Universite Paris-Saclay

COMPUTER VISION

Fair Apparent Age Estimation

The study of methods and techniques to overcome biases.

Pawel Budzynski

August 25, 2022

1 INTRODUCTION

In the following exercises transfer learning technique will be used to build an apparent age estimation model. The model will be based on famous *ResNet50* model as a backbone. The dataset comes from the challenge 2016 *Looking at People CVPR Challenge - Track 1: Age Estimation* (https://chalearnlap.cvc.uab.cat/challenge/13/track/13/description/). It contains 4065, 1482 and 1978 samples for train, validation and test respectively. Selected samples from the dataset were presented on Figure 1.

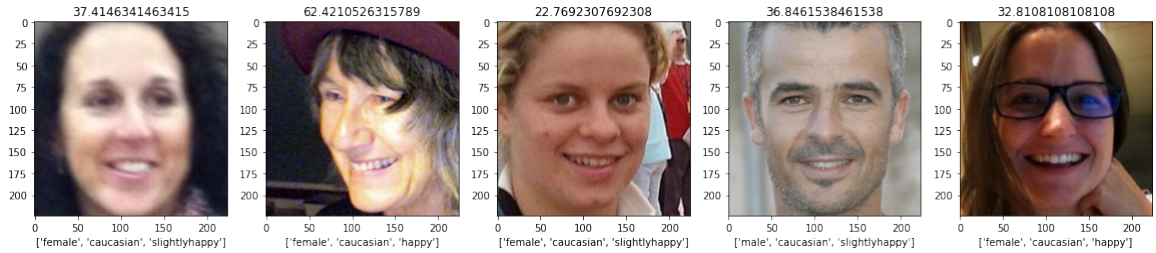

Figure 1: Random samples from the dataset.

Additionally, there is metadata attached to the dataset that contains additional gender, ethnicity and face expression labels. A brief look on the metadata distributions presented on Figures 2 and 3 gives a clue that the dataset is very unbalanced and the model trained may be biased. Because of that the project consists of two parts: Exercise 1 where the objective is to build the most accurate model and Exercise 2 where a number of experiments will be performed in an attempt to overcome the bias.

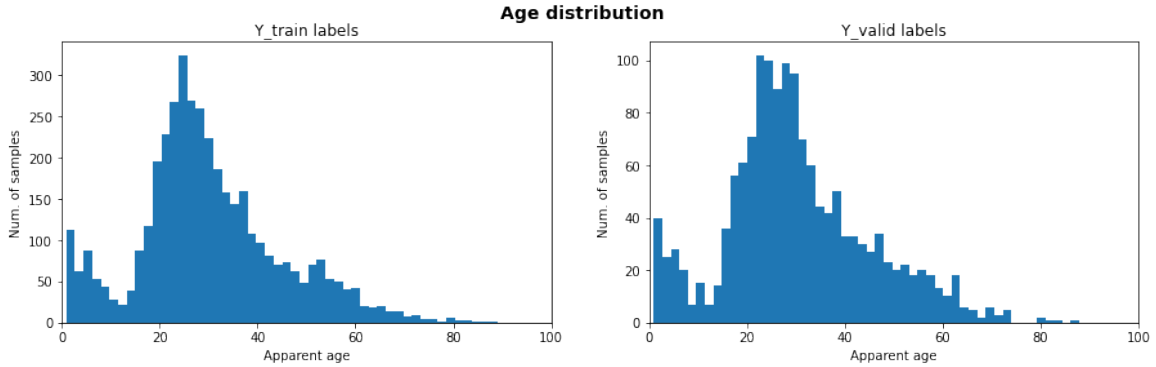

Figure 2: Distribution of samples with respect to the age for train and validation set.

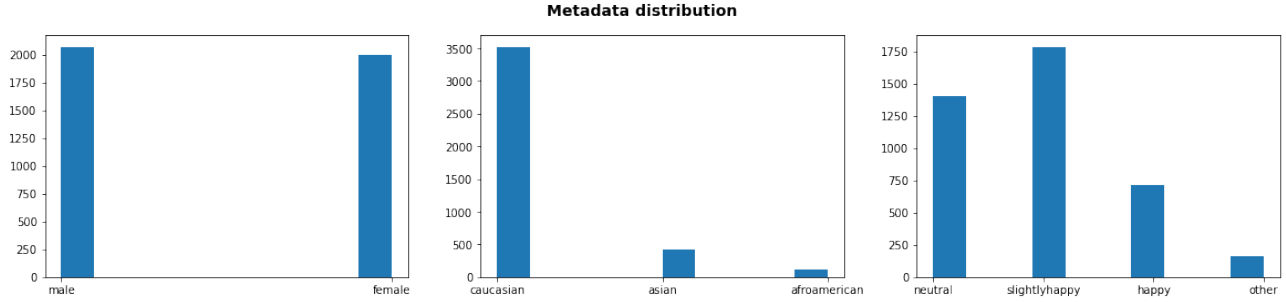

Figure 3: Distribution of metadata in the training set.

2 EXERCISE 1: AGE ESTIMATION MODEL

2.1 Summary of contributions

In this task we have been searching for the most efficient model to solve the problem of apparent age estimation. Starting with the baseline model provided, various tries have been made to improve the accuracy. Different settings of dense networks on top of the *ResNet50* have been checked, including experimentation with number of hidden layers, their size, activation functions, dropout and batch normalization. An effort has been made to replace the *ResNet50* backbone with *VGG16*, without significant benefit however. The network in this setting struggled to fit the dataset which may be caused by the fact that originally it was trained in *ImageNet* dataset for object detection while the domain of the task consists of human faces. Because of that we kept *ResNet50* as a backbone.

The architecture was slightly modified as the dense network was plugged directly to the earlier flattening layer with shape 2048. During the training process two optimizers have been tested together with various learning rates. *SGD* optimizer appeared to be slightly hard to tune as the learning curve was either very steep or flat. Adam optimizer was selected as the best one and used with a bigger learning rate than in the initial training. The learning rate was decreased during later steps to fine-tune the model. Since the original dataset is very unbalanced various data augmentation were applied to try to decrease the model biases regarding age, gender, ethnicity, and expression.

2.2 Experimental setup

One data augmentation strategy that was tried was first oversampling the categories of age that were underrepresented to half match the amount of the most represented category. It was matched to half due to Colab limitations. After this, Keras data augmentation function *ImageDataGenerator*, was used on the whole balanced set, applying rotations up to 30 degrees, width shifts, height shifts, shear, zoom, and horizontal flips. Keeping the above mentioned architecture, the network was trained and the results can be found in the next section as **A3** strategy Data Augmentation 1 (**DA1**).

Second data augmentation strategy was employed independently with the aim to make a balanced augmented train set regarding the different age ranges. First, the number of images in the train set belonging to each of the age ranges was computed, after which we determined the number of times we needed to augment each of the minority age range subsets to make a balanced augmented train set. Each image in the minority age range classes was augmented the necessary number of times, using a custom augmentation function, which performs the following image augmentation techniques: changing the brightness, image rotation, translation, zoom, gaussian blur and horizontal flip. Finally, the stage 2 of network training was performed using the augmented train set and the results are shown in **A4** strategy Data Augmentation 2 (**DA2**).

In Table 1 the results of the selected experiments are presented, without the data augmentation employed, with the following training strategies: (**S1**) – the first layers of the network are frozen; and (**S2**) – all layers of the network are set as trainable.

- **Model A**: ResNet(flatten_1) -> Dense(200, relu) -> Batch normalization -> Dropout(0.4) -> Dense(30,relu) -> Dropout(0.4) -> Sigmoid;
  Adam optimizer.

- **Model B**: ResNet(dim_proj) -> Dropout(0.4) -> Dense(200, relu) -> Dense(10, relu) -> Sigmoid;
  SGD optimizer.

Model A3 and **Model A4** represent the experiments with the two described augmentation techniques, used in the stage 2 of training the **Model A**.

Table 1: Comparison of the results obtained for selected models.

Model	Learning Rate	Training Strategy	Gender Bias	Expression Bias	Ethnicity Bias	Age Bias	MAE
A1	1e-3	S1	**0.1849**	0.6693	0.7407	9.3983	9.0990
A2	1e-5	S2	0.2011	0.3353	0.4904	**4.0602**	7.4408
B	1e-5	S1	0.9536	1.7809	1.1897	13.4204	11.3701
A3	1e-4	DA1	0.9377	0.3606	1.2064	4.5095	8.5879
A4	1e-5	DA2	0.7037	**0.1897**	**0.4141**	5.5101	**6.8682**

2.3 Discussion of the results

As can be seen in Table 1, the model that worked the best in terms of model biases is **A2**. Also, performing data augmentation to try balancing classes within different age ranges (**A4**) gave better results than upsampling the data and then performing data augmentation on all images of the dataset (**A3**). MAE was further reduced compared to the **A2** model using the **DA2** augmentation technique, from 7.44 to 6.86. The **DA2** augmentation model also achieved a decrease in the values of expression and ethnicity biases, while gender bias was substantially increased, from 0.20 achieved with the **A2** model to 0.70. Age bias was also slightly increased, from 4.06 to 5.51. This shows that by augmenting the data with respect to only one attribute, the achieved results can be better with respect to some of the training goals, while for others it will not be the case, which can also lead to the overall performance being better for the model without data augmentation employed. Possible better overall results could be achieved if the augmentation technique took into account balancing the data with respect to multiple attributes, for instance age range and gender or other attributes.

2.4 Final remarks

In this exercise we have managed to successfully train several neural networks for age prediction. Although the original dataset was highly unbalanced the models trained on it managed to achieve accuracies in the range 6.9 to 9.1 of error. Further trials to debias the model and improve the score gave some results: for example, some of the biases went down, however the others increased (Model **A2** vs **A4**). More work should be done to make sure the model bias decreases without a negative influence on the overall accuracy, by exploring other types of data augmentation, or it is important to try weighted loss too.

3 EXERCISE 2: BIAS REDUCTION

3.1 Summary of contributions

In this exercise the architecture of the network was fixed to a version that had the best performance in Exercise 1. It is a network based on the *ResNet50* backbone with dense layers on top of it. Various techniques have been tested with the aim to improve the model performance and decrease biases. Apart from training parameters such as learning rates and early stopping we have also experimented with *Reduce Learning Rate on Plateau* callback provided by Tensorflow. Multiple approaches have been tried to customize the training process. As presented in the later parts of the report, we have implemented several variants of training samples weighting, custom loss functions based on MSE and bias computation and multiple output networks.

3.2 Experimental setup

Regarding weighted loss, several strategies were implemented to calculate the weights to be passed to the training algorithm. One group of strategies involved calculating weights independently for each categorical variable, and then creating linear combinations of these, that is, one option was the mean of the weights, and another option was weighted averages, where the weights sum to one. The second group of strategies comprised first the combination of several categorical variables into one variable, and then the calculation of weights for these combined categories. In each strategy, several groups of variables were tested, with age group and ethnicity always included, and face expression and gender added in some of them. In this stage of training the learning rate was reduced by half when the validation MAE reached a plateau for 5 epochs, and the training was carried out over the best model obtained in our previous work on the second stage (fine tuning of the whole network without data augmentation). One last strategy tried was to add random normal noise of small variance (0.2) to the best strategy found in the previous steps. The results can be seen in Table 2.

Table 2: Comparison of results of different weighted loss strategies.

Variables	Weight Calculation	Gender Bias	Expression Bias	Ethnicity Bias	Age Bias	MAE
AEF	IM	0.472	0.975	0.762	5.534	7.083
AE	IM	0.160	**0.401**	**0.228**	4.014	7.060
AEFG	IM	0.278	0.499	0.241	3.505	7.053
.6*A.3*E.1*F	IWM 631	0.605	0.700	0.290	**3.234**	6.767
.5*A.3*E.2*F	IWM 532	0.789	0.539	0.556	3.636	7.003
AGEF	C	0.681	0.431	1.328	4.179	8.180
AE	C	**0.073**	1.280	0.843	5.100	10.165
.6*A.3*E.1*F	IWM	0.406	0.545	0.556	6.310	**6.683**

A=Age, E=Ethnicity, F=Facial Expression, G=Gender, I=Independent weight, M=Mean, W=Weighted, C=Combined, R=Random noise

The experiment with custom loss was initially performed for age bias as it was easy to compute given only the golden values. A custom loss function has been implemented and used during the training such that the final loss is a linear combination of age bias and mean squared error. Since computation of other biases requires metadata dependencies, their implementation would

require much more changes as a tensor of ground truths would have to be modified to include additional data. An approach has been made to combine custom loss function with the best performing weighting strategy.

Table 3: Comparison of models trained using age bias as custom loss function component.

Loss Function	Training Strategy	Gender Bias	Expression Bias	Ethnicity Bias	Age Bias	MAE
MSE (BaseTask1)	2	0.201	**0.335**	0.490	4.060	7.441
0.3*Age_bias + 0.7*MSE	2	**0.195**	0.459	0.695	4.932	7.197
0.5*Age_bias + 0.5*MSE	2	0.532	0.738	**0.379**	4.118	**6.869**
Age_bias + 0.15*MSE	2	0.582	0.619	0.609	**3.081**	7.144
0.5*Age_bias + 0.5*MSE	2+ IWM631	0.872	0.761	1.114	3.881	7.358

The experiment with multiple output models was conducted with the aim to solve the task as both the regression and classification problem. The last, output layer of the original network used in Exercise 1 was replaced with an output layer that consists of two outputs - one that treats the task as regression problem and minimizes the mean squared error (MSE) loss, and the other one that solves the classification task, with softmax activation function in the output layer and categorical cross-entropy used as the loss function. During training, the final loss was computed as the sum of the losses of two output branches of the model. Two classification strategies were tested: (1) each age was treated as a separate class, meaning that the network was trained to classify the person's age as one of 100 separate classes; (2) classification was performed according to 4 age ranges, which were defined the same way as age ranges used for the computing age bias and the age weights. Both models were trained either with or without class weights added to the loss computation in the phase 2 of training. The obtained results can be seen in Table 4. As the problem was also approached as a classification task, the classification accuracy, calculated with respect to different age classes, was given in a separate column.

Table 4: Comparison of results of different multiple output regression/classification models.

Model	Weight Calculation	Gender Bias	Expression Bias	Ethnicity Bias	Age Bias	MAE
100 classes	none	0.289	0.559	1.100	5.115	10.607
100 classes	AEF - IWM 631	0.226	0.674	1.029	2.836	8.717
4 classes	none	0.205	**0.518**	1.508	8.641	10.091
4 classes	Age only	0.439	1.230	1.766	**2.246**	8.625
4 classes	AEF - IWM 631	0.157	0.745	0.668	2.357	**8.545**
4 classes	AE - C	**0.116**	0.917	**0.593**	2.528	9.671

3.3 Discussion of the results

When it comes to weighted loss strategy, according to Table 2, it can be observed that the best result regarding MAE and age bias is achieved by the model trained with weighted average of Age, Ethnicity and Facial expression with weights of 0.6, 0.3 and 0.1 respectively. On the other hand, the strategy with combined categories did not provide very good results in terms of MAE and biases in general, except for the last model which showed the lowest gender bias.

The experiment with incorporating specialized biases in the custom loss function gave interesting results. As presented in Table 3, we have achieved a desired outcome as age bias is decreasing when its weight in the loss function is increasing. However the rest of the biases may suffer in the process of reducing age bias. Since computing the other biases requires metadata dependencies it would require to merge the metadata into the label tensor to use other biases in the loss computation. Furthermore, modifying the label tensor to contain encoded metadata would also require implementing a custom metric for the network. Hence, a try to incorporate other biases was abandoned.

The multiple output model which achieved the lowest value of MAE is the 4 class model with AEF - IWM 631 weighting strategy employed. This model also achieved the greatest accuracy for the classification sub-task. However, it did not achieve the best value on any of the bias scores. The lowest age bias was achieved using the 4 class model with age weights employed. The lowest facial expression bias was achieved using the 4 class model without adding weights to the loss, while the lowest values of gender and ethnicity bias were achieved using the 4 class model with combined age and ethnicity weights.

3.4 Final remarks

The lowest value for MAE was achieved using AEF - IWM 631 weighting strategy, with the value 6.767. Experiments with multiple output models resulted in higher values of MAE achieved on the test set for all models employed, compared with the values achieved by weighted loss or custom loss strategies, except for the AE-C weighting strategy, which lead to greater MAE value than any of the weighted multiple output models. However, the lowest age bias was obtained by the multiple output strategy with 4 age classes and AEF - IWM 631 weighted loss, with the value of 2.246. The lowest values for all other bias categories were achieved by one of the weighting techniques, namely: (1) gender, by AE-C, with value 0.073; (2) facial expression, by AE-IM, with value 0.401; (3) ethnicity, by AEFG-IM, with value 0.241. The proposed strategies managed to achieve better results than the base model used in Exercise 1. They obtained lower values of MAE and all bias categories, except for the facial expression bias, which still has the lowest value achieved by the base model in Exercise 1, equal to 0.335. The future work would include combining the most successful strategies into one, in order to additionally improve the results. Also, additional attention should be paid on developing the model that would minimize the facial expression bias, as its value has not been improved by any of the proposed strategies.