Digital Image Processing — Spring 2025

Lab 5: Histogram Processing — Report

Summary

In this lab, I performed histogram-based image processing using Python libraries including OpenCV, scikit-image, NumPy, and Matplotlib.

In Part 1, I applied histogram equalization to a grayscale image using cv2.equalizeHist(), which redistributed pixel intensity values across the full 0–255 range and significantly improved the image contrast. Comparing the histograms before and after showed that the concentrated intensity values were spread out, making previously dark or flat regions more visible.

In Part 2, I extended this to a color image by splitting it into its B, G, and R channels, equalizing each independently, and merging them back, which boosted overall brightness but introduced slight color shifts due to independent channel processing.

In Part 3, I used skimage’s exposure.match_histograms() to perform histogram matching, where the intensity distribution of a source image was transformed to resemble that of a brighter target image, effectively changing the visual tone while preserving the original scene structure.

In Part 4, I combined both techniques by first equalizing the source image and then applying histogram matching on the result, which produced a more refined contrast adjustment by flattening the distribution first and then reshaping it to match a desired reference.

Overall, this lab demonstrated how histogram manipulation is a powerful and flexible tool for controlling image brightness and contrast in both grayscale and color contexts.