Assignment for Image Recognition and Object Detection 25-26

In the final assignment of the "Visión por Computador" 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.*
- Compare object detectors based on single-stage and two-stage strategies.

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

Annotations for train are available for download, whereas test annotations will remain private. Test evaluation can be done through Codabench benchmark platform. Only one member of each group must submit to each competition a zip file containing the test results generated in the json file.

Practical Assignment

Image Recognition

Design and train Neural Nets (NNs) to deal with the "xview_recognition" benchmark. Compare results obtained using ffNNs and CNNs. Train popular architectures (e.g., GoogLeNet, VGG, ResNet, etc.) from scratch and compare 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.

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/10658/
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. Build a CNN to deal with the "xview_detection" benchmark. Train popular 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.).

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.

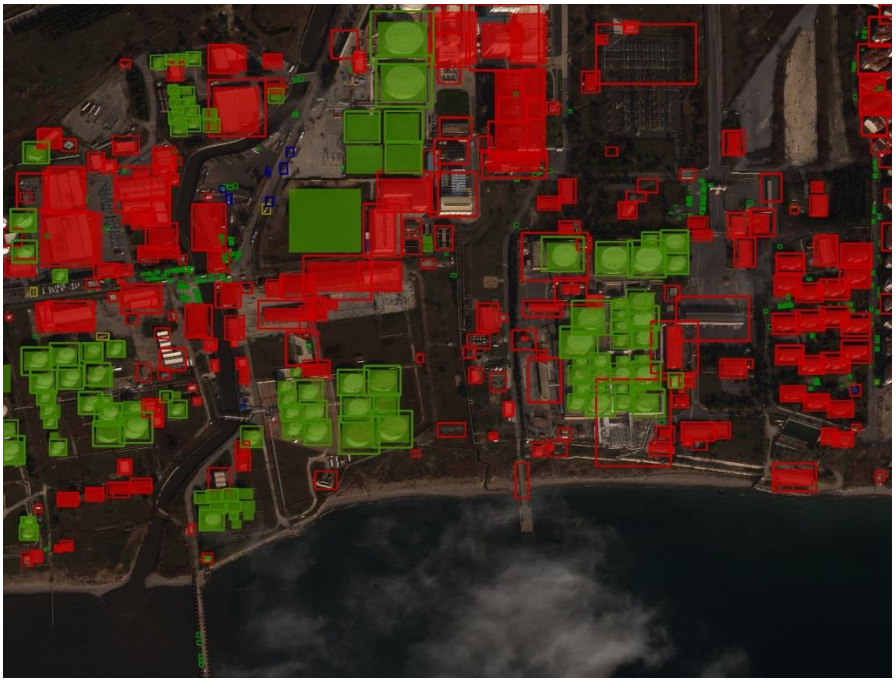

Figure 2. Sample image from the "xview_detection" test subset. The bounding boxes and contours represent predictions and annotations respectively, using a different colour for each category.

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/10661/
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.
- Experiment with at least 3 custom feedforward neural network architectures, 3 custom convolutional neural network architectures, and 3 popular architectures both with and without transfer learning, as well as 2 object detection models.

- Intermediate and final neural network architectures, including optimization algorithm, regularization methods (dropout, data augmentation, etc…), number of layers / parameters, and performance obtained with your models on the testing data set.
- The process that you have followed to reach your final solution on both detection and classification tasks. Check the evolution of hyperparameters (learning rate, scheduler, etc.) during the optimization by using a validation subset and decide when to stop training. Plots of the evolution of training and validation losses and accuracy performances. You must discuss and compare the results properly.

In the submission via Moodle, attach your **Jupyter Notebook file (.ipynb)** with the results of computations attached to the code which generated them.

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