Image Processing Techniques using OpenCV

Objective

The goal of this project is to understand and implement the logics of GAN. On this task, we were asked to use OpenCV and Matplotlib to explore through various functions of `cv2` and plot them as an output. We also used the webcam to capture real-time images and apply various filters on it.

Various Functions of OpenCV (cv2)

Capturing Image from Webcam

I explored OpenCV's feature of capturing real-time images through my webcam. I used `VideoCapture()` to capture the image. Since the image is captured in BGR format, it was converted to RGB for visualization using Matplotlib.

HSV Color Space Conversion

I converted the RGB image into HSV (Hue, Saturation, Value) color space and split it into individual channels (H, S, V). Hue represents the color, Saturation its intensity, and Value the brightness.

Histogram Equalization

To enhance contrast, I converted the image to grayscale and applied histogram equalization. This helps in improving the overall contrast of an image.

Binary Inversion Thresholding

Binary inversion thresholding was applied to the grayscale image. Pixels below a threshold (128) were turned white, and others black. This is critical for document analysis and shape segmentation, and is also used in OCR and background subtraction.

Reducing to 4 Gray Intensity Levels

Here, I reduced the grayscale image to 4 intensity levels by grouping intensity values into 4 bins. This creates a stylized image and helps reduce noise and simplify features.

Laplacian and Scharr Filters

Two edge detection techniques were used: Laplacian and Scharr. Laplacian uses second-order derivatives, while Scharr is an enhanced Sobel filter providing smoother, more accurate edges.

Noise Removal with Median Filtering

Salt-and-pepper noise was simulated and removed using a median filter. Median filtering preserves edges while effectively removing noise.

Unsharp Masking for Sharpening

I applied unsharp masking by subtracting a blurred version of the image from the original, enhancing fine details and edges.

LAB Color Space Analysis

The image was converted to LAB color space and split into L (lightness), A (green-red), and B (blue-yellow) channels. LAB is perceptually uniform and useful for color correction and comparison.

Final Output

Below are the results of the applied filters and techniques:

Figure 1: Final Filter Grid Output

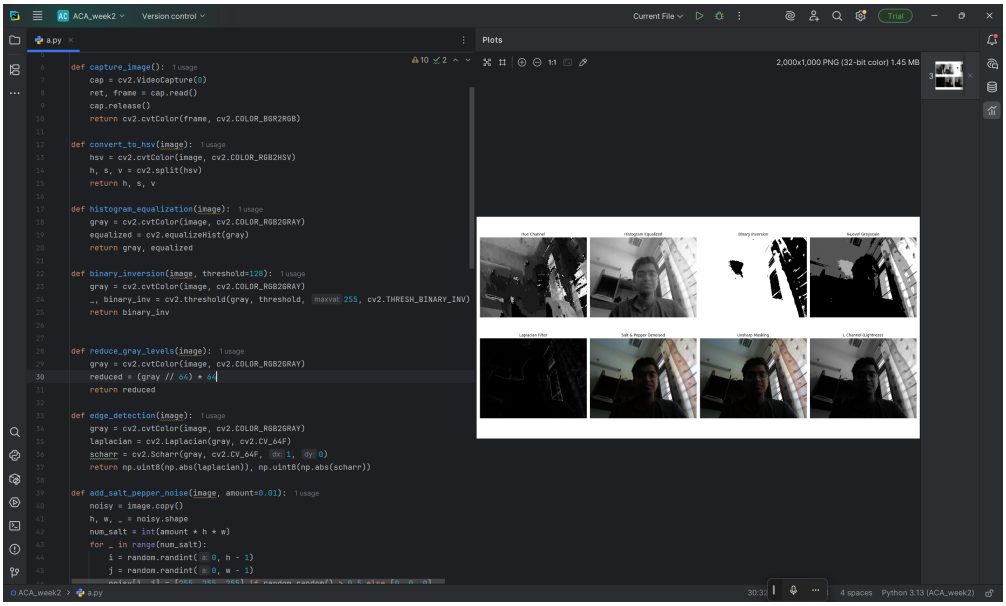

Figure 2: Screenshot of Code