Assignment for Image Recognition

In the final assignment of the "Deep learning" course, you will solve different problems using deep models.

The goals of the assignment are:

- Develop proficiency in using Tensorflow/Keras for training Neural Nets (NNs).
- Put into practice acquired knowledge to optimize the parameters and architecture of a *feed-forward Neural Net* (ffNN), in the context of an image recognition problem.
- Put into practice NNs specially conceived for analysing images. Design and optimize the parameters of a *Convolutional Neural Net* (CNN) to deal with previous image classification task.
- Train popular architectures from scratch (e.g., GoogLeNet, VGG, ResNet, ...), and compare the results with the ones provided by their pre-trained versions using *transfer learning.*

Database

xView (http://xviewdataset.org/) is a large publicly available object detection data set, with approximately 1 million objects across 60 categories. It contains manually annotated images from different scenes around the world, acquired using the WorldView-3 satellite at 0.3m ground sample distance. There are 846 annotated images in total. For this practice, we divide these annotations into 761 and 85 images for training and testing respectively.

From now on, we discard most categories selecting the 12 most represented classes. For image recognition, we crop previous images using their annotated bounding boxes to extract a subset of objects of interest. In this way, we collected **21377 and 2635 objects for training and testing** respectively. The resulting images are resized to 224x224. Follow the link below to download the classification benchmark "xview_recognition":
https://drive.upm.es/s/4oNHlRFEd71HXp4

Note that all experiments must be done using the same train/test protocol to make fair comparisons. We uploaded both benchmarks into the common /home/*/PROJECT folder at "Magerit" to save bandwidth and disk space in each "Cesvima" account (see ln -s command to create a symbolic link into your workspace).

Practical Assignment

Image Recognition

Design and train Neural Nets (NNs) to deal with the "xview_recognition" benchmark. Compare results obtained using ffNNs and CNNs. You can start your work from the sample notebook provided below, and improve its performance (see ffNN_example.ipynb).

To address this problem, you must decide on:

- Number of layers and number of units in each layer of your NN.
- Optimization algorithm and parameters to train the network.
- Check the evolution of these parameters during the optimization by using a validation subset and decide when to stop training (note that the test data set can only be used to evaluate the model).

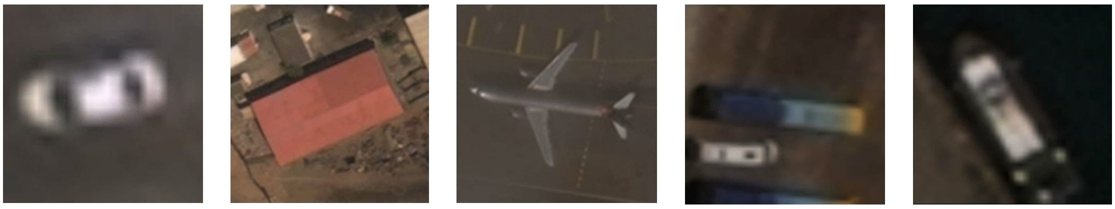
It is also worth mentioning that there is a severe class imbalance. This difference in class frequencies affects the overall accuracy and should be considered when training the model.

Category	Count	Frequency
Cargo plane	635	2.97%
Helicopter	70	0.32%
Small car	4290	20.06%

Bus	2155	10.08%
Truck	2746	12.84%
Motorboat	1069	5.00%
Fishing vessel	706	3.30%
Dump truck	1236	5.78%
Excavator	789	3.69%
Building	4689	21.93%
Storage tank	1469	6.87%
Shipping container	1523	7.12%

Table 1. Class frequencies in our “xview_recognition” train subset.

Compare your architecture and results with other popular architectures (e.g., GoogLeNet, VGG, ResNet, etc.). Additionally, train some of these popular architectures from scratch, and compare the results with the ones provided by their pre-trained versions using transfer learning.



Small car Building Cargo plane Truck Fishing vessel

Figure 1. Sample objects of different categories acquired from the “xview_recognition” benchmark.

Presentation of results

You must prepare a report (.pdf) describing:

- The problems and data sets (briefly).
- The process that you have followed to reach your final solution, including your intermediate results. You must discuss and compare these results properly.
- Final network architectures, including optimization algorithm, regularization methods (dropout, data augmentation, etc…), number of layers / parameters, and performance obtained with your model on the testing data set, including the plots of the evolution of losses and accuracy performances on both data sets.
- It would also be very valuable your feedback on the use of “Cesvima” or “Google Colab" services.

In the submission via Moodle, attach your Python (.py) or Jupyter Notebook (.ipynb) source file, including in the report all results of computations attached to the code that generated them.

The assignment must be done in **groups of 3 students maximum**. Each team must submit one submission before **Tuesday, April 16th, 23:55h.**