OPT 1 – COMPUTER VISION (2022 / UB)
<u>REPORT:</u> **Task Optional**

Group members:

Johnny Núñez Cano (1), <jnunezca11@alumnes.ub.edu>, johnnynunez
Qijun Jin (2), <qjinjinj9@alumnes.ub.edu>, QJ

1. SUMMARY OF CONTRIBUTIONS

In this practice, we are trying to predict the ages of the people, given a set of metadata such as gender, ethnicity, and facial expression. We will apply the ResNet50 and a more current model Inceptionv2S as our backbone model of the neural network. In addition to this model, it will be nested with a set of fully-connected layers to improve the prediction's accuracy. More concretely, we will set up some enhancements over the following points:

1 - Data augmentation (new version respecting practice 1)

We apply new data augmentation that has been generated by looking at the standard deviation and mean of the dataset. This approach would predict the ages more accurately between the training set and the validation set. With respect to categories, the bias is now more similar but with a penalized, more data, and more time to train.

2- Sample weights (similar to practice 2 but with more refinement)

It is a method used in some machine learning models to assign a customized weight for each group of the training sample that can benefit the model to classify more properly and make a decision boundary.

As we are facing this task with unbalanced datasets, there are certain groups that are more popular than others, so in this case, we have to weightless those popular groups in front of others. By providing these weights, those rarely-seen groups that could be classified by age, ethnicity, facial expression, and gender will have more weights. We have looked in Keras about the definition of sample weights as:

“sample_weights functionality adds more importance to some samples than others during training that the loss function used by the model is weighted per sample, not per class. It changes the way the loss is calculated.”

We use new sample weights experiments to give more weights to those categories that we have not been able to fix in the data augmentation, such as those with fewer quantity categories or less balanced.

3 - Model and hyperparameters

We have refined our model by considering several combinations of aggregated layers. In concrete, we have tried to aggregate layers of different sizes and fully-connected layers within some dropout layers. Moreover, we have also set several values for learning rate with

$1e^4$, $5e^{5,}$ and $1e^5$, but the result of the model has not improved at all. Therefore, we have tried to change the type of the loss by combining with early stopping patience up to 20. As we did not overcome the baseline but were very close, we opted to use InceptionV2S, a new neural network from Google with much better accuracy and faster training than a Resnet. In addition, we keep more or less the same parameters.

4 - Classification layers (Important on InceptionV2S)

We have kept the same layers classification for the Resnet50 but the advantage of Inception is that we can add convolutional layers, and then the classification layers.

2. EXPERIMENTAL SETUP & DISCUSSION OF THE RESULTS

M1 - Best Model Practice 1:

This is the model from practice 1.

M2 - Best Model Practice 2:

This is the model from practice 2.

M3 - Best Resnet Model Optional with data and samples improvements:

This is the model in which all of the layers are trainable with previous last layers trained. The last layers are trained with the configuration of stage II. In this case, we use the new sample weights and data augmentation.

M4 - Best InceptionV2 Model Optional with data and samples improvements:

This is the model in which all of the layers are trainable with previous last layers trained. The last layers are trained with the configuration of stage II. In this case, we use the new sample weights and data augmentation. M4 and M5 have the same data augmentation and sample weights (10 groups).

Conv2D	BatchNorm	GlobalAvgPool	Dropout	Nodes	Dropout	Nodes
(7,7)	-	128	0.2	512	0.2	256

Comparison of error for all the experiments:

Model	Learning rate	Training strategy	Gender bias	Expression bias	Ethnicity bias	Age bias	MAE
M1	1e-5	2	**0.044955**	0.162736	0.539348	3.056905	6.340045
M2	1e-5	2	0.0194	0.16	0.489	3.078	6.853

M3	1e-5	2	0.058	**0.048**	0.405	2.9	6.670045
M4	1e-5	2	0.04844	0.052295	**0.364755**	**1.213997**	**4.083535**

Table 1: Comparison table of error and bias, results obtained for all models

As we can observe in the table, our best model M4 is the last one which is trained with InceptionV2S with all the layers as trainable and with sample weights customized. In comparison with model M4 which is Resnet50, we can say that the new model is very nice which only two dense classification layers [512,256] but with one convolution layer respect Resnet50. In all cases, we use first stage training last layers with only data augmentation and then with data augmentation and sample weights. In this case, we obtain the best result in codalab.

Plots of the experiments:

For this first comparison, we have plotted the training curves using the models M4-variant and M4 with data augmentation.

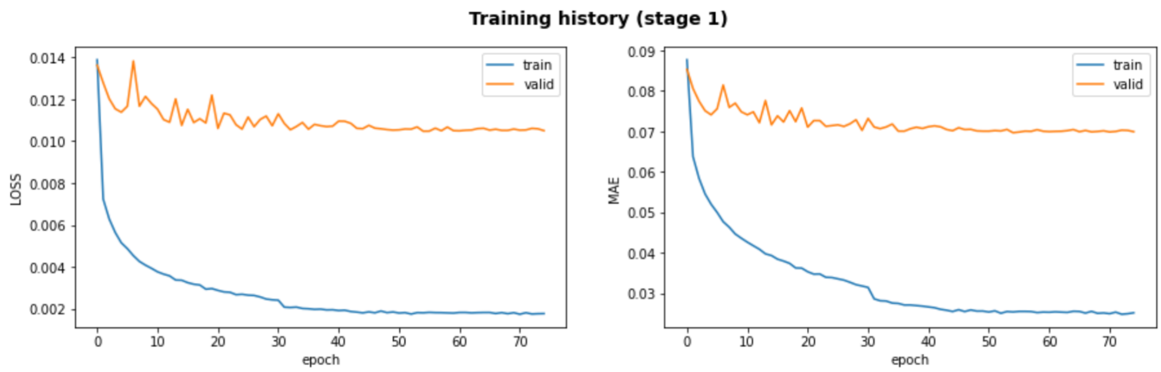

Figure 1: illustrative example of training curve on M5.

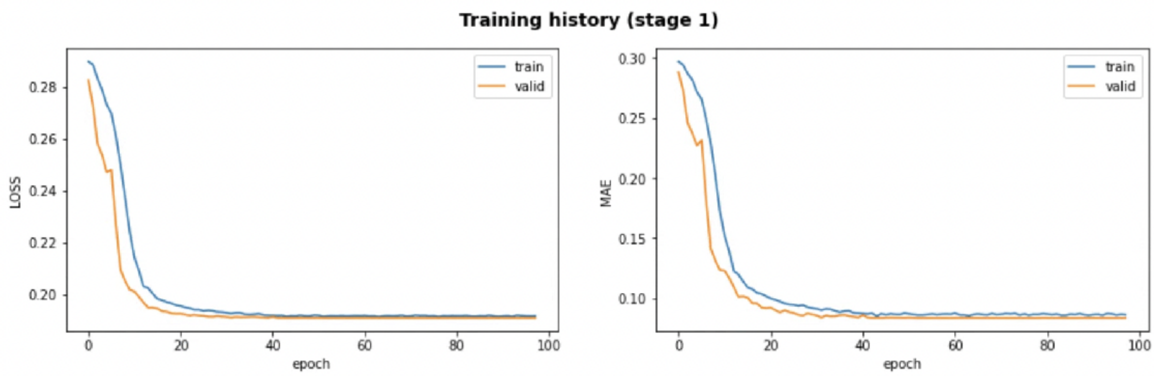

Figure 2: illustrative example of training curve on M4.

As we can see in the above plots, both models are able to be learned. With respect to the one, is Inception and we can see that de val_mae and loss can't decrease too much respect loss. Even so, these results are quite good. We see that the val-loss of Inception is lower than that of Resnet50.

This figure shows the pre-trained model without sample weights:

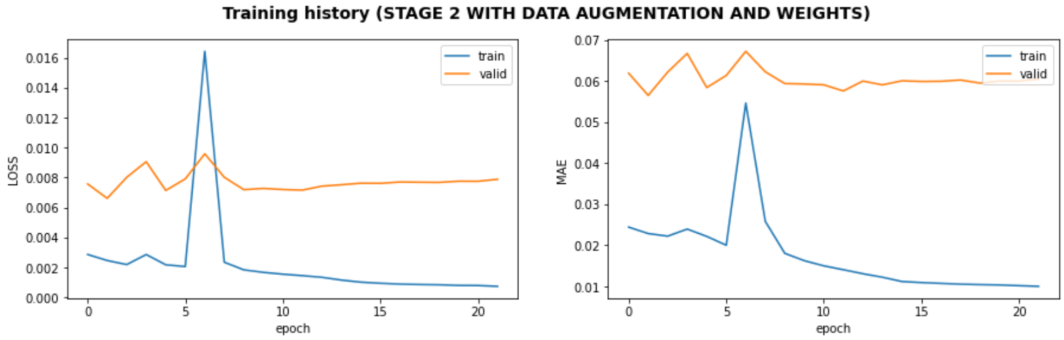

Figure 3: illustrative example of training curve on M5.

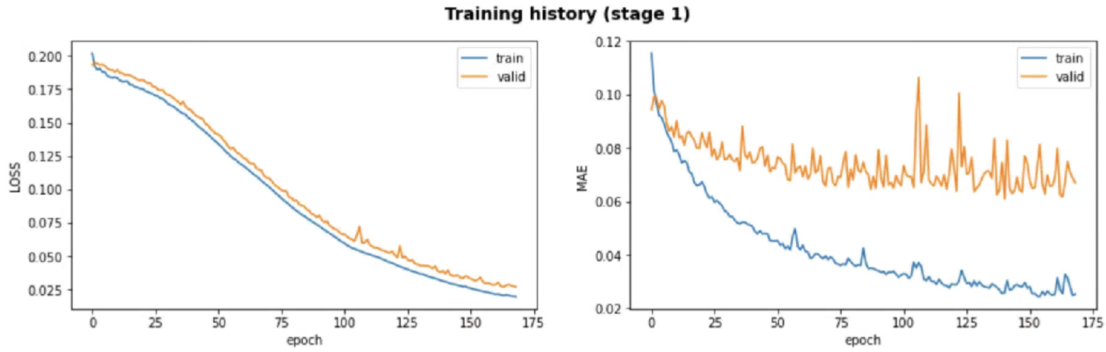

Figure 4: illustrative example of training curve on M4.

As we can see in the figures, with InceptionV2 we obtain better results with minor epochs around 30-40 epochs but more time in each epoch. This model is better than Resnet50. To clarify, this table from Inception M5 is not the best result that we have obtained, because is stochastic and we try a few attempts to obtain a mae of 3. After applying, more regularization, we obtain MAE around 4 exactly. Be careful not to train BatchNormalization layers on this model, otherwise, the previous training will go to hell. This is because it uses non-trainable weights to keep track of the mean and variance of its inputs during training.

3. FINAL REMARKS

To conclude, we want to show the progress of all deliveries. We can see that it improves a lot of results from practice 1. We think that applying data augmentation is good only if it is possible to be done and the technique of applying sample weights is good to refine the last touches made on the data augmentation. We enjoyed this practice as we have learned techniques that we have never seen before and we have been able to learn a lot about data augmentation, weights, and hyperparameters. Also how to do fine-tuning, modify trainable layers, the trainable BatchNormalization problem, and hyperparameters as well as learning rate decay techniques.