

Image Processing Project

Requirements

Project Requirements for Image Processing

1. Project Scope

- Build an Image Processing Application using Python only (Tkinter, PyQt5, PySide6, or any Python GUI library).
- Apply at least 6 image processing operations learned in class.
- Each group may have up to 4 students maximum.
- Total marks: 15 marks
- The project discussion will be held on **Thursday 18/12/2025**

2. Image Input Requirements

- Students must upload images from their local computer.
- Supported file types: JPG, PNG, BMP
- **Optional Rule** Using image URLs (online photos from the internet) is allowed.

3. GUI Requirements

- The GUI must include:
 - Button to **upload image**
 - Display for original image
 - Display for processed image
 - Buttons / dropdown menus for selecting operations
 - Sliders or numeric input fields for parameters (e.g., filter size, noise level)
 - Button to save the processed image
- The interface must be clean, organized, and easy to use.

4. Required Image Processing Operations (Choose at least 6)

A. Image Enhancement

- Histogram Equalization
- Contrast Stretching
- Smoothing Filters (Average, Gaussian)
- Sharpening Filters (Laplacian, High-Boost)

B. Noise & Restoration

- Add Noise (Salt & Pepper, Gaussian)
- Noise Removal (Median Filter, Adaptive Filter)

C. Color Processing

- Grayscale Conversion
- RGB Channel Extraction
- Color Enhancement

D. Segmentation & Morphology

- Thresholding (Global / Adaptive)
- Otsu's Method
- **Morphological Operations:**
 - Erosion
 - Dilation
 - Opening
 - Closing
- Contour Detection / Object Segmentation

E. Frequency Domain

- DFT Visualization
- Low-pass / High-pass Filtering.

5. Implementation Requirements

Use **Python (OpenCV)** for all image processing operations.

Code must be:

- **Modular**
- **Readable and well-commented**

Structured into functions or classes

6. Bonus Options (3 marks)

- **Real-time camera processing**
 - Add a button that opens the webcam.
 - Apply filters in real time (edge detection, smoothing, enhancement).
- **Frequency domain filtering and visualization**
 - Show magnitude spectrum of the image (DFT/FFT).
 - Apply low-pass / high-pass filters.
 - Reconstruct image
- **Automatic saving of processed images in a structured folder**
 - Add a feature to automatically save processed images after applying operations.
 - Organize saved images in a structured folder system, e.g.:
 - Separate folders for each operation or image type
 - Optionally, include timestamps in filenames for uniqueness
 - Allow the user to choose the base folder for saving all processed images.
- **Combining multiple operations into a single pipeline (batch processing)**
 - Allow users to select multiple operations to apply in sequence.
 - Apply all selected operations automatically in the specified order.
 - Optionally, allow batch processing for multiple images at once.
 - Display the final result of the pipeline and allow saving..