University of Wollongong
School of Computing and Information Technology
CSCI446/946 Big Data Analytics
Spring 2024

Assignment 3
(20 marks)

Aim: This assignment aims to provide students with essential experience in conducting image analytics. In this assignment, you should

- procedure big data analytics by following Big Data Analytics Lifecycle,
- appropriately choose, apply and evaluate models/algorithms and analytics techniques to complete the analysis tasks,
- understand and exploit the knowledge and skills learned in this subject.

Group work: You are to work as part of a group on this assignment. Each group is to work independently from other groups. Grouping is the same as for the previous assignments. All group members are expected to contribute, as equally as possible, to this assignment. All your answers to this assignment must be accompanied with suitable justifications and explanations.
Please plan before starting the assignment, then keep a detailed digital work log and timesheet for each group member. One submission per group only.

Penalties: If a group member fails to make sufficient contribution, the member could be awarded zero marks. Claims of less or no contribution should provide evidence like a work log. Plagiarism of any part in this assignment will result in zero marks being awarded to the whole group.

Background
With the proliferation of digital images and videos, the ability to process, understand, and extract information from visual data has become increasingly critical to Big Data Analytics. Images and videos are among the major contributors to the "volume" in Big Data. Advanced algorithms, especially deep learning models, allow computers to extract knowledge from images. This has numerous applications in areas like surveillance, self-driving cars, medical image understanding, healthcare, sentiment analysis, security, surveillance just to give a few examples.

Incorporating image analysis into Big Data Analytics solutions requires expertise, robust computational resources, advanced algorithms, and often specialized hardware to process the data efficiently. As technology continues to advance, the applications and importance of image analysis within Big Data Analytics will only grow. It is increasingly common to use large pre-trained models as feature extractors thus reducing the amount of data that needs to be processed further. These extracted features can then be used to build a specific model for a new task or for a new set of images. In this assignment you will learn to use features, extracted from pre-trained models, as means to solving a Big Data analytical problem.

Preliminaries
Read through the lecture slides, lab instructions and the recommended readings. Conduct relevant background studies. Review all the lab tasks and study all the sample programs and procedures therein so that you fully understand the techniques and know how to perform them. For this assignment you can use any publicly accessible toolbox or library for Python. Your submission must include the source code file(s) which, when run, would re-create all your results.

About the Datasets
Since it was introduced in 2009 [1], ImageNet has been the gold standard benchmark for evaluating image recognition models. With over a decade of constant refinement and competition, practitioners have achieved vast improvements in performance with the current leaderboards topping out at around 90% top-1 accuracy. See https://paperswithcode.com/sota/image-classification-on-imagenet for a comprehensive breakdown.

With the validation set of ImageNet being the main performance indicator for this decade of refinement, many within the research community suspect that this has led to the learning algorithms, neural network architectures and training regimes overfitting to the particular quirks of the data generation process of these 50000 images. To confirm this hypothesis and ensure that benchmarks capture the generalisation performance of image recognition models, many alternatives to the ImageNet validation set have been proposed. One such alternative is the ImageNetV2 [5] validation set and, of particular interest, the split derived by the "MatchedFrequency" sampling strategy. Hence forward, we will refer to the original validation set of ImageNet as test set 1 and the "MatchedFrequency" split of the ImageNetV2 validation set as test set 2.

NOTE: There exists extensive work on ImageNet and its variants. Copy from any public project will lead to zero mark for Assignment 3.

Practice Task and Preparations
It has been a consistent observation across all fields of deep learning that the learnt features of models trained on similar, or even different tasks, transfer surprisingly well to other tasks. At best, these features can be used directly, by simply training a classification head on the down-stream task and leaving the deep features fixed, and, at worst, they serve as a good starting point for initialising neural networks for further training on the down-stream task. Typically, the features immediately prior to the final classification layer are used but features from any layer of the pretrained network can be used with it being observed that features toward the middle of the network generalise better when the downstream task is very different to that of the pretraining task [4].

For the main task of this assignment, we will be supplying you with already extracted, pretrained features for a very large image recognition model. You can learn how to extract these features yourself. This practice task has been prepared wherein you will have to extract the deep features from a much smaller pretrained model [3] for the 10000 images of test set 2.

The steps for this task are as follows:

- Download all data (**large size, 8.24GB**!) to be used from https://uowmailedu-my.sharepoint.com/:u:/g/personal/leiw_uow_edu_au/Ece1YBv-NfxCs0ZZAiTD2H8Bncovq5_NJqWSq6H7Vf5BGQ?e=Ea8Tnp . This is only accessible to students with UOW emailing account.
- Unzip the download data to prepare for the following tasks.
- Clone the 'https://github.com/huggingface/pytorch-image-models' github repository to your working directory ('git clone https://github.com/huggingface/pytorch-image-models.git').
  – If you do not have git you can install it or just download and unzip the archived repository 'https://github.com/huggingface/pytorch-image-models/archive/refs/heads/main.zip'
- Install the packages listed under 'requirements.txt' and pandas.
  – The easiest way to do this would be using conda like we did in the first lab: 'conda install pytorch torchvision cpuonly -c pytorch', 'conda install -c huggingface safetensors huggingface hub' and 'conda install -c anaconda pandas'
- Use the 'validation.py' file (it is released with this instruction file) to replace the existing one in this directory. Familiarise yourself with some of its contents as this is the main file we will be using (use diff to see the changes).
  – The features are being extracted to a csv in a very inefficient manner for the sake of simplicity but it would generally be advisable to use native pytorch data structures.
- Adapt the validation.py script to extract features from the final layer before the 'head' of the model. Your features should be 176 dimensional not 8624.
- Extract 'imagenetv2-matched-frequency.tar.gz' archive you download in Step 1 and extract it somewhere on your machine.
- Run 'python validate.py /YOUR/PATH/TO/imagenetv2-matched-frequency-format-val/ --device cpu --model efficientformerv2_s0.snap_dist_in1k'
- Extract the archives you download in Step1 that contain 'train_efficientformerv2_s0.snap_dist_in1k.csv' and 'val_efficientformerv2_s0.snap_dist_in1k.csv'.
- Train a classifier of your choice on the dataset in 'train_efficientformerv2_s0.snap_dist_in1k.csv'.
- Evaluate your classifier on 'val_efficientformerv2_s0.snap_dist_in1k.csv' and the features in 'v2_efficientformerv2_s0.snap_dist_in1k.csv' which you created.

Main Task

For the main task of this assignment, you have been supplied with the pretrained features from a very large image recognition model [2] on the ImageNet training set, test set 1 and test set 2 in the files you download in Step 1 above 'train_eva02_large_patch14_448.mim_m38m_ft_in22k_in1k.csv', 'val_eva02_large_patch14_448.mim_m38m_ft_in22k_in1k.csv' and 'v2_eva02_large_patch14_448.mim_m38m_ft_in22k_in1k.csv', respectively.

The steps for this task are as follows:

- Use whatever methods you can to obtain the best classification performance on the two test sets.
  – At this stage you should probably create your own validation set from the training set for hyperparameter tuning.
- Analyse the performance gap on these two test sets and attempt to identify what factors are causing the performance to be lower on 'v2_eva02_large_patch14_448.mim_m38m_ft_in22k_in1k.csv' vs 'val_eva02_large_patch14_448.mim_m38m_ft_in22k_in1k.csv'.
- Test the hypothesis derived from your analysis to refine your classifier.
  – The unattainable ideal is that your refined classifier achieves the same accuracy on both test sets, but it is fine if you don't manage to improve the test set 2 results at all so long as all your analysis and efforts are reported.

Your analysis should focus on the deep features such as identifying which features work well for one test set but not the other, clustering them etc. You can also analyse the content of the images that were used to generate each set of deep features. You can find a "path" to each image as a variable of each sample in the provided CSVs.

1. In the case of V2, you already have a local copy of those images from the previous task so you can view them there.
2. For training and test set 1, you can download those from https://huggingface.co/datasets/ILSVRC/imagenet-1k/tree/main, but it is not recommended as that is around 120GB. Instead, you can use online search tools https://app.activeloop.ai/ and track down the images using the supplied path.
3. In case you cannot track down the ImageNetV2 paper here is a direct link: http://people.csail.mit.edu/ludwigs/papers/imagenet.pdf as it will be important for your analysis.

Whatever you do, DO NOT use any data from either test set when training your classification models as that is very likely to result in data leakage, thereby voiding any results you obtain and the analysis based on those results.

Report

Write a report which details your work on this assignment as follows:

- The design of a Big Data analytics project that follows the Big Data Analytics Lifecycle. Clearly frame the problem. (3 marks)
- Description of how you conducted the "main task". Presentation and analysis of results. (10 marks)
- Incorporate visualizations of data and results. (5 marks)
- Drawing of conclusions, description of what you have learnt from your work on this assignment, and outlook. (2 marks)

The report needs to be well-organized and written in a seamless fashion. It must be the collaborative result of the group. The report must not be a simple concatenation of segments written by individual members of the group. Cite referred articles and programming resources. Do not include code or extracts of code in your report.

References

[1] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE. 2009, pp. 248–255.
[2] Yuxin Fang et al. “Eva-02: A visual representation for neon genesis”. In: arXiv preprint arXiv:2303.11331 (2023).
[3] Yanyu Li et al. “Rethinking vision transformers for mobilenet size and speed”. In: arXiv preprint arXiv:2212.08059 (2022).
[4] Kevin Lu et al. “Pretrained transformers as universal computation engines”. In: arXiv preprint arXiv:2103.05247 1 (2021).
[5] Benjamin Recht et al. “Do imagenet classifiers generalize to imagenet?” In: International conference on machine learning. PMLR. 2019, pp. 5389– 5400.

Submission:

The submission link for Assignment 3 is on the subject’s Moodle site. Only one submission per group. **The submission must be a zip file named “A3GroupX.zip”, where ‘X’ is your group number. The ZIP file must remain under 200 MB in size and contain a report (mandatory) and code (mandatory).** The content of the ZIP file must be a report in .pdf format, and code files in .py

Important:

- The report must be in a single file and in .pdf. The title page must list the full name and student ID of all members in the group. Clearly indicate the contribution (in percentage points) made by each member.
- The report does not have a page limit.
- Marks will be deducted for incomplete or vague descriptions.
- Sufficient, suitable, and legible annotation shall be provided in your code to make it easy to understand. Marks will be deducted for untidy code, code that is difficult to read, code that does not run, or code that does not reproduce the results in your report.

Note:

Failure of your code to run may attract zero marks. Plagiarism of any part in your code, or any part in your report will attract zero marks for this assignment. It is the responsibility of the group to ensure that your submission does not contain plagiarised material. You may be requested to demonstrate and explain your program or explain your answer in the report. Marks are deducted if you are unable to offer an explanation. Marks will be awarded for correct design, implementation, style, completeness, level of comprehension, depth of analysis, clarity, and justification. The assessors will use the UoW Grading guidelines when determining the number of marks.

---- END----