COMPUTER VISION (2023-2024)

REPORT: **Task 2**

Group members:

Francesc Josep Castanyer Bibiloni, <frcastab37@alumnes.ub.edu>, Xisco354
Sergio Hernández Antón, <shernaan7@alumnes.ub.edu>, shernaan7
Ana Victoria Galindo, <avictoga7@alumnes.ub.edu>, Ana

1. MODEL DESCRIPTION

As it was advised to us, we decided to maintain VGG19 as our backbone. We decided to do this in order to be able to focus the task on working on the best modified loss. Moreover, it permits us to do fair comparisons with the results of Task 1. As this task does not focus on improving performance trying new backbones but rather on handcrafting a custom loss, we believe this is the most well-suited approach.

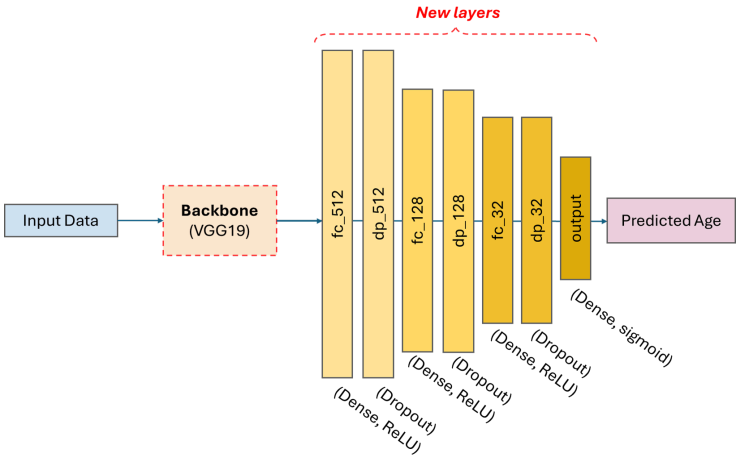

Figure 1: Illustration of our Task 1 final model.

2. BIAS MITIGATION STRATEGY

To handle bias mitigation without using data augmentation, we assigned weights to different attributes (which are: gender, expression, ethnicity and age) based on their frequencies in the dataset. The rationale behind this is to assign higher weights to less frequent attributes and lower weights to the more frequent ones. The reason behind this is that these less frequent attributes may represent minority groups that could be underrepresented in the dataset and thus require more emphasis to avoid bias. To do this, we used the inverse of the frequency of each attribute. In essence, this gives more importance to attributes that are less common in the dataset. After this step, the resulting weights are divided by the number of attributes. This ensures that the weights are scaled appropriately to reflect the overall contribution of each attribute to the dataset (in an ideal case, all weights are equal to 1). Finally, we computed the geometric mean of the weights. We decided to use it because it provides a balanced approach to weighting, ensuring that no single attribute dominates the weighting process. By taking the geometric mean, the goal is to achieve a fair distribution of weights across all attributes.

Recall that the geometric mean is the following:

$G(w_1, \ldots, w_n) = \sqrt[n]{w_1 \cdot \ldots \cdot w_n}$

Where $w_1, \ldots, w_n$ are the weights for the n different attributes.

Also, we tried a final approach, modifying the weights in terms of the result obtained, as it is more important how the model behaves on each category, rather than its frequency. We weren't able to get a better result mitigating biases, as we will explain in the results section, so we stuck to the geometric mean method.

3. TRAINING STRATEGY

In a similar way than with section 1, we decided to stick with the training strategy proposed in task 1 to allow fair comparisons with our task 1 results. Hence, we keep using a 2 stage strategy, maintaining 'Adam' as our optimizer, 'MAE' as our loss and the rest of hyperparameters (batch size of 16 and a learning rate of 10^{-5}). However, the only exception was the number of epochs due to the size reduction of the training set (2845 images) compared to the one of Task 1 (4048 images). This is why we decided to decrease the number of epochs of the second stage from 50 to 30 (also, same as in the training kit to be able to compare).

4. EXPERIMENTS AND RESULTS

Before choosing to use the geometric mean, we did some experiments. In all of them we started with the basic weight computation: computing the initial weights as the inverse of the frequency of each attribute normalized as explained before. When we had the values, we followed six different approaches.

In our initial approach, the function calculates the weights as the product of weights associated with each category for each attribute (gender: male or female, ethnicity: afroamerican, asian or caucasian, etc.). It essentially multiplies the weights of all categories to form the final weight for each instance. Then, we decided to normalize these weights, we scaled them so that they fall within a specific range, making them easier to compare. The normalization formula transformed the weights to a range between 1 and 20. After this second approach, we decided to try a different way of normalization. In this case, weights were adjusted based on their relationship to the mean, potentially emphasizing certain instances over others depending on their deviation with respect to the mean. This was computed by applying a transformation to convert the interval [min(weight),mean(weight)] to [1,10.5] and the interval [mean(weights),max(weights)] to the interval [10.5,20] lineally by parts.

After the previous experiments, we decided to change our approach and stopped trying to normalize the weights and focus more on finding the most suited weight average. Hence, instead of multiplying the weights together, we calculated their average across all attributes. The function computed the weights as the arithmetic mean of the weights associated with each attribute for each instance. As this approach showed promise, we tried to see what happened if we used the geometric mean, which provides a balanced approach to weighting and prevents any single attribute from dominating the weighting process. Finally, we checked what happened using the harmonic mean, which tends to mitigate the impact of extreme values and is useful when ensuring fairness in the weighting process.

In Table 1 we can see the results when running these experiments on the validation set. For each experiment one can simply see the bias for each attribute, the average bias and the Mean Absolute Error. Focusing on finding the strategy with less bias, we can see that the experiment number 5 was the most successful. It had the lowest bias for the face expression, ethnicity and age, while maintaining low the gender bias. It makes the model have the best average bias, showing the most successful performance in order to minimize biases. Furthermore, it is important to mention that this strategy also has the best value of MAE out of all the other experiments. For these reasons, we decided to choose strategy 5 as the best one of the six we tried

	Gender (bias)	Expression (bias)	Ethnicity (bias)	Age (bias)	Avg bias	MAE
Product	0.3168	0.5999	1.3035	2.9129	1.28327	6.4060
1st scaling [1,20]	0.0956	**0.4574**	0.5909	3.1055	1.06235	5.1545
2nd scaling [1,20]	0.1508	0.5687	0.7304	2.2411	0.9228	5.2756
Arithmetic mean	0.0543	0.5205	0.8767	2.8351	1.07165	5.2166
Geometric mean	0.1296	0.4844	**0.3858**	**1.8250**	**0.7062**	**5.0092**
Harmonic mean	**0.0517**	0.6334	0.61	3.1449	1.11	5.0645

Table 1: Results for the six first experiments with the same settings **on the validation set**. Better results are highlighted in bold.

Once we decided in which way to compute the weights in terms of the frequency of classes, we tried to fine-tune the model training it with modified weights in terms of the performance of the model in each class. Our approach was to modify the original weights in terms of the relative distance between their class' MAE value and the mean MAE value in each category. For example, if a sample is a male, as the model performed worse on males (5.0713 compared to 4.9417 on females) then the relative distance of males' MAE to the mean is $(MAE_1 - \overline{MAE})/\overline{MAE}$ = 0.01294, then the weight of every male sample is increased by a 1.29% of its original value. This is done in all classes, having the most impact in the age category and also in face expression and ethnicity categories.

On Table 2 we can see different results of the fine-tuning method with modified weights, changing different values of the learning rate (lowering it as we are not trying to train a new model but to modify ours in order to improve it). We can see that in none of the experiments, biases are reduced. In fact, they are being increased, so finally we abandoned this idea of modifying the weights.

	Gender (bias)	Expression (bias)	Ethnicity (bias)	Age (bias)	Avg bias	MAE
No fine-tuning	0.1296	0.4844	**0.3858**	**1.8250**	**0.8001**	5.0092
Fine-tuning w/ learning rate 1e-5	0.0161	**0.4335**	0.6128	2.9169	0.9948	**4.9453**
Fine-tuning w/ learning rate 1e-7	**0.008**	0.4663	0.3933	2.6816	0.8873	4.9523

Table 2: Fine-tuning results with different learning rates **on the validation set**. Better results are highlighted in bold.

On Table 3 we can compare the starting kit with and without data augmentation versus what we did on Task 1 and what we have done for Task 2. Notice that the comparison would not be fair with the starting kit provided to us, as it is trained on the VGGFaces2 dataset. To be able to compare them, we trained the starting kit model with the exact same configuration, but with the model's weights pretrained on the Imagenet dataset.

	Gender (bias)	Expression (bias)	Ethnicity (bias)	Age (bias)	Avg bias	MAE
Starting-kit - Task 1	0.2313	0.3078	0.6164	5.8252	1.7452	6.6971
Starting-kit - Task 2	**0.2165**	0.2955	0.7069	4.2909	1.3775	6.3704
Final model - Task 1	0.5895	**0.2607**	**0.09**	4.3621	1.3256	**5.4672**
Final model - Task 2	0.7585	0.3771	0.3124	**3.5754**	**1.2559**	5.7518

Table 3: Comparing the results of the final model using data augmentation (Task 1) vs custom loss (Task 2) **on the test set**. All models were pre trained with the Imagenet dataset to perform fair comparisons. Better results are highlighted in bold.

5. FINAL REMARKS

In short, we have tried six different approaches to compute the weights. After choosing the one using the geometric mean as the best, we tried to fine-tune it by modifying the weights depending on their deviation with respect to the mean, although we did not improve our biases even after reducing the learning rate. Finally, we compared our model with the final model of Task 1 and with both versions of the starting-kit (but training it on ImageNet to perform fair comparisons). The results were satisfactory, as our two models show better performance lowering the biases than the proposed baseline. We cannot really choose one above the other, as depending on the metric one surpasses the other. In particular, the model of Task 1 leads in MAE but the one of Task 2 leads in average bias. Overall, we have proven our two models perform better than the baseline in terms of reducing the biases, which was the objective.