Indian Institute of Science Education and Research Bhopal

Computer Vision(DSE312/ DSE602/ ECS320)

Assignment-1

Deadline: 09-09-2025, 11:59 PM Max mark: 15

Please follow the instructions carefully.

1. **All questions are mandatory. Plagiarism and copying from anywhere (similar submission) can debar you from this course and invite the academic dishonesty policy.**
2. **Implement all algorithms purely in Python without using specialised libraries like OpenCV or PIL for the processing. You may use libraries for basic operations (like loading an image), but the algorithms should be coded from scratch.**
3. **Comment on your code extensively to explain your logic and the steps you are implementing.**
4. **Display both the original and processed images to compare results.**
5. **Make a short 7-minute video and explain your code.**
6. **A report reflecting on what you have learned, prepared in LaTeX (.tex) format. Visualisation of the output must be there along with other necessary details.**
7. **You must submit a report (.pdf), codes (.py or .ipynb), and a video explaining your code and the concept behind it.**
8. **Use your own photo as an image for all the questions given.**

1. Write a program using OpenCV to read an RGB image, convert it into grayscale, visualise the individual colour channels (R, G, B), and compute as well as display the histogram for each channel. ***(Marks:2)***
2. Using OpenCV, read an RGB image, add Gaussian noise to it (e.g., mean = 0, variance = 10–20). Then, on the noisy image:
    1. Convert it to grayscale.
    2. Visualise the individual colour channels (R, G, B) and the grayscale image.
    3. Compute and display the histogram for each channel (R, G, B) and for the grayscale image.

    All plots must be clearly labelled. ***(Marks:3)***

3. Apply three different filters of sizes 3×3, 5×5, and 7×7 on an image with and without Gaussian noise. Perform edge detection using the following methods:

    1. Sobel edge detector
    2. Edge detection using the Sobel operator in both horizontal and vertical directions
    3. Laplacian edge detector

    Compare and describe the differences observed in edge detection results across the filters, kernel sizes, and noise conditions. ***(Marks:5)***

4. Implement the Canny edge detector from scratch by following these steps:

    1. Apply Gaussian smoothing.
    2. Compute gradient magnitude and orientation using Sobel filters.
    3. Apply non-maximum suppression.
    4. Apply hysteresis thresholding (use two thresholds).

    Apply your implementation to both a clean image and a noisy image. Show how noise affects the results. ***(Marks:5)***