Applied Machine Learning Systems ELEC0132 (18/19): 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:

i. The assignment leverages elements covered in:
   1. the AMLS lectures,
   2. the AMLS lab sessions, and
   3. 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 preprocess it as necessary.

ii. 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 for image classification.

iii. Implement your solution in your preferred programming language, e.g., Matlab, Python, C/C++, Java, etc. However, please note that the labs of the module will be based on Python.

iv. 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.

Goals 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

Dataset

A dataset with 5000 labeled images is provided for model training, validation and testing. It is designed specifically for this assignment and consists of 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/).

Tasks

A. Some noisy images have been added in the dataset described above (mainly images of natural backgrounds) that you are expected to detect and remove from the training data using an algorithm (i.e., not manually).

B. You should decide how to divide the dataset into training, validation and testing subsets, and provide some justification for your partition.

C. You must train ML models to perform the following classification tasks:

Binary tasks

1. **Emotion recognition:** smile or no_smile.
2. **Age identification**: young or old.
3. **Glasses detection:** with or without.
4. **Human detection**: real or not (i.e. real human or cartoon/avatar).

Multiclass tasks

5. **Hair colour recognition:** blond, ginger, brown, grey, black, bald

Note: All the images are of the same format and size. Further information can be found on the dataset download file accessible at https://drive.google.com/open?id=1NgP2jQakFHibIhpevDLshodWw-L52yXi.

Material Preparation and Submission Instructions

Code

Use your UCL github account (or create an account) to start a git repository named AMLSassignment/. Make sure to backup 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.

Inference on the Test Set

Create one .csv file per task (named task_X.csv, where X is the classification task number) containing predictions on the given test data set. The format should be the following:

<average inference accuracy>
<name of test file1>,<prediction for file1>
<name of test file2>,<prediction for file2>
…

That is, the files contain two comma-separated columns and on each row the name and prediction of each test data file. The very first row contains the average inference score: how accurate you believe your predictions are. See example file task_1.csv for reference.

Paper

The paper must include the following sections: abstract, introduction, algorithmic and implementation descriptions, experimental validation, conclusion and references. Your final report must be in the PDF format of an CVPR-style paper (http://cvpr2018.thecvf.com/submission/main_conference/author_guidelines). It must not exceed 6 double-column single-spaced pages including figures and tables, but excluding references (which have to appear on separate pages). In addition, you are allowed up to 4 pages for supplementary material (such as methodology clarifications, additional results, other tests, etc.)

Submission Instructions

Your assignment must be submitted in the AMLS Moodle page as a single .zip file no later than **11:59pm, Jan 7th, 2019** via the Moodle submission page of the module https://moodle-1819.ucl.ac.uk/mod/turnitintooltwo/view.php?id=836603. In your report, you must also include a link to your code in a repository that is publicly accessible, but the link is hidden (e.g., public Dropbox or Google Drive link, or similar). Your code **must** also include the CSV files mentioned in the *Inference on the Test Set* section, such that we can independently confirm the inference accuracy of your solution.

Assessment Criteria

Overall, each report will be judged based on its quantitative and experimental quality, the qualitative description of the proposed methods, and the volume of work that was carried out to produce the results. The inference accuracy achieved on the test set will also be taken into account, but only as a secondary criterion.

In order to achieve a PASS mark (40%-59%), your report should contain the following sections/material:

- **Abstract**. This section provides a brief overview of the methodology/results presented in the report.
- **Introduction - Problem Statement**. This section introduces the problem, along with the description of the dataset associated with the problem. The dataset description summarises the data (content, size, format, etc.) and describes any *data preprocessing* that was applied.
- **Proposed Algorithms.** This section describes the algorithmic approach used to solve the problem. You may opt to use a single learning algorithm to solve the problem or multiple ones, but bear in mind there are page limitations and that you should explain your rationale behind your choices. That is, the algorithmic description must detail your *reasons for selecting a particular model*.

- **Implementation**. This section must provide the name and use of *external libraries* (if any) and explain how the *model parameters* were selected. In particular, this section should include a thorough discussion on the training convergence and stopping criterion (it is recommended that learning curves graphs be used to this effect).
- **Experimental Results**. This section describes and discusses your results, comparing them to other approaches in the literature or variations of your ML solutions. Additionally, this section should include *accuracy prediction scores on a separate test dataset, provided by the module organisers, but not used during your training and validation process*.
- **Conclusion**. This last section summarises the findings and suggests directions for future improvements.

In addition, the code you submit must:

- Be **readable** and **well documented**. Each class and function should be accompanied by comments describing their use. Additionally, any block of code that implements a complex part of a function should be commented.
- **Compile** and **run**. You should provide a README file which details instructions on how to compile and run your code using the data set.

In addition to the above points, for a GOOD PASS (60%-69%) or DISTINCTION (70%+) mark, the following work is expected:

- Add a "**Related Work**" section to your report that summarises the latest research on the topic, discussing the merits / disadvantages of the different approaches.
- Implement and compare **more than one model** for solving the tasks and add **relevant discussion** comparing the results. Some of the models can be pretrained models from literature.