Age and Gender Detection

From Face Data

Hrayr Muradyan

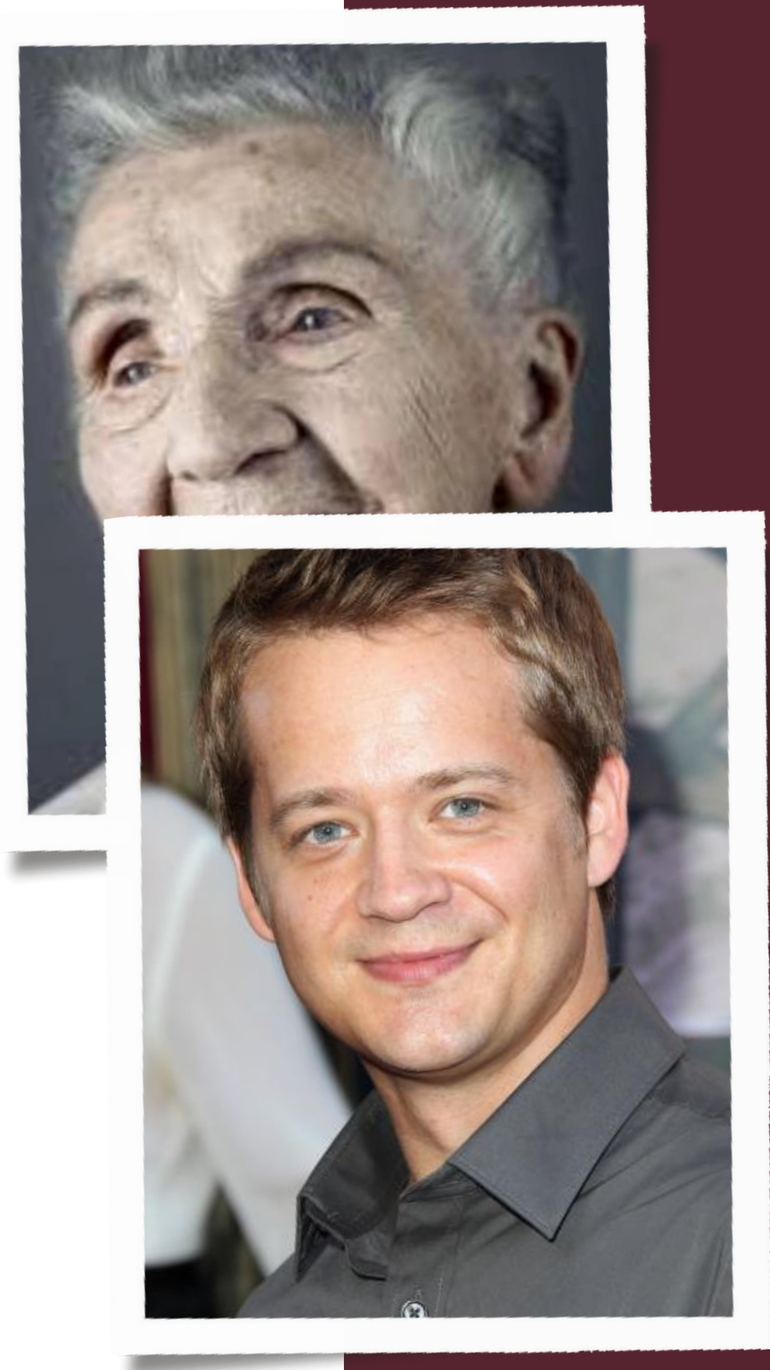
AMERICAN UNIVERSITY OF ARMENIA

Why to identify gender and age?

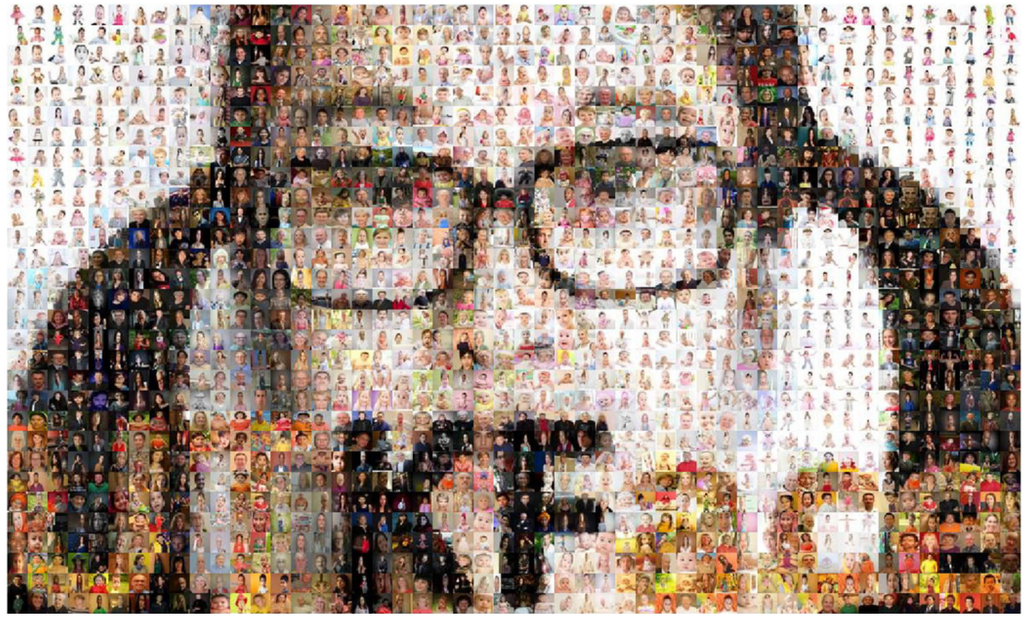
Security: In security systems, age and gender detection can aid in identifying individuals and understanding crowd compositions, which is crucial for safety and monitoring purposes.

Healthcare and Elderly Care: In healthcare, this technology can assist in monitoring elderly patients, tailoring care according to age and gender-specific needs.

Age-Restricted Content Control: It helps in ensuring that age-appropriate content is shown to users, especially crucial for online platforms hosting violate content.



The Dataset



UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over **20,000** face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc.

Source: https://susanqq.github.io/UTKFace/

Annotations

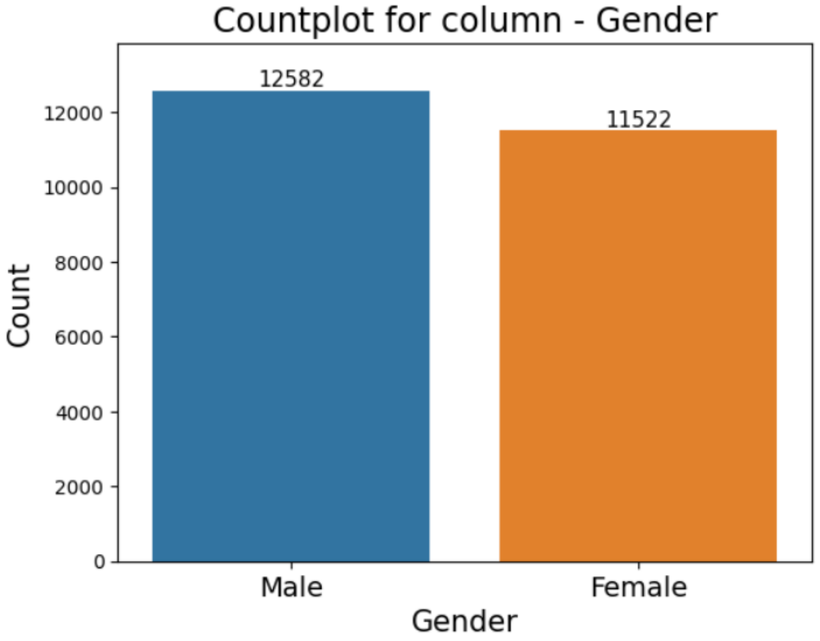
Each image information is contained in the filename:

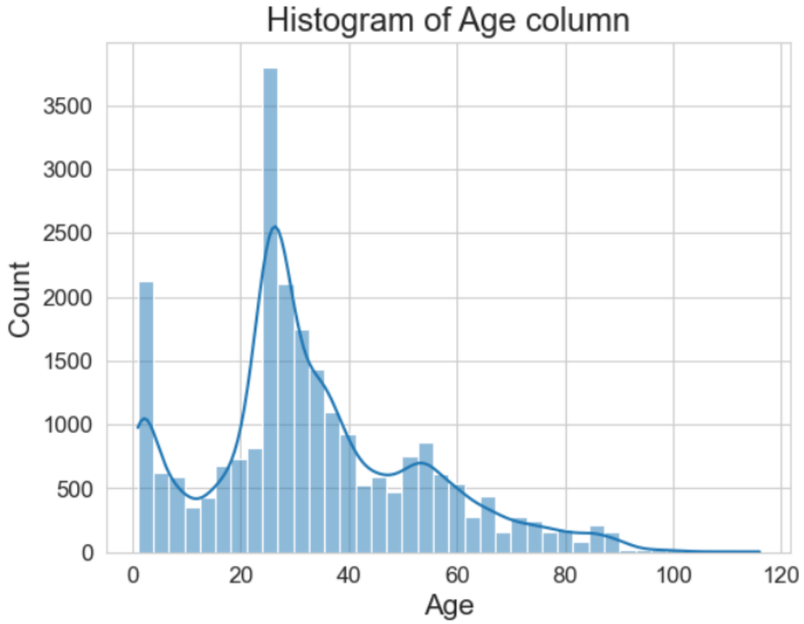
[AGE]_[GENDER]_[RACE]_[DATE&TIME].JPG

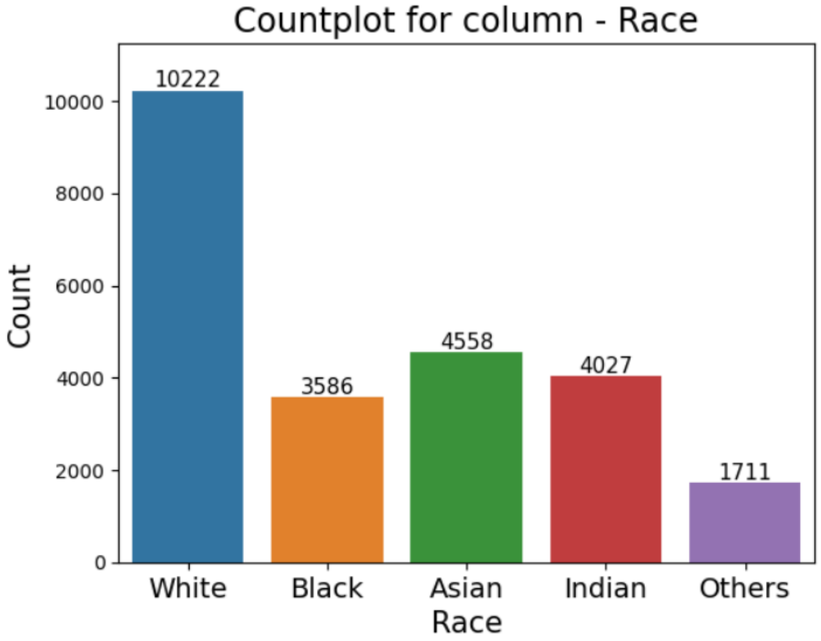
- **[AGE]** is an integer from 0 to 116, indicating the age
- **[GENDER]** is either 0 (male) or 1 (female)
- **[RACE]** is an integer from 0 to 4, denoting White, Black, Asian, Indian, and Others

Data Analysis

There are mostly middle-aged people, with approximately equal distribution of gender.

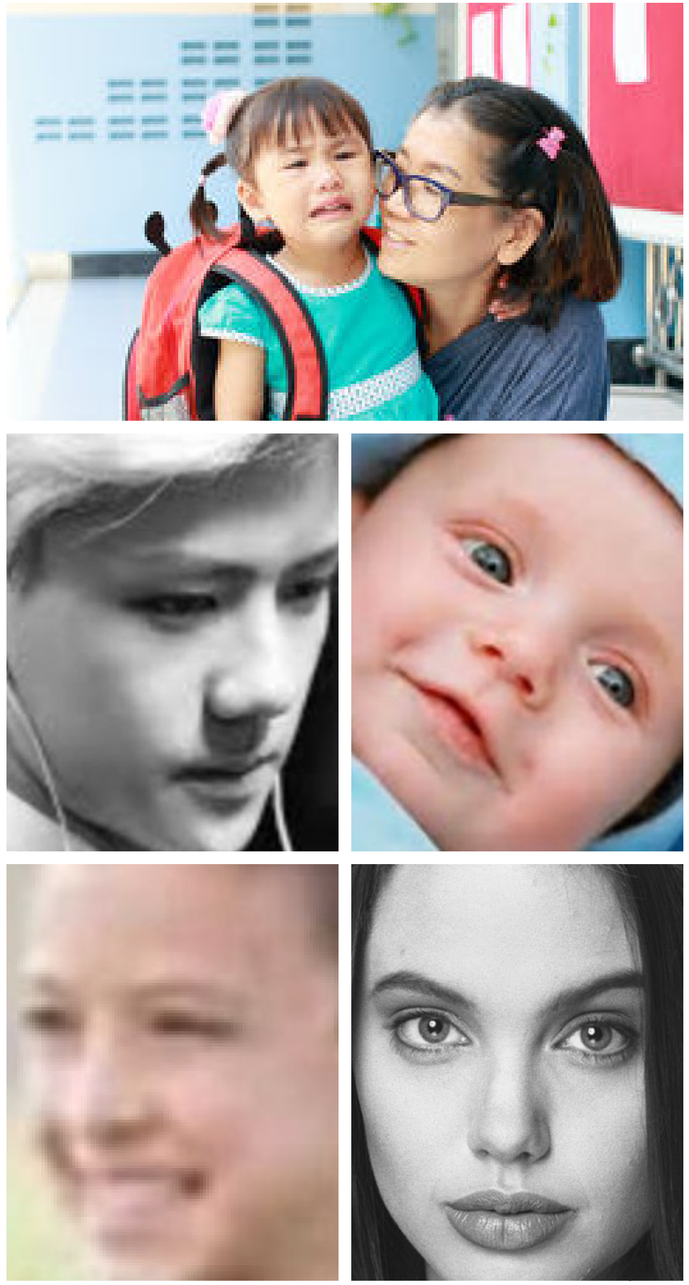






Specifications

- Rotated images
- Bad quality images
- Two or more faces
- Side looking images
- Black and white images
- Wrong Images



Observations

Wrong Data

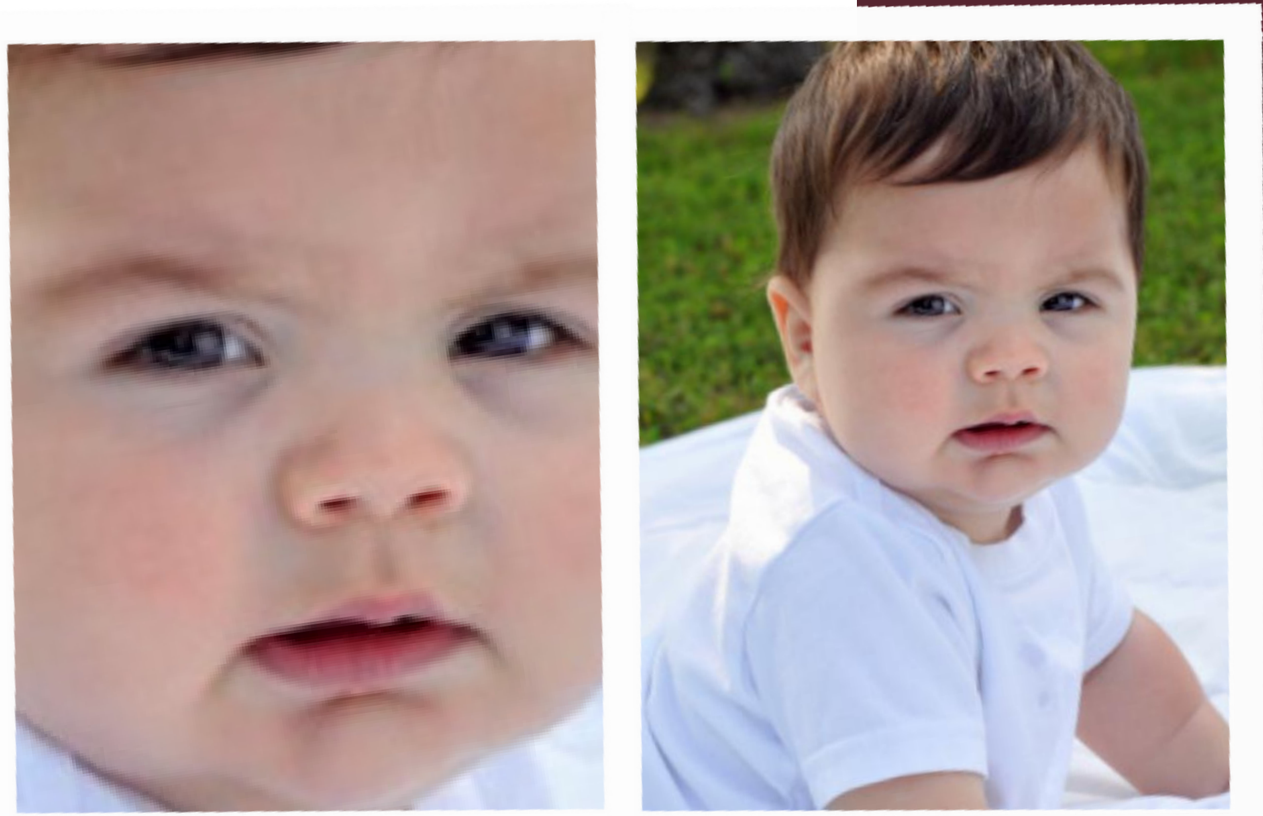
Wrong data were deleted from the main dataset.

```
images\39_1_20170116174525125.jpg
images\61_1_20170109142408075.jpg
images\61_3_20170109150557335.jpg
images\61_3_20170109150557335.jpg
```

Crop Faces

The first step of the project is to **crop the faces** from the images, to concentrate the predictions more on the face of the person.

Two or more face images, or images were no face was found, **were all removed**.



Face detection is done using library MTCNN.

Resizing the Face Images

224x224 Image

Resizing is needed to be able to **train** fixed-size-input CNNs.

Before resizing the images, the quality of the images were checked with certain threshold to be sure too **low-quality images are not used**.

All images were resized to **224x224** resolution. White paddings are added for the empty pixels to make all the images square

Saving in NumPy Files

All images are read and save into a numpy file for more convenient use.

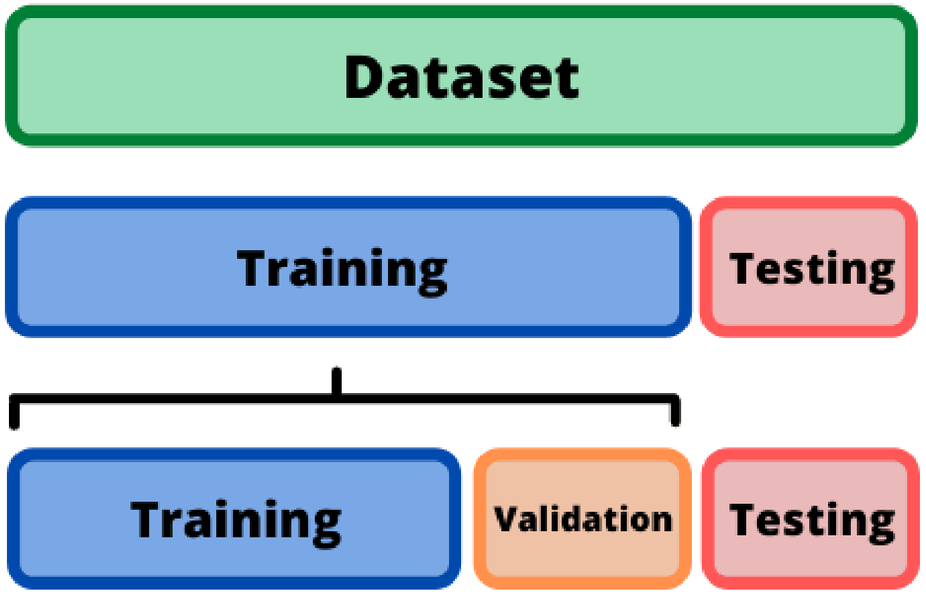
The observations are saved in batches, for example 5000-observations per file.

The labels are extracted and saved in a csv file.

The final total image data is of shape ()

Train-Val-Test Split

The dataset is divided into training, validation and test sets. The data is saved in separate folders.



Train Data - is used for estimating the parameters of the model.

Validation Data - is used for model selection, hyperparameter tuning, assessing the bias-variance tradeoff.

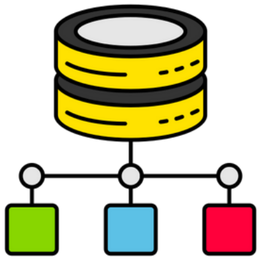
Test Data - is used for final evaluation of the final model.

Problem Requirements

Age Estimation

Gender Prediction

1) Creating the dataset



Creating the Dataset

Torch DataLoader → GetItem(Index)

- **Image** (Index) with swapped dimensions
- **Age** (Index)
- **Gender** (Index)

```
class TorchDatasetCreator(Dataset):
    def __init__(self, dataset):
        self.dataset = dataset

    def __len__(self):
        return len(self.dataset[0])

    def __getitem__(self, idx):
        image = self.dataset[0][idx]
        image = transforms.ToTensor()(image)
        age = self.dataset[1][idx]
        gender = self.dataset[2][idx]

        return image, age, gender
```

Train Loader

Test Loader

Validation Loader

Problem Requirements

Age Estimation

Gender Prediction

1) Creating the dataset

2) Cost - MSE or MAE

2) Cost - Binary Cross-Entropy

Cost Function

Cost/Loss function is different for both problems, because one case is a regression problem, the other one is a classification problem.

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log (1 - \hat{Y}_i))$$

Classification

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Regression

Problem Requirements

Age Estimation

Gender Prediction

1) Creating the dataset

2) Loss - MSE or MAE

2) Loss - Binary Cross-Entropy

3) Optimizer

Optimizer

Adam Optimizer

Is a go-to optimizer in many deep learning frameworks and applications due to its robust performance across a wide range of problems.

It's an extension of the basic gradient descent algorithm that incorporates several additional features to improve the efficiency and performance of the optimization process.

Easy to implement, is computationally efficient, has little memory requirements, needs less hyperparameter tuning, etc.

Diederik P. Kingma, Jimmy Ba (Dec 22, 2014). Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980

Problem Requirements

Age Estimation

1) Creating the dataset

2) Loss - MSE or MAE

3) Optimizer

4) Trainer function

Gender Prediction

1) Creating the dataset

2) Loss - Binary Cross-Entropy

3) Optimizer

4) Trainer function

Trainer Function

For each epoch

Running loss = 0

For each batch

Predict each batch	Optimizer updates the parameters
Calculate Loss	Zero out the accumulated gradients
Calculate the gradients	Add the loss to running loss

Calculate the loss for validation data

Calculate the accuracy for validation data (For classification)

Print the results of the epoch

Problem Requirements

Age Estimation

1) Creating the dataset

2) Loss - MSE or MAE

3) Optimizer

4) Trainer function

5) CNN

Gender Prediction

1) Creating the dataset

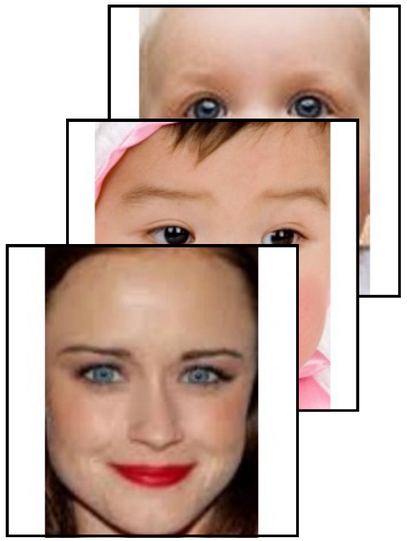
2) Loss - Binary Cross-Entropy

3) Optimizer

4) Trainer function

5) CNN

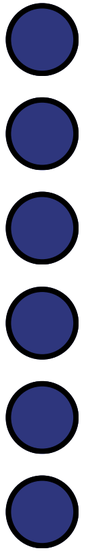
Convolutional Neural Network



ResNet34

+

FairFace Pretrained



Sigmoid (Classification)

+

Predictions

Adding new FC layer at the end

Problem Requirements

Age Estimation

Gender Prediction

1) Creating the dataset

2) Loss - MSE or MAE

2) Loss - Binary Cross-Entropy

3) Optimizer

4) Trainer function

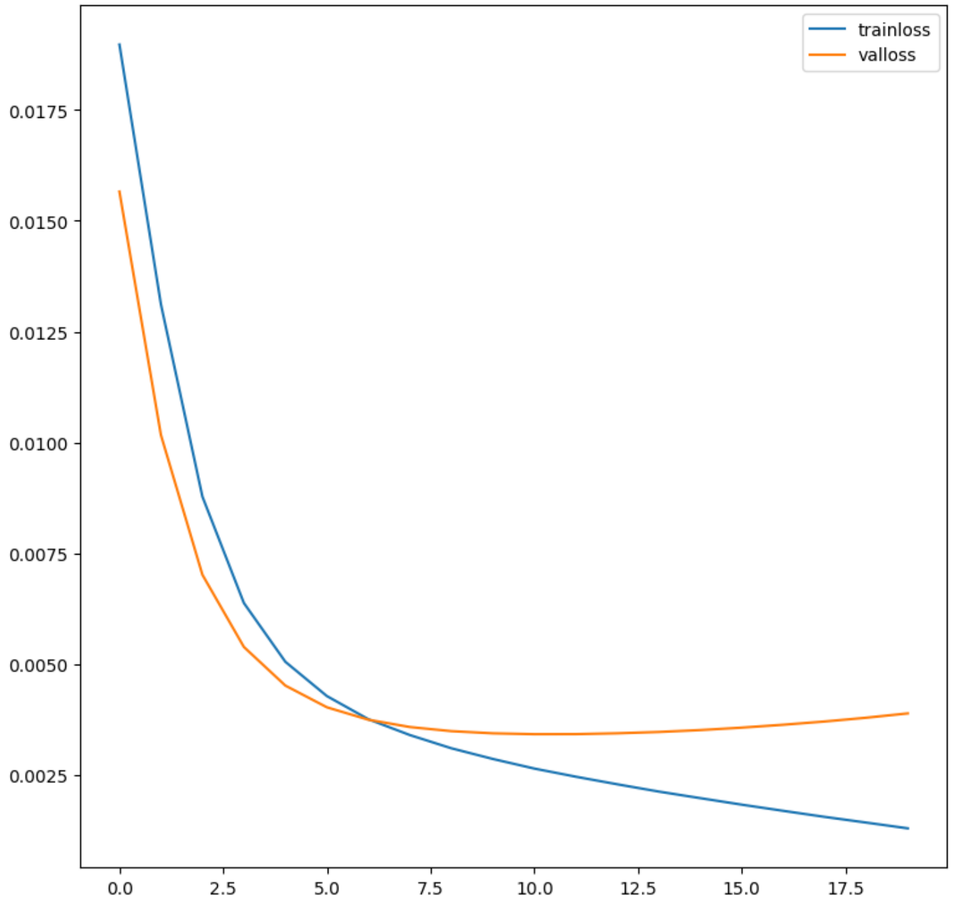
5) CNN

6) Results

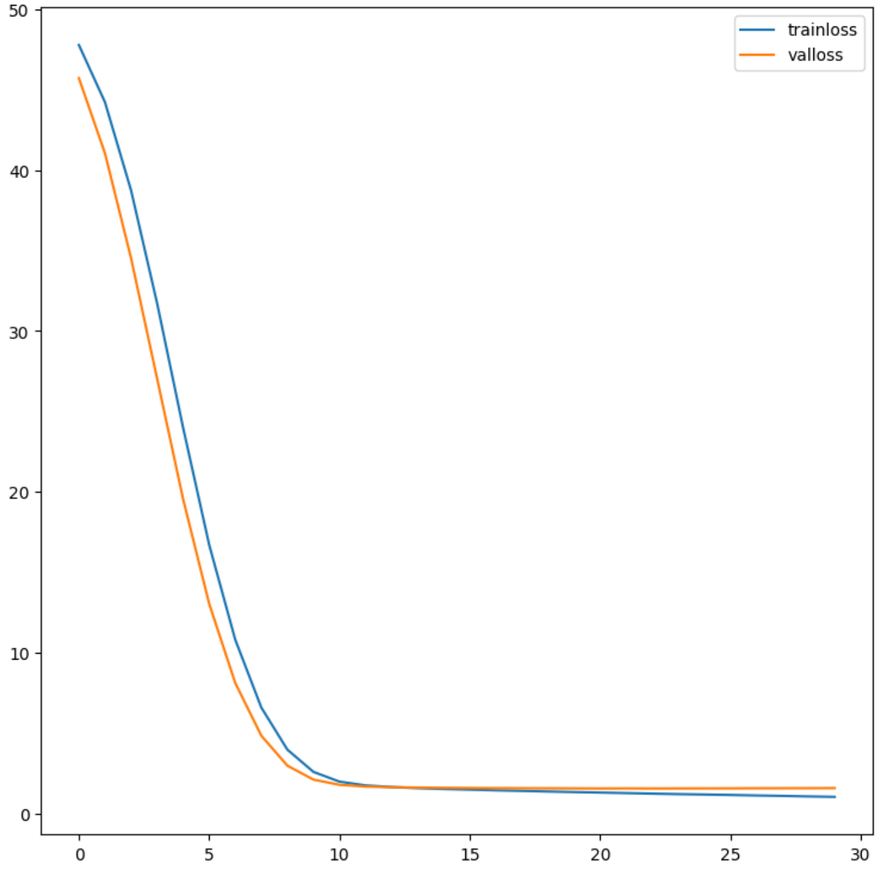
Results

(Learning Curves)

Gender Prediction



Age Prediction



Results

(Learning Curves)

Gender Prediction

Train Accuracy = 97.74%

Validation Accuracy = 96.1%

Test Accuracy = 96.05%

Age Prediction

Train Accuracy = 70.8%

Validation Accuracy = 63.4%

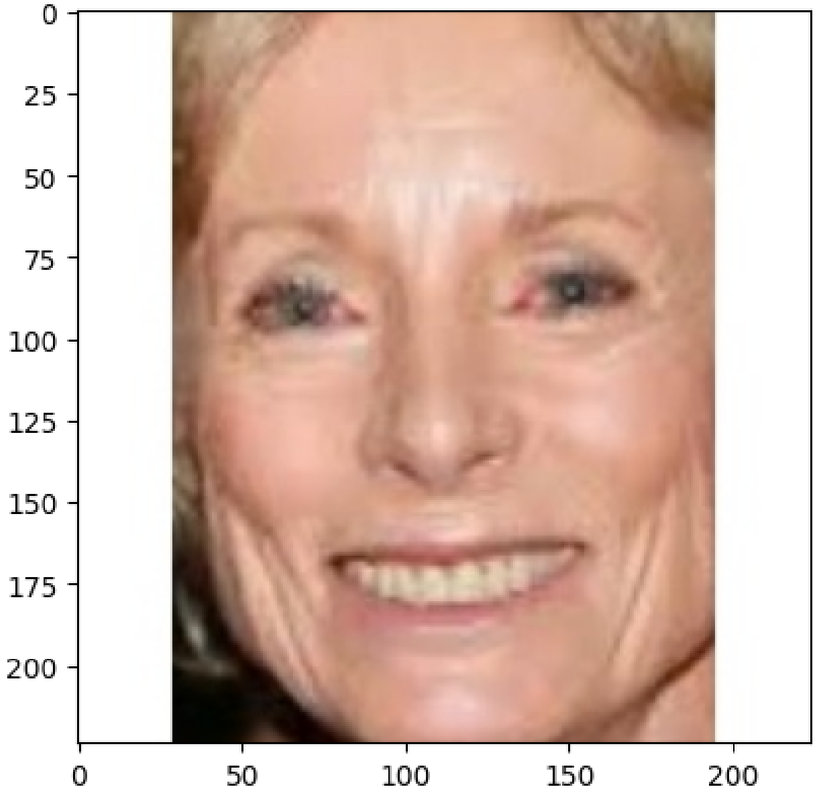
Test Accuracy = 63.2%

Train MAE = 4.22 years

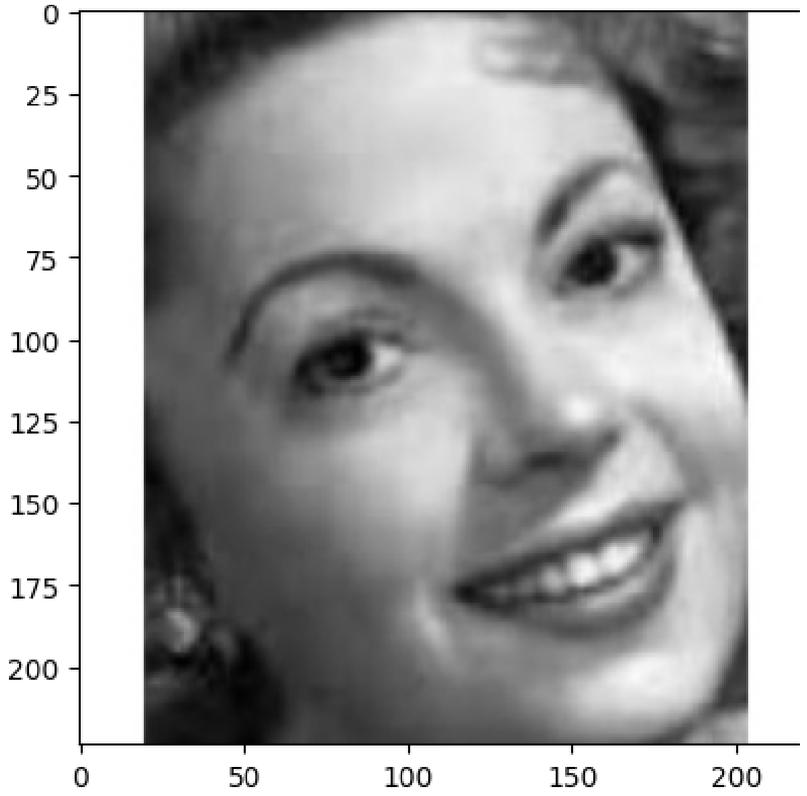
Validation MAE = 5.24 years

Test MAE = 5.18 years

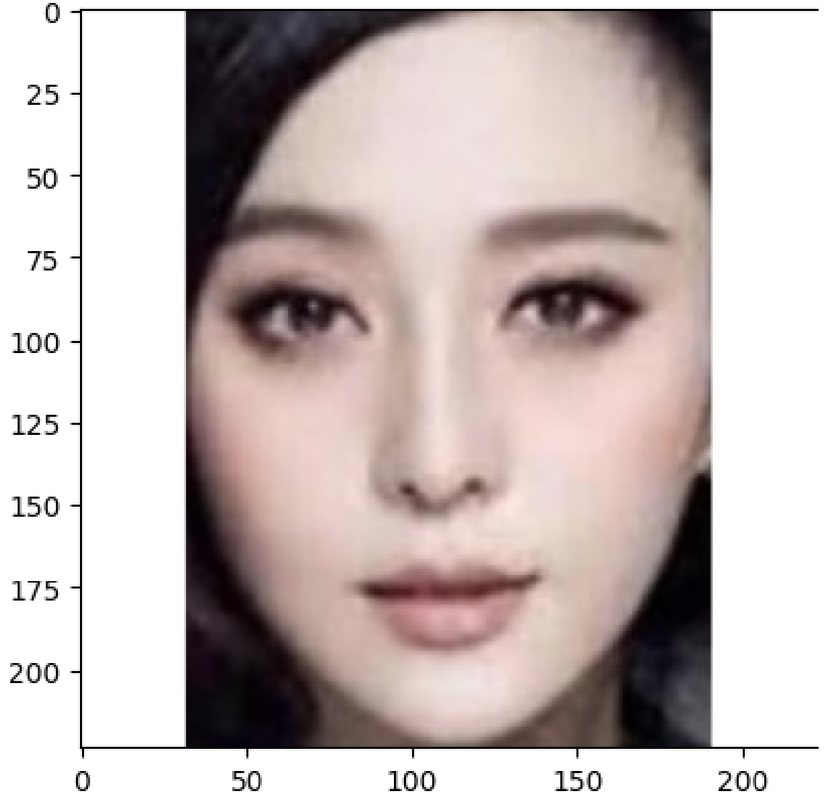
Wrong predictions for Gender




```
--------------------------------------------------
Correct label = Male
Predicted label = Female
Error = 0.9980271458625793
--------------------------------------------------
```

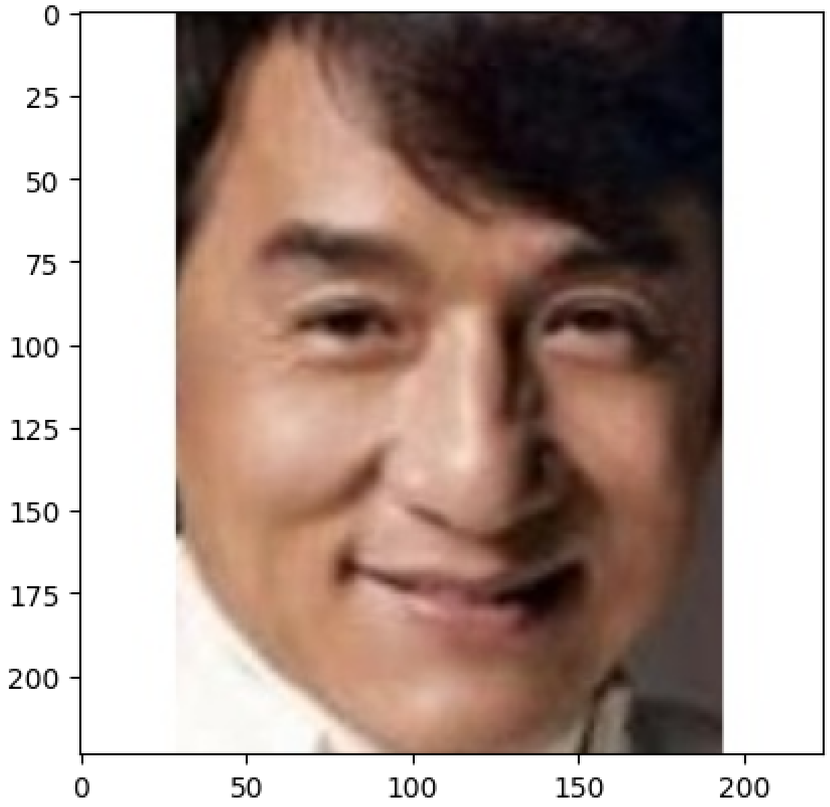



```
--------------------------------------------------
Correct label = Male
Predicted label = Female
Error = 0.9950774312019348
--------------------------------------------------
```

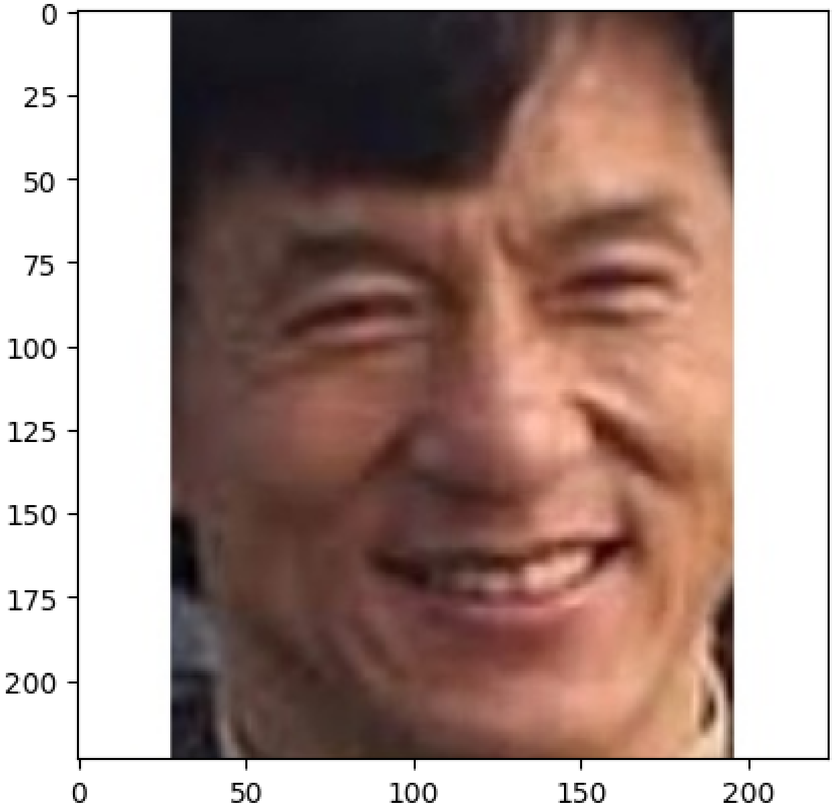



```
--------------------------------------------------
Correct label = Male
Predicted label = Female
Error = 0.991824209690094
--------------------------------------------------
```

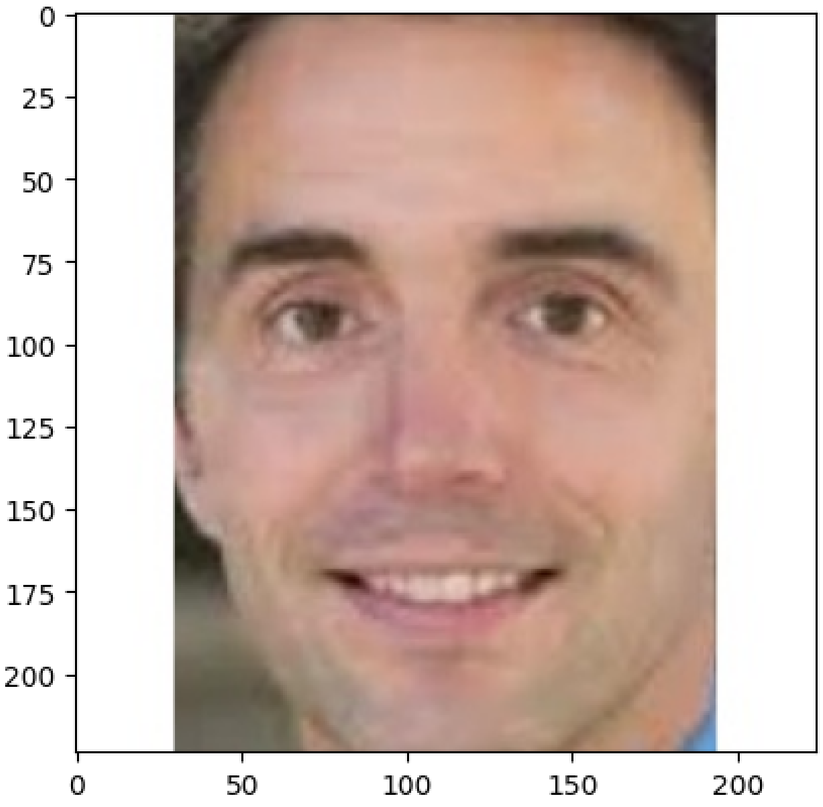
Wrong predictions for Age




```
----------------------------------------------------
Correct label = 59.0
Predicted label = 35.63019561767578
Error = 23.36980438232422
----------------------------------------------------
```




```
----------------------------------------------------
Correct label = 35.0
Predicted label = 58.19916915893555
Error = 23.199169158935547
----------------------------------------------------
```




```
----------------------------------------------------
Correct label = 51.0
Predicted label = 33.14531326293945
Error = 17.854686737060547
----------------------------------------------------
```

Thank you!