Age and Gender Classification Using Convolutional Neural Networks

Candidate Number: 06581
University of Bath

November 28, 2025

Model files:
Model A: https://drive.google.com/file/d/1Ov8IhEn_XFFpLteQ4srk8unwCviGJrUR/view?usp=sharing
Model B: https://drive.google.com/file/d/1MiKz51OFIISj58-R-AD_452BhweScJI5/view?usp=sharing

1. Introduction

The goal of this project is to classify a person's gender and estimate their age from a single RGB facial image. Both tasks are challenging because faces vary widely across individuals and conditions, including differences in race, pose, lighting, and expression. Age estimation is particularly difficult since people age at different rates and display age related features in different ways. For example, some individuals gain visible wrinkles or white hair earlier than others, which increases variation and makes the task difficult. Gender classification is generally easier because certain facial characteristics tend to differ between males and females, although these cues are not universal. It is also important to acknowledge that automatic gender prediction raises ethical concerns for people who do not identify within a binary category. Any system that attempts to infer gender from appearance risks reinforcing restrictive ideas about how gender should look, so these considerations should be recognised when developing models that work with images of people.

The dataset used in this project is a subset of 5000 images taken from the *UTKFace* collection, which contains more than 20000 facial images labelled with age, gender, and ethnicity. All images in the subset were provided already, cropped, and resized to 128×128 pixels in RGB format. Each sample includes an integer age label and a binary gender label. The *UTKFace* dataset was originally introduced in Zhang, Song, and Qi, 2017 and is available only for non-commercial academic research.

Traditional fully connected neural networks and shallow classifiers are not well suited for visual data because they treat each pixel as an independent feature and therefore ignore spatial locality. Convolutional neural networks address this by applying learnable filters that capture edges, textures, and facial structures at multiple levels of abstraction. Pooling operations then reduce the spatial dimensions while preserving important information, which helps CNNs generalise well to unseen faces.

This project aims to:

1. Design and train a custom CNN model from scratch for gender classification and age prediction on the UTKFace subset.
2. Implement and fine-tune a pretrained CNN using transfer learning to compare performance and training efficiency.
3. Analyse and interpret the performance of both models with respect to accuracy, mean absolute error (MAE), and training stability.
4. Discuss challenges such as dataset imbalance, model overfitting, and computational limitations.

The project does not attempt to achieve cutting edge performance; rather, its aim is to provide a comparative analysis of two paradigms in deep learning: building a CNN from scratch versus fine-tuning an existing one.

Prior work supports the expectation that a finetuned network will outperform a model trained from scratch when only a limited dataset is available. Özbulak, Aytar, and Ekenel, 2016 examined age and gender prediction on the Adience benchmark and compared a task specific CNN trained from scratch with two transferred models based on AlexNet and VGGFace. Both transferred models achieved clearly higher accuracy than the scratch model, with gains of about 7% for age classification and about 4.5% for gender classification. The authors noted that the task specific model was trained from scratch using a limited amount of data and concluded that reusing a deep pretrained representation provides stronger features and better generalisation. They also reported that transferring from a closely related domain such as face recognition yields the largest benefit. This evidence aligns with the design of the present project and supports the expectation that the pretrained model will achieve better performance.

The remainder of this report is structured as follows. Section 2 describes the design and implementation of the custom CNN model, including data preparation, augmentation, and training parameters. Section 3 presents the transfer-learning approach using a pretrained network and compares its performance with the baseline model. Section 4 discusses and concludes results, limitations, and potential improvements.

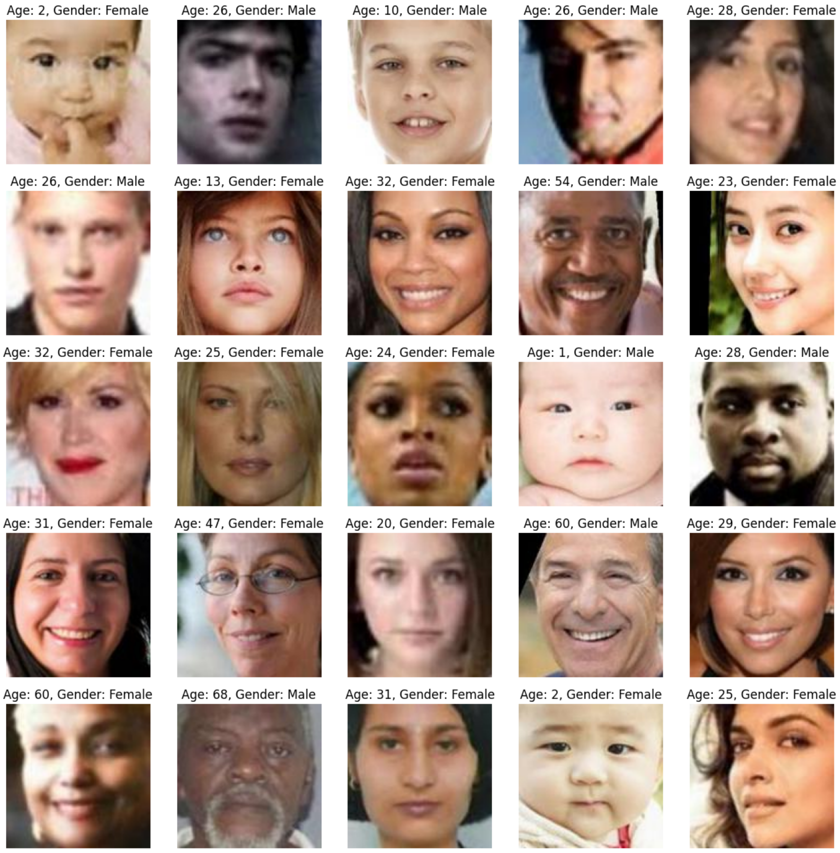


Figure 1: Sample images from the dataset used for training and evaluation.

2. Custom CNN

2.1. Dataset Analysis

The dataset used in this project consists of 5000 sampled facial images from the *UTKFace* dataset. Each sample includes an integer age label and a binary gender label, where males are encoded as zero and females as one. All images were provided in pre-aligned and cropped RGB format with a resolution of 128×128 pixels.

Although the age labels range from 1 to 116 years, the dataset contains only 96 distinct age values, as shown in Figure 2, and their frequencies vary greatly as well. The mean number of samples per age is about 52, but the median is only 32, which indicates that many ages have very few examples. The most noticeable concentration of samples occurs around ages 1 to 3 and 24 to 26, while most other ages are under-represented. For instance, age 26 appears 517 times, whereas age 99 appears only once, and ages such as 97 and 101 to 115 do not appear at all. This strong imbalance, together with the long tail of extremely rare ages, limits the model's ability to learn a smooth and generalisable regression function across the full age range.

A further consideration for the age task is the choice of prediction formulation. There are two common approaches in the literature. The first is to treat age as a continuous variable and train a regression model. The second is to convert age into discrete groups, for example ten year bins, and train a classification model over these grouped categories. The bin based approach has the benefit of increasing the number of samples available for each group, which can stabilise training and reduce the effect of in-

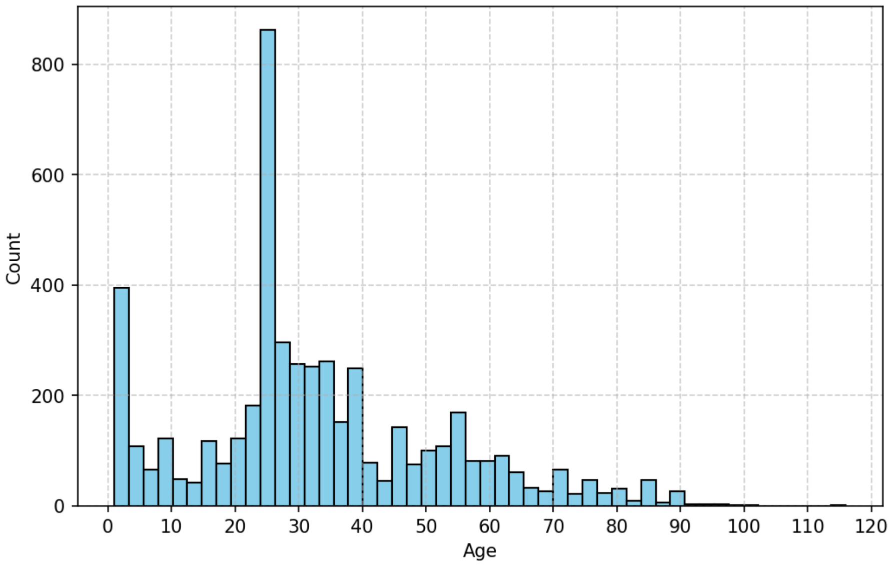


Figure 2: Age distribution within the dataset subset used for training and validation.

correct individual age labels. However, it removes fine detail by forcing all ages inside a group to share the same label, and it introduces boundaries that do not reflect the continuous nature of age. For this reason, the regression approach is more suitable for this project. It preserves all available information and allows the model to learn a smooth relationship across the entire age range.

In contrast to the age distribution, the gender distribution is relatively balanced. The dataset contains 2416 male images and 2584 female images, giving a median of roughly 2500 samples per gender class. This balance is likely to support strong performance in the binary gender classification branch of the model.

A further limitation is the presence of label noise. The UTKFace dataset was collected by scraping publicly available images, and the age and gender labels were likely produced by an automatic prediction system with only limited manual checking. This means that some samples may be incorrectly labelled, especially for age. Such noise can affect the reliability of the regression results and should be considered when interpreting the model's performance.

This may help explain some bias in the dataset, however, understanding this lies beyond the project scope. Class imbalance can hinder model generalisation, particularly in age regression, and therefore, the approach taken to the age problem is one of regression rather than classification, whereas binary classification is used for the gender problem.

2.2. Data Preprocessing

The training and validation sets were split to have images of 4000 and 1000, respectively. To improve model robustness and reduce overfitting, random data augmentations were introduced through Keras preprocessing layers:

- `RandomFlip` used in the horizontal direction, which randomly mirrors the image from left to right or right to left
- `RandomRotation` with a factor of 0.10, which randomly rotates the image by up to 36 degrees in

either direction

- RandomZoom with a factor of 0.10
- RandomTranslation with a factor of 0.10 for both height and width
- RandomContrast with a factor of 0.10, adjusting each pixel value according to $(x - \mathit{mean}) \cdot \mathit{contrast\ factor} + \mathit{mean}$

These augmentations intentionally modify the appearance of the training images. Although the transformed samples may not always reflect typical real world conditions, they help the model learn from both normal and extreme cases. Augmentations can temporarily distort the distribution of the dataset and produce images that appear unnatural. However, because the transformations vary randomly across batches and epochs, their effects average out over time. This is also the reason small factor values were chosen.

For this reason, the augmentations were applied inside the model rather than directly altering the stored training data. This ensures that each training pass receives a different augmented version of the same image, which increases the effective diversity of the dataset without requiring additional data collection.

The prefetch step was set to the automatic tuning option. This allows the input pipeline to load the next batch while the model is training on the current one. This process can reduce the overall step time and keeps the training process supplied with data at the right pace.

2.3. Model Architecture

The CNN consists of four convolutional blocks, as shown in Figure 3. Each block contains a convolutional layer followed by batch normalisation, a ReLU activation layer, and a 2×2 max pooling layer. All convolutional layers use 3×3 kernels with the padding option same, which adds zeros around the input so that the spatial dimensions of the output remain unchanged. This helps preserve information near the image boundaries, where important features may be lost during convolution.

The ReLU activation function is a simple and effective choice that returns the input when it is greater than 0 and otherwise outputs 0. It is widely used because it is computationally inexpensive and supports stable optimisation.

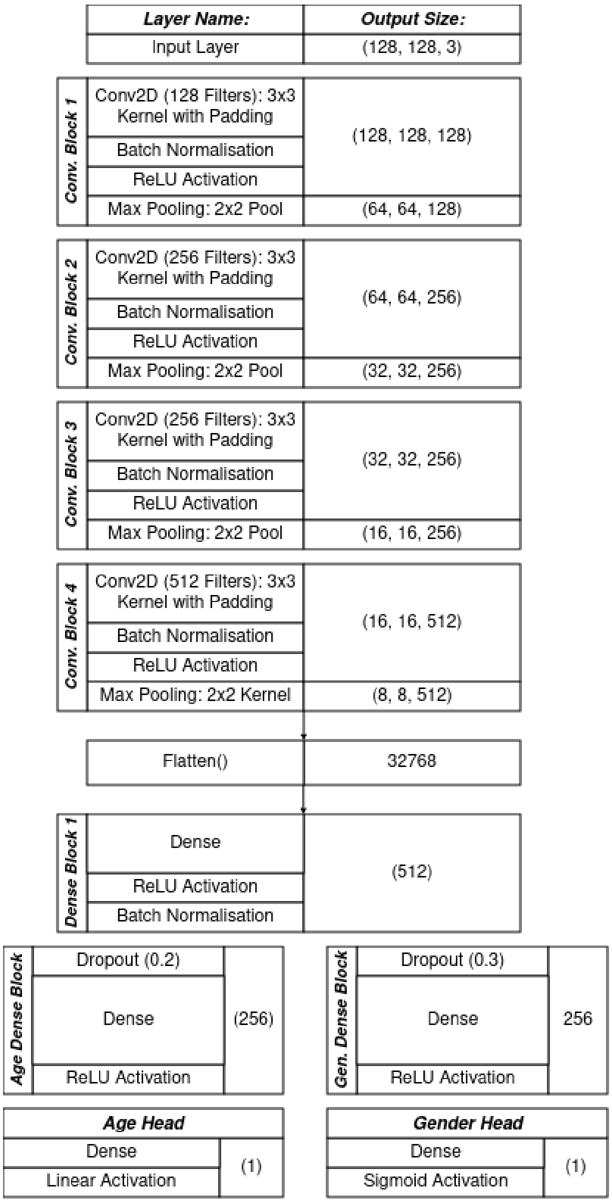
Although flattening greatly increases the number of trainable parameters, it preserves all feature values by concatenating the entire set of feature maps into a single vector. In this project, the validation curve was more stable when using Flatten compared with GlobalAveragePooling2D. The resulting representation is passed through a fully connected layer of 512 neurons with ReLU activation, followed by dropout values of 0.20 for the age head and 0.30 for the gender head. Each head contains a Dense layer of 256 units, followed by a ReLU activation.

Different dropout values were used in the first Dense block of each output branch, and each head was given its own separate second Dense layer. This separation allows the two tasks to regulate overfitting independently. Providing each head with its own Dense layer also enables the network to learn task specific features rather than forcing both outputs to rely on a shared representation.

For the gender prediction task, the sigmoid activation is appropriate because it maps the output to a range between 0 and 1, which aligns with the binary nature of the target. For the age regression task, a linear activation was used, since age is a continuous variable. The linear function does not restrict the output range, allowing the network to produce any real valued age prediction.

Batch normalisation was applied before the activation function in each block. This follows the standard design in modern convolutional networks, where normalisation is applied to the linear output of a layer and the activation is applied afterwards. Although the reverse order appears in some architectures, the batch normalisation then activation arrangement is widely used and was adopted here for consistency. The only exception was the Dense layers with 256 units in the two output branches, where performance was observed to be better without batch normalisation.

The bias term in each convolution and Dense layer was disabled because these layers were immediately followed by batch normalisation. Since batch normalisation includes its own learned offset, any bias from the previous layer would be redundant. Removing these unused bias parameters reduces computational cost without affecting the behaviour of the model.



	Layer Name:	Output Size:
	Input Layer	(128, 128, 3)
Conv. Block 1	Conv2D (128 Filters): 3x3 Kernel with Padding	(128, 128, 128)
	Batch Normalisation	
	ReLU Activation	
	Max Pooling: 2x2 Pool	(64, 64, 128)
Conv. Block 2	Conv2D (256 Filters): 3x3 Kernel with Padding	(64, 64, 256)
	Batch Normalisation	
	ReLU Activation	
	Max Pooling: 2x2 Pool	(32, 32, 256)
Conv. Block 3	Conv2D (256 Filters): 3x3 Kernel with Padding	(32, 32, 256)
	Batch Normalisation	
	ReLU Activation	
	Max Pooling: 2x2 Pool	(16, 16, 256)
Conv. Block 4	Conv2D (512 Filters): 3x3 Kernel with Padding	(16, 16, 512)
	Batch Normalisation	
	ReLU Activation	
	Max Pooling: 2x2 Kernel	(8, 8, 512)
	Flatten()	32768
Dense Block 1	Dense	(512)
	ReLU Activation	
	Batch Normalisation	

Age Dense Block			Gen. Dense Block		
	Dropout (0.2)	(256)		Dropout (0.3)	256
	Dense			Dense	
	ReLU Activation			ReLU Activation	

Age Head		Gender Head	
Dense	(1)	Dense	(1)
Linear Activation		Sigmoid Activation	

Figure 3: Overall architecture of the custom convolutional neural network with four convolution stages, a fully connected head, and two output branches for gender and age prediction.

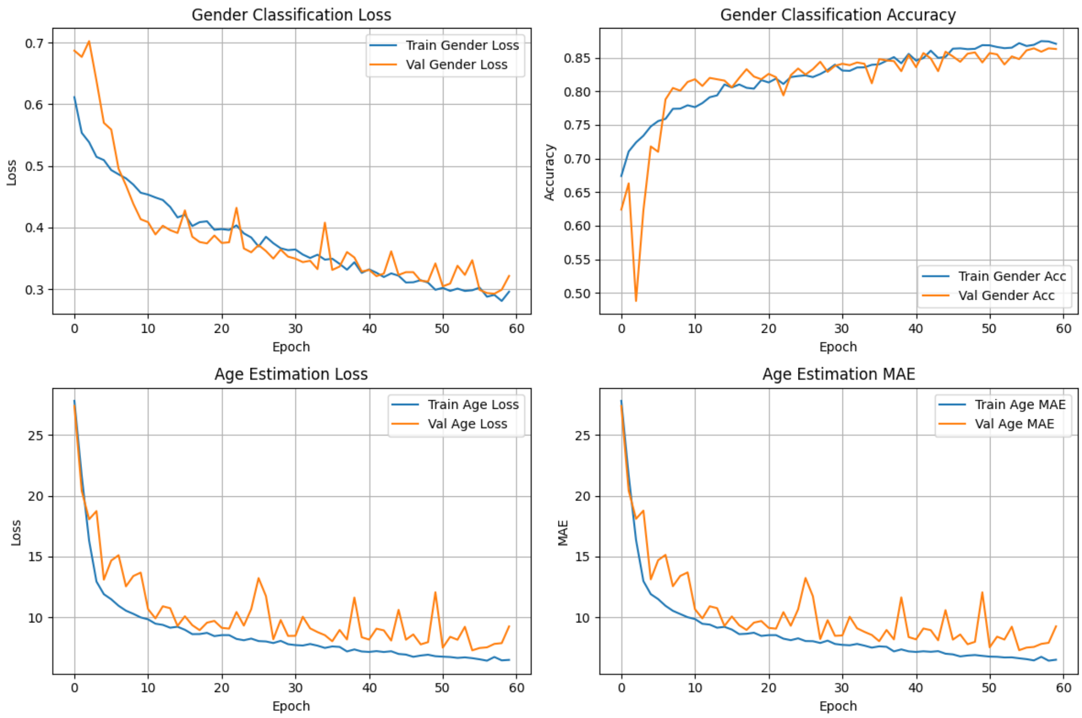


Figure 4: Training and validation curves for gender classification and age estimation for the custom CNN.

2.4. Training Configuration

The model was compiled with the `AdamW` optimiser (Loshchilov and Hutter, 2019), which extends the `Adam` method by decoupling weight decay from the gradient update. The learning rate was set to 5×10^{-5} and the weight decay to 1×10^{-5}. The network was trained for 60 epochs with a batch size of 64.

Binary cross entropy was used for the gender classification task, which is appropriate for a two class output. For the age regression task the loss function was mean absolute error (MAE), as it is more robust to outliers than mean squared error. This makes MAE suitable for this dataset, which contains irregular age values and possible labelling errors as discussed earlier.

2.5. Results and Observations

The training and validation losses for the gender classification task reached final values of 0.30 and 0.32, respectively. The training loss decreased steadily throughout training, and the validation loss followed the same trend, with only a small amount of divergence after about 40 epochs in the gender accuracy plot. The final classification accuracies were 87$\%$ for training and 86$\%$ for validation, these values are relatively close, showing that the model generalised well and did not suffer from major overfitting. For the age regression task the training MAE decreased smoothly until reaching 6.4, indicating that the model successfully fitted the training distribution. In contrast, the validation age MAE exhibited large fluctuations throughout training and ultimately settled at approximately 8.5 years. This curve also deviated from the training curve with a noticeable 2.1 year gap, indicating overfitting in the age regression. An MAE of 8.5 years means that the model's age predictions deviate from the labels by an average of 8.5 years in absolute terms.

3. Pre-trained CNN

3.1. Model Adaptation and Architecture

While the custom CNN established a baseline for performance, training deep architectures from scratch on limited data often leads to overfitting and may not generalise well. Transfer learning addresses this limitation by leveraging representations learned from large, diverse datasets such as ImageNet. This allows faster convergence, improved accuracy, and reduced dependence on data.

VGG16, developed by the Visual Geometry Group at the University of Oxford, was selected for its well established architecture and strong performance. The model contains 16 weight layers, including 13 convolutional layers and 3 fully connected layers, with a total of 138.4 million parameters. (Simonyan and Zisserman, 2015)

The convolutional base of VGG16 was imported with ImageNet weights and the original classifier was removed so that a new head could be attached. In this project, the new head used `Flatten`, consistent with the approach taken for the custom CNN. Although flattening creates a high dimensional representation and can lead to a very large number of parameters in some architectures, the resulting trainable parameter count in this configuration amounted only to 4.46 million. This made the model relatively inexpensive to train compared with the custom CNN, while still retaining all spatial features present in the VGG16 feature maps.

The new head followed a structure similar to that of the custom CNN. The flattened representation was passed through a Dense layer of 512 units, followed by batch normalisation and a ReLU activation, and then dropout values of 0.40 and 0.30 for the age and gender branches, respectively. Each branch then connected to a Dense layer of 256 units with ReLU activation, followed by a final linear output for age and a sigmoid output for gender.

3.2. Training Configuration

The input images were scaled back to the 0 to 255 range expected by VGG16 and then passed through the `preprocess input` function provided by the VGG16 implementation. This function applies the standard ImageNet normalisation. Apart from this step, the data preparation followed the same procedure described for the custom CNN in Section 2. Training was carried out in two stages:

1. **Stage 1: Head training** The convolutional base was kept frozen and only the newly added Dense layers were trained. The optimiser was AdamW with a learning rate of 1×10^{-4} and weight decay of 1×10^{-5} for 10 epochs.

2. **Stage 2: Fine-tuning** The last 4 out of 16 layers of the VGG16 model were unfrozen and trained with a reduced learning rate of 1×10^{-5} and weight decay of 1×10^{-6} for 25 additional epochs.

3.3. Results and Observations

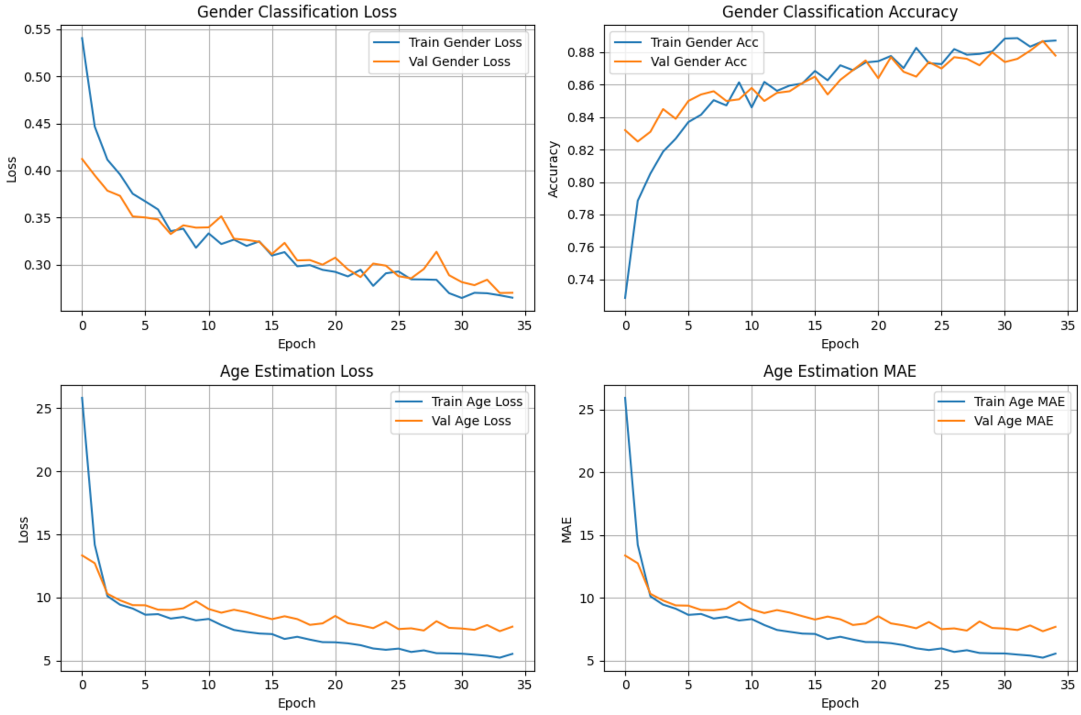


Figure 5: Training and validation curves for gender classification and age estimation for the finetuned VGG16 model.

As shown in Figure 5, the training and validation curves for the finetuned VGG16 model are noticeably smoother than those of the custom CNN. For gender classification, the training, and validation accuracies reached 89% and 88%, with almost identical loss curves of 0.26 and 0.27. These close values indicate good generalisation and improved stability compared with the baseline model.

For age estimation, on the training, MAE decreased steadily to 5.4. The validation MAE diverged early in training and then plateaued at about 7.6, indicating that the model began to overfit once the head became specialised. Even with this divergence, the final MAE values represent a clear improvement over the custom CNN with less noise.

Overall, the finetuned VGG16 model achieved higher gender accuracy, lower age MAE, and smoother optimisation behaviour. The training curves also converged more quickly and with less variance between training and validation metrics, confirming the benefit of transfer learning for small datasets.

4. Summary and Conclusion

The project compared a custom CNN built from scratch with a finetuned VGG16 model for joint gender classification and age estimation. Both models were trained on the same 5000 image subset of UTKFace, allowing a direct comparison of their behaviour.

The custom CNN established a clear baseline. Gender classification reached a validation accuracy of 0.86, with training and validation curves closely aligned, indicating good generalisation. The age regression task proved more challenging. Although the training MAE decreased smoothly, the validation MAE fluctuated and settled around 8.5, reflecting the imbalance in the age distribution and the presence of label noise. These results are reasonable for a model trained from scratch on a small dataset, but they also highlight the limits of learning all features directly from the available data.

Fine-tuning VGG16 led to consistent improvements across both tasks. The gender accuracy increased to 0.88 and the corresponding losses remained very close for training and validation, suggesting more stable optimisation. For age estimation the training MAE reached 5.4, and although the validation curve diverged and plateaued at about 7.6, it still outperformed the custom CNN. The smoother optimisation curves and faster convergence demonstrate the benefit of reusing features learned from large scale datasets such as ImageNet.

There are several limitations, with the most significant one being the amount and quality of data. The dataset is highly imbalanced, especially for ages above 60, which affects the regression task and leads to the underestimation of extreme ages. The total number of images is also modest, limiting the range of facial variation seen during training. If the full set of more than 20000 images had been available for use, it is very likely that both models would have achieved noticeably better performance.

Another limitation is the reduced image resolution. The UTKFace images were provided in 128 by 128 format for this project, but the original dataset includes faces at 200 by 200 resolution. Using the higher resolution images would almost certainly improve performance, since finer details such as wrinkles, texture, and small facial features are easier for a model to capture at larger scales. However, training with larger images increases memory usage and computational cost, and these were important constraints in this project. This also highlights a broader point in deep learning, which is that available compute strongly influences what models can be trained, what input sizes can be used, and how far an architecture can be scaled. With more powerful hardware, both models in this study could have been trained at higher resolution and with more extensive hyperparameter exploration.

The evaluation in this project used accuracy and MAE only, and future work could make use of additional metrics to capture more detailed aspects of performance. Further improvements could come from cleaning the dataset, balancing under-represented age groups, or sourcing external images to fill the gaps in the distribution.

In conclusion, this project shows that both approaches are capable of learning meaningful patterns from facial images, but transfer learning offers a clear advantage when data is limited. The custom CNN demonstrates that a compact architecture can provide a strong baseline, while the finetuned VGG16 model achieves higher accuracy, lower error, and more stable training by making use of features learned from a much larger dataset.

References

Loshchilov, I. and Hutter, F., 2019. *Decoupled weight decay regularization* [Online]. arXiv: 1711.05101 [cs.LG]. Available from: https://arxiv.org/abs/1711.05101.

Özbulak, G., Aytar, Y., and Ekenel, H.K., 2016. *How transferable are cnn-based features for age and gender classification?* [Online]. arXiv: 1610.00134 [cs.CV]. Available from: https://arxiv.org/abs/1610.00134.

Simonyan, K. and Zisserman, A., 2015. *Very deep convolutional networks for large-scale image recognition* [Online]. arXiv: 1409.1556 [cs.CV]. Available from: https://arxiv.org/abs/1409.1556.

Zhang, Z., Song, Y., and Qi, H., 2017. *Age progression/regression by conditional adversarial autoencoder* [Online]. arXiv: 1702.08423 [cs.CV]. Available from: https://arxiv.org/abs/1702.08423.