Streamlit Image Processor Project - Live Code Explanation Script

# 1. Project Introduction

Hi everyone, welcome to my project walkthrough. This is an advanced image processing app built using Streamlit and OpenCV. It allows users to upload or capture an image, apply transformations like grayscale, binary, edge detection, blur, invert, and even AI-based enhancement. Let's walk through the code and understand how everything works.

# 2. Import Libraries

These are the libraries I'm using:  
- streamlit for building the UI  
- cv2 (OpenCV) for image processing  
- os to manage file directories  
- PIL for image conversion  
- io for handling in-memory files  
- time to generate unique filenames  
- numpy for image arrays  
- fpdf to create PDF files with the processed image

# 3. Create Upload Folders

This ensures that folders for uploaded and processed images exist. If not, Python creates them automatically.

# 4. Setup Streamlit Page

This part sets up the Streamlit UI with a title and description so users know what the app does.

# 5. Input Method Selection

This allows users to either upload an image or take a new photo using their webcam.

# 6. Handle Upload or Camera Input

Depending on what the user selects, this code displays a file uploader or a camera input. It also allows cancelling the camera if needed.

# 7. Decode Image

Once we get the image from either method, we use OpenCV and NumPy to convert it into a format we can process.

# 8. Select Processing Options

This radio button lets users choose the type of processing they want to apply.

# 9. Optional Brightness/Contrast

Users can optionally adjust brightness and contrast before processing. Sliders let them fine-tune the settings.

# 10. Save Input and Process Image

If an image is available, we save it to the upload folder and make a copy to preserve the original.

# 11. Apply Brightness and Contrast

Here we apply brightness and contrast changes using OpenCV’s convertScaleAbs function.

# 12. Apply Selected Processing

This is the main processing block.   
- Grayscale converts image to black & white  
- Binary applies thresholding  
- Edge Detection uses the Canny algorithm  
- Blur applies a Gaussian filter  
- Invert reverses image colors  
- AI Enhance sharpens the image using a kernel filter.

# 13. Display in Columns

We create 3 columns:  
- Left: Original image  
- Center: Processed image with download options  
- Right: Image info like size and channels

# 14. Download Buttons

Processed images can be downloaded as PNG or PDF. The PIL library is used to save PNGs and fpdf for PDF generation.

# 15. Show Image Info

This section provides basic metadata about the uploaded image.

# 16. Conclusion

So that’s how the Advanced Image Processor works. It covers key aspects of OpenCV image transformations, real-time UI interaction with Streamlit, and offers useful features like brightness/contrast adjustment and file downloads. This project has been a great way for me to apply my skills in a practical, user-friendly tool.