Applied Machine Learning Systems ELEC0134 (20/21)

Assignment

General Overview

The AMLS assignment comprises individual code writing, training and testing 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 tasks on a provided dataset. 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 your preferred programming language, e.g., MATLAB, Python, C/C++, Java, etc. However, please note that the weekly exercise 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 programming skills.
- To further develop your skills and understanding of machine learning systems.
- 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

Datasets

We provide two datasets which are designed specifically for this assignment and contains pre-processed subsets from the following datasets:

1. **CelebFaces Attributes Dataset (CelebA)**, a celebrity image dataset (S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: A Deep Learning Approach", in *IEEE International Conference on Computer Vision (ICCV)*, 2015)
2. **Cartoon Set**, an image dataset of random cartoons/avatars (source: https://google.github.io/cartoonset/).

The datasets you are going to use in this assignment are:

1. **celeba**: A sub-set of CelebA dataset. This dataset contains 5000 images. It is going to be used for task A1 and A2.
2. **cartoon_set**: A subset of Cartoon Set. This dataset contains 10000 images. It is going to be used for task B1 and B2.

The datasets can be downloaded via following link:

https://drive.google.com/file/d/1wGrq9r1fECIIEnNgI8RS-_kPCf8DVv0B/view?usp=sharing

A separate test set will be provided one week before the deadline.

Tasks

The machine learning tasks include:

A. **Binary tasks (celeba dataset)**
   - A1: Gender detection: male or female.
   - A2: Emotion detection: smiling or not smiling.

B. **Multiclass tasks (cartoon_set dataset)**
   - B1: Face shape recognition: 5 types of face shapes.
   - B2: Eye color recognition: 5 types of eye colors.

You should design separate modes for each task and report training errors, validation errors, hyper-parameter tuning. You are allowed to use the same model/methodology for different tasks, but you must explain the reason behind choices. If you tried several models for one task, feel free to show them in your code and compare the results in the report.

Report and Code Format, and Marking Criteria

Report format and template

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

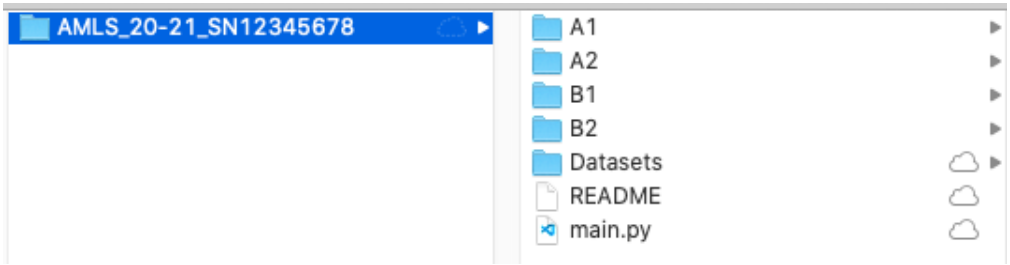
Your report should be no longer than **8 pages** (including the reference). You are allowed to append an additional supplementation material to your report with no longer than **4 pages**.

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_20-21 _SN12345678.pdf

Code criteria

You should write your code in modules and organize them in the following fashion:



- Keep **'Dataset'** folder empty while submitting your code. Use this folder for your programming 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. Then, your project should look like:
    - AMLS_20-21_SN12345678
        - A1
        - A2
        - B1
        - B2
        - Datasets
            - cartoon_set
            - celeba
            - cartoon_set_test
            - celeba_test
        - main.py
        - README.md
- The '**A1**', '**A2**', '**B1**' and '**B2**' folders should contain the code files for each task.
- Pre-trained models (especially for deep learning models) are allowed to be saved in the folder for each task.
- The **README** file should contain:
    - 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 recommend format for **README** file is markdown (**.md**). **.txt** is acceptable too.
- We should be able to run your project via '**main.py**'. An example structure of '**main.py**' has been provided for your reference.

- You are NOT going to upload your code and dataset to Moodle. Please refer to Submission session 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		60%	CORRESPONDING CODE	40%
Abstract		5%		
Introduction		7%		
Literature survey		15%		
Description of models (Use flow charts, figures, equations etc. to explain your models and justify your choices)	Task A1	2%		
	Task A2	2%		
	Task B1	2%		
	Task B2	2%		
Implementation (the details of your implementation, explain key modules in your code.)	Task A1	3%	Correct implementation	7%
	Task A2	3%	Correct implementation	7%
	Task B1	3%	Correct implementation	7%
	Task B2	3%	Correct implementation	7%
Experimental Results and Analysis	Task A1	2%	Reasonable results	3%
	Task A2	2%	Reasonable results	3%
	Task B1	2%	Reasonable results	3%
	Task B2	2%	Reasonable results	3%
Conclusion		5%		

It should be noted that – whereas we expect students to develop machine learning models delivering reasonable performance on tasks A1, A2, B1 and B2 – 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**: See Moodle.
- **Report submission**: you should only submit your report on Moodle:
- **Code submission**: You must include a link to your code in a repository that is publicly accessible in your report, but the link is hidden (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_assignment20_21/

  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.