DLCV – Challenge 2
Convolutional neural networks for classifying dogs and cats

In this challenge, your task is to define and train different convolutional neural network models that are able to distinguish between images of dogs and images of cats.

CATS AND DOGS

Part 1 – Data preparation

Download the *Cats and Dogs* dataset[1], available on Moodle. This dataset consists of 2,000 images for training (1,000 dogs and 1,000 cats); and 1,000 images for validation (500 dogs and 500 cats).

Process the *dataset* to obtain new sets for validation and testing. To do this, you should separate the 1,000 images from the original validation set into two new sets, one that will be used for validation and the other one for testing, each with 500 images. After this process, you should have 2,000 images for training, 500 for validation and 500 for testing.

Use *Tensorflow/Keras* functions `keras.utils.image_dataset_from_directory(...)`[2] and `keras.utils.split_dataset(...)`[3] to ease the construction of the *datasets*. In this classification problem you can use a binary classification approach, which may have some implications in the parameterization of the construction of the *datasets* from the file system.

Check the distribution of the classes in the validation and test sets obtained. Each should ideally contain 250 images of dogs and 250 images of cats. Given the pseudo-random processing involved, this ideal distribution may not be reached, but make sure that classes distribution is not too far from the ideal.

---

[1] It is a subset of the images published in: https://www.kaggle.com/c/dogs-vs-cats/data
[2] https://keras.io/api/data_loading/image/
[3] https://keras.io/api/utils/python_utils/#splitdataset-function

Part 2 – Custom convolutional network

Define a convolutional neural network that allows you to classify images into *Dog* or *Cat*. The proposed network architecture, image size, the approach followed (binary or multiclass classification), use of techniques aiming to reduce *overfitting*, as well as other parameterizations, are entirely up to you. Similarly to the previous challenge, it is suggested to use *callbacks* to ensure that you can save and restore the best models obtained during the training processes.

Part 3 – Transfer learning

Use transfer learning to improve the results achieved in Part 2. To do this, you should use two pre-trained CNNs of your choice, from those provided by *Tensorflow/Keras*[4]. The only restriction is that the two networks must come from two different families. Check the documentation to determine whether you need to apply some kind of pre-processing to your images (e.g., normalizing pixel values, resizing images, swapping color channels, etc.).

Part 4 – "Mini-Report"

Write a "mini-report" to briefly describe what was done, as well as to show and comment on the results related to parts 1, 2 and 3 of the challenge. You should also include in the report:

- examples of test set images where your best model failed the classification;
- results on images of dogs and cats that are pets of your group members. If no one in the group has a dog or cat, you could ask somebody else for a dog/cat image or search for images on the internet (what matters is that these images are not present in the *dataset*).

The "mini-report" should be about 5 pages long (without accounting for the cover or annexes).

Delivery and assessment

The delivery of the challenge is carried out through Moodle. One of the group members should submit the work, delivering the **developed *scripts* (or *notebooks*)**, a pdf with the **"mini-report"** and, eventually, files with the best models of the trained networks. The deadline for delivery is **March 30 (Sunday) at 11:59 pm**.

The grade will be on a scale of 0 to 10 and feedback will be provided through Moodle. Part 1 is worth 10% of the challenge grade and parts 2, 3 and 4 are worth 30% each. Late submissions are penalized with one point for each day after the deadline.

In doubt, feel free to contact the teacher in the classes or by email.

---

[4] https://www.tensorflow.org/api_docs/python/tf/keras/applications