Assignment for Image Recognition and Object Detection 25-26

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).
- Experiment with *feed-forward Neural Nets* (ffNNs) by exploring different architectures, optimizers, loss functions, and training strategies in the context of an image recognition problem. Analyse the effect of *regularization techniques* on model generalization and performance.
- Put into practice NNs specially conceived for analysing images. Experiment with custom *Convolutional Neural Nets* (CNNs) to deal with previous image classification task, and compare results with popular model (e.g., GoogLeNet, VGG, ResNet) trained from scratch and using *transfer learning*.
- Experiment with *object detection models*, comparing single-stage and two-stage detectors.

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.

For object detection, we extract cropped images of size 640x640. We focus only on **4 object categories**, and discard cropped images containing fewer than five objects of interest. As a result, we will only use **7606 and 852 images for training and testing** respectively for the image detection task. Follow the link below to download the detection data set "xview_detection": https://drive.upm.es/s/P7nEf3Bygns7tbM

For image recognition, we crop previous images using their annotated bounding boxes to extract a subset of objects of interest. We focus only on **13 object categories**. In this way, we collected **18746 and 2365 images 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/2DDPE2zHw5dbM3G

We also uploaded both to the shared /home/y222/PROJECT folder at Magerit to save bandwidth and disk space (use *ln -s* to create a symbolic link). Note that training annotations are available for download, while test annotations remain private. Test evaluation must be done through Codabench benchmark platform. Only one member of each group must submit to each competition a zip file containing the test results in a json file.

Practical Assignment

Image Recognition

Experiment with different NNs to deal with the "xview_recognition" benchmark using custom ffNNs and CNNs, by exploring architectural choices and regularization techniques. Compare their performance with popular CNN architectures trained from scratch and using transfer learning.

Small car | Building | Cargo plane | Truck | Fishing vessel

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

You can start your experiments from the sample notebook provided below, and improve its performance (see ffNN_example.ipynb).

The Image Recognition challenge 25-26 is hosted on: https://www.codabench.org/competitions/13701/
Participants will have to login on Codabench using the UPM institutional email (@alumnos.upm.es)

Object Detection

In this task, you will take image classification to the next level, by recognizing multiple objects of different classes within a single image. Experiment with different single-stage object detectors (e.g., YOLO, SSD, RetinaNet, etc.) and compare results with two-stage object detectors (e.g., Faster R-CNN, R-FCN, etc.) to deal with the "xview_detection" benchmark.

We recommend the use of the KerasCV or Ultralytics libraries for training object detection models. Note that the computational requirements needed to train an object detector are too much higher than those required for the image recognition task.

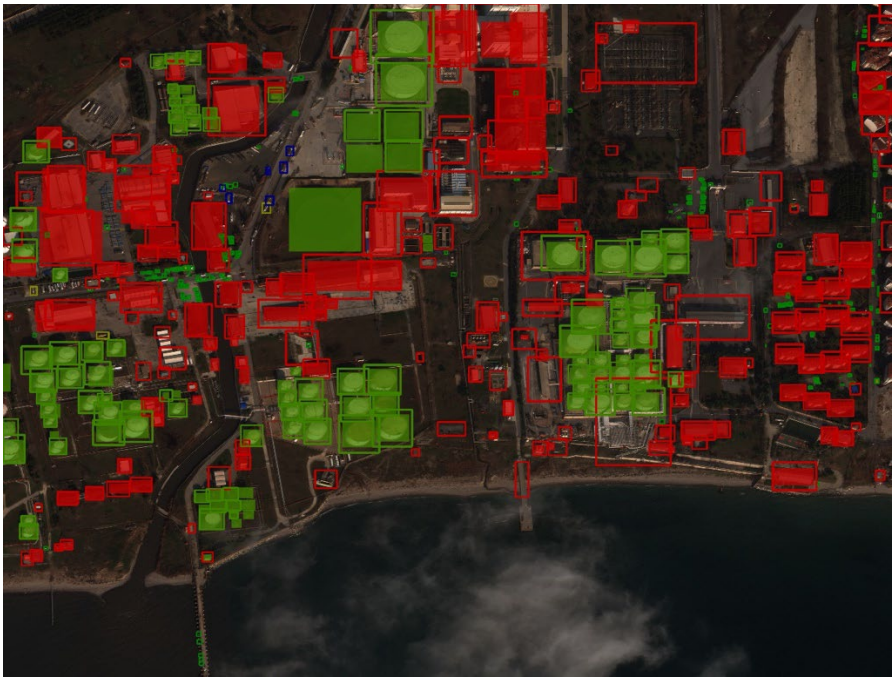

Figure 2. Sample image from the "xview_detection" test subset. The bounding boxes and contours represent predictions and annotations respectively, using a different colour for each category.

You can start your experiments from the sample notebook provided below, and improve its performance (see detector_example.ipynb).

The Object Detection challenge 25-26 is hosted on: https://www.codabench.org/competitions/13702/
Participants will have to login on Codabench using the UPM institutional email (@alumnos.upm.es)

Presentation of results

You must prepare a **report (.pdf)** describing:

- The problems and data sets. It would also be very valuable your feedback on the use of "Cesvima" resources.
- Experiment with at least 3 custom feedforward neural network architectures, 3 custom convolutional neural networks, and 3 popular architectures both trained from scratch and using transfer learning,

considering different strategies such as freezing part of the network or fine-tuning all layers, as well as at least 2 object detection models, including single-stage and two-stage detectors. Note that all experiments must be done using the same train/test protocol to make fair comparisons.

- A detailed discussion of the experimental process, including the analysis of intermediate results obtained while tuning architectures and hyperparameters using a validation subset to decide when to stop training. Include plots of the evolution of training and validation losses and accuracy, as well as evaluation metrics such as confusion matrices, precision-recall curves, or other relevant metrics to properly assess and compare model performance.

In the submission via Moodle, attach your **Jupyter Notebook file (.ipynb)** corresponding only to the final selected model of each phase, including all code and its execution outputs.

The assignment must be done in **groups of 3 students maximum**. Each team must submit one submission before **Monday, April 13th, 2026, 23:55h.**