Assignment 2

Multi class image classification and Transfer Learning

Goal: The aim of this assignment is to learn how to tackle a multi label image classification. We will also study the effect of transfer learning, when it is used as feature extraction or for fine tuning. In the first section of the assignment, we will see how a deep pre-trained CNN works with a small dataset when the network is set to be trained (feature extraction). In the second part of the assignment, the fine-tuning phase we will see how the network behave when it is set to use the weights of image net dataset.

Deadline: TODO

Tool: Tensorflow / keras, sklearn, matplotlib, sklearn. Look at the instruction folder "Getting started" to install all the necessary packages and the environment.

Dataset: The dataset is about the most stolen car in USA. The dataset is composed by 5911 images dived in 10 classes as follow:

Class	Quantity
Chevrolet impala 2008	875
Chevrolet Silverado 2004	815
Dodge ram 2001	499
Ford F 150	879
GMC sierra	427
Honda accord 1997	647
Honda civic 1998	812
Nissan Altima 2014	342
Toyota Camry 2014	330
Toyota Corolla 2013	285

You can download the dataset from Kaggle: https://www.kaggle.com/datasets/abhishektyagi001/vehicle-make-model-recognition-dataset-vmmrdb/download?datasetVersionNumber=1 an account is required.

- Download directly the dataset from the seafile: https://seafile.utu.fi/f/7fb495a3fcdb47c283f2/?dl=1
- As alternative and easier way, we prepared the same dataset ready to use in Moodle: "pMost_Stolen_Car.zip".

Tasks: There is a Jupyter notebook template "Assignment21_template.ipynb" with instructions that you need to follow in order to tackle this lab. In summary you have to:

1- Prepare the data, use the function provided to load the data, as target size of the images use (112, 112, 3). Normalize the data. Make one hot encoding to the labels (use the OneHotEncoder function from sklearn). Use the train test split function from sklearn library to split the data. Use test split size of 0.3.

2- Build a base CNN with the following characteristics:
    a. Input layer
    b. As base model use VGG16:
        i. Weights: imagenet
        ii. Include_top: False
        iii. Input_shape the target shape described in point 1.
    c. Add a flatten layer
    d. Add a Dense layer with 512 units and a dropout layer with 0.2 unit.
    e. Add a Dense layer with 256 units and a dropout layer with 0.2 unit.
    f. Add the final classifier with the correct number of units and the suitable activation.

3- Set the layer block5_conv2, block5_conv3, block5_pool trainable.

4- Train the model with batch size of 32 and 15 epochs.

5- Evaluate the model and record the score of accuracy.

6- You can make a confusion matrix at this point (is optional it is little bit tricky to do here).

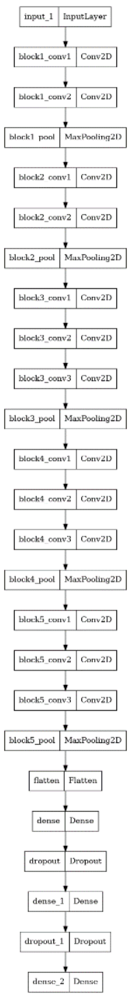
7- Make and visualize some predictions.

8- Load again the CNN and set all the base model layers to not trainable.

9- Train the model with batch size of 32 and 15 epochs.

10- Evaluate the model and record the accuracy score. What happen? Why?

11- Make and visualize some predictions.



Instruction for Submissions: You must convert your notebook to a PDF file and then submit the
PDF file to Moodle. There are two ways for converting a notebook to PDF:

- Install PyPDF2 by running pip install PyPDF2
- you can manually convert the jupyter notebook to HTML (File -> Download as -> HTML (.html)), save the HTML page as a PDF

Important:
- Please make sure that the submitted notebook has been run and the cell outputs are visible.
- The describtion of each task should be added into the notebook as a comment in order to make your code easier for understanding.

Assignment Evaluation: The notebook is evaluated on 0-10 points scale (0 means that the work is
rejected).