**Lab Report: Advanced Computer Vision - Lab 4**

**Student Name:** Anmol Pandey  
**Student ID:** E22CSEU1069  
**Course:** Advanced Computer Vision  
**Lab Number:** 4

## ****1. Objective****

The objective of this lab was to explore histogram analysis techniques in image processing. The tasks focused on computing, visualizing, and manipulating histograms to enhance image contrast using histogram equalization and CLAHE (Contrast Limited Adaptive Histogram Equalization).

## ****2. Introduction****

Histograms are a fundamental tool in image processing, representing the distribution of pixel intensity values. This lab covers:

* **Histogram Computation for Grayscale and RGB Images**
* **Histogram Visualization and Probability Histograms**
* **Histogram Equalization for Global Contrast Enhancement**
* **CLAHE for Local Contrast Enhancement**

## ****3. Implementation Details****

### ****3.1 Libraries Used****

The following Python libraries were used:

* cv2 (OpenCV): For image processing and histogram equalization.
* numpy: For numerical computations.
* matplotlib.pyplot: For plotting histograms.

### ****3.2 Histogram Computation****

import cv2

import numpy as np

import matplotlib.pyplot as plt

from google.colab.patches import cv2\_imshow

def compute\_histogram(image, is\_gray=False):

if is\_gray:

hist = cv2.calcHist([image], [0], None, [256], [0, 256])

else:

hist = [cv2.calcHist([image], [i], None, [256], [0, 256]) for i in range(3)]

return hist

**Explanation:**

* Uses cv2.calcHist() to compute histograms.
* Supports both grayscale and RGB images.

### ****3.3 Histogram Plotting****

def plot\_histogram(hist, title, color=None):

plt.figure()

plt.title(title)

plt.xlabel('Intensity Value')

plt.ylabel('Pixel Count')

if color:

for h, c in zip(hist, color):

plt.plot(h, color=c)

else:

plt.plot(hist, color='black')

plt.xlim([0, 256])

plt.show()

**Explanation:**

* Plots grayscale histograms in black.
* Plots RGB histograms with corresponding color channels.

### ****3.4 Probability Histogram****

def plot\_probability\_histogram(hist, title, color=None):

plt.figure()

plt.title(title)

plt.xlabel('Intensity Value')

plt.ylabel('Probability')

if color:

for h, c in zip(hist, color):

plt.plot(h / np.sum(h), color=c)

else:

plt.plot(hist / np.sum(hist), color='black')

plt.xlim([0, 256])

plt.show()

**Explanation:**

* Normalizes histograms to display probability distributions.

### ****3.5 Histogram Equalization****

def apply\_histogram\_equalization(image):

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

equalized = cv2.equalizeHist(gray)

return equalized

**Explanation:**

* Converts the image to grayscale.
* Applies histogram equalization to enhance contrast.

### ****3.6 CLAHE (Adaptive Histogram Equalization)****

def apply\_clahe(image):

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))

enhanced = clahe.apply(gray)

return enhanced

**Explanation:**

* Uses CLAHE for local contrast enhancement while limiting noise amplification.

## ****4. Results and Observations****

The implemented code produces the following outputs:

1. **Histograms of Grayscale and RGB Images**: Displays intensity distributions.
2. **Probability Histogram**: Normalized distribution for analysis.
3. **Histogram Equalization Output**: Enhances global contrast.
4. **CLAHE Output**: Locally enhances contrast while preserving details.

**Observations:**

* Histogram equalization improves contrast but may over-enhance some regions.
* CLAHE provides better local enhancements, avoiding excessive noise amplification.

## ****5. Conclusion****

This lab introduced essential histogram-based techniques for image enhancement. These methods are widely used in medical imaging, object detection, and feature extraction.

## ****6. Future Scope****

* Experiment with different CLAHE parameters to optimize enhancements.
* Apply histogram equalization to color images using the LAB color space.
* Compare CLAHE with other contrast enhancement techniques.
* Use histograms for image segmentation and object recognition.

**End of Report**