Prof. Cláudio Jung

Introduction to Deep Learning for Visual Computing (INF01050 and CMP620)

Programming Assignment 1

Goals: The main goal of this assignment is solve a multilabel image classification problem using a CNN. The students will have to process and prepare the training data, and then implement, train and evaluate a simple CNN with transfer learning to perform image classification.

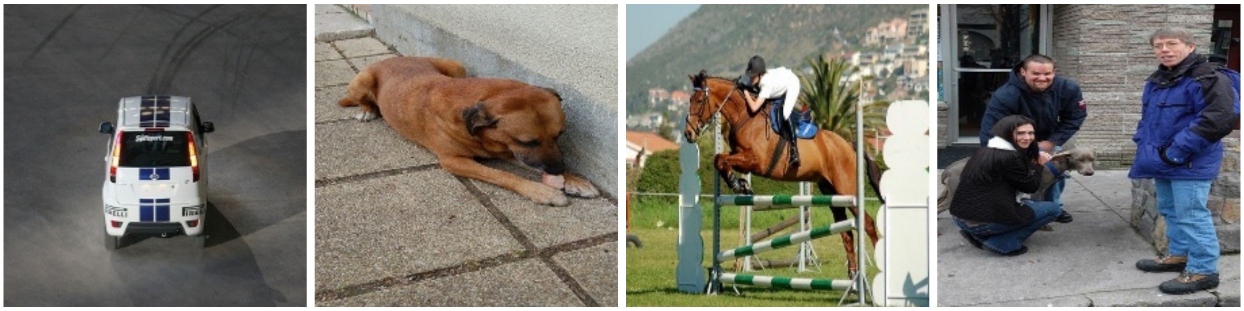
Description of the task

The Pascal VOC dataset contains class-label annotations for several images, such as cat, dog, person, car, etc. These labels **are not** mutually exclusive, since one image might contain instances of different categories. For this assignment, we created a small subset from the original VOC dataset focusing on two categories only: *dog* and *person*. The selected images were resized to a default resolution of 224 × 224 pixels, and annotations are provided (for each image) regarding the the presence or not of instances from the two categories in the image. More precisely, we provide a zip file (`https://www.inf.ufrgs.br/~crjung/fdl/modified_voc.zip`) containing a set of images and a csv file `annotations.csv`.

The csv file contains four fields for each image:

- `image`: contains the name of the image file;
- `dog`: contains a binary label (zero or one) regarding the presence of a dog in the image;
- `person`: contains a binary label (zero or one) regarding the presence of a person in the image;
- `split`: defines if the image belongs to the train or test splits.

An example of images and the corresponding labels for `dog` and `person` is shown Figure 1.



(a) person:0, dog:0 (b) person:0, dog:1 (c) person:1, dog:0 (d) person:1, dog:1

Figura 1: Examples of images in the dataset and the corresponding annotations.

Based on the information provided above, you should:

1. Download the .zip file and extract it.
2. Based on the structure of the provided csv file, organize the data (images and labels) for **training** and **testing** the models.
3. Implement in `Keras` a simple CNN-based classifier that explores a pre-trained MobileNet backbone, which musty be kept **frozen** when training the model, and a suitable *classification head* with trainable weights.

4. Choose a **loss function**, an **optimizer**, a **learning rate** and the **number of epochs** that you think is the most appropriate for this task. If you think that **overfitting** is a problem, think of mitigation strategies. **Justify** your choices experimentally or theoretically.

5. Evaluate your model **quantitatively** by computing the per-category precision and recall rates also **qualitatively**, by showing some visual examples of classification results (good and bad).

Handing in your Assignment

This assignment should be carried out preferably in groups of two students. Provide the requested implementations and analyses to the questions in a single *Google Colab notebook* – provide the link in the appropriate Moodle activity. **Number** each question in the markup section of the notebook, and do not forget to grant editing permissions to "anyone with the link", or share it directly with me (crjung@gmail.com). You should hand in your assignment by **May 4, 2025**.

Please use the "Template" Colab Notebook available at `https://colab.research.google.com/drive/1s90AwwcpdbeUlEOVNgpJEQ69BT-Fowy0#scrollTo=-KxuOOs6WgyD` as a starting point. It basically downloads and unzips the dataset.