analyze image data.

### **Objectives:**

* **Image Negation:** To create an inverted color version of the input image.
* **Thresholding:** To convert grayscale images to binary images using a specified threshold value.
* **Inverse Thresholding:** To invert the binary thresholding operation and produce the opposite binary image.
* **Brightening and Darkening:** To adjust the brightness of the image, making it lighter or darker based on user input.
* **Bit-Plane Slicing:** To extract specific bit planes of the image and visualize their contribution to the overall image.

These operations help in understanding the underlying structure of digital images and how image pixels can be manipulated for various applications.

### 

### **Theory:**

#### **1. Image Negation:**

Image negation refers to inverting the colors of an image. Each pixel's color value is subtracted from 255 (for an 8-bit image), producing an image where light areas become dark and dark areas become light. This technique is often used for aesthetic or analytical purposes, such as enhancing certain features in an image.

#### **2. Thresholding:**

Thresholding is a method of segmentation that converts a grayscale image into a binary image. Every pixel in the image is compared against a threshold value, and if the pixel's value is greater than the threshold, it becomes white (255); otherwise, it becomes black (0). This is commonly used in image analysis to distinguish objects from the background.

#### **3. Inverse Thresholding:**

Inverse thresholding is the opposite of normal thresholding. Pixels greater than the threshold value are set to black (0), and those less than or equal to the threshold are set to white (255). This technique is useful in certain image processing applications where the background needs to be dark and the objects need to be highlighted in white.

#### **4. Brightening and Darkening:**

Brightening and darkening operations involve adjusting the pixel values of an image by adding or subtracting a constant value. Brightening increases the intensity of the image, while darkening reduces it. These techniques are useful in enhancing images taken under different lighting conditions or for creative visual effects.

#### **5. Bit-Plane Slicing:**

Bit-plane slicing is a technique where each bit of the pixel values in an image is treated as a separate image. Each bit-plane corresponds to a particular level of precision in the image. For instance, the least significant bit (LSB) controls the finer details of the image, while the most significant bit (MSB) contains the more prominent parts. This technique can be used to analyze the importance of different bits in representing an image.

### 

### **Methodology:**

The implementation of these image processing operations was done using Python and the OpenCV library. The following steps outline the process:

1. **Loading the Image:** The input image is read in grayscale mode using OpenCV's cv2.imread() function.
2. **User Input for Operation Choice:** The user is prompted to choose one of the image processing operations: negation, thresholding, inverse thresholding, brightening, darkening, or bit-plane slicing.
3. **Image Processing:** Based on the user's choice, the appropriate function is called to perform the selected operation on the image. Some operations (e.g., brightening, darkening, and bit-plane slicing) require additional parameters like intensity values or bit-plane numbers.
4. **Displaying Results:** The original and processed images are displayed using cv2\_imshow(), a function provided by Google Colab for visualizing images. The matrix values of the original and processed images are also printed to the console for debugging or analysis purposes.

### **Google Colab Link for the Code:**

### [Assignment1.ipynb](https://colab.research.google.com/drive/1lN27WTnfX4HNVr_growCN_a8-1skoK0w?usp=sharing)

### **Conclusion:**

This project demonstrates the practical application of basic image processing techniques using OpenCV. Through operations like image negation, thresholding, and bit-plane slicing, users can manipulate and analyze images in different ways. These techniques are fundamental in fields such as computer vision, medical imaging, and digital forensics. By adjusting various parameters and understanding the effects of these operations, we can gain deeper insights into the structure of digital images and their potential uses in real-world applications.