Lab2 Video Classification

Jorge González Villacañas

I. INTRODUCTION

This laboratory report represents the work done in both lab assignments of the course in 'Aprendizaje Profundo para Procesamiento de Señañes de Imagen y Vídeo'. First one goes about understanding and executing two baseline video classificaction methods. The next assignment deals with the improvement of those baselines by a fine tuned convolutional neural network (CNN) previously trained on Imagenet, in this case video classification is dealt as frame by frame classification.

II. METHOD

For the first assignment we evaluate two naive baseline video classification methods. Not intended to obtain the best results, but in order to gain a insight and to have some first results to compare against when testing other video classifiers. Baseline will be a random classifier, which given a video it randomly picks from all possible categories, and a fixed classifier which always sets the same category to all videos.

For the second assignment we propose to fine tune a convolutional neural network previously trained on Imagenet. The base convolutional network is an Inception network [1]. As it will be detailed in the following section, the top of the inception model is replaced by a new untrained dense layer which will be retrained over our dataset. Video classification can then be done through frame by frame classification. This approach may be also be referred as transfer learning, as we are using information learnt on a completely different dataset (Imagenet) and using it to initialize our neural network for a new video-frame classification task.

III. DATA

The dataset used for the assigments is a subset of UCF101 'Action Recognition Data Set' of videos collected from YouTube. The categories selected are : ApplyEyeMakeup, ApplyLipstick, Archery, BabyCrawling, BalanceBeam, BandMarching, BaseballPitch, Basketball, BasketballDunk, BenchPress, Biking, Billiards, BlowDryHair, BlowingCandles, BodyWeightSquats, Bowling, BoxingPunchingBag, BoxingSpeedBag, BreastStroke, BrushingTeeth. We will further use subsets of this subset, calling UCFX the subset of videos from categories from first one (ApplyEyeMakeup) to X category. Example if UFC5 will be all videos from categories: ApplyEyeMakeup, ApplyLipstick, Archery, BabyCrawling, BalanceBeam. We will use UCF5, UCF10, UCF15 and UCF20.

IV. IMPLEMENTATION

The following is an explanation of how all three methods were implemented.

A. Random Baseline

This method simply picks a random category from all possible and assigns it as the video label. It will serve as a measure to compare if our method is better than 'just random'. We have executed the random baseline and averaged it over a 100 executions.

B. Fixed Baseline

For this method, given any video, always the same label is returned. Fixing the output label to the video category with the most videos in it will yield the best result (see section V). It will serve as a measure of how well balanced our video classes are. A good structured dataset should not have an excess of data of certain category as it may introduce a biase when training a neural network model.

C. Fine tuned CNN

The core of the second assignment was to implement the code for fine tuning the Inception neural network on a our subset of UCF101 dataset.For that purpose we followed the steps:

- Removing top layer: last layer is removed from the Inception neural network. We subsequently add a global average pooling layer, and two dense layers. First dense layer reduces the feature representation to the size of 1024, and the last dense layer outputs as many neurons as category classes we want to predict. A the top we add a Softmax layer for outputting the probabilities of belonging to each category.
- Training top layers: in order to efficiently train the new neural network we start by freezing those layers that still contain weights from Imagenet. We only train the last two newly stacked dense layers for 10 epochs over the training set. Using RMSprop optimizer and learning rate of 0.001.
- Training mid and top layers: once the top layers have been trained a little, they no longer have random weights. We then proceed to unfreeze mid layers of the Inception network and lower the learning rate to 0.0001 in order to train them for 100 epochs.

D. Metrics

We will use the following metrics to compare method performance:

- Accuracy: classification accuracy. Given a video with its category and the predicted label, this metrics checks whether prediction and true category are the same. Accuracy will be then averaged over all videos in test data.

E. Running the code

In order to execute the experiment we have two main Python files:

- "random vs fixed mode ", it runs through UCF5 dataset and classifies all videos according to the random and fixed mode methods.
- "train cnn ", for the second assignment we needed to complete the "get model" function code. This file deals with Inception fine tuning. Number of classes must be specify in the code, so to adjust the number of output neurons in the last dense layer, and therefore classify videos just into those selected categories.

V. RESULTS AND ANALYSIS

In this section we show the results obtained from executing the various methods for video classification previously explained. We further discuss and analyse them.

The following table shows the accuracy of all proposed methods.

UCF5	
Method	**Accuracy**
Random mode	19.99%
Fixed 'ApplyEyeMakeup'	22.52%
Fixed 'ApplyLipstick'	17.70%
Fixed 'Archery'	22.52%
Fixed 'BabyCrawling'	20.50%
Fixed 'BalanceBeam'	16.77%
Fined tuned CNN	90%

The following table compares the CNN method trained with 5, 10, 15, and 20 categories.

CNN				
Number of classes	**Train Acc**	**Train loss**	**Val Acc**	**Val loss**
5 (UCF5)	0.9000	0.3086	0.9750	0.2059
10 (UCF10)	0.9500	0.3895	0.9500	0.4062
15 (UCF15)	0.7875	0.9561	0.8875	0.6661
20 (UCF20)	0.7625	1.3278	0.7125	1.2127

The following subsections analyse and discuss the results.

A. Random VS fixed mode

Which mode gets the best results? Why? The mode which gets the best results is fixed mode 'Archery' and fixed mode 'Apply Makeup'. Because they contain the most videos, 290 each. Random mode will converge to 0.2 accuracy as it picks uniformly over all five categories considered.

Which one gets the worts results? Why? The mode which gets the worst results is 'BalanceBeam', as it is the mode containing the least videos.

B. CNN vs Baseline

When fine tuning the Inception network over the UCF5 dataset we get a training accuracy of 90% and a validation accuracy of 97%. Taking into account the validation accuracy we clearly see that the CNN outperforms the baseline methods.

C. Training CNN with different number of classes

As it can be seen in the tables, the more categories the less accuracy it gets. This may be due to features getting entangled in feature space, and network having more trouble discerning to which category they belong.

VI. CONCLUSIONS

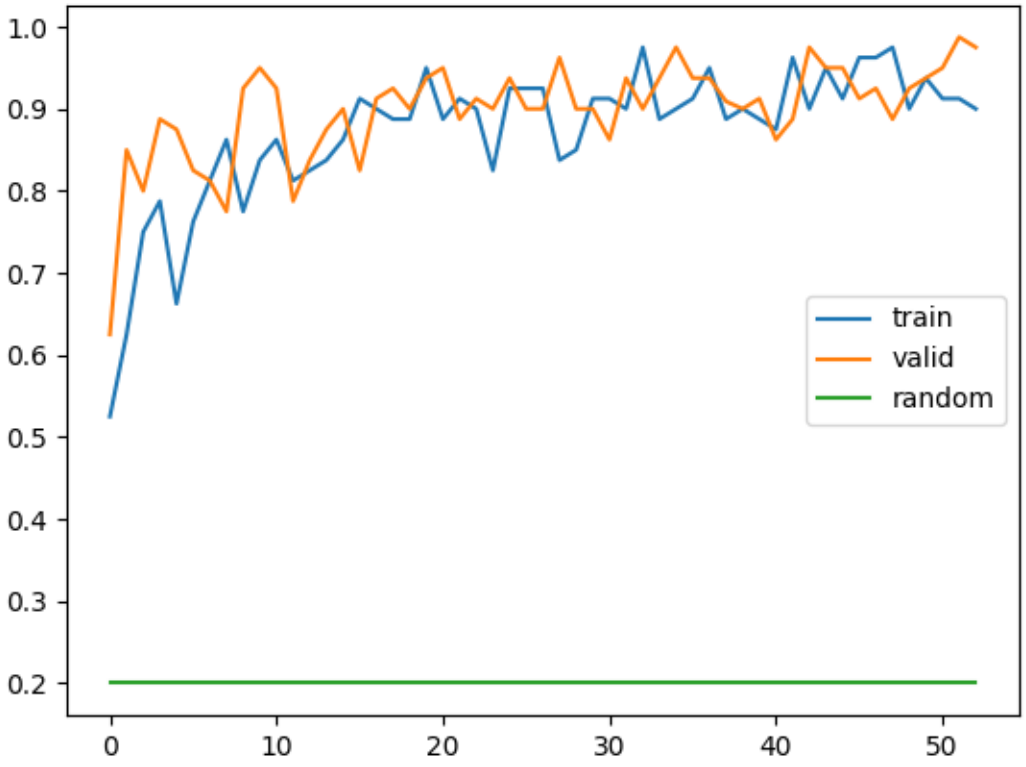
Having done the experiment, we can conclude that using pretrained models when possible and fine tuning them is a sure way to speed up model convergence. In future work, we should include some study about how to merge frame classification into video classification: average, most probable category most votes and so on.

REFERENCES

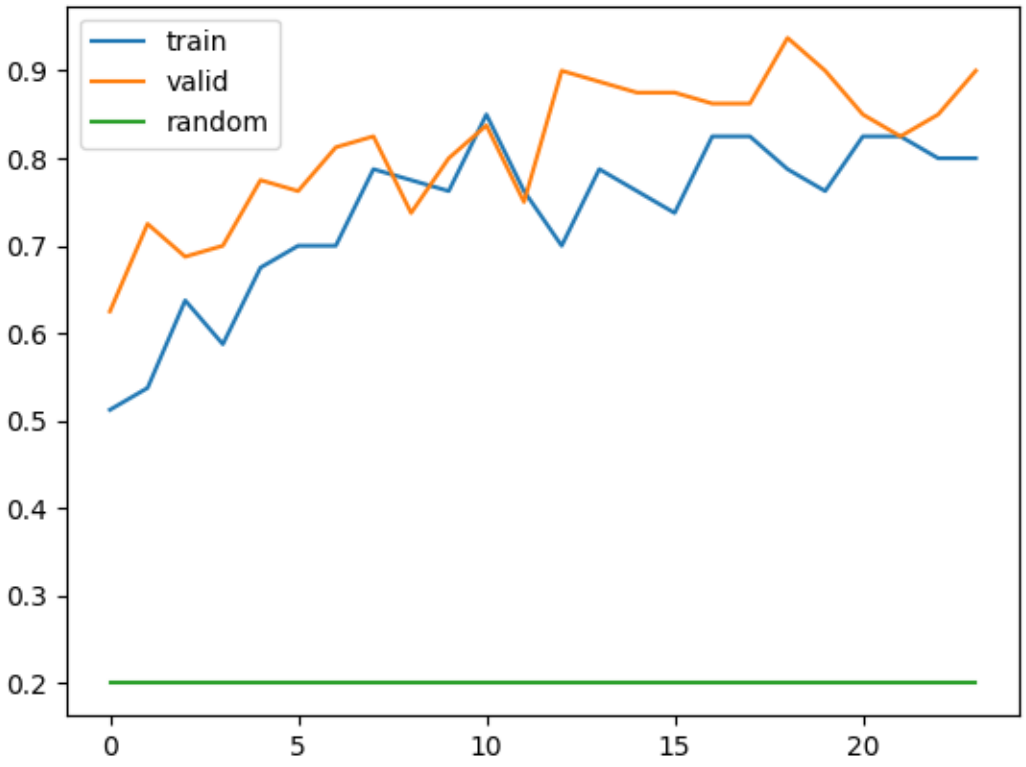
[1] Szegedy and Wei Liu and Yangqing Jia and Pierre Sermanet and Scott Reed and Dragomir Anguelov and Dumitru Erhan and Vincent Vanhoucke and Andrew Rabinovich, Going Deeper with Convolution, arXiv 2014

VII. ANEXO

Next we add the plots of training execution of the Inception network over 5, and 15 classes to give an insight into how training differed in terms of number of accuracy converge, when dealing with different number of classes.



Plot from training CNN with 5 classes



Plot from training CNN with 15 classes