Mandril.JPG

Lab 2: Image Enhancement – Spatial Domain (Use Python + OpenCV)

1. Image Thresholding:

- Load the grayscale image from LAB 1.
- Apply (i) Binary thresholding (ii) Binary Inverse thresholding to your image using a threshold value of 127.
- Observe the effect of each method on the resulting image and their pixel values. Explain

2. Bit-Plane Slicing:

- Load the grayscale image from LAB 1 and convert it into its binary representation.
- Implement bit-plane slicing to extract specific bit-planes (e.g., MSB, LSB) and visualize each bit-plane separately and observe the contribution of each bit to the overall image representation.
- Compress the image from an 8-bit image to a 3-bit image by combining any 3 planes of choice. Is this a good method for image compression? Explain.

3. Smoothing with Low-Pass Filter:

- Load the **color** image from **LAB 1.**
- Implement a low-pass filter (e.g., average filter) to smooth the image and reduce noise. Use different filter sizes (n = 3, 5 and 7), and observe their effects on image blurring.
- Compare the results with and without smoothing and discuss the trade-offs between smoothing and loss of image details. Explain

4. Smoothing with Median Filter:

- Load a noisy (salt and pepper: SP= 0.2) image of **mandril.jpg.** (PROVIDED)
- Implement a median filter to remove the noise while preserving edges and details.
- Experiment with different kernel sizes (3,5,7) and observe their effects on noise reduction and image quality.

Reference:

1. https://docs.opencv.org/3.4/d4/d13/tutorial_py_filtering.html

2. https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html

3. https://www.kaggle.com/code/siddheshmahajan/opencv-basics-bit-plane-slicing

4. https://docs.opencv.org/3.4/d4/d13/tutorial_py_filtering.html#:~:text=medianBlur()%20takes%20the%20median,pepper%20noise%20in%20an%20image.