Image Processing Toolkit Submission

Name: Bompally Divya Rao
Roll No: 22671A7311
Course: AIML-A
Subject: Computer Vision
Task: Assignment 3 - Image Processing Toolkit Submission

---

1. Introduction

This assignment involves creating an interactive **Image Processing Toolkit** using Python and Streamlit. The toolkit allows users to upload images and perform a variety of image processing operations including color conversions, geometric transformations, filtering, morphological operations, enhancements, edge detection, and saving the processed image in various formats. This assignment demonstrates understanding and implementation of core computer vision concepts using OpenCV and Streamlit for an interactive UI.

---

2. Objectives

- Develop a user-friendly interface for uploading and displaying images.
- Implement common image processing operations such as color space conversions, transformations, filters, morphology, enhancements, and edge detection.
- Allow saving the processed image in different formats with adjustable compression.
- Understand the integration of OpenCV image processing functions within a web application framework (Streamlit).
- Gain experience with real-time image manipulation and session management.

---

3. Tools and Libraries Used

- **Python 3.x**: Programming language used.
- **Streamlit**: For building the interactive web app interface.
- **OpenCV (cv2)**: For image processing operations.
- **NumPy**: Handling array operations and transformations.
- **PIL (Pillow)**: Image saving and format conversions.

- **Matplotlib (only imported but not extensively used)**: Could be used for plotting (not used directly here).

---

4. Functionalities Implemented

4.1 Image Upload and Display

- Users can upload images in formats like JPG, JPEG, PNG, BMP.
- Original and processed images are displayed side-by-side for easy comparison.

4.2 Image Information Extraction

- Displays filename, dimensions (width x height), number of channels, file size (in KB), and color mode (Color/Grayscale).

4.3 Color Space Conversions

- RGB ↔ BGR
- RGB ↔ HSV
- RGB ↔ YCbCr
- RGB ↔ Grayscale

4.4 Geometric Transformations

- Rotation (angle and scale adjustable)
- Scaling (X and Y scales adjustable)
- Translation (X and Y translation adjustable)
- Affine Transform (preset points)
- Perspective Transform (preset points)

4.5 Filtering

- Gaussian Blur
- Mean Filter (Averaging)
- Median Filter
- Sobel Filter (edge detection)

- Laplacian Filter (edge detection)

4.6 Morphological Operations

- Dilation
- Erosion
- Opening
- Closing

4.7 Image Enhancement

- Histogram Equalization (grayscale and color images)
- Contrast Stretching
- Sharpening (using convolution kernel)

4.8 Edge Detection

- Sobel (with adjustable dx, dy)
- Canny (with adjustable thresholds)
- Laplacian

4.9 Saving Processed Image

- Supports saving in JPEG, PNG, and BMP formats.
- Provides download button with proper mime type and filename.

---

5. Code Structure and Logic

- **Session State Management:** Uses Streamlit's session state to persist images and related data.
- **Image Normalization:** Ensures images are correctly normalized for display (0-255 uint8).
- **Function-Based Operations:** Each processing task is modularized into functions for easy maintenance.
- **User Interface:** Sidebar controls are organized by operation categories (File, Color, Transformation, Filtering, Morphology, Enhancement, Edge Detection, Compression).
- **Dynamic Parameters:** Interactive sliders and dropdowns allow users to fine-tune operation parameters.

- **Image Processing Pipeline:** Applies selected operations sequentially on the original image and updates the processed image.
- **Page Refresh:** Uses st.rerun() to update displayed images after processing.

---

6. Challenges Faced

- Managing color space conversions consistently between OpenCV (BGR) and PIL (RGB).
- Handling different image types (grayscale vs color) for processing functions.
- Ensuring user interface responsiveness while performing potentially expensive image operations.
- Maintaining image quality during saving and compression.

---

7. Future Enhancements

- Add support for custom affine and perspective transform points through UI input.
- Implement additional filters like bilateral filtering, edge-preserving filters.
- Add histogram and pixel value visualizations.
- Add batch processing for multiple images.
- Incorporate AI-based enhancements or segmentation models.
- Improve UI/UX for better accessibility and operation previews before applying.

---

8. Conclusion

This assignment provided hands-on experience in building an end-to-end image processing web application using Streamlit and OpenCV. It strengthened my understanding of fundamental image processing techniques and their practical application, as well as skills in developing interactive applications with real-time updates. This toolkit can serve as a foundation for further advanced computer vision projects.

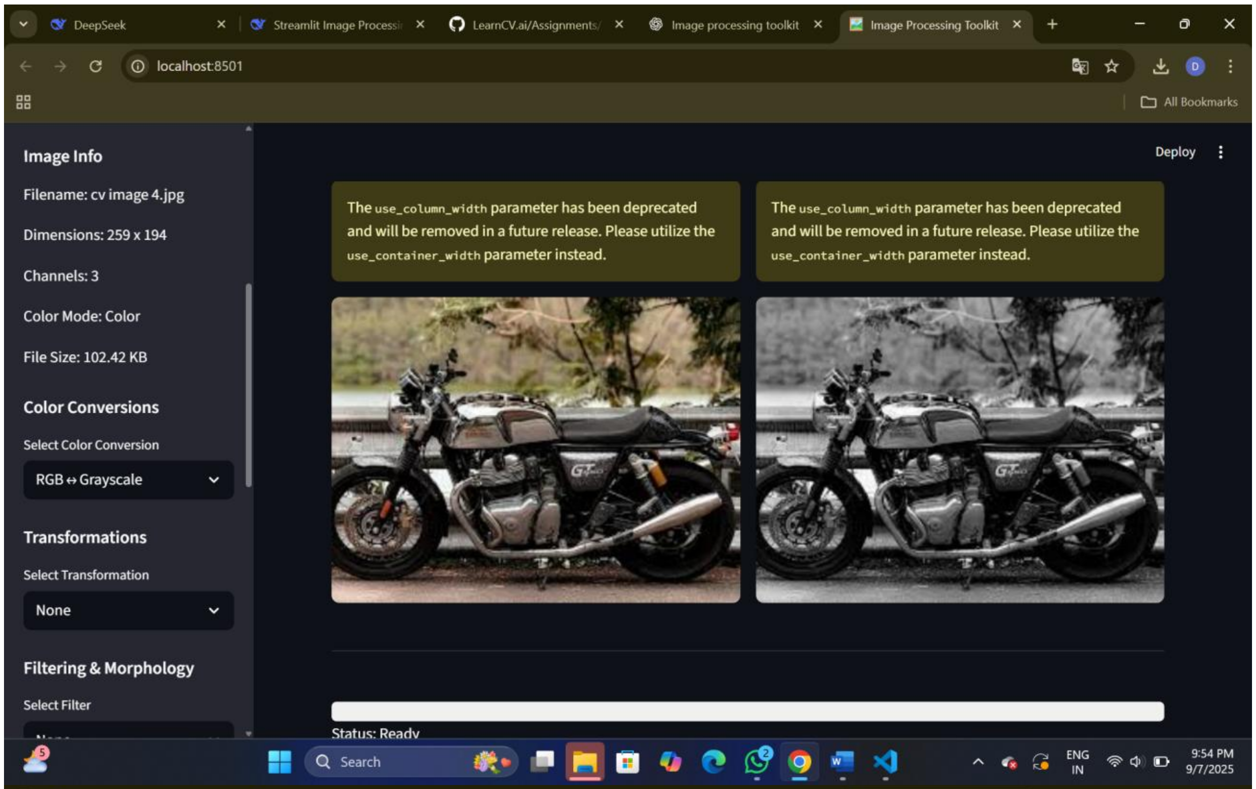


Image Info

Filename: cv image 4.jpg

Dimensions: 259 x 194

Channels: 3

Color Mode: Color

File Size: 102.42 KB

Color Conversions

Select Color Conversion: RGB ↔ Grayscale

Transformations

Select Transformation: None

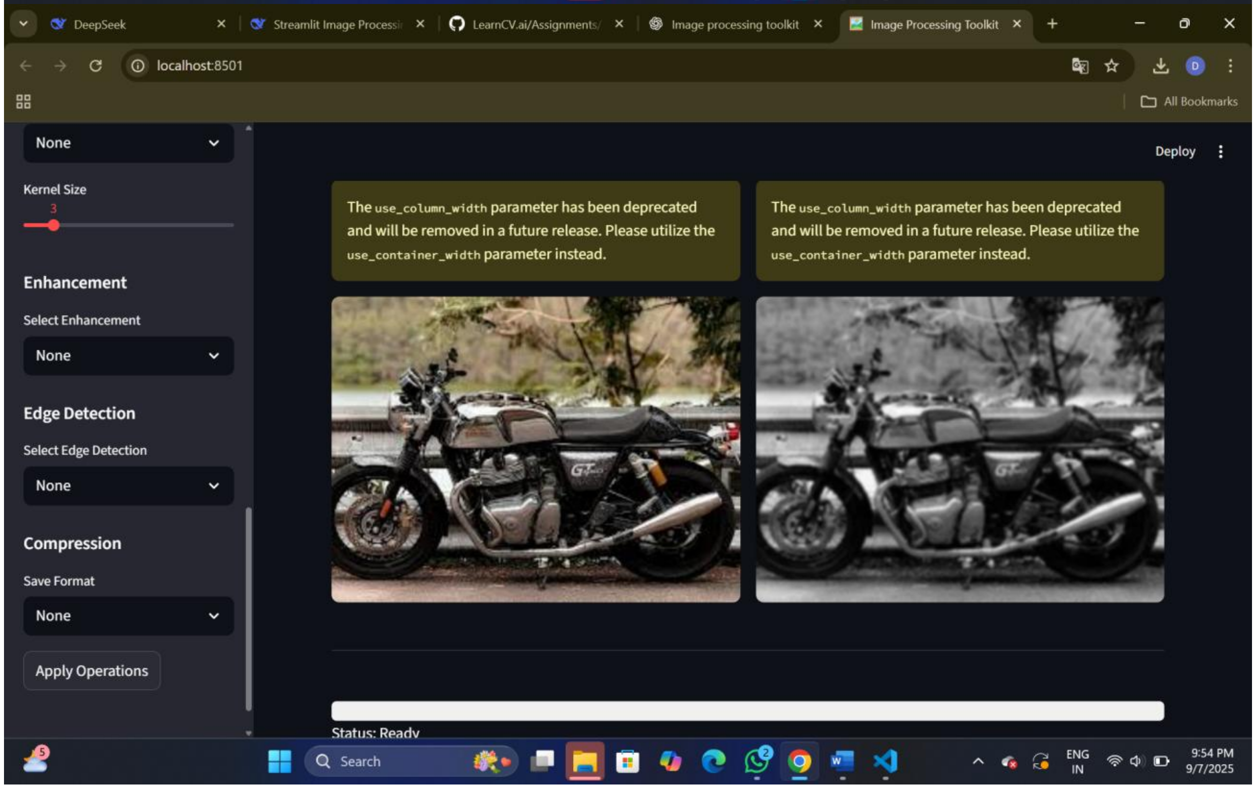
Filtering & Morphology

Select Filter

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Status: Ready



None

Kernel Size: 3

Enhancement

Select Enhancement: None

Edge Detection

Select Edge Detection: None

Compression

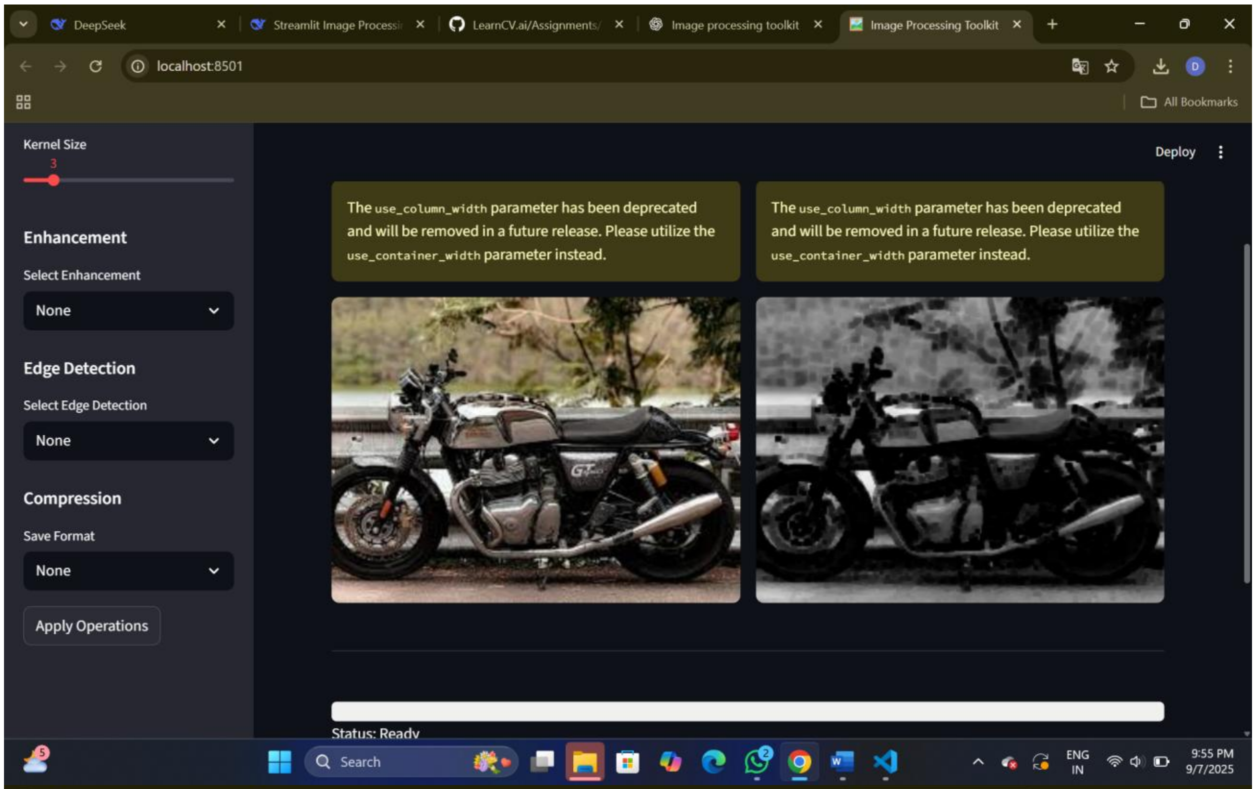
Save Format: None

Apply Operations

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Status: Ready



Kernel Size

3

Enhancement

Select Enhancement

None

Edge Detection

Select Edge Detection

None

Compression

Save Format

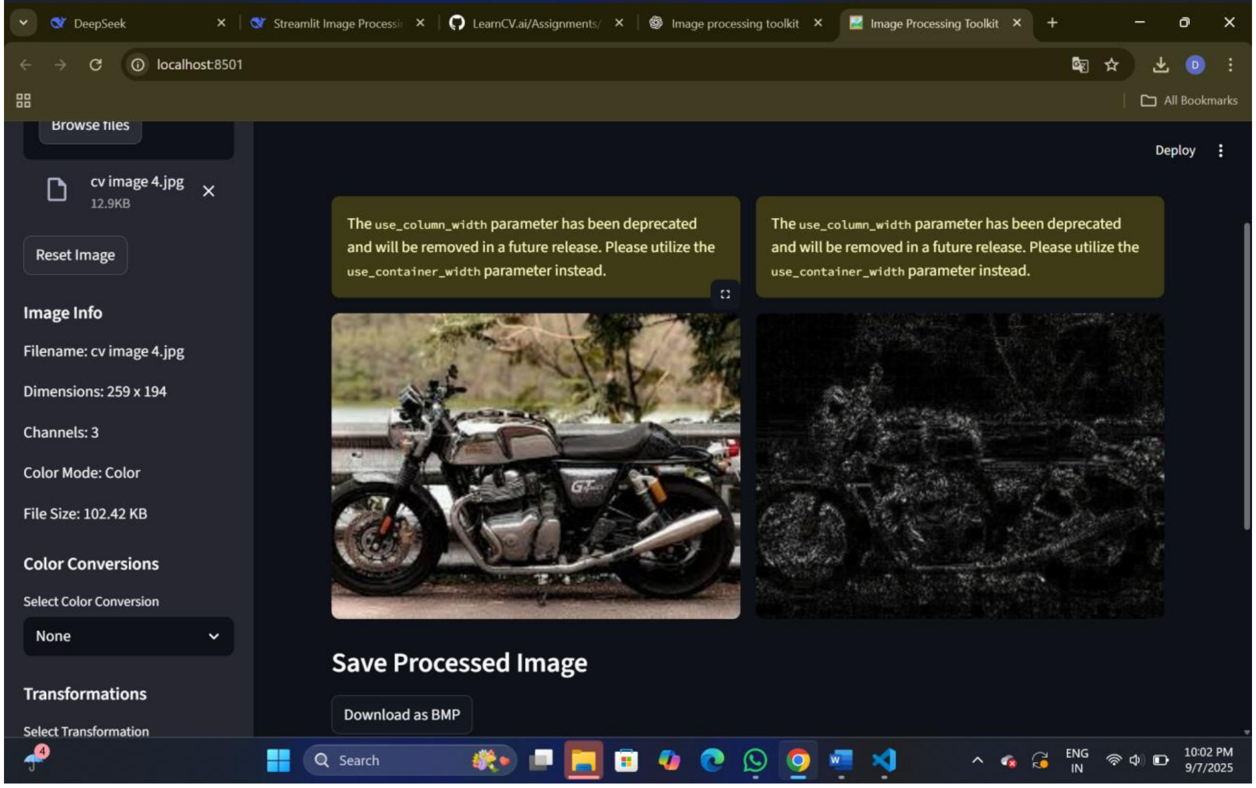
None

Apply Operations

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Status: Ready



Browse files

cv image 4.jpg
12.9KB

Reset Image

Image Info

Filename: cv image 4.jpg

Dimensions: 259 x 194

Channels: 3

Color Mode: Color

File Size: 102.42 KB

Color Conversions

Select Color Conversion

None

Transformations

Select Transformation

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Save Processed Image

Download as BMP