Assignment 1B

CAB420, Machine Learning

This document sets out the two (2) questions you are to complete for CAB420 Assignment 1B. The assignment is worth 10% of the overall subject grade. All questions are weighted equally. Students are to work individually. Students should submit their answers in a single document (either a PDF or word document), and upload this to TurnItIn.

Further Instructions:

1. Data required for this assessment is available on blackboard alongside this document in *CAB420_Assessment_1B_Data.zip*. Please refer to individual questions regarding which data to use for which question.

2. Answers should be submitted via the TurnItIn submission system, linked to on Blackboard. In the event that TurnItIn is down, or you are unable to submit via TurnItIn, please email your responses to `cab420query@qut.edu.au`.

3. For each question, **a concise written response** (approximately 2-3 pages) is expected. This response should explain and justify the approach taken to address the question (including, if relevant, why the approach was selected over other possible methods), and include results, relevant figures, and analysis. **Python Notebooks, or similar materials will not on their own constitute a valid response to a question and will score a mark of 0.**

4. Python code, including live scripts or notebooks (or equivalent materials for other languages) may optionally be included as appendices. **Figures and outputs/results that are critical to question answers should be included in the main question response, and not appear only in an appendix**.

5. Students who require an extension should lodge their extension application with HiQ (see `http://external-apps.qut.edu.au/studentservices/concession/`). Please note that teaching staff (including the unit coordinator) cannot grant extensions.

Problem 1. Training and Adapting Deep Networks. When training deep neural networks, the availability of data is a frequent challenge. Acquisition of additional data is often difficult, due to logistical and/or financial reasons. As such, methods including fine tuning and data augmentation are common practices to address the challenge of limited data.

You have been provided with two portions of data from the Street View House Numbers (SVHN) dataset. SVHN can be seen as a 'real world' MNIST, and although the target classes are the same, the data within SVHN is far more diverse. The two data portions are:

1. A training set, `Q1/q1_train.mat`, containing $1,000$ samples total distributed across the 10 classes.
2. A testing set, `Q1/q1_test.mat`, $10,000$ samples total distributed across the 10 classes.

These sets do no overlap, and have been extracted randomly from the original $SVHN$ testing dataset. Note that the training set being significantly smaller than the test set is by design for this question, and is not an error.

Your Task: Using these datasets you are to:

1. Train a Linear One vs One SVM (with $C = 1$), on the provided abridged SVHN training set.
2. Design/select a DCNN architecture and using this one architecture:
   (a) Train a model from scratch, using no data augmentation, on the provided abridged SVHN training set.
   (b) Train a model from scratch, using the data augmentation of your choice, on the provided abridged SVHN training set.
   (c) Fine tune an existing model, trained on another dataset used in CAB420 (such as MNIST, KMINST or CIFAR), on the provided abridged SVHN training set. Data augmentation may also be used if you so choose.
3. Compare the performance of the four, considering the accuracy, training time and inference time (i.e. time taken to evaluate the models), using the provided test set.

All models should be evaluated on the provided SVHN test set, and their performance should be compared. DCNN architectures and pre-trained models may be taken from lecture examples or practical solutions. Your selection of model may take computational constraints into consideration.

You have been provided with sample python code that will load and prepare the dataset for training for both the One vs One SVM, and DCNN. Sample code also illustrates how you can time a process with Python, and how to resize images and convert them to greyscale if this is required for your approach. Note that any pre-processing (i.e. resizing of images, colour conversion) should be applied for all models (SVM and DCNNs) to enable a fair comparison.

Your final response should include sections that address the following:

- Discussion and justification of design choices for the neural network method (e.g. network design, type of augmentation, data used to pre-train the network), and any other key parameters or considerations relating to model training.
- Discussion of computational considerations that informed the chosen model, including hardware availability (i.e. presence or absence of a GPU, available memory) how these factors influence your choices.
- Comparison between the four trained models (SVM and three DCNNs), which considers both performance, training time, and inference time. Evaluation should us appropriate metrics and/or visualisations to highlight any differences in performance between the models.

Problem 2. Person Re-Identification. Person re-identification is the task of matching an image of a person (a *probe* sample) to a *gallery* of previously seen people. The problem can be seen as a ranking or retrieval task, in that the *gallery* samples are ranked based on their similarity to the *probe* samples. Like many biometrics tasks, a common approach involves the use of dimension reduction techniques such as PCA and/or LDA, or deep learning and Siamese networks, to learn a compact sub-space in which samples can be compared. Ideally, this sub-space will be such that samples that belong to the same subject will lie close to one another.

Person re-identification (and performance for other retrieval tasks) is commonly evaluated using Top-N accuracy and Cumulative Match Characteristic (CMC) curves. Top-N accuracy refers to the percentage of queries where the correct match is within the closest N results, and is measured by ranking *gallery* samples based on their similarity to the *probe*, and determining the location of the true match within the ranked list. Ideally, the top result (i.e. the closest *gallery* sample to the *probe*) will be the same subject. A CMC curve plots the top-N accuracy for all possible values of N (from 1 to the number of unique IDs in the dataset).

You have been provided with a portion of the Market-1501 dataset [1] (see `Q2/Q2.zip`, a widely used dataset for person re-identification. This data has been split into two segments:

- `Training`: consists of the first 300 identities from Market-1501. Each identity has several images. In total, there are $5,933$ colour images, each of size $128x64$.

- `Testing`: consists of a randomly selected pair of images from the final 301 identities. All images are colour, and of size $128x64$. These images have been divided into two directories, `Gallery` and `Probe`, with one image from each ID in each directory.

Using the `Training` dataset, a model to extract a compact representation of a sample can be trained. The resultant transform can then be applied to the `Testing` set to transform samples to a lower-dimensional representation, at which point samples can be matched. The testing set is broken into `Gallery` and `Probe` sets. To match samples, each samples in the `Probe` set can be compared to each image in the `Gallery` set, and based on the distance between pairs of probe and gallery samples the most similar instances can be identified.

Your Task: Using this data, you are to:

1. Develop and a **non-deep learning** method using one of the dimension reduction methods covered in Week 6 for person re-identification. The method should be evaluated on the test set by considering Top-1, Top-5 and Top-10 performance. A CMC (cumulative match characteristic) curve should also be provided.

2. Develop and evaluate a **deep learning based** method for person re-identification, using one of the methods covered in Week 7. The method should be evaluated on the test set by considering Top-1, Top-5 and Top-10 performance. A CMC (cumulative match characteristic) curve should also be provided.

3. Compare the performance of the two methods. Are there instances where the non-deep learning method works better? Comment on the respective strengths and weaknesses of the two approaches.

You have been provided with sample python code to:

- load all images, and to transform them into a vectorised representation for non-deep learning methods;
- resize images and transform the images to greyscale, which you may or may not wish to use;
- build image pairs or triplets for use with metric learning methods;
- plot a CMC curve given a set of distances between gallery and probe samples.

Your final response should include sections that address the following:

- What pre-processing (i.e. resizing, colour conversion) you apply to the data and why. Note that you do not need to pre-process the data, in which case you should explain why you are using the data 'as is'.
- Details of the selected approaches, including justification for their selection. Your justification should consider the nature of the problem, constraints imposed by the data, and computational considerations.
- An evaluation that compares the two methods, reporting Top-1, Top-5 and Top-10 accuracy, and a CMC curve. Your evaluation should consider instances where performance differs between the two methods, and comment on the respective strengths and weaknesses (including computational efficiency/runtime) of the two approaches.

References

[1] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. USA: IEEE Computer Society, 2015, p. 1116–1124.