DIGITAL IMAGE AND VIDEO PROCESSING LAB - EXPERIMENT NO. 6

HISTOGRAM EQUALIZATION AND MATCHING

Submitted by Unnati Singh (21EC39027)

Overview:

This experiment involves implementing and analysing image processing techniques, specifically histogram equalization and histogram matching. Histogram equalization enhances image contrast by redistributing pixel intensity values, making details more visible, while histogram matching adjusts the histogram of an image to resemble that of a reference image, ensuring similar intensity distributions.

Files Included:

1. `Exp_6_21EC39027.py`: Python script containing the modular functions for histogram equalisation and matching.
2. `Exp_6_21EC39027.ipynb`: Jupyter Notebook for interactive execution and visualization of the filtering techniques.
3. `Input_images/`: Directory containing the input images.
4. `output/`: Directory containing the output images.
5. `dip_exp6.pdf`: PDF containing the problem statement.
6. `README.pdf`: This README file.

Functions Overview:

1. `displayim(i1, i2, i3=None, t1="Original Image", t2="Equalised Image", t3=None):`
   - **Purpose:** Display up to three images side by side with optional titles for each image.
   - **Working:** The function arranges and resizes images with titles, handles optional third image and titles, and displays them using OpenCV.
   - **Output:** A side-by-side display of the provided images with specified titles.

2. `get_hist_table(image):`
   - **Purpose:** Generate a histogram table for a grayscale or RGB image, including pixel values, counts, probabilities, and CDF.
   - **Working:** For grayscale images, it computes pixel frequencies, probabilities, and cumulative distribution. For RGB images, it does this separately for each channel.

- **Output:** A DataFrame containing pixel value, count, probability, and CDF for each channel or the grayscale image.

3. `plot_histogram(img):`
   - **Purpose:** Plot the histogram of a grayscale or multi-channel image.
   - **Working:** For grayscale images, it plots a single histogram of pixel values. For RGB or multi-channel images, it plots histograms for each channel separately.
   - **Output:** Displays one or more histograms, showing the distribution of pixel intensities for the image's channels.

4. `histeq(image):`
   - **Purpose:** Perform histogram equalization on a grayscale or RGB image.
   - **Working:** It adjusts pixel intensities based on the cumulative distribution function (CDF) to improve contrast, applying the transformation to HSV space in the case of coloured images.
   - **Output:** A new image with equalized histogram, enhancing its contrast, returned as an 8-bit image.

5. `histogram_matching(source_image, target_image):`
   - **Purpose:** Match the histogram of a source image to that of a target image to make their intensity distributions similar.
   - **Working:** It computes the cumulative distribution functions (CDF) of both images and maps pixel values from the target to the source image's histogram for each channel.
   - **Output:** An image with pixel intensity distribution adjusted to match the target image, returned as an 8-bit image.

6. `histogram_matching_hsv(source_image, target_image):`
   - **Purpose:** Match the histogram of a coloured source image to that of a coloured target image to make their intensity distributions similar.
   - **Working:** It computes the cumulative distribution functions (CDF) of both images and maps pixel values from the target to the source image's histogram in HSV space.
   - **Output:** An image with pixel intensity distribution adjusted to match the target image, returned as an 8-bit image.