Image Processing Assignment 2

Write a program for Histogram Display and Equalization

Aditya Chavan
Roll No: 231080019
T.Y B.Tech I.T
Date: September 27, 2025

1 Aim

To write a Python program to display the histogram of an image and perform histogram equalization using the OpenCV library.

2 Theory

Histogram Equalization is a technique in image processing used to enhance the contrast of images. It spreads out the most frequent intensity values to improve the global contrast, especially in images that are dark or light.

2.1 Key Concepts

1. **Histogram:** A histogram is a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each intensity value.
2. **Histogram Equalization:** The process of adjusting image intensities to enhance contrast. It aims to achieve a uniform histogram for better visual quality.
3. **Grayscale Equalization:** It modifies the intensity distribution of grayscale images.
4. **Color Image Equalization:** Typically applied to the Value (V) channel in HSV or Lightness (L) channel in LAB color space to avoid color distortion.
5. **Cumulative Distribution Function (CDF):** Used to map the old pixel values to new ones to redistribute intensity values more uniformly.

2.2 Advantages

- Enhances contrast in low-light or low-contrast images
- Highlights hidden details
- Simple and computationally efficient

2.3 Applications

- Medical Imaging (e.g., X-rays)
- Satellite and Aerial Image Analysis
- Machine Vision and Surveillance

3 Algorithm

1. Read the input image (grayscale or color)
2. If grayscale:
   - Use `cv2.equalizeHist()` directly
3. If colored:
   - Convert BGR to HSV
   - Equalize only the Value channel
   - Merge channels and convert back to BGR
4. Display and compare original and equalized images and their histograms

4 Implementation

The implementation was done in Google Colab using Python and OpenCV.

4.1 Required Packages

Install the required packages using the following command:

```
pip install opencv-python matplotlib scikit-image numpy
```

4.2 Python Code

Google Colab Notebook

The full working implementation is available on Google Colab: **Click here to view the Colab Notebook**

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
from google.colab.patches import cv2_imshow
```

```
path = "bki.jpeg"
img = cv.imread(path)
cv2_imshow(img)
cv.waitKey(0)
cv.destroyAllWindows()
```

Figure 1: Original Image

```
hist,bins = np.histogram(img.flatten(),256,[0,256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()
plt.plot(cdf_normalized, color = 'b')
plt.hist(img.flatten(),256,[0,256], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.show()
```

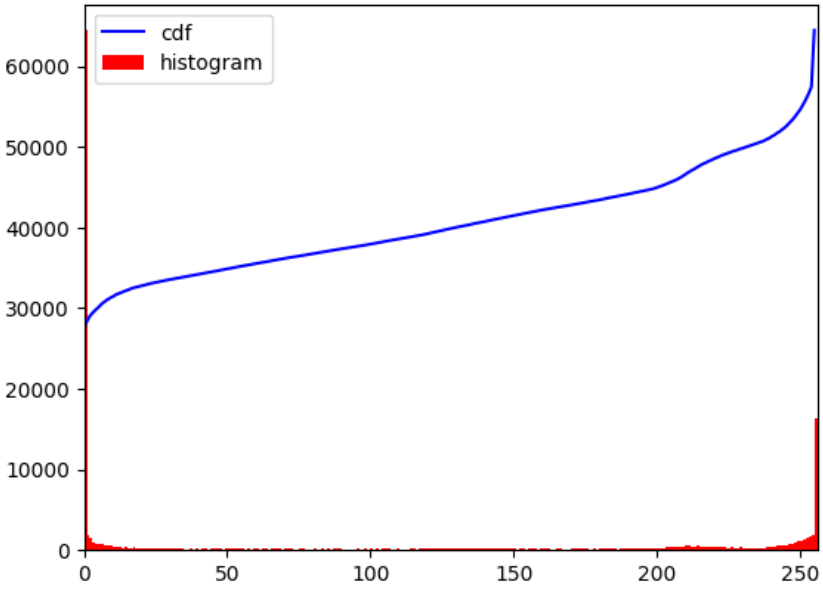

Figure 2: Histogram and CDF of Original Image

```
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
equ = cv.equalizeHist(gray)
cv2_imshow(equ)
```

Figure 3: Histogram Equalized Output Image

5 Conclusion

- Histogram Equalization is an effective method for image contrast enhancement that uniformly distributes intensity values across the full dynamic range
- For grayscale images, the technique directly applies to the single intensity channel, improving overall visibility and detail representation
- For color images, equalizing only the Value (V) channel in HSV color space prevents color distortion while maintaining natural color appearance
- The implementation utilizes OpenCV functions such as `equalizeHist()`, `cvtColor()`, and `split()/merge()` for efficient image transformations
- Visual comparison of original and equalized images along with their histograms clearly demonstrates the contrast improvement achieved through this technique
- This method serves as a fundamental preprocessing step in computer vision applications, medical imaging, and general image enhancement tasks due to its simplicity and computational efficiency