Image Processing Report

Objective

This report aims to present the observations and comparisons obtained after applying various image processing techniques, organized into three sections: **Intensity Transformations (T1)**, **Histogram Processing (T2)**, and **Spatial Filtering (T3)**. For each task, transformations, filters, and convolution operations were applied to specific images using Python as the implementation language. The results obtained in each exercise can be viewed directly in Google Colab or by running the code locally, as the output remains consistent in both cases. The Google Colab link is:
https://colab.research.google.com/drive/1H1ZNKP5-u5aDWLJQjRWPBpOYRoVHiGrO?usp=sharing.

Section T1: Intensity Transformations

T1.1 - Negative Image

In this task, a negative transformation was applied to the image, where each pixel intensity value was inverted using the formula.

- **Observations**: The negative image transforms bright areas into dark ones and vice versa, highlighting contrast areas inversely. This technique is useful for visual analysis of hidden details in brighter regions of the original image.

T1.2 - Logarithmic Transformation

A logarithmic transformation was applied to expand low-intensity values and compress high-intensity values. This is especially useful for enhancing contrast in dark images.

- **Observations**: The logarithmic transformation brightens the darker areas, making previously hidden details more prominent. However, very bright areas tend to lose contrast.

T1.3 - Gamma Correction (T1.3a and T1.3b)

Gamma corrections with different values (γ) were applied to adjust pixel intensity non-linearly.

- **Observations T1.3a (γ < 1)**: The resulting image is darker as lower intensity values are compressed.
- **Observations T1.3b (γ > 1)**: The image brightens considerably, highlighting details in darker areas.

T1.4 - Contrast Stretching

Contrast stretching expanded the dynamic range of the image, better distributing intensity levels.

- **Observations**: Contrast stretching significantly improves areas where intensity levels were limited, allowing greater detail in shadows and highlights. The result is a more balanced and visually appealing image.

T1.5 - Intensity Slicing

Intensity slicing was used to highlight a specific range of intensities (between 100 and 200) and attenuate the rest.

- **Observations**: The transformed image clearly shows the areas within the slicing range, while other regions appear black. This technique is useful for highlighting specific features in medical or data analysis images.

T1.6 - Bit-Plane Slicing

The 8-bit planes of the image were extracted, showing each bit's contribution to the image formation.

- **Observations**: Higher bit planes (6 and 7) contain the most structural information, while lower bit planes (0 and 1) are noisier and contain less relevant information.

Section T2: Histogram Processing

T2.1 - Histogram Calculation

A function was implemented to calculate the histogram of an image, showing the distribution of intensity values.

- **Observations**: The histogram reflects the tonal distribution in the image. High-contrast images have a more dispersed histogram, while low-contrast images have values clustered in a narrow range.

T2.2 - Transformation Histograms

Histograms were calculated for all intensity transformations performed in Section T1.

- **Observations**:
  - **Negative image**: The histogram is inverted compared to the original image.
  - **Logarithmic transformation**: The histogram is more concentrated in lower levels, indicating dark values have expanded.
  - **Gamma correction**: The $\gamma > 1$ image showed greater dispersion towards high values, while the $\gamma < 1$ image was more concentrated in low values.
  - **Contrast stretching**: The histogram spread across the entire range of values, significantly improving contrast.
  - **Intensity slicing**: The histogram showed a peak in the selected range, with values outside the range at 0.
  - **Bit-plane slicing**: Each bit-plane has a different histogram, reflecting the amount of information contained in each plane.

T2.3 - Histogram Equalization

Histogram equalization was implemented, and the results were compared to the original image.

- **Observations**: Histogram equalization increased the image's contrast by redistributing intensity values more evenly. This resulted in a more visually balanced image, though fine details might be lost due to increased contrast in some areas.

Section T3: Spatial Filtering

T3.1 - Low-Pass Filtering with Box Kernels

A low-pass filter with box kernels of sizes 3x3, 11x11, and 21x21 was applied.

- **Observations**:
  - **Kernel 3x3**: Slightly smooths the image while preserving details.
  - **Kernel 11x11**: Provides more noticeable smoothing, removing some fine details.
  - **Kernel 21x21**: Significantly smooths the image, making it much blurrier. This size is useful for noise removal but may result in the loss of important information.

T3.2 - Gaussian Filtering

The process in T3.1 was repeated using Gaussian kernels with different σ values.

- **Observations**: Gaussian filtering produces a more natural smoothing effect compared to box filtering. As σ increases, the image smooths more, but edges are better preserved compared to equivalent box kernels.

T3.3 - Median Filtering

Median filtering was applied with different kernel sizes.

- **Observations**:
    - **Kernel 3x3**: Removes noise without significantly affecting details.
    - **Kernel 5x5**: Removes more noise but starts to smooth details.
    - **Kernel 9x9**: Removes a large amount of noise but also significantly smooths image details. It is ideal for images with high noise levels.

T3.4 - High-Pass Filtering

Sobel, Prewitt, Roberts (first derivative), and Laplacian (second derivative) operators were applied.

- **Observations**:
    - **Sobel**: Detects edges with smoothing, preserving details while minimizing noise.
    - **Prewitt**: Similar to Sobel but less precise in edge detection.
    - **Roberts**: Detects sharper edges but is more susceptible to noise.
    - **Laplacian**: Highlights edges with higher precision but also amplifies noise. It is recommended to use this operator along with a prior low-pass filter.

Conclusions

This image processing analysis showed that intensity transformations, histogram processing methods, and spatial filters can significantly enhance contrast, reduce noise, and highlight key features in images. Depending on the specific objectives (e.g., noise reduction, edge enhancement, or contrast improvement), each technique has its advantages and limitations. Logarithmic transformations, histogram equalization, and median filters proved to be the most effective for enhancing visual quality under various circumstances, while high-pass filters and derivatives were effective for edge detection.