6G7Z1003 High-Performance Computing and Big Data

(1CWK50) HPC TensorFlow Task (50%)

Dr Ryan Cunningham (Lecturer, Data Science)
Department of Computing and Mathematics
Faculty of Science and Engineering
Manchester Metropolitan University

(Assessment Overview)

In this assessment you are tasked with developing an image classification system using Python and TensorFlow, and a publicly available dataset (Caltech-256). You are expected to produce the following deliverables:

1) **Your Python code script or notebook** with detailed comments (or formatted text in a notebook). Your Python code should use Keras data generators to stream images from the provided dataset, build a model, train that model, and provide full testing (all images in the validation set) results in the form of Cross Entropy and Accuracy. Your code should implement your best model as per tasks below. When developing your model, you must:

- Develop your own model – ***do not use a pretrained model***
- Validate on a random 10% of the total number of samples in Caltech-256
- Track all experiments (learning curves, hyperparameters) with TensorBoard

2) **A short report (500-1000 words) on experimentation** with different models and model hyperparameters (including image augmentation), using TensorBoard.

Your report should contain the following sections:

(Introduction) A brief summary on the experiments you have successfully completed, and a discussion and conclusion on your main findings.
(Hyperparameter Tuning with TensorBoard) A brief description of the experiments you have carried out, with justification and interpretation of your main findings. You should include figures showing training curves via TensorBoard. You should also include a table of results, summarising performance of each model/test.
(Optimal Model Evaluation) A brief description of your chosen model with justification for the architecture (plus summary: ***model.summary()***). You should comment on your model's testing performance and speculate on the reasons for this, suggesting possible improvements that could be made. You should include figures of your model architecture via TensorBoard. You should include figures, with accompanying interpretation, showing good and bad predictions from your model.

3) **A short report (500-1000 words) on explainability** of your chosen model (as per task 2 above).

Your report should contain the following sections:

(Introduction) A brief summary on the visualisation methods you have successfully applied, and a discussion and conclusion as to whether these methods were informative as tools to gain insight into your model's decision making.
The following sections should contain a brief explanation of what you did, and some relevant figures with accompanying interpretation:
(Feature Map Visualisation)
(Max Feature of Image)
(Deep Feature Visualisation)
(Saliency Mapping)

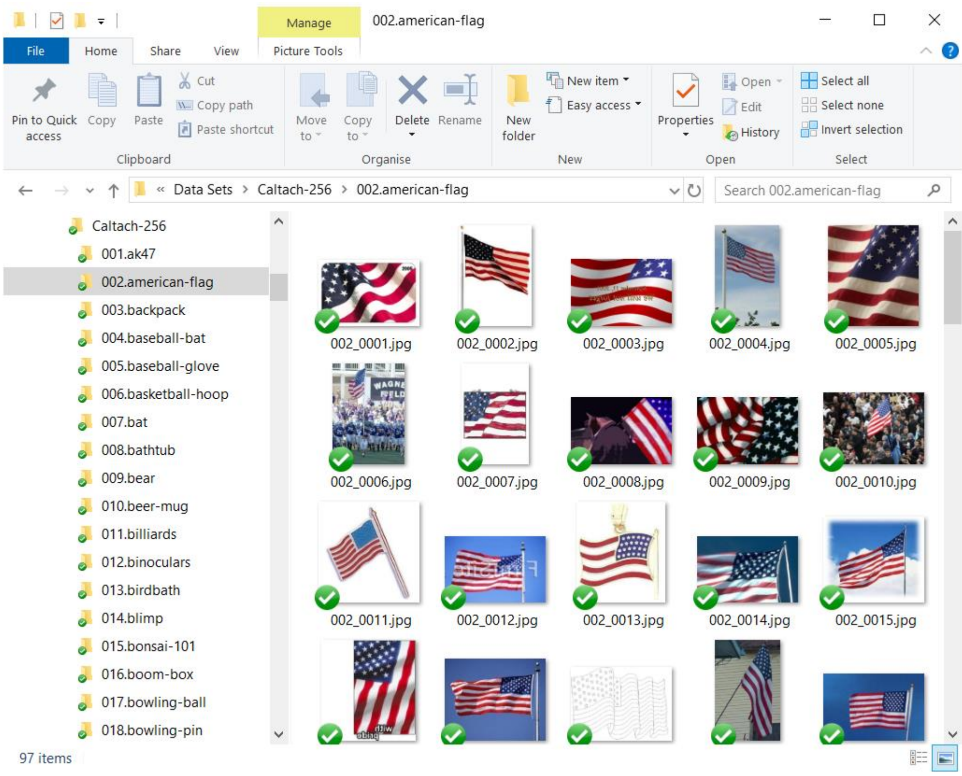
(The Dataset) Caltech–256

This section of the assignment brief is to introduce you to the Caltech-256 dataset which you will use to train and validate your classification model. Read this section carefully and refer to the original paper for more details: https://authors.library.caltech.edu/7694/.

The Caltech 101 data set consists of a total of 9,146 images, split between 101 different object categories, as well as an additional background/clutter category. Each object category contains between 31 and 800 images. Caltech-256 is an improvement in that it has a total of 30,607 images, split between 257 different object categories (256 objects, plus a 'clutter' category), where the minimum number of images per category has increased from 31 to 80.

(Download) Instructions

On the unit Moodle page, download the caltech-256.zip file and extract it into your working directory. ***Do not modify the contents, or folder structure.*** Once extracted, inspect the folders and images. You should see 257 folders, where the name of each folder represents the category of images within. For example, below you can see images in the 'american-flag' category:



(Important Differences) Caltech-256 *vs* MNIST/CIFAR-10

The first difference is that there are many more categories in Caltech-256 compared with MNIST and CIFAR-10. This discrepancy makes it a more challenging task, but more importantly this should be reflected in the dimensions of the output layer of your model (i.e. one neuron per category). It should become clear that the images in Caltech-256 are high-resolution and have inconsistent aspect ratios, where MNIST and CIFAR-10 contain low-resolution images all of the same 1:1 aspect ratio. This is not a problem, as the Keras data generator will resize any image it loads to the dimensions specified via the ***target_size*** parameter. This will obviously change the aspect ratio of the images, but unfortunately Keras provides no interface to crop images to the correct aspect ratio before resizing. Although this can be achieved by implementing your own generator or preprocessor, ***it is not recommended for this assignment***.

(Testing your Classifier) Expected Performance

The authors of Caltech-256 computed some benchmark metrics which are useful for guidance. The performance measures summarised below (figure 1), struggle to approach 40%. A good deep learning model, trained from scratch (no pretraining), only using this data should achieve at least 40%.

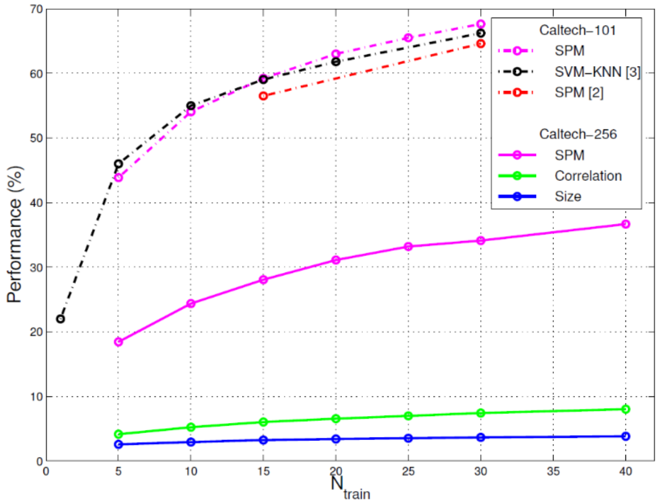


Figure 1. Caltech-101 and Caltech-256 performance summary. The solid lines show the performance of various image classifiers, trained on increasing number of images. The dashed lines are a different dataset and thus should be ignored.

(Tips) For Improving Performance

- Do not down-sample your images to less than 64×64 pixels. Try 128×128 or higher (be wary of training time).
- Add image augmentation (Week 3 – Bonus Tasks)
- Use dropout in each layer. Not too much, else you will cripple the training.
- Use batch normalisation
- Favour depth over number of features
- Try smaller batch sizes (optimal batch sizes are reported at < 5 samples). Be wary of training time though.

Dr Ryan Cunningham, E142, ryan.cunningham@mmu.ac.uk

(TensorBoard) Presenting Your Graph

For your report you should utilise TensorBoard to create a graphic which details your optimal model architecture. Below is an example of a well-presented graph: convolution, batch norm, dropout and pooling layers have been grouped into blocks using ***name_scopes***. Within TensorBoard, blocks can be expanded (by double clicking on them) to reveal their contents. All blocks have the same structure, but a different number of units per convolution layer, which is reflected in the name of the block (e.g. Block_32 has convolution layers with 32 units). You should organise your graph in a similar way – good presentation of your model will be awarded.

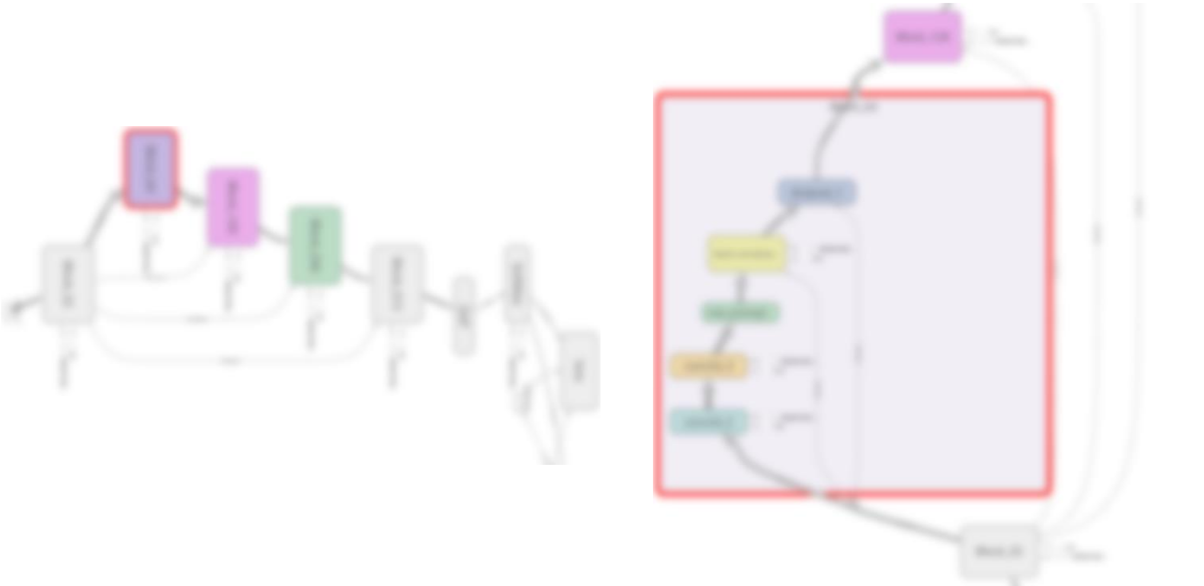


Figure 2. Example TensorBoard graph schematic example. *Left:* shows the structure of the whole network, arranged in blocks. *Right:* One of the blocks has been expanded (by double-clicking on a block) to reveal what kinds of layers are contained within. This graphic has been intentionally blurred to prevent copying of the network structure.