Report: Machine Learning 2

University of Bath

CNN Models on

Age and Gender Classification

ID	Contribution	Agree with Contribution [Y/N]
249580664	50%	Y
249248029	50%	Y

Link to Model A: Click Here

Link to Model B: Click Here

Introduction

This assignment includes training two CNN models (custom and pre-trained) based on a subset of UTKFace dataset, consisting of 5,000 face images with annotations of age, gender, and ethnicity, covering a large variation in pose, facial expression, illumination, etc.

The target is to estimate a person's age and predict their gender based on face images. The age ranges from 0 to 116 and the gender is either 0 (male) or 1 (female).

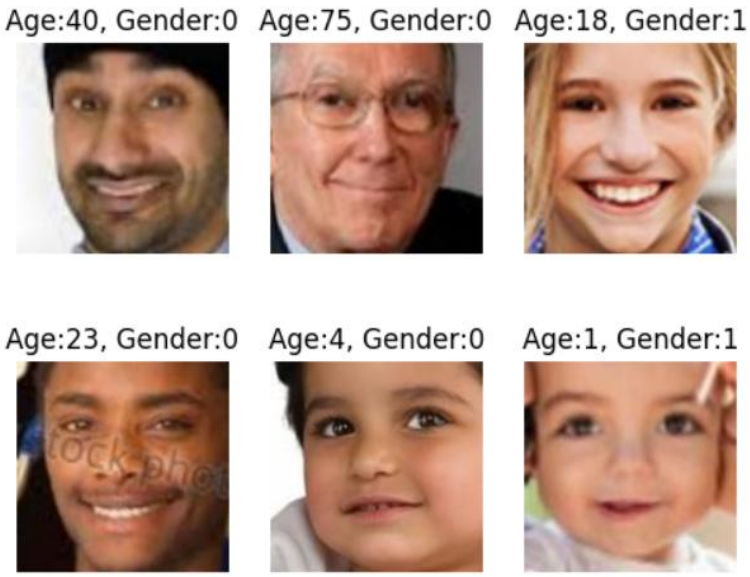

Figure 1: Example Images

The data is split into training and validation sets.

Since the image in the dataset is centred, we don't use shear and shift, and only apply horizontal flip, rotation, and zoom for data augmentation.

Figure 2: Example Augmented Images

The Custom CNN

Architecture

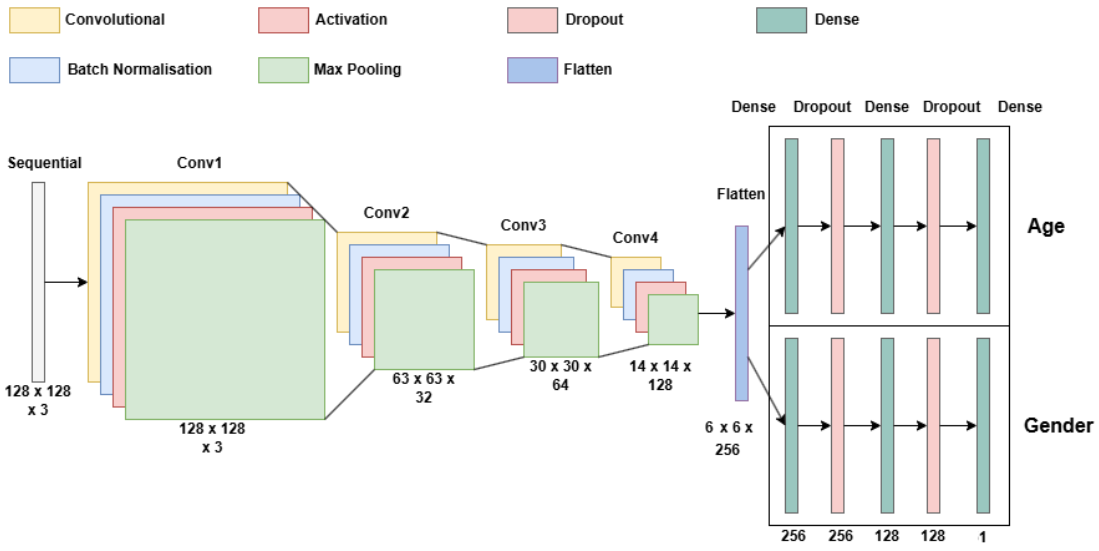

Figure 3: Architecture of Custom CNN

The sequential layer is employed for data augmentation.

The feature map size is restricted to 10 x 10 before the fully connected layer. To satisfy this restriction, we use zero padding in convolutional layers and stride 2 in max pooling layers to reduce the spatial dimensions of feature maps. The convolutional and max pooling kernel size is set to 3 x 3 and 2 x 2 respectively.

Then the output size after each convolutional block is computed as:

$$\text{Output size} = \frac{(n - 3 + 1 - 2)}{2} + 1 = \frac{(n - 4)}{2} + 1$$

After 4 blocks, the feature map dimensions progress as follows:

$$128 \rightarrow 63 \rightarrow 30 \rightarrow 12 \rightarrow 6$$

Where 6 x 6 meets the requirement of less than 10 x 10.

Batch normalization is applied between each convolutional layer and activation function to stabilize the training process. It is not applied to fully connected layers as it works similarly to dropout.

Flatten layer is applied as a transition between convolutional blocks and fully connected layer.

The dense layers are separated by gender and age. The dropout rate is set to 0.3. L2 regularization with lambda = 0.01 is applied in each dense layer.

For activation functions, ReLU is used in all convolutional and dense layers, and for output layers, sigmoid is used for gender binary classification and linear regression for age prediction.

Total parameters	Number of layers
5,174,530 (19.74 MB)	31

Training process

We chose Adam optimizer as it is good for binary classification and linear regression tasks.

A lower learning rate in a complex model generally improves performance by allowing more gradual convergence. Given the model's complexity, we set the learning rate to 1e-4, lower than the default value.

The loss functions and metrics are:

	Age	Gender
Loss	MAE (Mean Absolute Error)	Binary Cross-Entropy
Metric	MAE (Mean Absolute Error)	Accuracy

We use the same loss function and metric to avoid potential problems caused by a different loss function. For example, using MSE as a loss function could result in excessively large loss, as it squares the errors in age from 0 to 116.

To prevent overfitting, we used early stopping. If the validation loss does not decrease after 10 epochs with a tolerance of 0.01, training is stopped, and the best weights corresponding to the lowest validation loss are retained.

We use standard 32 batches and 100 epochs. The training dataset size is 4000, which means 125 weight updates per epoch.

After training, the model stopped early at epoch 76, with the best weights achieved at epoch 66. The running time is 19ms/step.

Performance

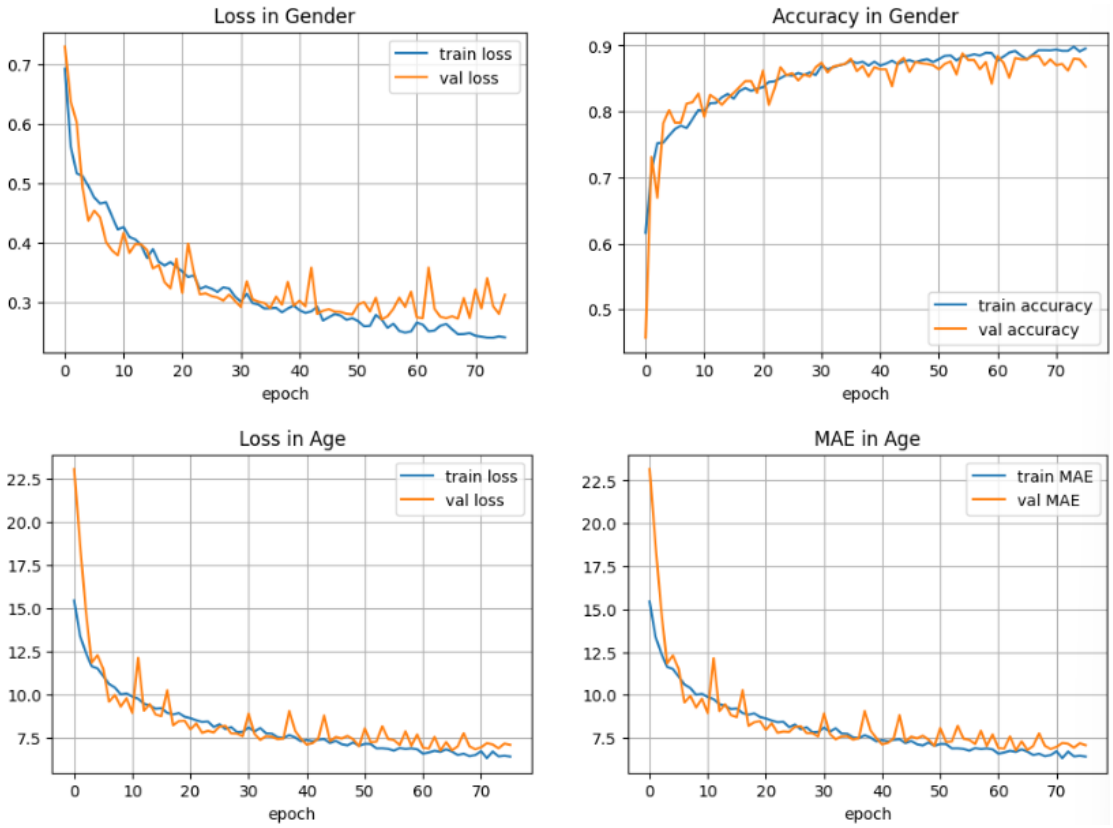

Figure 4: Learning Curve of Custom CNN

All curves plateau at a low value after several epochs, indicating the model is converged. It achieves 87% accuracy in gender classification and 7.1 MAE in age estimation, which is strong compared to the previous year's best result (90.2% accuracy and 5.66 MAE).

The consistent decrease in the train and validation loss for age suggests no overfitting or underfitting.

After 50 epochs, the training loss decreases in gender while the validation loss remains the same, indicating overfitting, possibly caused by poor regularization.

While the training curves are smooth, the validation curves show some fluctuations. This might be due to the small data size (4000), or a not representative validation dataset. It also could suggest mild overfitting, possibly caused by too many parameters. With 5 million parameters compared to only 4000 training samples, there will be a risk of overfitting if the data augmentation and regularization are insufficient.

The Pre-trained CNN

Architecture

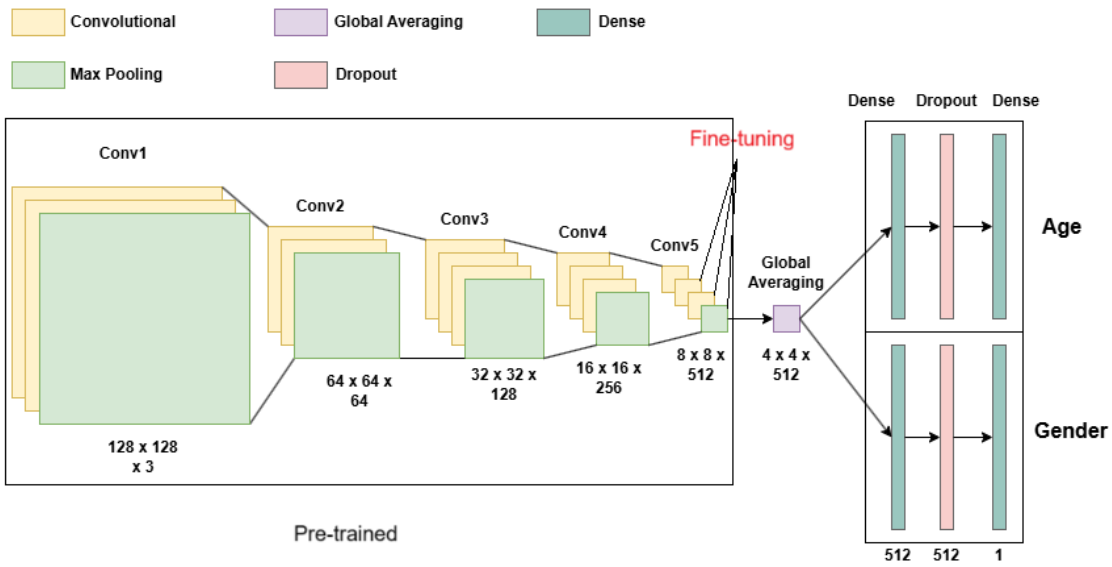

Figure 5: Architecture of Pre-trained CNN

We chose VGG16 model, pre-trained on ImageNet, as our base CNN.

The VGG16 consists of 5 convolutional blocks and top layers. We freeze the initial layers and fine-tune the last three layers.

The traditional transition layer between convolutional blocks and fully connected layers is flatten layer. However, in VGG16, the dimension of feature maps is large. Flatten layer would convert these maps into a long 1D vector, which could lead to many parameters, while global average pooling computes the average of each feature map, significantly reducing the spatial dimensions of the feature map and the number of parameters.

The mismatch of input size (224 x 224 x 3 for VGG16, 128 x 128 x 3 for our dataset) means that we must replace the fully connected layers of the pre-trained model by an untrained dense layer. The dropout rate is set to 50%. The regularizer and activation function are the same as custom CNN.

Total parameters	15,241,026 (58.14 MB)
Trainable parameters	7,605,762 (29.01 MB)
Non-trainable parameters	7,635,264 (29.13 MB)
Number of layers	25

Training process

Fine-tuning is typically done with a very low learning rate, and with the SGD optimizer rather than an adaptive learning rate optimizer such as RMSProp. We chose SGD optimizer, which refers to Mini-batch Gradient Descent, and set learning rate to 1e-4.

The loss function, metrics, batch size, and number of epochs are the same as custom CNN.

Compared to the dataset that VGG16 was originally trained on (1.4M), our dataset is significantly small (4000). The pre-trained model converges faster, and the training must stop earlier to prevent overfitting. We set patience to 5, and other parameters remain the same as custom CNN.

After training, the model stopped early at epoch 49, with the best weights achieved at epoch 44. The running time is 66ms/step.

Performance

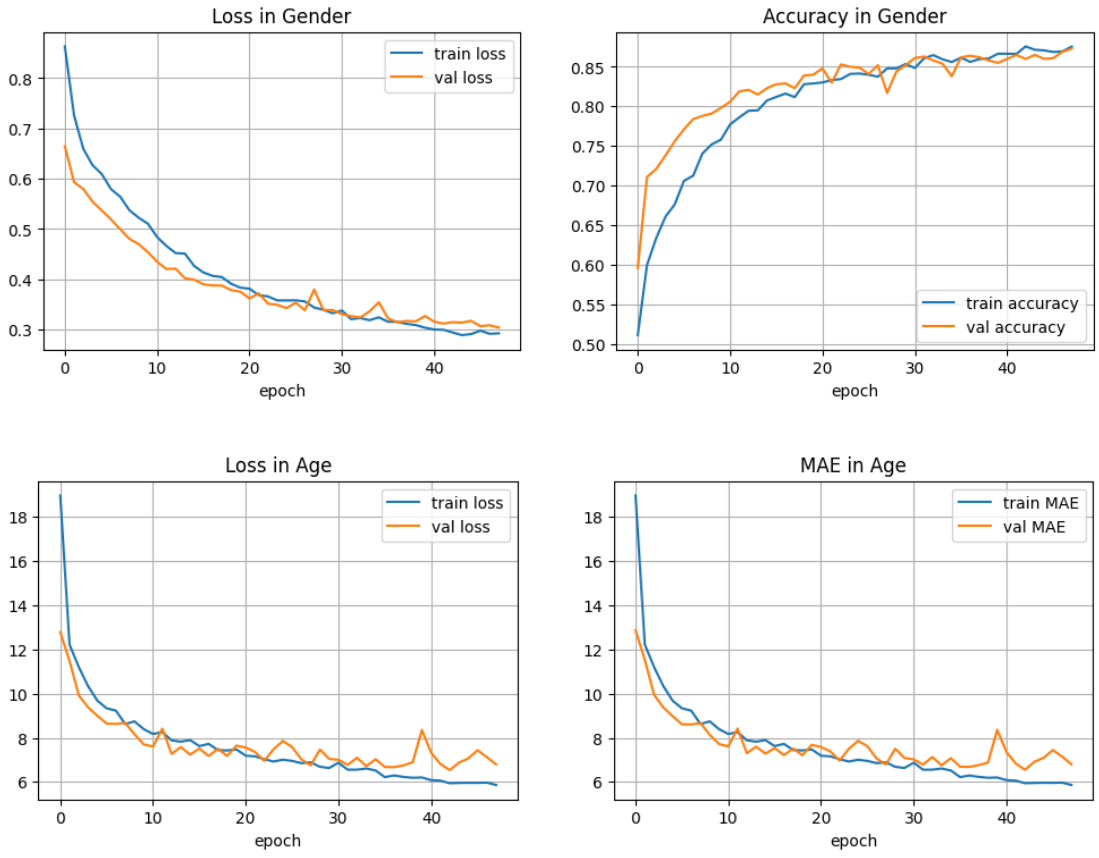

Figure 6: Learning Curve of Pre-trained C

The curves demonstrate that the model is converged. It achieves 87% accuracy in gender classification and 6.8 MAE in age estimation, which is strong compared to the previous year's best result (90.1% accuracy and 5.62 MAE).

The consistent decrease in the train and validation loss for gender suggests no overfitting or underfitting.

After epoch 35, the train loss decreases while the validation loss remains the same, indicating overfitting. This could be due to the higher complexity of age estimation compared to gender classification.

The training curves are smooth, and the validation curve fluctuates less than those of the custom CNN, suggesting less overfitting.

Summary and Discussion

We compare and discuss the two CNN models in three aspects: architecture, training process, and performance.

Architecture

	Custom CNN	Pre-trained CNN
Parameters	5 millions	15 millions
Complexity	Low	High
Total Layers	31	25
Number of Convolutional Blocks	4	5
Number of convolutional layers in each block	1	2 or 3
Number of Dense Layers	3	2
Transition Layer	Flatten	Global Average Pooling
Data augmentation	In the model	Outside the model
Feature Map Size	6 x 6 x 256	4 x 4 x 512
Original Input Size	128 x 128 x 3	224 x 224 x 3
Original Training Dataset Size	4000	1.4 million

Training Process

	Custom CNN	Pre-trained CNN
Optimiser	Adam	SGD
Learning Rate	1e-4	1e-4
Early Stopping	Yes	Yes
Stop Epoch	76	49
Best Epoch	66	44
Running time	19ms/step	66ms/step

Performance

	Custom CNN	Pre-trained CNN
Converge	Yes	Yes
Converge Speed	Slow	Fast
Result	87%, 7.1	87%, 6.8
Overfit/Underfit	Overfits in gender after 50 epochs	Overfit in age after 35 epochs
Validation curve	Strong Fluctuations	Small Fluctuations
Training curve	Smooth	Smooth

Discussion

The overall performance of pre-trained CNN is better than custom CNN, demonstrated by faster convergence speed, slightly better results, and smaller fluctuations in validation curves.

When there's insufficient data, the validation dataset will not be large enough to be representative, resulting in a bad validation curve.

In real life, it is unlikely to have enough data sampling.

Instead of training a CNN from scratch based on our small dataset, we could fine-tune a model pre-trained on the large dataset.

In this assignment, nearly half of the parameters in the pre-trained model are trainable. However, in real life, most of the parameters are pre-trained weights, which are non-trainable, and the trainable parameters will be significantly fewer than the parameters in custom CNN, resulting in reducing computation resources and running time.

Additionally, research suggests that when the dataset is small and similar to the original dataset of the pre-trained model, the pre-trained model performs better, otherwise the custom model performs better as it is trained on our dataset features.

References

Song, Y., Zhang, Z., Qi, H., 2018. *UTKFace* [Online]. Kaggle. Available from: https://www.kaggle.com/datasets/jangedoo/utkface-new [Accessed 14 March 2025].

V., 2023. *Analysis of Fine-Tuned vs. Custom Models: Impacts of Dataset Size and Similarity on Machine Learning Performance* [Online]. Medium. Available from: https://medium.com/@schu1678/analysis-of-fine-tuned-vs-29948766094a [Accessed 14 March 2025].

Simonyan, K., Zisserman, A., 2014, *Very Deep Convolutional Networks for Large-Scale Image Recognition*(v6) [computer program], Available from: https://doi.org/10.48550/arXiv.1409.1556 [Accessed 14 March 2025].