Lab2 : Image Processing

Tools: Python, OpenCV, NumPy, Matplotlib

1. Objectives

- Build on your previous work by diving deeper into image processing techniques.
- Develop a mini-pipeline that applies various transformations to an image.
- Document your process, observations, and any challenges in a short report.

2. Explore Color Representations

Task 1: Load and Display an Image

- Load an image using OpenCV
- Display it using OpenCV or Matplotlib
- Print the image dimensions (height, width, channels)

Task 2: Explore Color Representations

- Convert the image to grayscale and display the result
- Convert the image from RGB (BGR in OpenCV) to HSV
- Split and display the individual HSV channels
- Briefly describe the visual differences and significance of each channel
- List and convert the image into multiple color representations and display all of them.

Task 4: Geometric Transformations

- Scale the image (e.g., reduce by 50% or enlarge by 150%).
- Rotate the image by a fixed angle (e.g., 45°) using cv2.getRotationMatrix2D and cv2.warpAffine.
- Translate (shift) the image's position horizontally or vertically.
- Compare at least two interpolation methods (e.g., INTER_LINEAR vs. INTER_NEAREST) for each transformation.

3. Mini-Pipeline Project

- Combine the operations from Tasks 1-3 into one cohesive pipeline. For example, load an image, convert to HSV, extract a channel, apply a transformation, and display the original and processed images side-by-side
- Ensure your code is well commented to explain each step

4. Conclusion & Discussion

- Summarize the key findings from your lab tasks
- Discuss the effects of the various transformations on the image
- Reflect on any challenges encountered and insights gained
- Suggest potential improvements or extensions for future exploration
- Submit your Python code alongside a short lab report .Your report should explain your workflow, the outcomes of each task, and any insights or challenges you encountered.
- Include code snippets and screenshots where appropriate.