Project Report

Title: Image Processing Toolkit using Streamlit & OpenCV
Course: Image Processing & Analysis
Date: 07-09-2025

1. Problem Statement / Objective

The primary objective of this project is to design and implement an interactive image processing toolkit that enables users to:
- Upload images in multiple formats.
- Apply a variety of image transformations, enhancements, and filters in real time.
- Visualize both the original and processed images side by side.
- Save and export processed results in different formats.

2. Introduction

Image processing is a core area in computer vision, medical imaging, biometrics, and multimedia applications. The project integrates Streamlit (for the user interface) with OpenCV (for image processing) to develop a web-based, interactive platform.

The assignment tasks covered include:
1. Image upload, display, and format handling.
2. Basic operations such as color space conversions and geometric transformations.
3. Bitwise operations across multiple images.
4. Filtering, morphological operations, and edge detection.
5. Image enhancement techniques such as histogram equalization and sharpening.
6. Exporting processed images in different formats.

3. Requirements & Libraries Used

Functional Requirements
- A simple GUI accessible to non-programmers.
- Real-time feedback while applying transformations.
- Support for multiple image formats (PNG, JPG, BMP, TIFF).
- Download and save functionality for processed results.

Libraries Utilized
- Streamlit: Interactive web-based GUI.
- OpenCV (cv2): Core image processing operations.
- NumPy: Array and matrix operations.
- PIL (Pillow): Image conversion and export.
- io, os, datetime: File handling and metadata management.

4. Methodology

The system was designed in modular blocks:

a. Image Loading & Utilities
- Upload via st.file_uploader().
- Conversion of uploaded bytes into OpenCV-compatible arrays.
- Utilities for PIL ↔ OpenCV conversion, metadata extraction, and file export.

b. Image Processing Operations
- Color Conversions: RGB, BGR, HSV, YCrCb, Grayscale.
- Transformations: Rotation, scaling, translation, affine and perspective transformations.
- Bitwise Operations: AND, OR, XOR, NOT.
- Filtering & Morphology: Gaussian, median, mean, Sobel, Laplacian, dilation, erosion, opening,

and closing.
- Enhancement & Edge Detection: Histogram equalization, sharpening, and Canny edge detection.

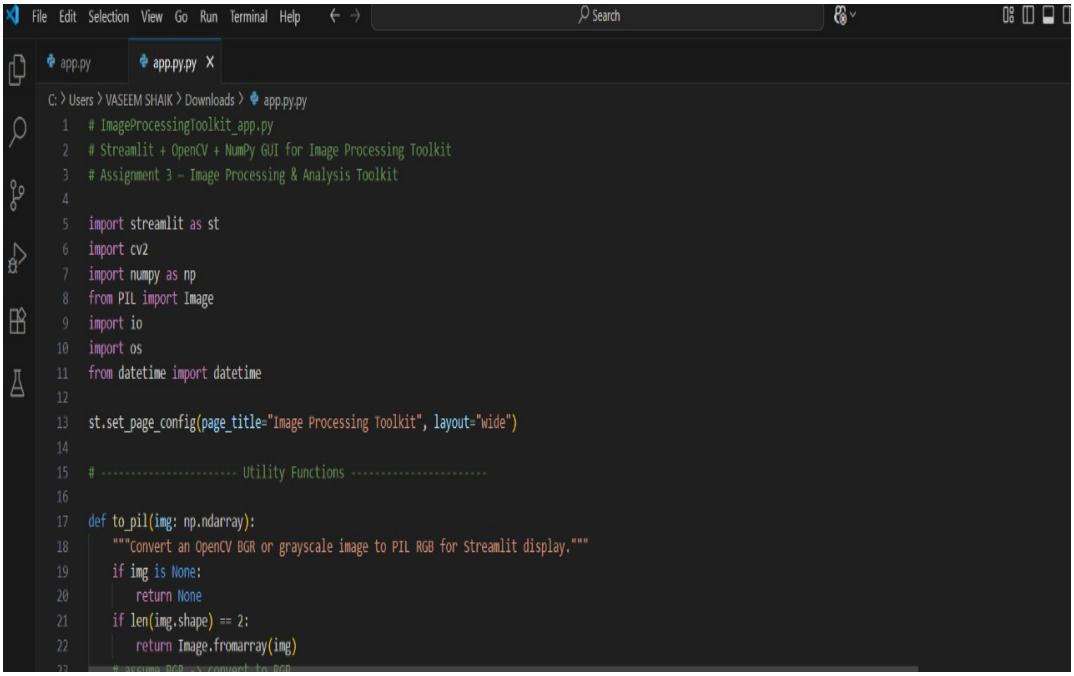
c. User Interface (Streamlit)
- Sidebar for operation selection and parameter adjustments.
- Dual-column display for original vs. processed images.
- Image information panel (dimensions, channels, size, format).
- Save/export options with adjustable JPEG quality.

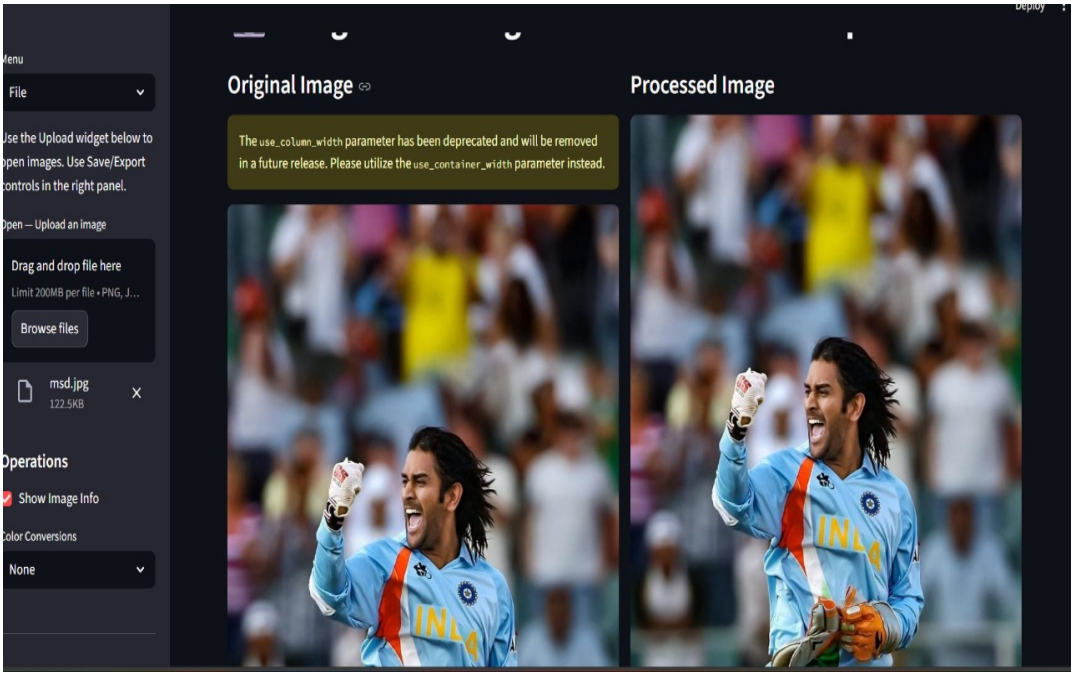
5. Results

The toolkit successfully allows users to:
- Upload and process images interactively.
- Apply multiple transformations and filters.
- Compare original and processed outputs in real time.

Examples:
- Canny Edge Detection – edges are highlighted effectively.
- Geometric Transformations – accurate scaling, rotation, and translation.

Screenshots of implementation:





6. Conclusion

Knowledge Gained:
- Practical application of OpenCV for image processing.
- Development of interactive applications using Streamlit.
- File handling, format conversion, and real-time user interface updates.

Challenges Faced:
- Managing images with varying color channels.
- Ensuring consistent BGR ↔ RGB conversions between OpenCV and PIL.
- Adjusting kernel sizes for filters.
- Maintaining interactivity without performance bottlenecks.

Future Enhancements:

- Integration of advanced filters (bilateral, non-local means denoising).
- Support for real-time video processing.
- Region-of-interest (ROI) based editing.
- Incorporation of deep learning-based techniques (super-resolution, segmentation).