COMPUTER VISION (2023-2024)
REPORT: **Task 2**

Group members:
Joel Dieguez (1), jdieguvi15@alumnes.ub.edu , joeldvd
Clàudia Valverde (2), clvalves7@alumnes.ub.edu , claudia4

1. MODEL DESCRIPTION

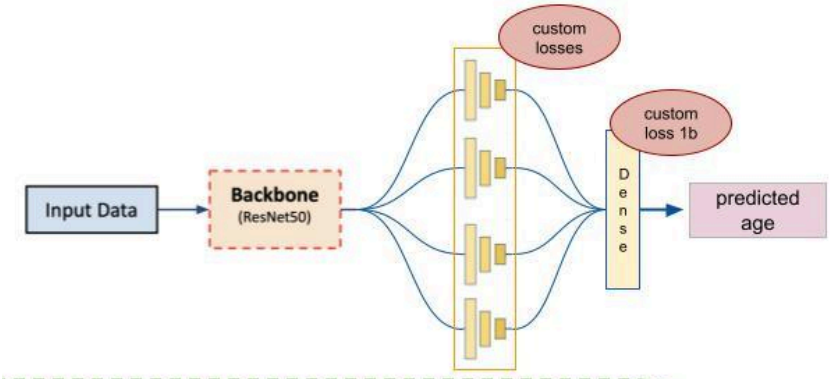

Figure 1: Illustration of the proposed model.

The proposed model architecture comprises three main components, improving on our previous work. Firstly, the data undergoes processing through the pre-trained ResNet50 backbone. Secondly, the embeddings obtained from ResNet50 serve as input for training four distinct models, each tailored with a unique custom loss function, designed to address the specific metadata groups (age, expression, ethnicity, or gender). Finally, we use a dense layer to consolidate the predictions and produce a singular predicted age for each sample. This model is trained using data augmentation and custom losses.

2. BIAS MITIGATION STRATEGY

This time around, we have decided to improve our previous bias mitigation strategy by adding a custom loss function. We have tried different approaches and we also have compared its performance with and without data augmentation.
All our custom losses take into account all metadata as well as age. We have used 4 different approaches:
Custom loss 1: we calculate the support i.e. the number of entries for each subgroup in metadata and ages and assign a weight according to the amount of representation. The weight is assigned using the following formula:

$$weight = (5 + total) / (5 + n * support)$$

where *total* is the total number of images in the dataset and *n* is the number of possible options (for example, for gender it would be 2, as it can be 'male' or 'female'). We decided to add 5 to the numerator and the denominator to avoid ever dividing by 0, even when the subgroup is empty. Then, for each entry in the dataset, we calculate the weight multiplying the values of each corresponding subgroup. For example, image 123 is a "46"-year-old "caucasian" "male" with a "slightly happy" expression, thus his assigned weight in the dataset will be 0.99 * 0.38 * 0.56 * 1.19 = 0.33.
This loss had 2 variations:

a. The ages are considered by year from 0 to 100, calculated rounding the numbers in the labels. In this case, we would have an *n* of 100.

b. The ages are grouped in: young: from 0 to 20, adult: from 20 to 40, senior: from 40 to 60 and old: older than 60.

Custom loss 2: After our first approach, we realized that the dataset was still very unbalanced. For example: even though the weight for caucasian people was very small, there were so many in comparison that, when we applied the other weights, the total accumulated for all caucasian people was still much bigger. Therefore we decided to consider the intersections of the subgroups. In the same example as before, for image 123 we won't have to multiply the weights, we have a precomputed weight for the intersection of "46"-year-old "caucasian" "slightly happy" "male" and all in this intersection have the same value. In this case the supports are much smaller and the *n* value much bigger.

a. Using the years from 0 to 100 was fastly discarded, as many intersections were empty and made the totals even more unbalanced.
b. The grouped ages worked fine. In this case, $n = 2 * 3 * 4 * 4 = 96$.

Custom loss 3: At this point we had found pretty good losses but the sums of the weights were still not perfect. We have seen that one of the main drawbacks of this dataset is how unbalanced it is regarding different metadata. Taking into account this, we wanted to test if forcing the total values would make the subsets to be equal (same amount between male and female, caucasian, asian and afro american, …). So far we found that balancing one characteristic would unbalance the other, so we decided to set the numbers manually. The values chosen are in the third column in **Table 1.**

	Male	Female		Happy	Slightly H	Neutral	Other
Loss 1b	0.99	1.01		1.45	0.56	0.73	5.78
Loss 3	0.35	0.5		2.	1.	1.	7.
	Caucasian	Afroamerican	Asian	Young	Adult	Senior	Old
Loss 1b	0.39	10.36	3.01	1.32	0.42	1.36	6.83
Loss 3	0.4	15.	4.	0.8	0.3	1.5	10

Table 1: Weights for the different subgroups from the Custom Loss 1b and 3. Custom Loss 1a is the same as 1b except for the age. Custom Loss 2 has too many weights for all the intersections to fit in the table.

2.1. BIAS MITIGATION FOR OUR FINAL MODEL

To train the experts in our final model, we used the same layer structure and training strategy but changed the loss to focus on mitigating the bias on some aspects of the data. The experts on gender, expression and ethnicity use their corresponding loss defined from the values in 1b plus the age weights from 1b (the model gave too poor results in age bias otherwise). For the expert in age, we used the age weights from 1a.

3. TRAINING STRATEGY

The model was trained in distinct stages:

1) Firstly we obtain the embedding representation of the pre-trained ResNet50 backbone.
2) These representations are imputed to each expert model where each of them is trained separately to predict the age using a different loss.
3) Finally, we add the final layer and everything is trained together using custom loss 1b.

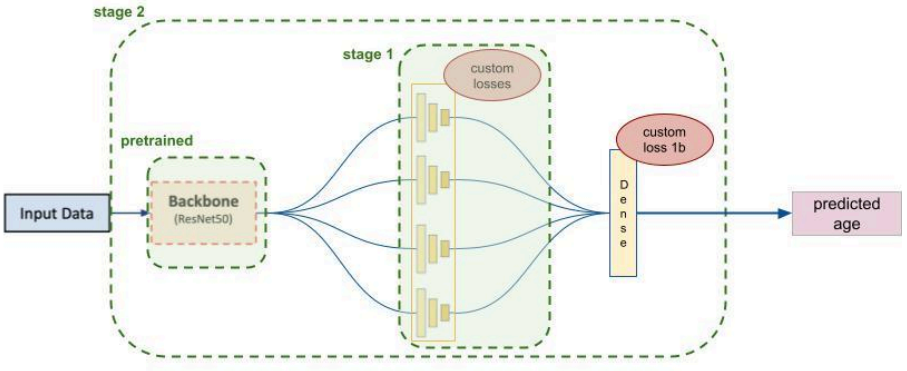

Figure 2: Training strategy of the selected model.

4. EXPERIMENTS AND RESULTS

EXPERIMENT 1: Custom Losses

For this experiment, we obtain the architecture proposed in the starting kit and change the sample weights with those computed by our custom losses. It takes into account all images in the dataset for the different metadata groups and computes the weights based on it. We test for both considering each round age as a group, and grouping by (young, adult, senior and old). As shown in **Table 2**, loss 1a has the best age bias, 1b is a bit worse overall and 2b is the more well-rounded, with good scores in all categories. Loss 3 has very good results in every field except age and a surprisingly bad Test MAE.

EXPERIMENT 2: Custom Loss + Data Augmentation

For the next experiment we decided to merge together the Data Augmentation developed in our previous work with the custom loss. As concluded in our first work, Data Augmentation gives more data to work with but does not solve the biases, as the dataset is too unbalanced. Therefore, we can use the custom losses to regulate the weights.

For the testing, we used custom loss 1b. This is due to the fact that 3 was specifically calculated for the original dataset and 1a, 2a and 2b would not work because the dataset after data augmentation had spikes of data that would disrupt the statistics for the weights. The results, in **Table 2**, are not very impressive, it worked better with just the custom loss.

EXPERIMENT 3: Ensamble of experts

This has been the proposed model as it is the most interesting one for the current task which we have thoroughly explained in other sections. To see the result please, see **Table 2.** An interesting fact about this model is that the experts sometimes excelled in other biases that were not expected.

EXPERIMENT 4: Custom Loss + data augmentation with experiment 3 from Task1

Inspired by the proposed experiment in Task1, we wanted to improve its performance as well as the strategy. Also, we wanted to see how our loss behaves using data augmentation. Thus we propose a modification to the original model from Task 1. One of the main drawbacks was that the 3rd stage was not really improving the performance, so we have decided to delete the addition of input of the original data again in that stage and train it with the augmented data as well as the custom loss 1. Finally, we train everything on the same data augmentation, for 20 epochs in stage 1, 30 in stage 2 and 10 in stage 3 which we incorporate custom loss 1 in this final stage. Also, we have decided to not split the

augmented dataset, the results show that with and without loss the model improves in respect to the proposed in Task 1, see **Table 2**.

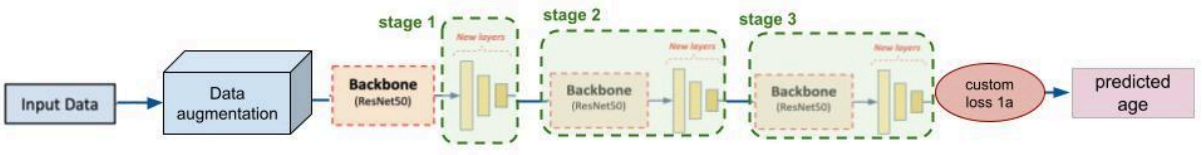

Figure 3: Architecture and training strategy of the modified model for experiment 4

Experiment	Version	Data Aug	Custom Loss	Gender (bias)	Expression (bias)	Ethnicity (bias)	Age (bias)	Avg bias	Test MAE
0	Baseline	No	No	0.154	**0.1023**	0.3441	3.1021	0.9256	4.8119
Task 1	extreme	Yes	No	**0.0401**	0.343203	1.1946	2.4392	1.0042	**4.5194**
Task 1	split mean	Yes	No	**0.0401**	0.396389	0.6124	3.8192	1.2170	5.1558
1	loss 1a	No	Yes	0.6457	0.484553	0.7166	1.2732	0.7800	4.6445
1	loss 1b	No	Yes	0.5190	0.4854	**0.1337**	2.6647	0.9507	4.7222
1	loss 2b	No	Yes	0.6319	0.4750	0.4955	**0.8688**	**0.6178**	4.5481
1	loss 3	No	Yes	0.3260	0.3802	0.8658	2.1790	0.9377	8.0118
2	loss 1b**	Yes	Yes	1.1281	0.4230	0.7724	1.1114	0.8587	6.0763
2	loss 1b**	Yes	Yes	0.6739	0.4932	0.79619	2.7593	1.1806	7.4551
3	*gender*	Yes	Yes	1.3762	0.5064	0.6330	1.5226	1.0095	6.3386
3	*ethnic*	Yes	Yes	1.3353	0.4345	0.9789	2.0862	1.2087	6.0249
3	*expression*	Yes	Yes	0.8927	0.4041	0.6839	1.4695	0.8625	5.8703
3	*age*	Yes	Yes	1.2491	0.5738	0.6314	1.6617	1.029	6.7205
3	overall	Yes	Yes	0.7665	1.7527	1.1630	4.0626	1.9362	8.9540
4	3 stages	Yes	Yes	0.2033	0.15428	0.246	3.331	0.983	5.0918

Table 2: Summary of test results for all the experiments.

5. FINAL REMARKS

We have noted several important observations during our experimentation. Firstly, dealing with an unbalanced dataset presents significant challenges. Attempts to address this imbalance through data augmentation (in Task 1) or custom loss functions (in Task 2) may not always yield satisfactory results due to the magnitude of inherent biases. In Task 2, our focus shifted towards mitigating bias scores, devising various custom loss functions to address biases and data imbalances simultaneously.

Finally, our proposed model wanted to improve previous results but it didn't reach expectations. It could be for various reasons, maybe it is overfitting, or that the addition of data augmentation + custom loss is not optimal, as the results in Experiment 2 are not great as well. Interestingly, we have found in Experiment 4 that the simpler the better, as we have adopted the proposal from Task 1 but we have done it simpler and the results have improved. Moreover, the best results overall come from the original model with custom loss 1b and no data augmentation.

Some things that we would have liked to explore but we did not have time nor computational resources are: explore various hyperparameters for the best-performing models, investigate alternative pre-trained models, and conduct comparisons with different backbones trained on diverse datasets.