Handwritten Number Recognition System (HNRS)

- **Due:** *11:59 pm 02/11/2025 (End of Week 12)*
- **Contributes** 50% of your final result
- **Group Assignment – Group of 3-4 students**

Summary

This project requires your group to implement and demonstrate a system that can recognise handwritten numbers. This is a problem that is simpler than the more complex and practical problem of handwritten text recognition. On the other hand, the problem is more challenging than the typical problem of Handwritten Digit Recognition. The problem can typically be broken down into several subproblems: *image acquisition*, *digit localization* (*separation*), *individual digit recognition*, and (possibly) *multi-digit number construction*.

Overall constraints:

The HNRS can use any machine learning technique or combinations of them. You should take advantage of existing libraries such as numpy, pandas, PyTorch, Tensorflow, Keras, etc.

- At the minimum, the HNRS will have to be able to train ML models for handwritten digit recognition using the MNIST dataset and give meaningful predictions based on these models. More info about MNIST dataset: https://git-disl.github.io/GTDLBench/datasets/mnist_datasets/
- A GUI will be available for the user input and showing the output. However, a good project will also enable the user to set parameters/hyper-parameters, choose one of the applicable models, visualise the results, etc.

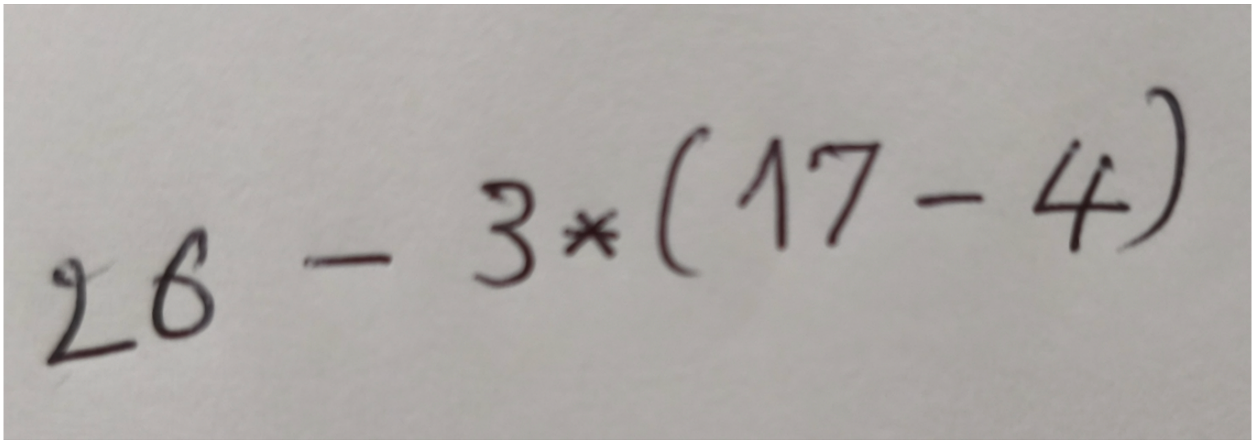
System requirements

- For image acquisition, there should be options to allow both the *automatic creation of the image of a number from a folder of images of individual digits* and *loading the input image from an image file*.

- **Basic Requirement:** The system can follow the following overall methodology to tackle this problem:
  1. (**Task 1**) *Image Preprocessing:* Image preprocessing may involve the following: resizing the image to match with the format recognisable by your ML model, grayscaling (and, perhaps, binarization). **Note:** The preceding list of tasks are examples of what image preprocessing may involve. Your project may choose different image preprocessing techniques as long as it works for your project. *For example*, you may not need binarization and simply use the grayscale value for each pixel of the image.
  2. (**Task 2**) *Image Segmentation*: Image segmentation is the process of partitioning an image into separate parts (e.g., an image of a number consists of the images of the digits forming that number, or an image of a handwritten text consists of the images of the words forming that text, etc.) By recognising each sub-image containing a digit or a word, the entire image can be recognised.
  3. (**Task 3**) *Machine Learning (ML) model representation and training*: The images of each single digit can be classified using a ML model. This is one of the basic tasks in ML. Your task is to learn about different approaches. While Convolutional Neural Networks (CNNs) are one of the most established models, the students are required to learn about different techniques to allow them to compare the performance of different techniques and/or models. For this task, you are encouraged to use libraries such as TensorFlow, Keras or Pytorch.
  4. (**Task 4**) *Evaluation and test your solution*: The performance of different techniques/models should be comprehensively tested and evaluated. The evaluation should be performed on both the single-digit images and images of multi-digit numbers.

- **Extension.Option1:** Can the technique and the model you have developed be extended to perform handwritten text recognition where the text can contain any characters (alphabetical and numerical)?

- **Extension.Option2:** Can the technique and the model you have developed be extended to perform recognition of simple arithmetic expression containing the digits 0 - 9 and also +, –, ∗, ÷, (,), and then calculate the result. For instance, an image may look like this:



Project requirements

- Source code maintained on Git based VCS (Github/Bitbucket/GitLab/…). You must provide read-only access to the tutor/lecturer
- Running illustrative demo of a working prototype (please refer to **Marking Scheme** for details on functionality that needs to be implemented)
- Project report (8-10 pages) that includes the following sections
  - Cover Page (with team details) and a Table of Contents (TOC)
  - Introduction
  - Overall system architecture
  - *Data preprocessing*: What are the techniques that have been investigated and how do they compare? What is the selected technique?
  - *Image segmentation*: What are the techniques that have been investigated and how do they compare? What is the selected technique?
  - *Handwritten Digit Recognition*: What are the techniques that have been investigated/experimented with and how do they compare? What is the selected technique?
  - *Overall system functions and performance*: Results of evaluation and testing your system.
  - Some critical analysis of the implementation, and
  - Summary/Conclusion.
- Presentation (video).

Marking Scheme

Requirements	***Mark***
Task 1: Research and experiment with different preprocessing techniques and select the appropriate technique for your project. Implement the chosen technique.	8
Task 2: Research and experiment with different image segmentation techniques and select the appropriate technique for your project. Implement the chosen technique.	8
Task 3: Research and experiment with different ML techniques for image recognition/handwritten digit recognition and implement them to evaluate different methods. Select the best one to integrate into the final solution for the project.	24
Task 4: Integration of all components into a single AI solution for a handwritten number recognition system. Performing comprehensive evaluation and testing of the ML model and the system performance.	10
GUI: A GUI to allow your user to interact with your system and the system to acquire the input image. The image acquisition task is part of this task.	10
Project Report	10
Project Presentation (Video, 6-8 minutes)	10
	80
Research Component (can be done by the whole team, a sub-team, or an individual) There are many potential extensions in this project. **Extension (Option 1)** and **Extension (Option 2)** are two examples but there are many more. Choose one and get your tutor's approval then complete it very well.	Up to 20
	100/100
You need to follow good programming practice (e.g., well-designed, well-structured codes with clear and helpful comments). Failure to do so get penalty.	**Up to -20**
You need to demonstrate the progress you make every week to your tutor by following a clear project plan the team have agreed with your tutor after the team has been formed. Failure to do so will get a penalty.	**Up to -80**

NOTE: Individual marks will be proportionally adjusted based on each team member's overall contribution to the project as indicated in the '*Who did what*' declaration.

To enforce the weekly progress updates and also to provide a project schedule, the project team will be required to apply the following project plan. Note that **Tasks 1**, **2**, **3** and **Task GUI** can be done in parallel. **Task 4** can only be done after **Tasks 1**, **2**, and **3** have been completed.

Weeks	Activities	Deduction
1-2	Students form team and select their project topics	NA
3-4	Every student needs to work on a subtask. The team can work on **Task 1** and **Task 2** in this sprint (2 sub-teams). Each sub-team should investigate at least 2 different techniques for their assigned task. Then, each sub-team should proceed to implement at least 1 technique to solve its Task. We expect that every student will contribute in the forms of code and/or documents AND **commits to Github**.	Up to -16
5-6	Every student needs to work on a subtask to either complete the implementation for **Task 1**/**Task 2** or the research for **Task 3**. Since Task 3 is the most important one in this project, every student must be involved in the research and implementation for Task 3. The team will need to work on different techniques/models for Task 3. Each student will need to demonstrate their research and experimentation in **Task 3**. We expect that every student will contribute their research and/or development in the forms of code or documents or both AND **commits to Github**.	Up to -16
7-8	Every student needs to work on a subtask for implementing the ML models for digit recognition. The team will need to perform comprehensive evaluation and comparison of different techniques/models that have been implemented. The team is required to produce the evidence to show that the selected model is the best one among those that the team had implemented. We expect that every student will contribute to the development and experimentation in the forms of code and/or documents AND **commits to Github**.	Up to -16
9-10	Every student needs to work on a subtask for completing or extending the project. The subtasks include: Developing the GUI for the project; Integrating all components into a single functional system; Testing the code with many test cases; Writing documentation for the software. Some students who are aiming for D/HD will be working on the extension options mentioned above. We expect that every student will contribute to the project in the forms of code and/or documents AND **commits to Github**.	Up to -16
11-12	Some students may have completed the project at this stage with clear evidence of their contributions. Some other students may be completing their extensions for the project or working on the research components. Some students simply do extra testing to ensure the quality of their software. Most students are working on the project report and video presentation. We expect that students will clearly demonstrate their contributions in the forms of code and/or documents and/or videos.	Up to -16

Submission

At least one member of the team must submit the entire project (code + report) as a .zip file to Canvas by 11:59pm of 02/11/2025. Create a single zip file with your code and a working version of your system. Standard late penalties apply – 10% for each day late, more than 5 days late is 0%.

You must also provide your tutor read only access to your git repository within 1 week of forming teams.

The video **(6-8 minute duration)** should be submitted to Canvas by at least one member of the team by 11:59pm on Tuesday 04/11/2025.