Practical No.3

Aim :

Read any image. Display the histogram, Equalized histogram, and image with equalized histogram

Objective:

To learn to plot image in the form histogram and apply histogram equalization**.**

Theory:

What we can explore from Histogram? Why Histogram equalization.

We can deduce a great deal about the appearance of the image from its histogram –
i) In a dark image, the gray levels would be clustered at lower ends

ii) In an uniformly bright image, the gray levels would be clustered at the upper end

iii) In a well contrasted image, the gray levels would be well spred out over much of the image.

Applications:

For image enhancement

a) Displaying Histogram of input image:

1.Read and load the input RGB image.

2.Convert the image to grayscale using the cv2.cvtColor() function.

3.Obtain the histogram of the grayscale image using the cv2.calcHist() function. Set the following parameters:

Input image

Channels: [0] (since the image is grayscale).

Mask: None (since we require the histogram of the entire image and not a specific part).

histSize: [256] (represents the bin count; 256 in case of grayscale image).

range: [0, 256] (the range of values that a pixel can have).

4. Plot the histogram using the matplotlib library.

Histogram :

The histogram of an image is a plot of the number of occurrences of gray levels in the image against gray level values. Histograms are the basis for numerous spatial domain processing techniques. It plays an important role in enhancement of perceived brightness and contrast of an image. It specifies the number of pixels having each gray level, but gives no hint as to where those pixels are located within the image. The histogram of an N × M image is defined as the percentage of pixels within the image at a given gray level

$$h_i = \frac{n_i}{MN} \quad \text{for } 0 \le i \le G_{max}$$

where n_i is the number of pixels at gray level i,
NM is the total number of pixels within the image and
G_{max} is the maximum gray level value of the image.

Histogram Equalization :

The histogram technique that is used to enhance the brightness and contrast of an image is histogram equalization. The goal of histogram equalization is to distribute the gray levels withinan image so that every gray level is equally likely to occur. In other words, histogram equalization takes an image's histogram and produces a new image with a histogram that is uniformly distributed. Histogram equalization will increase the brightness and contrast of a dark and low contrast image, making features observable that were not visible in the original image. Since histogram equalization distributes an image's gray levels uniformly about the range of gray levels, all images will have approximately the same brightness and contrast, hence allowing images to be compared equally without a bias due to perceived contrast and brightness differences.

Procedure:

1. Read and load the grayscale image generated in the previous step.

2. Obtain the equalized histogram of the grayscale image using the cv2.equalizeHist() function.

3. Obtain the corresponding histogram which can be displayed for the equalized histogram using the cv2.calcHist() function. Set the parameters as set above.

4. Plot the equalized histogram using the matplotlib library

5. Display the image corresponding to the equalized histogram using cv2 library.

Conclusion:

We have successfully read the image and display the histogram, Equalized histogram,and image with equalized histogram