CSE438: Image Processing Lab Manual

Lab-02: Advanced Image Processing Operations

Objective:

To implement advanced image processing techniques such as contrast stretching, bit plane slicing, logarithmic and power-law transformations, histogram matching, and histogram equalization**.**

1. Perform Contrast Stretching on an Image

Theory:
Contrast stretching enhances the dynamic range of an image by stretching the intensity values to a desired range. This is particularly useful for improving visibility in low-contrast images.

Implementation Steps:

1. Convert the input image to grayscale (if necessary).
2. Identify the minimum and maximum intensity values in the image.
3. Apply contrast stretching using a linear transformation.
4. Display the original and contrast-stretched images.

2. Apply Bit Plane Slicing on an Image

Theory:
Bit plane slicing separates an image into its individual bit planes, helping to analyze the contribution of different bits to the image. Higher-order bits contribute more to the image details-while lower-order bits capture finer variations.

Implementation Steps:

1. Convert the image to grayscale.
2. Extract different bit planes (from MSB to LSB).
3. Display the bit planes separately.

3. Change the Contrast of an Image using Logarithmic and Power-law Transformations

Theory:

- Logarithmic Transformation: Enhances low-intensity values more than high-intensity values, making dark areas more visible.

- Power-law (Gamma) Transformation: Controls the brightness and contrast of an image using a nonlinear transformation, where different gamma values produce different effects.

Implementation Steps:

1. Convert the image to grayscale.
2. Apply logarithmic transformation.
3. Apply the power-law transformation with different gamma values.
4. Display the transformed images.

4. Perform Histogram Matching (Specification)

Theory:
Histogram matching transforms the histogram of an input image to match a reference image, ensuring similar brightness and contrast characteristics.

Implementation Steps:

1. Load the input and reference images.
2. Compute the histograms of both images.
3. Apply histogram matching to adjust the input image.
4. Display the original, reference, and transformed images along with their histograms.

5. Change the Contrast of an Image using Histogram Equalization

Theory:
Histogram equalization improves the contrast of an image by redistributing the intensity values to achieve a more uniform histogram. This technique is effective for images with poor contrast.

Implementation Steps:

1. Convert the image to grayscale.
2. Compute the histogram of the original image.
3. Apply histogram equalization to enhance contrast.
4. Display the original and equalized images along with their histograms.

Functions Used:

1. **Imadjust() – Performs contrast stretching.**
2. **Bitget()– Extracts individual bit planes.**

3. **log and power functions() – Apply logarithmic and power-law transformations.**
4. **imhistmatch ()– Performs histogram matching.**
5. **histeq ()– Applies histogram equalization.**