Face age prediction

Olaf Werner

January 2024

Contents

1 Introduction

The aim of the project is to train an model which can predict person age. This is difficult task as an people can age differently based on genetics and lifestyle. Also conditions under which photo was taken may impact the results.

2 Data Description and Preparation

We have our dataset UTKFace from Kaggle [4]. It consist of 23708 jpgs containing faces. People on this photos are of various ages, genders and ethnicity. Ages ranges from 1 to 116. Possible races are: White, Black, Asian, Indian, and Others. Genders are Male or Female.

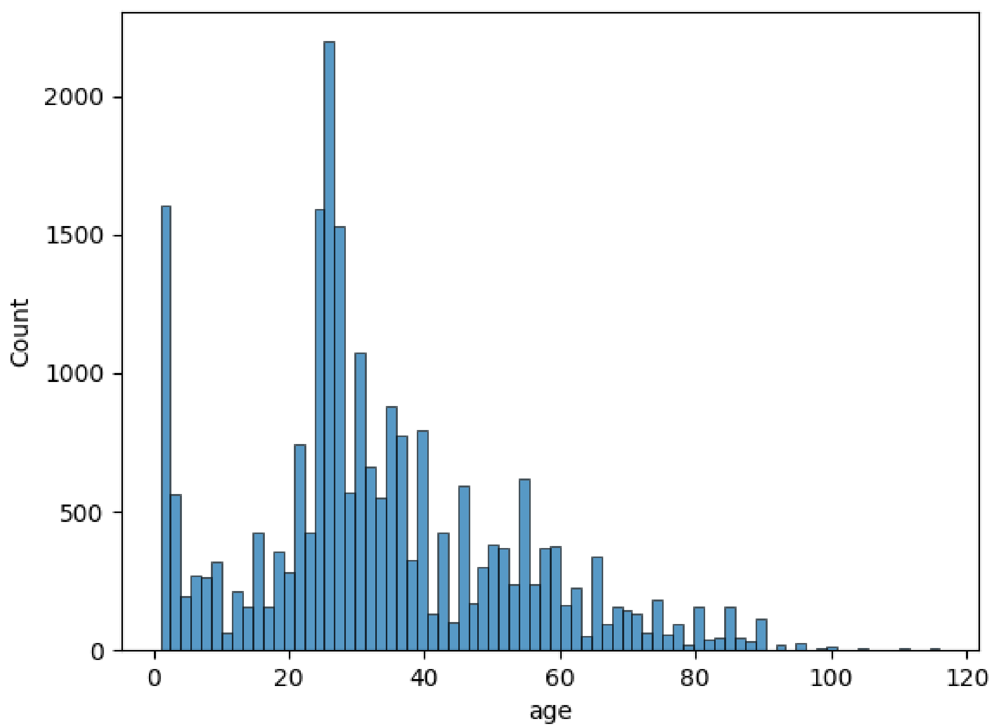

Figure 1: Age distribution

2.1 Data splitting

We split our data in following proportions:

- Train: 18966 images, 80% of total dataset
- Validation: 2371 images, 10% of total dataset
- Test: 2371 images, 10% of total dataset

Before splitting data we sort it by age. Thanks to that ages will be distributed more uniformly across all subsets.

2.2 Data Augmentation and Sampling

We will use following basic augmentations during training, testing and validation:

1. Change size: we will reshape to (224,244)
2. Random horizontal flip: we will horizontally flip image with 50% chance
3. Random brightness: we will increase or decrease brightness of the image by up to 20% .

Because data is imbalanced we will use balancing sampling for training. This means that each age class will have equal chance to appear in given training batch. Also we will cap maximum age to 100 as one of our benchmarks models do that and classes above 100 are very rare. We rescale to (224,244) because our benchmark model needs this shape.

3 Models

3.1 EfficientNet

We will use EfficientNet-B0 [3]. This network is originally for image classification. We will perform transfer learning by replacing model head with one for predicting person age. This will be an regression task instead of classification.

We picked this network because it is as name suggest efficient. Authors of the network checked how to scale width, depth and resolution in such a way that best results are obtained.

3.2 VGG DEX

As a benchmark model we will use VGG age classifier from DEX trained on the IMDB-WIKI dataset [5]. This model maximum age prediction is capped to 100 as it classification model.

We picked this model as a benchmark because this model won an competition [2] and is used in cycle-GAN project [1].

3.3 Median

As a basic benchmark we will use median age as a predictor. Median age for the training dataset is 29.

4 Training

For training our prediction model we will use following set up:

- seed: 123
- max_num_epochs: 10
- batch_size: 32
- learning_rate: 0.001
- optimizer: Adam
- training loss function: MSE
- validation loss function: MAE

We will be performing validation check 10 times per epoch. As a check point we will be saving our best model based on validation loss.

5 Results

5.1 Error during training

As we can see on (Fig:2) our model reached convergence as best validation loss is no longer dropping. We are also better than median. Our best validation MAE is 6.54.

5.2 Distribution of predictions

Looking at the distribution of predictions of benchmark model VGG (Fig:3) and EfficientNet (Fig:4) we can see that our model is more uniform in its predictions than benchmark model. This is caused by our training routine which encouraged more uniform sampling from different classes. On the other hand our benchmark model is more jagged in its predictions. Whats worse it has troubles with age of babies as almost no predictions are for this class. This is caused because its IMDB-WIKI dataset [2] was collection of actors and celebrities so babies were underrepresented.

5.3 Final test MAE

To make comparison between benchmark model more fair we also check how it compares to it when in our test set include only adults (age at least 18). Even in this case we can see that our model is the best (Table:1).

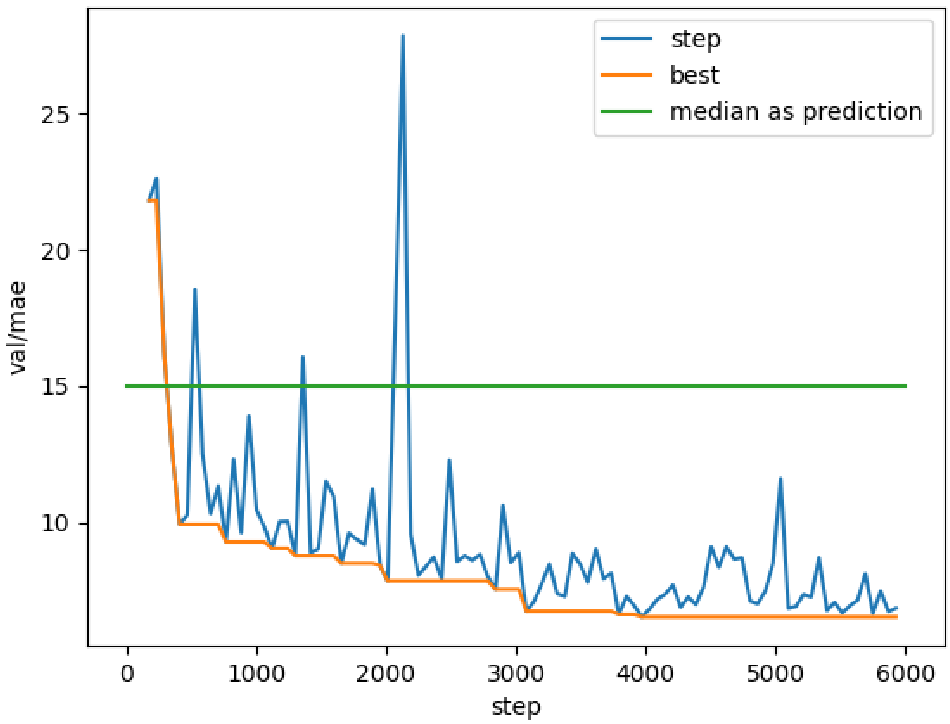

Figure 2: Progress of training

Model	Test MAE (All)	Test MAE (Adults)
EfficientNet	6.75	7.25
VGG	11.52	10.20
Median	14.96	13.28

Table 1: Final results

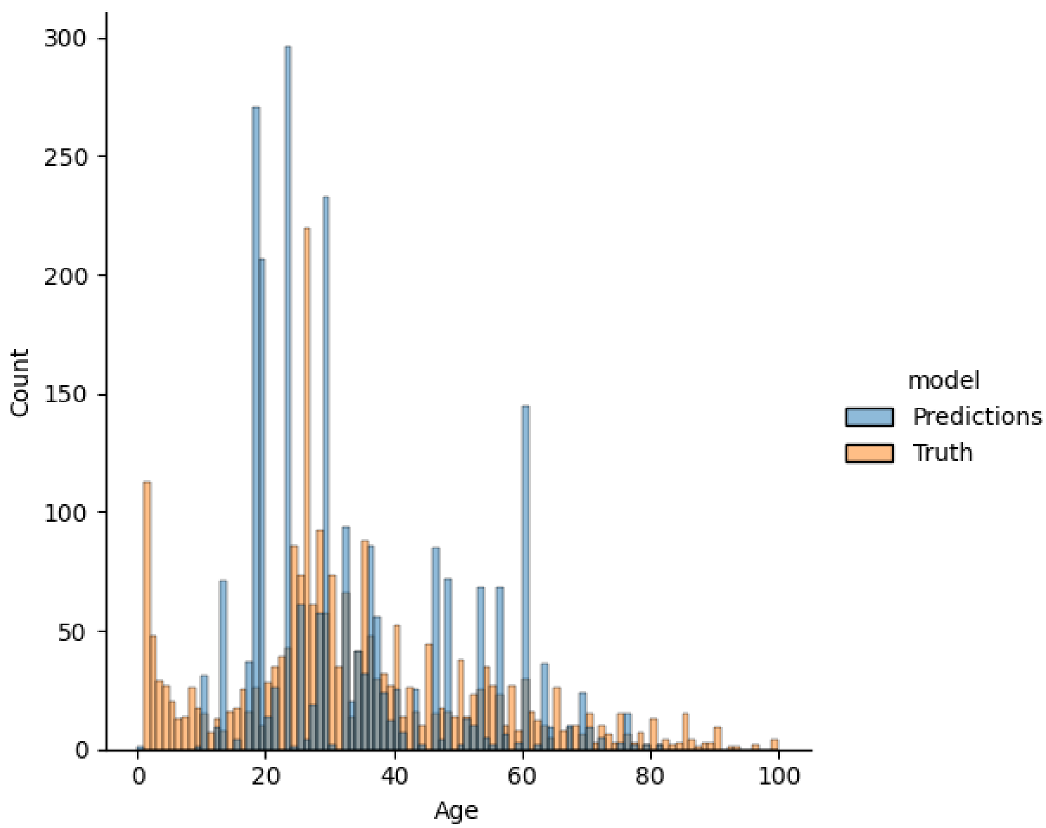

Figure 3: Distribution of predictions of VGG

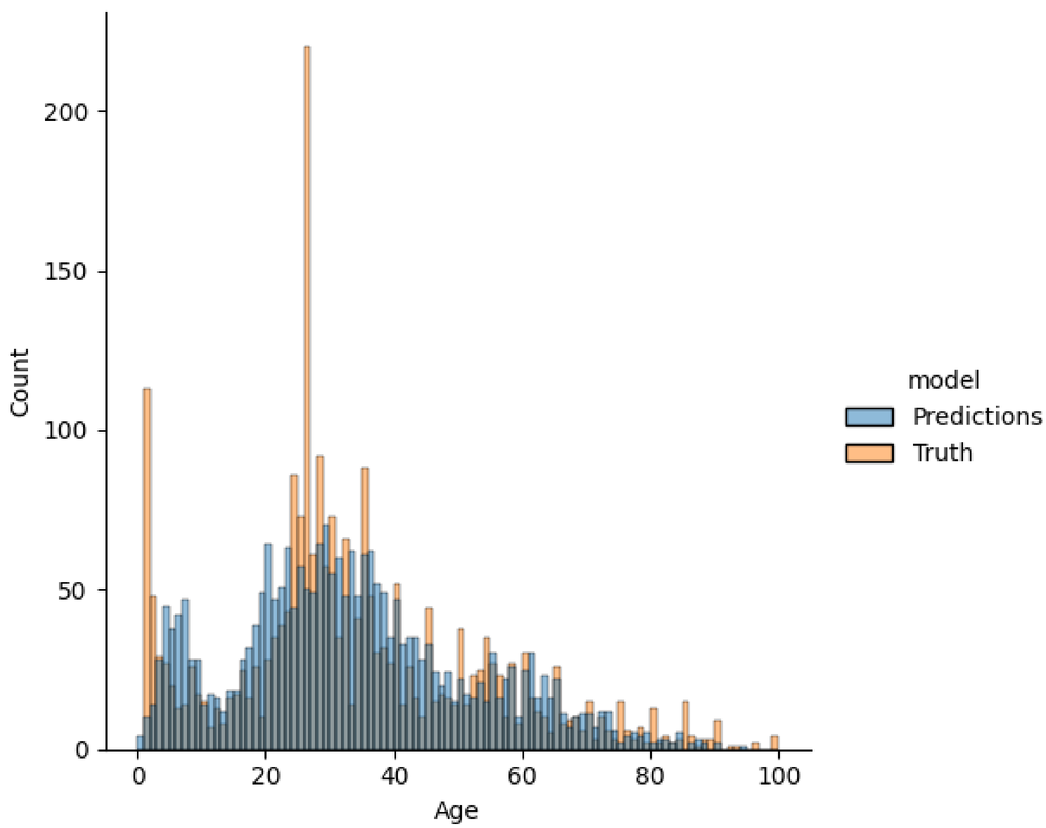

Figure 4: Distribution of predictions of EfficientNet

6 Summary

We managed to get better results on our particular dataset better results than state of the art model which won the LAP challenge [2]. This only highlights how important dataset is for given model and that generalization especially when trained in biased context is hard to achieve.

7 Used libraries

We done our work in Python Here is the list of the most important libraries:

- torch: Deep learning framework
- torchvision: pytorch dedicated for computer vision
- pytorch-lightning: pytorch wrapper for streamlining model development
- efficientnet-pytorch: pytorch implementation of EfficientNet
- seaborn: visualization library
- pandas: data wrangling library

8 Source Code

Project git repository is under this link `https://github.com/wernerolaf/Age_Detector`.

References

[1] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. "Only a Matter of Style: Age Transformation Using a Style-Based Regression Model". In: *ACM Trans. Graph.* 40.4 (2021). URL: `https://doi.org/10.1145/3450626.3459805`.

[2] Rasmus Rothe, Radu Timofte, and Luc Van Gool. "Deep expectation of real and apparent age from a single image without facial landmarks". In: *International Journal of Computer Vision* 126.2-4 (2018), pp. 144–157.

[3] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. arXiv: `1905.11946 [cs.LG]`.

[4] *UTKFace*. https://www.kaggle.com/datasets/jangedoo/utkface-new.

[5] Xu Yao et al. "High Resolution Face Age Editing". In: *CoRR* abs/2005.04410 (2020).