COMPUTER VISION (2023)

REPORT: **Task 1**

**Group members:**

*Leonardo Bocchi, <lbocchbo37@alumnes.ub.edu>, LeonardoBocchi*

*Jordi Segura Pons, <jsegurpo8@alumnes.ub.edu>, JordiSegura*

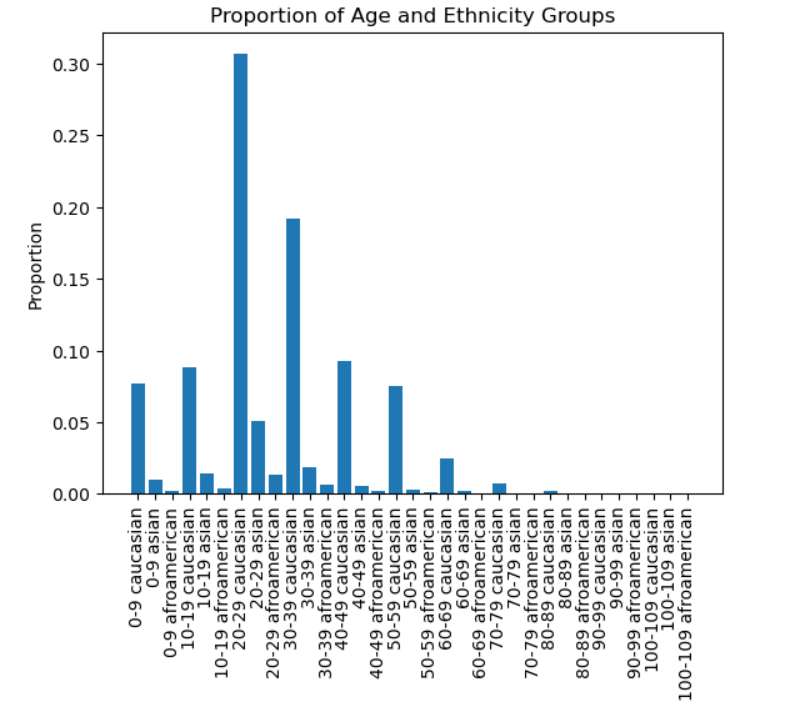
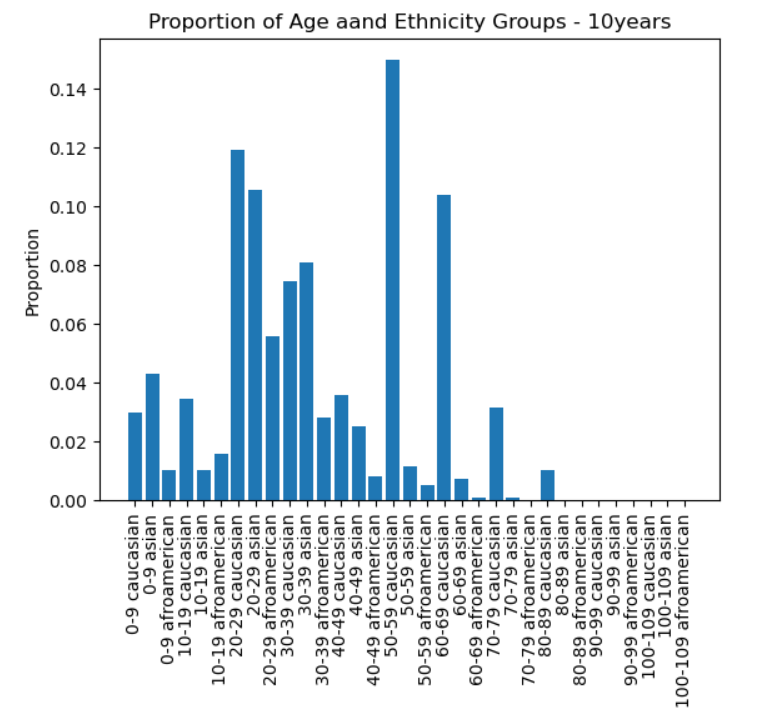
**1. SUMMARY OF CONTRIBUTIONS**

The model architecture we adopted makes use of the EfficientNet B3 model, with pre-training on ImageNet. We chose this model because it is particularly accurate while still being very efficient and it is easily accessible through tensorflow. EfficientNet-B3 has been shown to perform well on the Adience benchmark dataset for age estimation from facial images. EfficientNet models are known for their ability to achieve high accuracy with fewer parameters than other models. In particular, it uses a total of 10,783,535 parameters (compared to the 23,561,152 of ResNet) and the same input form as the one we are using.

Regarding the data distribution, the histogram showed an Age distribution very unbalanced, with a center-heavy distribution and few samples for larger and smaller values. We can also observe that the metadata has an unbalanced distribution, which will generate bias in our model. The gender attribute is balanced, but the ethnicity is tremendously unproportioned, with 85% of the sample belonging to the caucasian ethnic group. More or less, the same happens with expressivity.

Once we have studied the attributes and their distribution, we have decided to focus on the most unbalanced attributes: **age-ethnicity**. For this reason, we created a function to divide the dataset in bins of ages per ethnicity. In our case, we decided to make groups of 10 and 20 years for computational resources, but the smaller the bins, the more precise the proportions we would get. From now on we will treat the dataset by groups like: (0, caucasian), (0, afroamerican), …, (80, caucasian). (see Figure 1 for the proportions of the dataset).

Note that augmenting the age-ethnicity attributes might affect the other two, but since age is fairly distributed we settled on this first approach.

********

***Figure 1 Figure 2***

***1.1. Data augmentation strategy (in the case of task 1 or 3)***

Once we’ve studied the data distribution, we created 6 data augmentation functions. **Rotate**; which applies a random rotation to an image within the range of -45 to 45 degrees, **Flip**; which flips an image horizontally (along the y-axis), **Blur**; which applies Gaussian blur with a random kernel size of 3x3, 5x5, or 7x7 to an image, **Crop**; which crops a random portion of the input image with a random size between 0.8 and 0.9 of the original image size, and resizes it back to the original size, **Jitter**; which applies random brightness, contrast, saturation, and hue adjustments to an image in HSV color space, **Noise**; which adds salt and pepper noise to an image with a proportion of 50:50 and an amount of 0.5%.

Then, our data augmentation will consist of applying these functions to each of our bins proportionally. We created a variable called desired\_proportions which stores for each bin, which proportion we would like to have. We will only apply these transformations if our current group is below the desired proportion, we will generate a proportional number of images. (num\_augmented\_images)

Not only are we controlling the groups of age-ethnicity, but we are also applying both random number of transformations and random transformations, inside the scope of our functions, to the images. This can be seen in num\_augs, which is a random number between 2 and 4 transformations, later in DA2 was changed to 1 to 3, and augs, which are the random data augmentation functions chosen. We then store the new image, the correspondent labels, and metadata.

Some results of our work are that we have gone from a dataset of shape (4065, 224, 224, 3) to a new balanced dataset of shape (18900, 224, 224, 3) in the case of DA1. We have multiplied by 4,5x the size of our data with random augmentations. Doing so without taking into account both attributes, we would have unbalanced the other one.(see Figure 2).

***1.3. Training strategy***

The adopted training strategy consisted in the training of the whole model (EfficientNet and classifier) in one stage, making use of two training sessions, using a learning rate of 1e-3 and of 1e-4 respectively. The used optimizer is Adam and the training made use of an early stopping criteria with a patience parameter of 5. This strategy has been adopted after different experiments, which are detailed in the next section.

**2. EXPERIMENTS AND RESULTS**

The pre-trained model used in the Starting kit is pretty large, so we chose to try implementing EfficientNet B3. We did so for many reasons, from the seemingless implementation to the already shown accurate results in similar tasks and its efficiency. We started by testing its performance by adding an extremely simple classifier on top and training the whole model in one stage. (the experiment does not have the two models in the same conditions, but puts the new one at a disadvantage; we did this mainly because we wanted to see the performance of the representation part of the model, so we could assess the improvement of better classifiers later)

| ***Model*** | ***Learning rate*** | ***Training strategy*** | ***Gender bias*** | ***Expression bias*** | ***Ethnicity bias*** | ***Age bias*** | ***MAE*** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Starting[[1]](#footnote-0)* | *1e-4* | *2* | 0.168732 | *0.150445* | *0.926386* | *7.602568* | *10.472130* |
| *EffNet[[2]](#footnote-1)* | *1e-4* | *1* | 0.110675 | *0.259034* | *0.491582* | *3.858772* | *5.300690* |

All considered, the EfficientNet pre-trained model seems to be performing better while also using less than half the parameters of ResNet50.

Now, since we have also seen that training the pre-trained model gives promising results we would like to make the training less demanding. (looking forward to having to train on the larger augmented dataset) To do so, we tested solutions using SGD as optimizer and various learning rates (1e-2, 1e-3, 1e-4, 1e-5). Considering the results, we tried to divide the stage 1 training in two parts, refining the learning rate from 1e-3 to 1e-4. (It would probably be more optimal to define a function to refine the learning as a function of the number of epochs, we would like to experiment with that, but due to time constraints we implemented a simpler solution) Since we were running short on time, we also decided to add two layers to the classifier already, to hopefully reach a better result. On the same reasoning we then tried to add a second stage of training, in which only the representation part of the model was being refined with a learning rate of 1e-5 and a patience parameter of 15.

| ***Model*** | ***Learning rate*** | ***Training strategy*** | ***Gender bias*** | ***Expression bias*** | ***Ethnicity bias*** | ***Age bias*** | ***MAE*** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *EffNet* | *1e-4,* | *1* | 0.110675 | *0.259034* | *0.491582* | *3.858772* | *5.300690* |
| *Stage 1* | *1e-3, 1e-4* | *1* | 0.363686 | *0.073896* | *0.390308* | *3.188383* | *4.847096* |
| *Stage 2* | *1e-5 Adam* | *2* | *0.284434* | *0.131195* | *0.518158* | *3.887366* | *4.899098* |

The training time was reduced, from around 50 epochs to 30. (both models reached early stopping with a patience parameter of 5). Unfortunately, the second stage of training did not produce better results, despite training for 20 more epochs.

Finally, we decided to try one more train strategy that made use of 2 stages. The strategy consisted in a first stage in which we used the refining learning rate approach to train the whole model with the simplest classifier we implemented before, followed by a second stage in which we froze stage one, dropped the output layer, and added a much larger classifier, which was trained in the second stage. We felt this would yield good results since we were first refining the representation and then focussing on training a large classifier on it.

| ***Model*** | ***Learning rate*** | ***Training strategy*** | ***Gender bias*** | ***Expression bias*** | ***Ethnicity bias*** | ***Age bias*** | ***MAE*** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Stage 1* | *1e-3, 1e-4* | *1* | 0.363686 | *0.073896* | *0.390308* | *3.188383* | *4.847096* |
| *Stage 2* | *1e-3, 13-4* | *2* | 0.336174 | *0.421551* | *0.822292* | *3.040212* | *5.873725* |

Considering the results, we decided to use the one stage training strategy, hoping we could achieve better results on the bias scores using data augmentation and also considering the more efficient training time.

***2.1. Data augmentation implementation***

We tried implementing data augmentation using two different sets of parameters in our strategy.

| ***Model*** | ***Learning rate*** | ***Training strategy*** | ***Gender bias*** | ***Expression bias*** | ***Ethnicity bias*** | ***Age bias*** | ***MAE*** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *No DA****[[3]](#footnote-2)*** | *1e-4, 1e-3* | *1* | 0.363686 | *0.073896* | *0.390308* | *3.188383* | *4.847096* |
| *W/ DA1[[4]](#footnote-3)* | *1e-4, 1e-3* | *1* | *0.346159* | *0.285059* | *0.573059* | *3.177542* | *4.981238* |
| *W/ DA2[[5]](#footnote-4)* | *1e-4, 1e-3* | *1* | *0.450619* | *0.144195* | *0.156583* | *3.333091* | *4.967893* |

The results are not considerably better on the whole line. However, we managed to improve some scores without lowering the MAE too much. We are confident that using this method will allow to generate a better distribution and further improve the biases scores

**3. FINAL REMARKS**

All considered, we reached good results with the baseline model architecture and we have opted to follow clear and logical paths. The data augmentation strategy, despite thinking it is a good approach to balance the dataset did not allow us to obtain considerably better results on the biases scores.

Future work could be made trying to better balance the datasets, or including larger classifiers which would require a longer training time in order to optimize the scores on the biases and the MAE.

***Baseline + weighted loss with sample weights from started kit***

***gender\_bias: 0.276278***

***expression\_bias: 0.396767***

***eth\_bias: 0.140017***

***age\_bias: 3.582565***

***mae: 5.275298***

***Baseline + weighted loss with 1st attempt at personalized weights***

***gender\_bias: 0.381279***

***expression\_bias: 0.246949***

***eth\_bias: 0.283630***

***age\_bias: 6.336271***

***mae: 5.395432***

***Baseline + weighted loss with 1st attempt at personalized weights (proportional)***

***gender\_bias: 0.536273***

***expression\_bias: 0.225158***

***eth\_bias: 0.121925***

***age\_bias: 4.108331***

***mae: 5.128052***

***Trasformazione simile a quella dello starting kit***

***gender\_bias: 3.309571***

***expression\_bias: 2.464126***

***eth\_bias: 2.216465***

***age\_bias: 17.934510***

***mae: 13.248561***

***Actual proportional weights (finally)***

***gender\_bias: 0.099239***

***expression\_bias: 0.460264***

***eth\_bias: 0.338204***

***age\_bias: 4.115105***

***mae: 5.815720***

***1-0.8proportional weights***

***gender\_bias: 0.097882***

***expression\_bias: 0.317216***

***eth\_bias: 0.365985***

***age\_bias: 4.598607***

***mae: 5.836170***

***1-1.2proportions***

***gender\_bias: 0.309343***

***expression\_bias: 0.310098***

***eth\_bias: 0.817502***

***age\_bias: 4.406461***

***mae: 5.223919***

***Inverse Number of Samples (INS): 1/n\_samples\_group***

***gender\_bias: 0.516866***

***expression\_bias: 0.395914***

***eth\_bias: 0.096084***

***age\_bias: 3.909363***

***mae: 5.388470***

***Inverse of Squareroot of Number of Samples (ISNS): 1/sqrt(n\_samples\_group)***

***gender\_bias: 0.374354***

***expression\_bias: 0.214759***

***eth\_bias: 0.175823***

***age\_bias: 4.102623***

***mae: 5.098659***

***Effective Number of Samples (ENS): beta=0.9***

***gender\_bias: 0.344722***

***expression\_bias: 0.161563***

***eth\_bias: 0.201967***

***age\_bias: 4.040752***

***mae: 5.241138***

1. Starting: Starting kit 2 stage model [↑](#footnote-ref-0)
2. EffNet: EfficientNet B3 with simple classifier on top [↑](#footnote-ref-1)
3. No DA: model trained without using data augmentation on the training dataset [↑](#footnote-ref-2)
4. W/ DA1: model trained using data augmentation on the training dataset, attempt 1(20y bins) [↑](#footnote-ref-3)
5. W/ DA2: model trained using data augmentation on the training dataset, attempt 2 (refined parameters and 10y bins) [↑](#footnote-ref-4)