Applied Machine Learning Systems ELEC0134 (24/25)

Assignment

General Overview

The AMLS assignment comprises individual code writing, training and testing models on data, and an individual report in the form of a conference paper and (optionally) supplementary material. You are allowed to discuss ideas with peers, but your code, experiments, and report must be done solely based on your own work.

Assignment summary

1. The assignment leverages elements covered in:
    a. The AMLS lectures,
    b. The AMLS lab sessions, and
    c. Relevant research literature associated with machine learning systems.

    The assignment involves the realisation of various machine learning models on specified datasets. You are expected to go through the data, analyse it and/or pre-process it as necessary.
2. Using your ML knowledge acquired in the lectures and the labs, design solutions for each task described in the section *Assignment Description* below. You should also search the relevant literature for additional information, e.g., papers on state-of-the-art methods in machine learning.
3. Implement your solution in Python. Please note that the weekly exercises of the module will be based on Python, so you are encouraged to use this programming language too.
4. Write a report summarising all steps taken to solve the tasks, explaining your model and design choices. In addition, in the report, you should also describe and analyse the results obtained via your experiments and provide accuracy prediction scores on unseen data. Please refer to Report and Code Format and Marking Criteria section for more details about the report.

Goal of the assignment

- To further develop your skills and understanding of machine learning systems.
- To further develop your programming skills.
- To acquire experience in dealing with real-world data.
- To develop good practice in model training, validation and testing.

- To read state-of-the-art research papers on machine learning systems and understand the current challenges and limitations.
- To develop your writing skills by presenting your solutions and findings in the form of a conference paper.

Assignment Description

For this assignment, you will benchmark the performance of **two** machine learning models on a single medical image dataset from MedMNIST. The goal is to understand how model capacity, data augmentation, and training budget influence performance across both classical and deep learning approaches.

Dataset

BreastMNIST

Description: Grayscale ultrasound images of breast tumors (28×28 pixels).

Task: Binary classification — Benign vs. Malignant.

Use Case: Suitable for medical image classification and cancer detection.

Reference: Yang, J., Shi, R., Wei, D. et al., MedMNIST v2 – A large-scale lightweight benchmark for medical image classification.

The datasets are accessible at https://medmnist.com/.

Please obtain the training, validation, and test datasets in accordance with the original dataset. Feel free to use a portion of the training and validation datasets for model training. However, it is important to utilize the entire test dataset for conducting your testing and include the testing results in your report. You are welcome to use the machine learning model and feature extractor, if applicable, that you deem appropriate. The choice of models ranges from basic models such as SVM and Random Forest to neural networks, and more, depending on your preferences.

Goal

On the dataset, benchmark **two** distinct models and analyze how their performance changes with:

• Model capacity (complexity)

• Data augmentation

• Training budget (number of epochs, number of samples, or computational cost)

Requirements

Implement and train **one model from each category** (**two** models in total):

Category A:

1. Linear or Kernel SVM
2. Random Forest or XGBoost
3. k-Nearest Neighbors (k-NN) or Logistic Regression

Category B:

1. Neural Networks (ResNet)
2. Convolutional Neural Network (CNN)

Feature Pipelines (for Classical Models)

Use appropriate preprocessing and feature extraction steps (e.g., flattening, PCA, HOG features, normalization). Compare performance between raw and processed features.

Data Augmentation

Include at least two augmentation techniques suitable for medical images, such as:

• Random rotations, flips, or shifts

• Gaussian noise or blurring

• Brightness or contrast adjustments

Evaluation

• Use metrics appropriate to your dataset:

- Accuracy, precision, recall, F1-score for binary tasks (BreastMNIST)

• Discuss model complexity, overfitting behavior, and sensitivity to augmentation.

Report and Code Format, and Marking Criteria

Report format and template

We provide both latex and MS word templates in ***AMLS_assignment_kit*** (see Moodle). The instructions for each part are detailed in the template. For beginners in latex, we recommend overleaf.com, which is a free online latex editor.

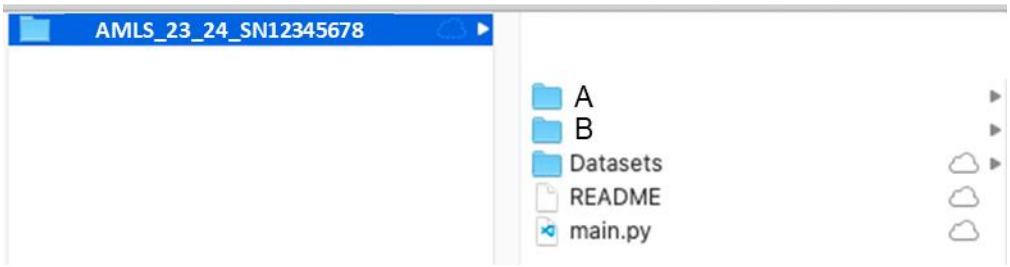
Your report should be no longer than **8 pages** (including the reference). You are allowed to append additional supplementary material to your report of up to additional **4 pages**.

Once you finish your report, please export it into a PDF document and name it with the following format (Using your SN number):

Report_AMLS_25-26 _SN12345678.pdf

Code criteria

- **Project Folder Structure**:
  - Name your project folder as **AMLS_25_26_SNXXXXXX** (replace "XXXXXX" with your student number). Organize your code into modules, following this structure:
    - **Folders**:
    - **Code:** Contains code files related to the project
      - One folder for model A
      - One folder for model B
    - **Datasets**: Leave empty for submission; it will be populated during assessment.
    - **Files**:
    - **main.py**: The main script to run the project.
    - **README.md**: Contains a project overview, file descriptions, and setup instructions.
  - This organization is essential for efficient assessment and ensures clarity in your project layout.



- Keep **'Dataset'** folder empty while submitting your code. Use this folder for your programming assignment only. If you need to pre-process the dataset, do not save the intermediate results or pre-processed dataset. Your final submission must directly read the files we provided.
- When assessing your code, we will copy-paste the dataset into this folder. Your project should then look like this:
  - AMLS_25-26_SN12345678
    - Code
    - Datasets (leave empty)
      - BreastMNIST
    - main.py
    - README.md

- Pre-trained models (especially for deep learning models) are allowed to be saved in the folder for each task.
- The **README** file should contain at least the following:
    - a brief description of the organization of your project;
    - the role of each file;
    - the packages required to run your code (e.g. numpy, scipy, etc.).

    The recommended format for **README** file is markdown (**.md**). **.txt** is acceptable too.
- Your project should be runnable from the terminal with the command '**python main.py' .**
- You are NOT going to upload your code and dataset to Moodle. Please refer to Submission section for more details.

Marking scheme

The mark will be decided based on both the **report** and **corresponding code**. In particular, we will mark based on following scheme:

REPORT	**80%**	**CORRESPONDING CODE**	**20%**
Abstract	5%		
Introduction	5%		
Literature survey	5%		
1. Description of models(Use flow charts, figures, equations etc. to explain your models and justify your choices) 2.Discussion of selected feature pipelines 3. Explanation of augmentation techniques	10% (Model A)		
	10% (Model B)		
Implementation (the details of your implementation, explain key modules in your code.)	10% (Model A)	Correct implementation	10%
	10% (Model B)	Correct implementation	10%
Experimental Results and Analysis	10% (Model A)		
	10% (Model B)		
Conclusion	5%		

It should be noted that – whereas we expect students to develop machine learning models delivering reasonable performance on the tasks – the assessment will not

be based on the exact performance of the models. Instead, the assessment will predominantly concentrate on how you articulate about the choice of models, how you develop/train/validate these models, and how you report/discuss/analyse the results.

Submission

- **Deadline**: Please see the ELEC0134 AMLS I Moodle page.
- **Report submission**: you should only submit your report on Moodle:
- **Code submission**: You must include in your report a link to your code in a repository that is publicly accessible. (e.g., GitHub, public Dropbox or Google Drive link, or similar).

  You are encouraged to use GitHub to save and track your project as we expect to see you progress your assignment gradually. Use your UCL GitHub account (or create an account) to start a git repository named **AMLS_assignment25_26/**
  Make sure to back-up your code on the git repository regularly and keep your repository private so it is not viewable by other students. Changes made after the assignment deadline will not be taken into account. The code should be well documented (i.e., each class and function should be commented) and an additional README.md file containing instructions on how to compile and use your code should be created in the repository. We reserve the right to test the code and we may ask you to provide us with your GitHub commit history evidencing how you gradually built and tested your solution.