CAI Classification Problem
Report 4: Implementation IV

Nil Crespo-Peiró, Paco Rahn, Amin Ranem

10. February 2021

1 Introduction

After modifying the initial GUI and the first hyperparameter tuning of the models, the following report deals with the final hyperparameter tuning of both models that will lead to the final results.

2 Implementation 4

In this section, the results of the model structures will be shown. Since the Code Base is already implemented, and the training of these models has already been started, the last hyperparameter tuning of the final models and their training is the focus of this week. Two different model structures have been used:

- AlexNet: Transfer learning to rebuild the so called ToolNet (part of EndoNet) from the corresponding paper.
- ResNet: Transfer learning using the ResNet-50 model

2.1 AlexNet/ToolNet (part of EndoNet)

As it has already been mentioned previously, this method will be used as a baseline since it was the one used by the authors of the paper Twinanda et al. 2016. The architecture in this model can be retrieved from the paper but it is essentially an AlexNet model where the last layer is modified.

2.2 ResNet

2.3 First training models

In this section, the first results regarding the training phase of the previously described models will be discussed.

Results of the AlexNet As it had been explained in the last report, we had some limitations on the Kaggle Kernels, which is the platform where the code has been executed, only 40 videos could be trained initially. The initial results (1-7) were also explained on the last report and were far from the expected values. In spite of that, the hyperparameter tuning phase had to be resumed in order to find a better configuration that leads to better results. The goal of that phase is to tune the parameters in order to obtain a high training accuracy in a dataset with very few data (5 videos), and therefore overfit on the small dataset. As one can see in the table, models 8-10 were trained with respect to this goal and an improved model – one better than the fifth model (which lead to the best results in the last report) – has been developed. Even though the model seems to achieve better results it's not regularizing well enough on large datasets. Due to the time restriction of this project, the model obtained in last report will be used as the final model for the training on the large dataset. So, the next objective is to finish training with the large dataset (40 videos) and corresponding high amount of epochs. This is the final phase of this project and the results (which will be shown in the next report) are very promising:

Model #	# videos	# epochs	learning_rate	batch_size	weight_decay	train_acc	train_loss	val_acc	val_loss	test_acc
1	37	30	1e-3	100	0.75	29.33%	0.003332	31.23%	0.003474	-
2	5	300	5e-3	200	0.5	21.45%	0.006631	6.85%	0.006808	-
3	5	60	1e-2	50	0.5	30.19%	0.01309	26.94%	0.01347	-
4	5	30	1e-3	50	0.9	17.81%	0.007252	6.486%	0.007888	-
5	5	60	1e-4	32	1e-3	98.89%	0.0002176	37.05%	0.01546	47%
6	20	60	1e-4	32	1e-3	96.76%	0.0004687	56.84%	0.01028	61%
7	37	10	1e-4	32	1e-3	89.30%	0.001431	72.99%	0.004042	61%
8	5	30	1e-4	32	5e-4	98.34%	0.0002409	45.77%	0.01377	46%
9	5	30	5e-5	32	5e-5	100%	5.323e-7	43.06%	0.03911	36%
10	5	30	5e-5	64	1e-5	100%	1.354e-7	46.89%	0.03258	59%
11	37	30	5e-5	64	1e-5	99.61%	3.064e-5	58.99%	0.007504	-

Table 1: All the specifications for all models used in the hyperparameter tuning phase can be observed

Results of the ResNet: In comparison to the last report we were able to improve the ResNet-50 significantly. The Kaggle-Kernels only provides 13GB of RAM when using a GPU. Therefore we were only able to use 40 videos at once. We used all 38 videos for training, where only 4 videos were used for validation, and 2 videos for testing. In the report from last week we came to the conclusion that the ResNet was not able to fit a small dataset where the AlexNet overfitted the training data very quick. By increasing the number of videos used for training from 5 to 40 the ResNet achieved a similar performance as the AlexNet. The ResNet seems to converge slower and does not tend to overfit the data as quick as the AlexNet. The final results for the ResNet are listed in Table 2.3 under `Model 8`. Unfortunately, the training process crashed after 23 Epochs, which might be due to the RAM limitation. Therefore we could not enter the testing stage with that model.

Model #	# videos	# epochs	learning_rate	batch_size	weight_decay	train_acc	val_acc	test_acc
1	37	30	1e-4	100	0.75	30.16%	38.63%	29% -
2	5	15	1e-4	30	0.01	58.61%	28.6%	30% -
3	5	15	1e-4	30	5e-3	60.2%	40.2%	43% -
4	5	10	1e-3	30	0.00025	67.22%	35.07%	40% -
5	5	10	1e-3	30	5e-5	70.2%	45.24%	53%
6	5	10	5e-4	30	0.000125	72.2%	41.62%	47%
7	5	10	5e-3	30	0	68.7%	51.2%	27%
8	40	20	1e-3	62	5e-5	-%	72.2%	-%

Table 2: Results for the ResNet-50

For the remaining days we will try to find a way to train the ResNet for more epochs and use the remaining 40 videos for testing. Therefore we need to split the training/testing task into two independent tasks due to the RAM limitation of the Kaggle Kernels. Furthermore we will save the model state after 20 epochs, which already lead to a good result, and testing that model (`Model 8`) on the remaining 40 videos.
Furthermore we want to test the models on the full videos in contrast to the random slice approach used during training.