

**PDE4434**

**2024/25**

**Assessment 2 – Computer vision task**

Assessment weight: 60%

Due: week 25

Deadline 18/04

h. 23:59 UK Time

Write a program for recognising cards from the game UNO. The program will receive input from file or from a standard camera and identify cards in the image or frame (one at a time). The task includes building the image dataset on which to perform the recognition process.

Students will submit in a ZIP file:

- 1) all the code
- 2) the image dataset
- 3) a readme file describing the submission content and any guidelines necessary to run and check the code
- 4) a short video of the final system running on their computer.

You can use any function from OpenCV, scikit-learn, TensorFlow or from libraries included in the Anaconda Python distribution. Keep in mind that sometimes implementing a function yourself can be faster / easier than making something external work as you wish. If you find useful information / examples on the web, please cite appropriate references.

**Data collection:** In building the image dataset, consider the characteristics of the setup, such as background and illumination. You are allowed (and advised to) make the visual processing as simple as possible. Please submit all images used to train and test the system.

**Classification:** Card recognition will identify both colour and number. You are encouraged to use classification based on machine learning techniques, but this is not a requirement, and you can implement specific code for recognising some features algorithmically.

**Documentation:** The code should easily run on a standard platform. Include an option (or alternative code) for running the program on saved files or camera stream. Any important information related to code execution, user interface, help, etc. has to be included in the readme file. Use of repositories (e.g. GitLab) is advised, although not required.

## Evaluation

- 30%    **Approach.** Soundness and correctness of the chosen approach; suitable and convenient choice of libraries and functions; demonstrated knowledge of computer vision techniques. The use of techniques covered in class is preferred and will be rewarded, compared to other possible approaches.
- 30%    **Implementation.** Appropriate and competent coding solutions; sensible program structure with functions and modularity; bugs and errors; safety checks; code clarity and readability. The three components of the code will be assessed: data collection, recognition from images, recognition from camera stream.
- 30%    **Performance.** Accuracy and reliability in recognising shapes and colours; robustness and generality; user interface and options; display solutions. Performance will be assessed for both image and video recognition.
- 10%    **Documentation.** File and function descriptions; suitable line comments; readme file.