CSCE 4603 Fundamentals of Computer Vision
Course Project Specification

Project Title : *Sudoku Solver*[1]
Weight : *30%*

Introduction.
This project is to be carried out in groups of 2 – 4 students. It is designed to cover a variety of the topics to be discussed in the lectures, and related topics to enrich student experience in the field of Computer Vision, and the creation of a practical solution to a real-life problem. The main problem at hand is to be able to parse a Sudoku puzzle grid from any real-life source via camera, then solving it.

Topics to be assessed.
1. Robust image preprocessing.
2. Reliable image enhancement and noise attenuation.
3. Extraction of puzzle information from image via morphological operations and Hough transform.
4. Applying the required geometric transformations to straighten the puzzle for optical recognition.
5. Manual implementation of OCR (***without*** machine learning algorithms – i.e., via pattern matching).

Detail of the task.
This project aims to incorporate the material discussed in the course in a practical application, encouraging students to explore said material in practice and how to apply it in an actual problem they are facing, such as that of the captured image of the sudoku puzzle. By preprocessing and cleaning the captured image, an undistorted, noiseless, binarized (grayscale), clear grid of scale-neutral digits should be obtainable and can be further processed with basic OCR to extract the values in the grid, and finally solve the puzzle.

The OCR portion of the project may not be fully explored in the course material, however; it aims to expand the knowledge of students in the field, and how to create a basic implementation of a new technology, then see how this technology has been improved in the real world. This is why emphasis is put on refraining from implementing OCR in the project with machine learning, since indulging in a basic implementation will give better intuition to why machine learning for such a use-case is good, instead of using it directly without much background on the subject.
The project is divided into two milestones, and students are required to work in groups of 2 – 4 to produce working code, demo, documentation, as well as converse about their progress periodically and share their findings and conclusions with their instructors and discuss the final output.

What you should hand in.
You will be handing in your work in the form of a well-documented git repository pushed and collaborated[2] on through GitHub throughout the semester over two milestones, encompassing the following:

---

[1] A real-life implementation of this project can be found in the 2009 iPhone application called "Sudoku Grab". You can refer to this blog post if you're interested in how its author implemented it.
[2] Make sure your commits are distributed, granular, and self-explanatory. This will be a critical point during discussion, take care!

Milestone 1 (due 22/11/25): (12 marks)

- *Preprocessing of the captured image. (4 marks)*
- *Outer frame isolation. (4 marks)*
- *Outer frame corners identification. (2 marks)*
- *Grid straightening into a square. (2 marks)*

Milestone 2 (due 4/12/25): (18 marks)

- *Basic OCR with pattern matching. (8 marks)*
- *Solving the extracted sudoku puzzle.[3] (2 marks)*
- *Documentation[4], demo[5], and discussion. (8 marks)*

Regarding academic misconduct.

The "*Academic Misconduct*" is defined as: 'any case of deliberate, premeditated cheating, collusion, plagiarism or falsification of information, in an attempt to deceive and gain an unfair advantage in assessment'. This includes attempting to gain marks as part of a team without contributing. The academic misconduct is considered very seriously, and any suspected cases will be investigated.

It is your responsibility to ensure that you understand what constitutes Academic Misconduct and to ensure that you do not break the rules.

[3] You do not need to implement the sudoku solution algorithm, you are allowed to look it up and integrate it with your application.
[4] Your documentation (readme.md) needs to be thorough and include the references you've referred to in your implementation.
[5] Your demo (video in readme.md) must be performed on test cases which shall be handed out prior to submission.