**Group Tis**

**Manuel Bressan, Federica Maddaloni**

For this challenge, since the problem was not addressed during the laboratory’s sessions, we looked on the Internet in order to understand how to approach it. The task is, in fact, a classification problem on 58 classes where the input is made of an image and a question, so we had to create two different models, a CNN for the images and an RNN for the text, and then combine them together*.*

A preliminary issue faced during the data preprocessing was that uploading all the images on the same folder on Google Drive caused Colab to crash because it could not handle so many items all together. So we had to split the dataset into 20 different folders, where the kth contains images from indexes (k-1)\*1500 to (k\*1500-1).

First of all, we created a custom Data Generator that split the dataset into training (20%) and validation (80%), preprocessed the images and tokenized the questions through a custom Tokenizer. We created the latter to tokenize and pad the questions, using 100 as maximum number of words in a sentence since we noticed that every question was very short.

Later on, we introduced also a custom Test Data Generator which preprocessed the images and tokenized the questions as well as the Data Generator for Training/Validation.

We tried several architectures for the CNN, like "vgg16" and "resnet50v2", and we reached the best results with "inceptionresnetv2". We noticed soon that there were still problems of memory, indeed Colab resulted in “OOM” with too large batch sizes, so eventually we settled with 32.

The final model, obtained in the class LSTM\_CNN, was the concatenation of the CNN and the RNN (for which we used an LSTM), followed by two dense layers: a ReLu and a Softmax.

Since the training time seemed too long, we tried reducing the size of the images before the training time: with half the size of the original images, we managed to reduce it significantly.

After 30 epochs we were quite satisfied with the val\_accuracy that reached a peak of 0.5800 and, since we were validating on a quite large dataset, we were expecting this score to be transposed also in the final test. However, we experienced a slight setback and we eventually got a final score of 0.53813.

Overall, we think that the result could have been improved if we had subdivided the classification task in two steps. In the first one we would have had a model that, solely on the basis of the question, would have narrowed down the possible answers (e.g. “Yes/No”, “Colour”, “Counting”, “Person” and “Other”). Then, in the second model, we would have proceeded in a similar way as we presented, with the main difference that the possible outcomes would have been only the ones decided by the first model. In this way we would have had to train more models but possibly more specialized on the single task.