Advanced Vision Mini-Project

January 2022

Abstract

This document describes the Advanced Vision assessed practical mini-project. The main goal is to perform image classification with limited data. We will be using the ImageNet dataset and artificially reduce the number of samples per class. The objective is to obtain the highest accuracy on this. This is part of a popular computer vision workshop and hence a good and novel project can result in a workshop paper! (https://vipriors.github.io/challenges/)

Task Background

Data: The dataset will consist of 1000 classes, with 50 images training samples each. The original Imagenet consists of more than 1M training images, 50,000 validation images and 100,000 test images. We use a subset of this.

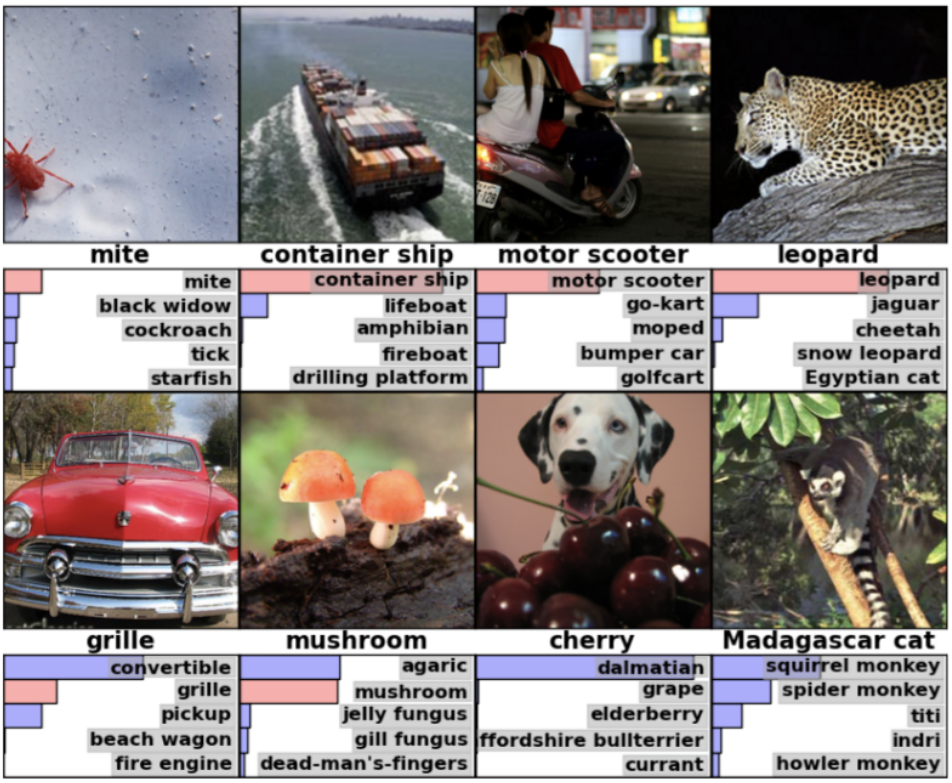
Task: The assignment consists of implementing deep learning approaches to obtain highest top-1 accuracy on this limited-data scenario.

Rules:

1. *Please note that this challenge does not allow using any pre-trained checkpoint, including any pre-trained backbone!* We provide a simple baseline ResNet50 trained end to end on the given samples.
2. This is an open-ended problem, so you have the freedom to choose which approaches you want to try.
3. The majority of the marks will be going to the report and the set of strategies tried.
4. However, you will also be evaluated on your performance on the class leaderboard.
5. You should split the training data into training and validation, but not touch the test data until the inference stage.
6. Feel free to use codebases available on the website, but there has to be changes in terms of the model and structure to obtain any marks. Make sure you reference these codebases as mentioned in the report section of this document.
7. You are supposed to keep the ResNet-50 as a backbone. Adding layers (eg, using ResNet-151) or switching to a different backbone (eg, DenseNet) is not allowed.

Data

In the figure below, you can see examples of various images from the dataset.



Resources

The data for this can be found at:
https://uoe-my.sharepoint.com/:f:/g/personal/s1960707_ed_ac_uk/ErcZ5obzYF5Bh_aKHG21JKMBcx3ZLMuB9wOrt19UAtTW0w?e=TbF7n3
The zip file contains 2 directories for training and testing with 1000 subdirectories each.
For GPU access: Google colab https://colab.research.google.com/
Baseline model: https://drive.google.com/file/d/1EgYFbyJqCmr1uwx0EKW7zKd8pBIBNxcw/view?usp=sharing (this can be used as the pre-trained model which roughly gives 26% accuracy)
You can also redefine your model or use the one defined on https://github.com/VIPriors/vipriors-challenges-toolkit/blob/master/image-classification/resnet_fconv.py to obtain a slightly higher 31% accuracy. This can be downloaded here: https://drive.google.com/file/d/1VqlV10kFDJsSwos1B7HsbHLSjw50GuGq/view?usp=sharing

Classifier Details (ResNet-50)

The baseline is a ResNet-50 model that is not pre-trained as mentioned earlier. Training this from scratch should give roughly 31% accuracy on the given dataset. You should use the same ResNet-50 model as your backbone network and build on that. Remember the higher the accuracy you get the better!

There are several sources for ResNet50 in either Matlab or Python:
matlab: https://www.mathworks.com/help/deeplearning/ref/resnet50.html
and pytorch: https://pytorch.org/docs/stable/torchvision/models.html

or pytorch: https://docs.fast.ai/quick start.html. You are welcome to use other sources.

Loading the baseline weights (ResNet-50) for the model with 26% accuracy

The trained weights are obtained by training the ResNet model with multiple GPUs. Hence loading these weights to the pre-trained model will lead to an error as seen below:

```
RuntimeError: Error(s) in loading state_dict for ResNet:
 Missing key(s) in state_dict: "conv1.weight", "bn1.weight", "bn1.bias", "bn1.running_mean", // and more
Unexpected key(s) in state_dict: "module.conv1.weight", "module.bn1.weight", "module.bn1.bias", "module.bn1.running_mean", "module.bn1.running_var", // an
d more
```

To fix this, add the following lines of code before you load the model with these weights:

```
from collections import OrderedDict
new_state_dict = OrderedDict()
for k, v in state_dict.items():
name = k[7:] # remove module.
new_state_dict[name] = v
model.load_state_dict(new_state_dict)
```

This should help to load the baseline model successfully.

Some General Directions for Solutions

Here are some ideas for things to try. However, you are most welcome to look up approaches of your own. Creativity also matters:

1. Augmentation: While simple augmentation techniques would only improve performance by a small margin, advanced techniques like cutout and mixup will improve performance by a larger margin. Follow this link for a brief tutorial on these methods: https://www.kaggle.com/kaushal2896/data-augmentation-tutorial-basic-cutout-mixup

2. Modifying loss functions: With standard cross entropy loss we obtain the baseline of roughly 45%. What about adding losses in the form of contrastive learning? Follow this link for a brief tutorial on contrastive learning: https://www.kaggle.com/debarshichanda/pytorch-supervised-contrastive-learning

3. You might explore some of the values of the hyperparameters associated with the fully connected layers, such as the number of layers, size of layers, learning rate, etc.

Your Report

Each team writes and submits a single report (10 pages long plus code in an appendix) that describes:

• The components that you selected and connected together.
• A detailed explanation of why you chose these components and an **ablation study** of their effects.
• Learning plots for the classifier.
• A table/plot on some hyperparameters used and their effect.
• Discussion of the class-wise performance.

• As an appendix, add the code that your team wrote. Do not include the code that was downloaded from other web sites, but include a statement about what code was used and where it came from.

Other Comments

- The assignment is estimated to take 30 hours coding/test and 5 hours report writing per person, resulting in a 10 page report plus the code appendix.

- You must do this assignment in teams of maximum 4. Use the following google sheet to enter your student ids. In case you do not have a team, just add your student id and I'll randomly allocate teams.

- **A single, joint, report is to be submitted**. Split the work so that each partner can do most work independently (i.e. split the work rather than duplicate it).

Assignment Submission

The deadline for submission is March 11th at 4pm. Name the submitted PDF file as: <student-number-1> <studentnumber-2><....> 1.pdf. Submit your report in PDF online using Learn.

The proportion of marks are explained in the table:

Issue	Percentage
1. Clear description and motivation of algorithms used, including creativity / innovation	40%
2. Performance on the dataset	25%
3. Clear MATLAB or Python code	5%
4. Discussion of results	30%

Publication of Solutions

We will not publish a solution set of code. You may make public your solution but only 2 weeks after the submission date. Making the solutions public before then will create suspicions about why you made them public.

Good Scholarly Practice: Please remember the University requirement regarding all assessed work. Details about this can be found at:
http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct

Furthermore, you are required to take reasonable measures to protect your assessed work from unauthorised access. For example, if you put any such work on a public repository then you must set access permissions appropriately (generally permitting access only to yourself, or your group in the case of group practicals).

Plagiarism Avoidance Advice

You are expected to write the document in your own words. Short quotations (with proper, explicit attribution) are allowed, but the bulk of the submission should be your own work. Use
proper citation style for all citations, whether traditional paper resources or web-based materials.
If you use small amounts of code from another student or the web, you must acknowledge the original source and make clear what portions of the code were yours and what was obtained elsewhere.
The school has a robust policy on plagiarism that can be viewed here:
http://web.inf.ed.ac.uk/infweb/admin/policies/guidelines-plagiarism .
The school uses various techniques to detect plagiarism, including automated tools and comparison against on-line repositories. Remember: ***a weak assignment is not a ruined career (and may not reduce your final average more than 1%), but getting caught plagiarising could ruin it.***

Late Coursework Policy

Please see university-set guidelines regarding late submission and penalties for the same: https://web.inf.ed.ac.uk/infweb/admin/policies/late-submission-coursework