An-Najah National University
Department of Computer Engineering
Digital Image Processing - 10636318
Second Semester 2024/2025
OpenCV Project

Part 1: Design and implement a complete pipeline that loads a real-world input image of your choice, applies various digital image processing techniques, and produces a report along with annotated results, according to the following instructions:

- **Input image**: Take a photo of any object using your own camera (or smartphone). Embed your name and student ID as a visible watermark in a random location within the image.
- **Tasks**:
  - Load the image in grayscale
  - Display its dimensions, color channels, and pixel value statistics (mean, min, max, etc.).
  - Modify the brightness of the grayscale image by applying the following equation

    $$s = c*r$$

    where:
    - s is the output gray level
    - r is the input gray level
    - **c is a random value where $0.4<c<2.0$**
  - Analyze the brightness of the resulting image by plotting and discussing its histogram.
  - Apply one or more suitable techniques to correct the brightness of the resulting image. Provide justification for your selected methods. Also, plot the histograms of the original and corrected images for comparison.
  - Write a function to add salt-and-pepper noise by randomly changing the values of some pixels to black or white.
  - Reduce the noise using both mean filter and median filter. Then, apply any additional filters necessary to correct artifacts that may result after noise reduction (e.g., sharpening filters). Compare the performance and visual outcomes of both filtering methods.
- **Deliverables**
  - Source code
  - Original input image
  - Output images showing intermediate and final results
  - Report (PDF):
    - Explanation of each step
    - Screenshots of outputs

    - A summary of challenges faced and justification for the techniques used, explaining why certain filters or transformations were chosen

Part 2: Write a program to read a video stream from your computer's camera. Allow the user to choose one of the following display modes:

1. **Edge detection**: Apply any suitable edge detection technique.
2. **Grayscale Quantization**: Reduce the grayscale levels to a user-defined number.
3. **Contrast enhancement:** Enhance contrast by applying histogram equalization frame-by-frame. Test your implementation by changing the brightness of your environment and observe how the processed video responds.
4. **Soft and polished appearance:** Apply a blurring filter to smooth the video frames. This helps reduce noise and minor facial imperfections, resulting in a soft and polished appearance.
5. **Cartoon Filter:** Apply creative effects like a cartoon-style filter for fun and practice. *Hint: This may involve converting frames to grayscale, applying smoothing, detecting edges, and performing quantization.*

Please submit your source code for both parts along with a report in PDF format for part 1. Adhere to the following regulations:

- Work in groups of two to complete the project.
- Only C, C++, or Python programming languages can be used, in conjunction with OpenCV.
- Resources: OpenCV Documentation
- Deadline: 11.05.2025