Ain Shams University
Faculty of Engineering

University of East London

CSE483 / CESS5004 – Computer Vision
Course Project Specification

Introduction.
This project is to be carried out in groups of ***1 – 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 **scan a barcode from a sample of pictures with various distortions** with a **robust and generic image preprocessing pipeline** using **only classical computer vision algorithms**.

Topics to be assessed.

1. Robust image preprocessing techniques.
2. Adaptive image enhancement and noise attenuation techniques.
3. Extraction of barcode from image via morphological operations / Hough transform / contours.
4. Applying the required geometric transformations to extract the region of interest (barcode bars).
5. Decoding the barcode (sample images are all encoded using the code 11 symbology).

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 a barcode to automatically scan the bars on the label emulating the behaviour of real-life barcode scanners. By preprocessing the captured barcode label image, an undistorted, noiseless, binarized (grayscale), clear barcode be obtained and decoded as shown in the linked guide above.

The barcode decoding portion of the project may not be fully explored in the course material, however; it aims to expand the knowledge of students, and how they can apply their obtained knowledge on basics of classical computer vision to implement a real-life standard through simple logic; in this case that is to decode the position of pixels (representing the bars of the barcode) and their varying widths to decode the data represented in the barcode. **You should not attempt to read the text representation of the data at the bottom of the barcode with machine/deep learning and should not use any external libraries other than those discussed in the course labs**.

The project is divided into two milestones, and students are required to work in groups 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.

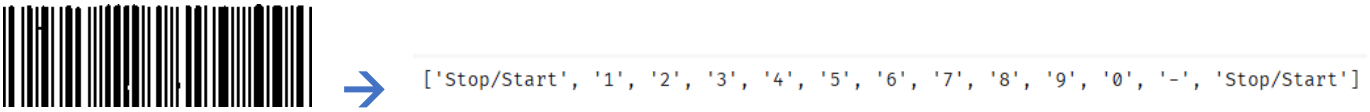
You will be handing in your work in the form of a well-documented git repository pushed and collaborated[1] on through GitHub throughout the semester over two milestones, encompassing the following:

*Milestone 1 (due **by the end of Week 9**):*

- *Generic preprocessing of the captured image (i.e., fixing all types of distortion). (15 marks)*
- *Cropping and straightening the image to contain only the bars of the barcode label. (5 marks)*

*Milestone 2 (due **by the end of Week 13**):*

- *Decoding the barcode data. (5 marks)*
- *Documentation[2], demo[3], and discussion[4]. (10 marks)*



Your final pipeline needs to be **generic** and **robust** enough, capable of passing as many test cases as possible without requiring any modifications or fine-tuning on-the-spot. *In other words, if-conditions that attempt to figure out which test case is currently being processed are not considered generic. Your code should only try to figure out the problem(s) in the current test case (e.g., salt-and-pepper noise, rotated barcode, etc.) and apply the appropriate fix(es) when applicable.*

Regarding academic misconduct.

The University defines "*Academic Misconduct*" 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 department takes academic misconduct very seriously and any suspected cases will be investigated through the University's standard policy. If you are found guilty, you may be expelled from the University with no award. It is your responsibility to ensure that you understand what constitutes Academic Misconduct and to ensure that you do not break the rules. If you have any confusion regarding what is required, please ask (*via Teams*).

Also, make sure you've read the general guidelines for use of artificial intelligence in any coursework.

[1] Make sure your commits are distributed, granular, and self-explanatory. This will be a critical point during discussion, take care!

[2] Your documentation (readme.md) needs to be thorough and include the references you've referred to in your implementation.

[3] Your demo (video or notebook) must be performed on test cases and contain results of all test cases (if possible). In case of test case failure, you will need to explain why this test case failed, and what you think could fix it but don't have time to try it.

[4] Any team member can be asked about any part of the project; hence all members should be prepared in the discussion.