Age Group And Gender Classification Using Convolutional Neural Networks

By:

Sanchit Gautam

Introduction

- Our work aims to develop a CNN model and train it on a large-scale image dataset for the task of age group and gender prediction.
- The UTKFace dataset consists of 23,708 images.
- Convolutional Neural Network is used for the purpose.
- Model is evaluated on the basis of performance by using accuracy and AUC score as the performance measures.

Motivation

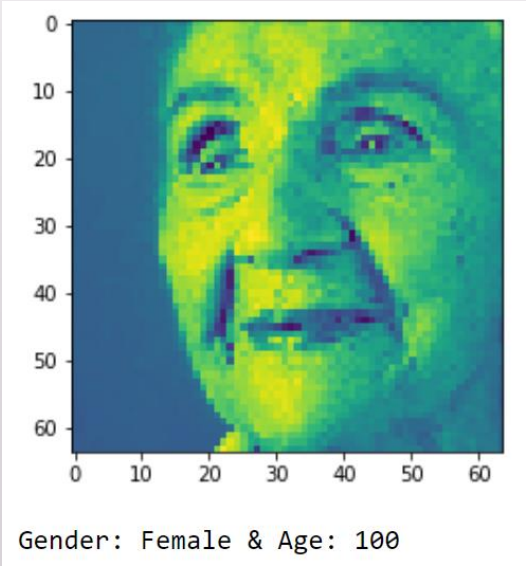
- From biometrics, security control to entertainment and commercial terminals, we find various applications of the task of automatic age and gender recognition.
- In this social media world, we can obtain a lot of information if we determine the age and gender from the images of a person.
- But a major challenge in this task is the nature of data itself, as there is a limited amount of data available on the internet having required age and gender labels.
- These are the reasons behind the motivation of building a simple CNN architecture for this task.

Dataset

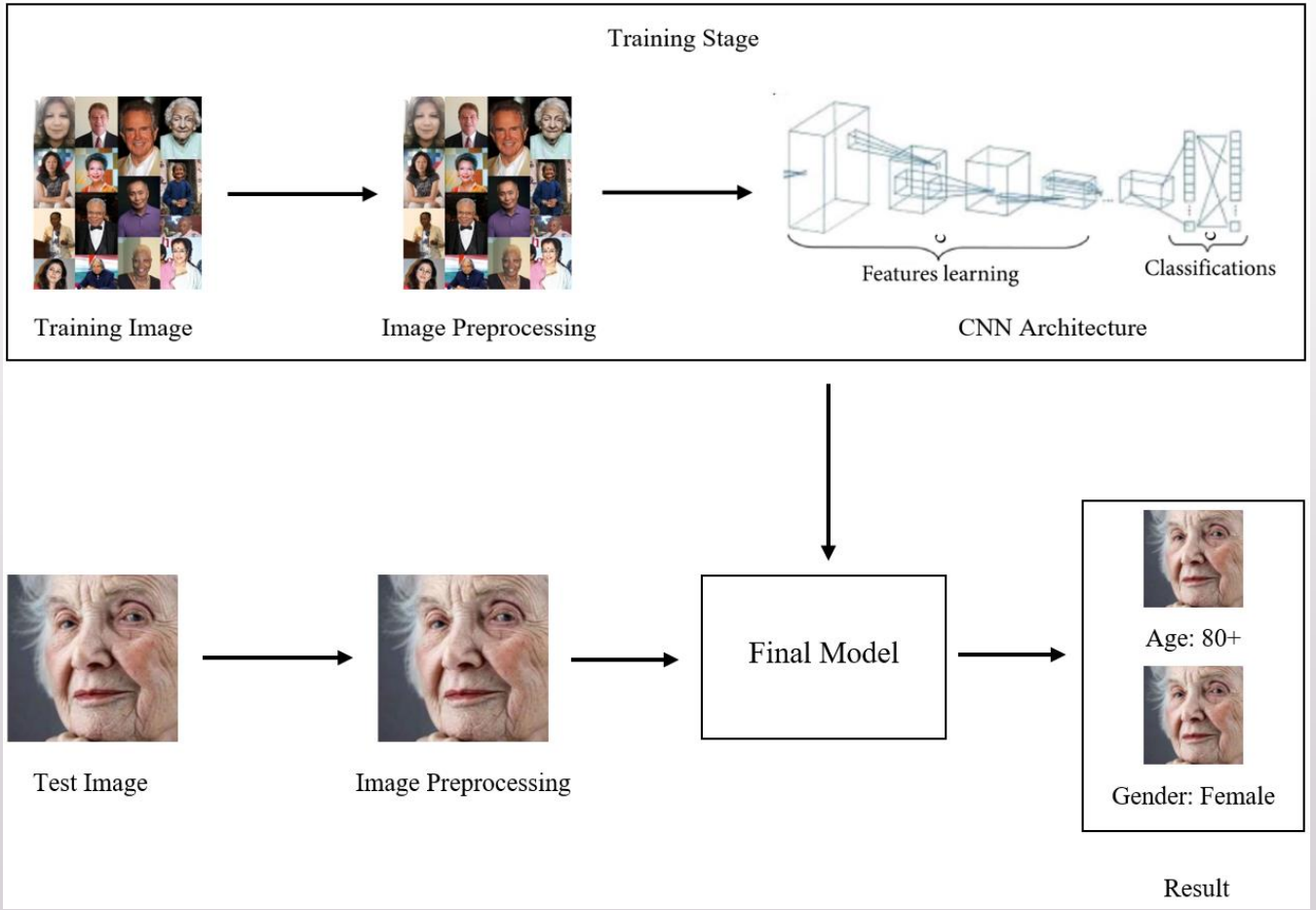
- Dataset I used is UTKFace images from https://susanqq.github.io/UTKFace/
- It is a large dataset consisting of face images with age span (ranging from 0 to 116 years old).
- The dataset contains 23708 face images with age, gender and ethnicity labels.
- This dataset contains images that covers large variations in pose, facial expression, resolution, etc.
- Labels for the images is saved in the file name in the following format ([age]_[gender]_[race]_[date&time].jpg).

Data Preprocessing

- We used cropped and aligned images are provided in the dataset.
- Performed resizing of the images to 64*64.
- Performed reshaping of the images to 64*64*1.



Experiment Design

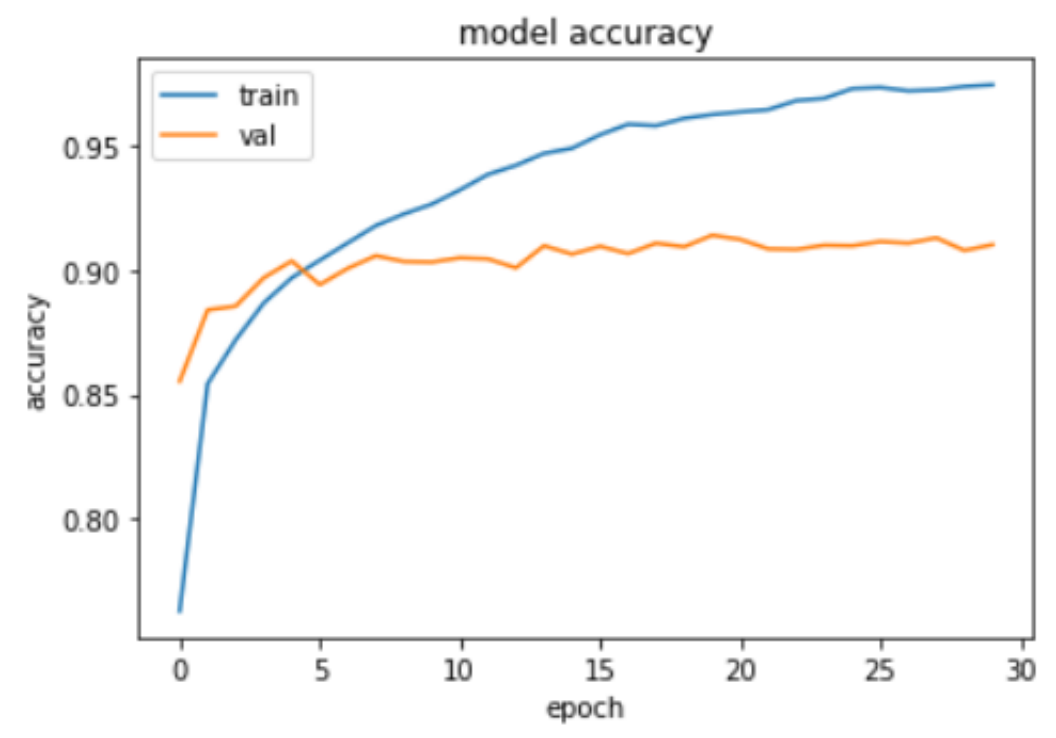


CNN Architecture

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 64, 64, 1)	0	
conv2d_1 (Conv2D)	(None, 62, 62, 32)	320	input_1[0][0]
conv2d_2 (Conv2D)	(None, 60, 60, 64)	18496	conv2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0	conv2d_2[0][0]
conv2d_3 (Conv2D)	(None, 28, 28, 128)	73856	max_pooling2d_1[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 128)	0	conv2d_3[0][0]
dropout_1 (Dropout)	(None, 14, 14, 128)	0	max_pooling2d_2[0][0]
flatten_1 (Flatten)	(None, 25088)	0	dropout_1[0][0]
dense_1 (Dense)	(None, 128)	3211392	flatten_1[0][0]
dense_3 (Dense)	(None, 128)	3211392	flatten_1[0][0]
dropout_2 (Dropout)	(None, 128)	0	dense_1[0][0]
dropout_3 (Dropout)	(None, 128)	0	dense_3[0][0]
dense_2 (Dense)	(None, 1)	129	dropout_2[0][0]
dense_4 (Dense)	(None, 1)	129	dropout_3[0][0]

Total params: 6,515,714
Trainable params: 6,515,714
Non-trainable params: 0

Results



Accuracy: 90.37%

AUC Score: 0.93

Conclusion and Future Work

- We have explored how well a simple CNN Model can perform on large scale data and which is a challenging task.
- There are some scopes to improve performance of this model in future:

1. Implementing data augmentation.
2. Further making the model more accurate and efficient.