Assignment 3

The purpose of this assignment is to apply your knowledge of edge detection and circle detection (a form of geometric primitive extraction) to segment and identify the circular boundaries between the pupil-iris and iris-sclera in eye images.

Background:
The human eye has distinct regions that can be segmented for various applications, notably in biometrics for iris recognition. Two of the most prominent boundaries are between the pupil and the iris, and between the iris and the sclera (white part of the eye). This assignment will test your ability to identify and demarcate these regions.

Dataset:
You will be provided with 5 eye images. Each image contains one eye.

Tasks and Mark Distribution:

Total marks = 26

1. **Circle Detection and Results Compilation (20 marks)**

For each eye image in the “Input_Images”:

- Detect the circle boundary between the pupil and the iris (2 marks per image).
- Detect the circle boundary between the iris and the sclera (2 marks per image).

Overlay the detected circles on the original images and save these segmented images in a designated folder named "Output_Images".

Also, save the corresponding edge maps in a designated folder named "Edge_Maps".

Note: Partial marks will be awarded if a circle is not precisely detected but a commendable edge map indicating the potential boundary is generated.

2. **Report (4 marks)**

Write a brief report discussing:

- Your approach and the algorithms used.
- Challenges faced and how you overcame them.
- Insights or observations from the results.
- Any potential improvements that could be made in the future.

The report should not be more than 2 pages of text (excluding the images).

3. Code Cleanliness (2 mark)

Your code should be:

- Well-commented to explain your logic.
- Organized in functions or classes as appropriate.
- Free from redundant or unnecessary lines of code.

Note: A clear and well-structured code can significantly aid in understanding and debugging. Aim for clarity and simplicity.

Deliverables

Submit a zip file containing:

- A folder named “Code” with the complete source code, ensuring it's executable on another system without additional modifications. This can include Python files (.py) or Jupyter Notebooks (.ipynb).
- A folder named “Edge_Maps” with the resulting edge images. Name your output images as “x_edge.tif”, where x is the corresponding input image's name. Non-compliance will result in marks deduction.
- A folder named “Output_Images” with the output images. Name these as “x_output.tif”, where x is the corresponding input image's name. Non-compliance will result in marks deduction.
- A PDF file named “Assignment 3_Report” for documentation.
- Note that a folder structure has been provided in the given zipped assignment for consistency.

Instructions

- You may use high-level functions from OpenCV, NumPy or other libraries.
- Your code must work effectively on unseen images. Use a consistent set of parameters that provides good results across all input images. **Avoid fine-tuning parameters specifically for each individual image.**
- You can also use code from the Internet as long as:
  a. You have significantly enhanced the original code.
  b. The source is properly cited in your report.
  c. Your report clearly describes how the code operates.
- Use Python version > 3.6 and OpenCV version > 3.4
- All code that you submit should either be original or, if sourced from the internet, properly cited as indicated above.
- Submit all materials to MLS in a single zip file.