

# Experiment 3: Image Enhancement using Histogram Equalization

## Aim

To enhance the contrast of a grayscale image using histogram equalization.

## Software Required

- Python
- OpenCV
- NumPy
- Matplotlib

## Theory

Histogram equalization is an image enhancement technique that improves image contrast by redistributing gray level intensities over the entire available range. It uses the cumulative distribution function (CDF) of the image histogram to map old intensity values to new ones.

The transformation function is given by:

$$s = (L - 1) \sum_{j=0}^r p_r(j)$$

where  $L$  is the number of gray levels and  $p_r(j)$  is the probability of occurrence of gray level  $j$ .

## Algorithm

1. Read the input grayscale image.
2. Compute the histogram of the image.
3. Normalize the histogram.
4. Compute the cumulative distribution function (CDF).

5. Map the old gray levels to new gray levels using the CDF.
6. Replace the original pixel values with the new values.
7. Display the enhanced image.

## Pseudo Code (Python Style)

1. Read the grayscale image into a variable `img`
2. Initialize an array `hist` with zeros of size 256
3. **for** each pixel value `p` in `img`:
  - Increment `hist[p]` by 1
4. Normalize `hist` by dividing each value by total number of pixels
5. Compute cumulative sum of `hist` to obtain `cdf`
6. **for** each gray level `i` from 0 to 255:
  - `new_value[i] = round(255 * cdf[i])`
7. **for** each pixel `p` in `img`:
  - Replace `p` with `new_value[p]`
8. Display the histogram equalized image

## Example

Consider an image with pixel values concentrated in a narrow gray level range. Histogram equalization redistributes these values across the full intensity range (0–255), resulting in improved contrast.

## Result

Thus, the image was successfully enhanced using histogram equalization, producing better contrast and visual quality.