# CM50265 - Group 1 - Cover Page

`Contribution grid submitted individually via Moodle and omitted from report.`

| Model Links | |
|---|---|
| Model A | https://drive.google.com/file/d/1MaIDjs3kznC8Wxq9RpixRH_Bz4_IiXcl/view?usp=sharing |
| Model B | https://drive.google.com/file/d/1BuZmakWXkTTTTiMnITJOY3AWKmgM2fAO/view?usp=sharing |
| Notebook | https://colab.research.google.com/drive/1rFKfHzzp0lgjYBCbP1GG0Ttn77acamw2?usp=sharing |

# CM50265 - Assignment 1 - Group 1 Report

## Age Estimation and Gender Classification CNN

Daniel Davis, James Whiffing

March 21, 2022

# Contents

# List of Figures

# List of Tables

# 1 Introduction

This report documents the creation of two CNN models which predict the age and gender of people based on an image (128px x 128px) of their face. The first model was created from scratch and learns entirely from the training data. The second is a finetuned model using a base trained on ImageNet. This base is combined with new output layers and retrained to predict age and gender. The perfomance of these models is analysed and compared.

Both of the CNN models were developed using Python 3 and Keras. Once trained, the models were saved; links to the models can be found on the cover-page for this report.
Development was performed using Google Colab, using a mixture of free cloud compute time (utilizing GPU resources) and running locally on our own machines.

# 2 Model A - CNN Created From Scratch

## 2.1 Architecture

Our model architecture is shown in FIGURE 1. It consists of a shared trunk of convolutional layers with relu activation. These layers are interspersed with maxpooling and batch normalisation layers. The output of this trunk is sent to different branches for age and gender prediction. The age and gender branches have the same architecture except for the number of filters in their final convolutional layers. Each branch has three convolutional layers interspersed with maxpooling and batch normalisation. The output of these convolutional layers is then flattened and fed into a 3 layer dense MLP. The final layer is a single node for the output. The output of the age branch has no activation and the output of the gender branch has a sigmoid activation.

This architecture was chosen because the first layers of a CNN usually extract generic low-level features which are common to many images. Therefore using a shared trunk allows the network to use fewer parameters and helps reduce over-fitting. The branches then allow the model to further extract higher level features specific to age or gender prediction. The convolutional layers extract features from the input, the maxpooling layers reduce the parameter count and the batchnorm layers help to prevent vanishing gradient issues. The dense layers are responsible for making sense of these extracted features in order to make predicitions.

Measures employed to reduce overffiting include the shared trunk as well as dropout and L2 regularisation on the dense layers.

## 2.2 Training

The model is trained to minimize a weighted average of losses from each branch. The loss for the gender prediction output is binary cross entropy because it is a binary classification problem and the loss for the age prediction is mean squared error because it is a regression problem. Mean squared error was chosen over mean absolute error in order to penalise larger errors more than smaller ones. The average loss is weighted 1:100 towards gender prediction because typical gender losses seen during training were approximately 100 times smaller than the age losses. This weighting was chosen so the model attempts to balance its training objectives without over-prioritising either age or gender. The model uses the Adam optimizer with a exponentially decaying learning rate to train for a total of 50 epochs.

Hyperparameters were tuned using the `keras_tuner` library to minimise the validation loss. After tuning the tuned values were fixed and the model was rebuilt using them. The following hyperparameters were tuned:

1. Greyscale: Whether to apply a greyscale filter to the gender branch, both branches, or neither.
2. Sigmoid Vs `BinaryCrossEntropy` With logits: whether to use a sigmoid activation on the gender branch or enable logits for the `BinaryCrossEntropy` loss function.
3. Initial Learning rate
4. Feature depth of final convolution layer

5. Dropout rate
6. Regularisation

An output of the tuning can be found in the appendix here Appendix C

### 2.3 Data

The model is trained on data from the dataset provided. The data is augmented using the Keras `ImageDataGenerator` to effectively increase the size of the dataset and prevent over-fitting. The augmentations used are rotation, zoom, and horizontal flipping. Pixel values were scaled between 0 and 1.

A 20% validation split was used. The training and validation data were subject to the same preprocessing.

### 2.4 Performance

Fig Figure 2a and Figure 2b show the training graphs. Our model shows good performance on both age and gender prediction. It achieves an accuracy of 93% for gender prediction and a mean absolute error of 5.6 for age prediction on the training data. However it can be seen that the model over-fits slightly and achieves slightly lower performance on the validation data, with an accuracy or 88% for gender prediction and mean absolute error of 6.8 for age prediction. The gender branch can be seen to over-fit to a greater extent than the age branch.

The trace of the final training epoch is below.

```
    Epoch 50/50
125/125 [==============================]
- 41s 325ms/step
- loss: 82.5588
- age_output_loss: 55.1034
- gender_output_loss: 0.1627
- age_output_mean_absolute_error: 5.5780
- gender_output_binary_accuracy: 0.9330
- val_loss: 131.3783
- val_age_output_loss: 90.0434
- val_gender_output_loss: 0.3111
- val_age_output_mean_absolute_error: 6.8350
- val_gender_output_binary_accuracy: 0.8821
```

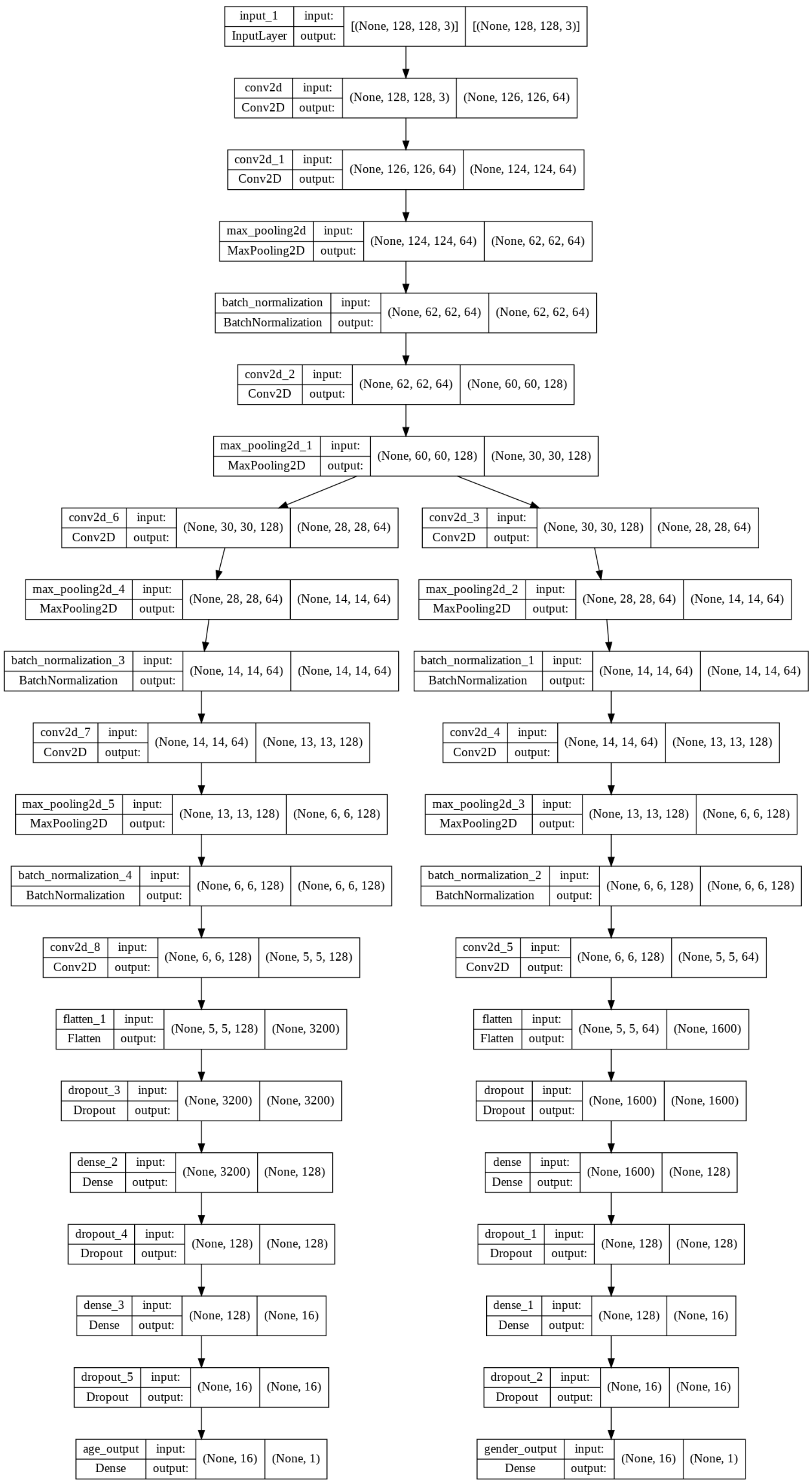
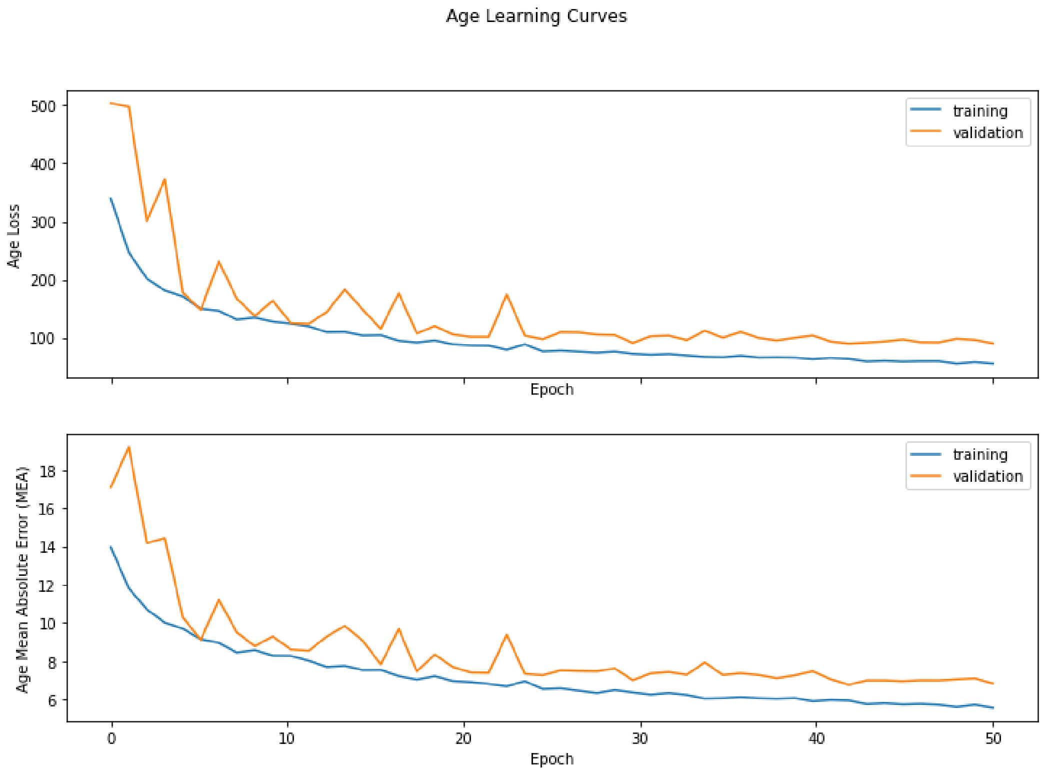
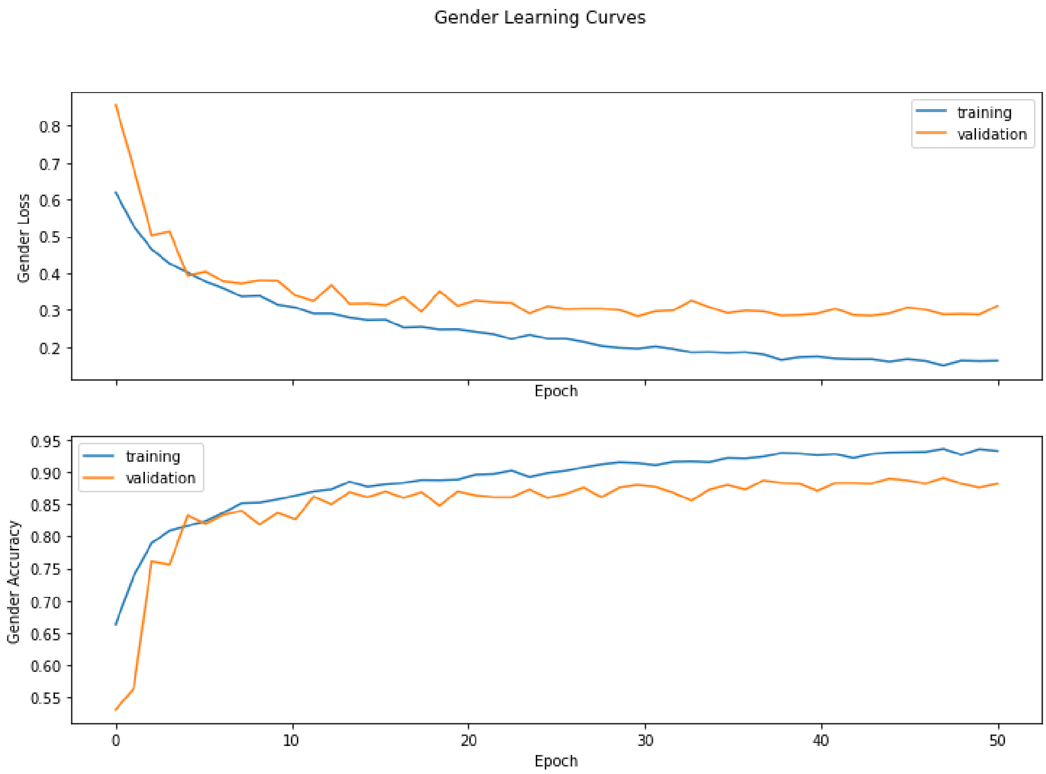



Figure 1: The architecture for Model A.




(a) The performance on age prediction for our model.



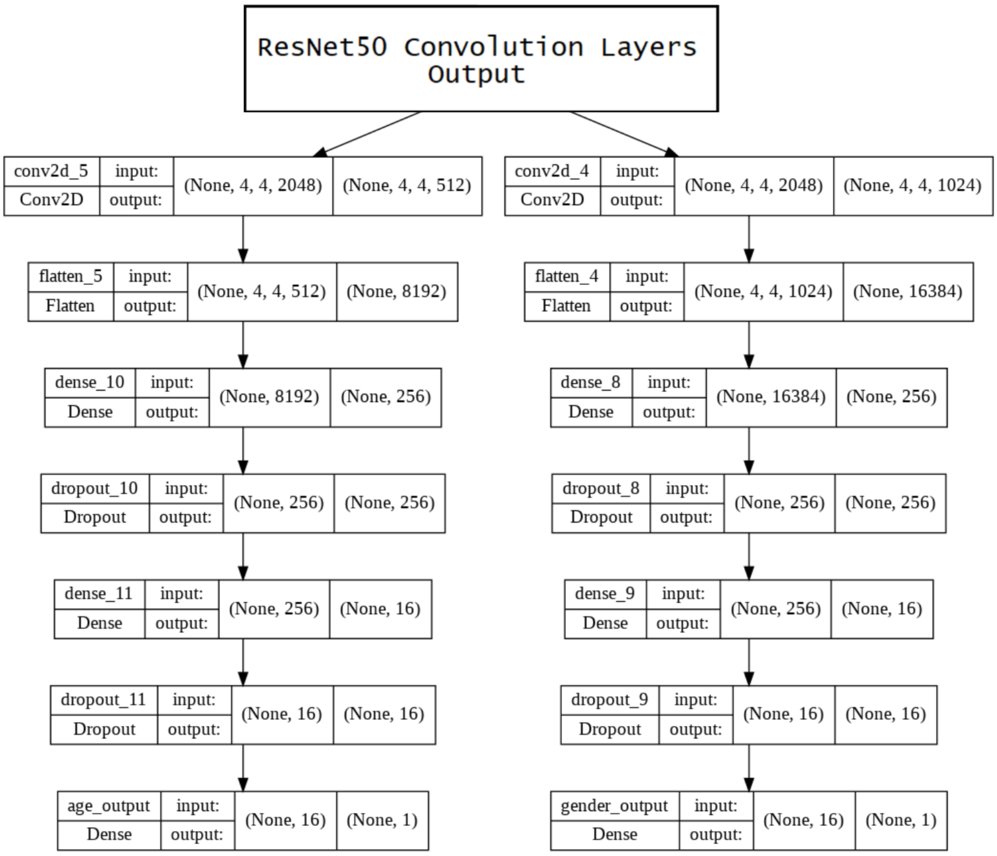

(b) The performance on gender prediction for our model.

Figure 2: Learning curves observed training Model A

# 3 Model B - Using Pre-Trained CNN

## 3.1 Architecture

For our pre-trained model we used ResNet50 as our base. This is because it is a relatively fast, lightweight and high performing model (with respect to ImageNet top 5 rating).
We used weights trained on ImageNet as our initial weights. These weights were not frozen in order to allow the model to learn features specific to age and gender prediction.
The output of this base model is fed into 2 branches (one for gender, the other for age). Each performs an additional padded convolution, before being flattened and passed on towards their own FCN layers (with one output layer and 2 hidden layers). All the FCN Dense layers made use of dropout and regularisation to reduce over-fitting.




## 3.2 Training

The pre-trained model was trained in the same manner as Model A, with the same losses, loss weights, and optimizer.

To limit overfitting a stop early callback was used to end training if the validation loss failed to improve over 5 consecutive epochs.

Hyper-Parameters were tuned in the same manner as for Model A.
Hyper-parameters tuned were:

1. Initial Learning rate
2. Flatten Vs Global Pooling: whether to flatten the 3D output from the final convolution layers
3. Feature depth of final convolution layer
4. Number of Units for first FCN layer
5. Dropout rate
6. Regularisation

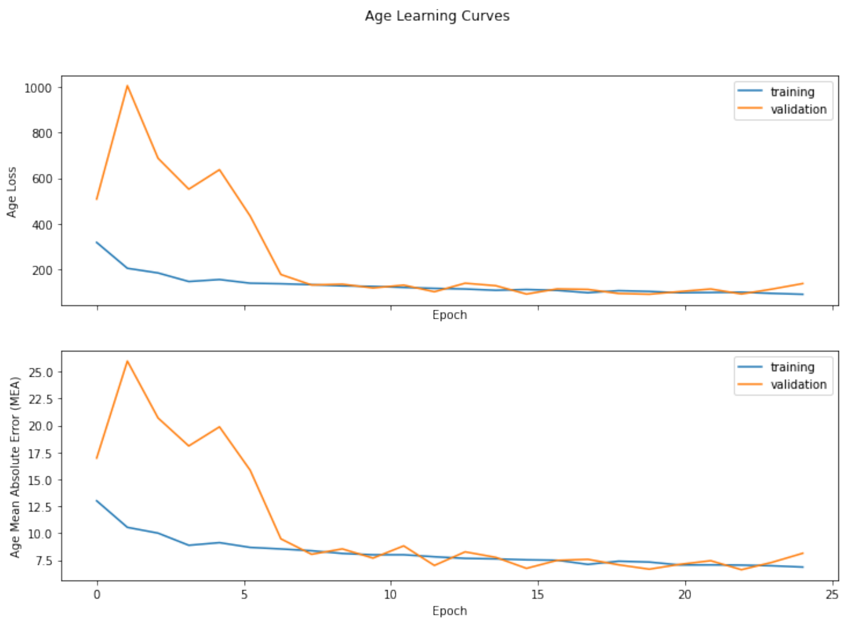
Once tuning completed, we rebuilt and trained the model with the tuned hyper-params fixed.

### 3.3 Performance

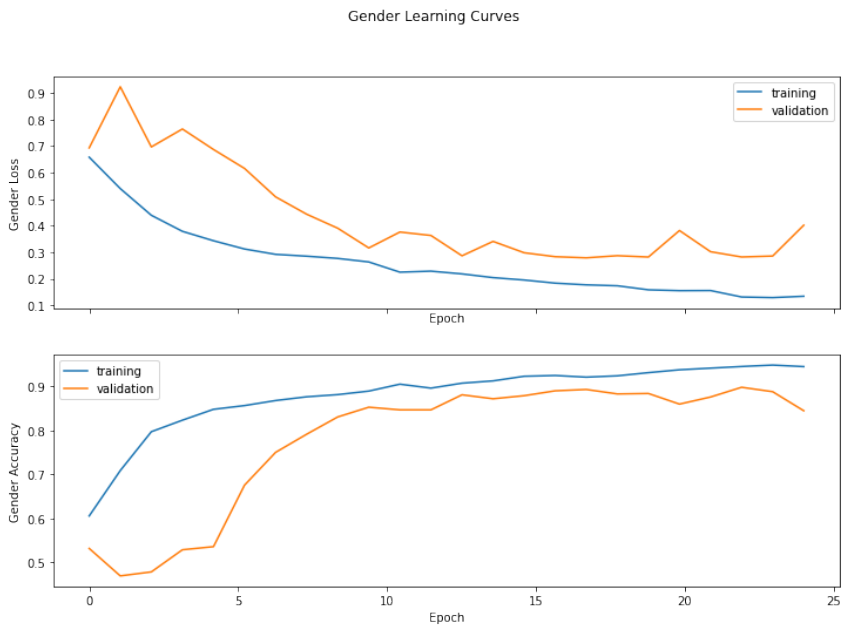
The trace of the final epoch is below.

```
Epoch 31/50
125/125 [==============================]
- 37s 296ms/step
- loss: 159.7228
- age_output_loss: 106.7853
- gender_output_loss: 0.1204
- age_output_mean_absolute_error: 7.3058
- gender_output_binary_accuracy: 0.9507
- val_loss: 154.9256
- val_age_output_loss: 86.5945
- val_gender_output_loss: 0.3250
- val_age_output_mean_absolute_error: 6.4584
- val_gender_output_binary_accuracy: 0.8871
```

The pre-trained model performed slightly better than our model on both tasks on the validation data. It achieved an accuracy of 89% on gender prediction and a mean squared error of 6.5 on age prediction. It can be seen in FIGURE **??** to overfit on gender prediction but not on age prediction.




(a) The performance on age prediction for the pre-trained model.




(b) The performance on gender prediction for the pre-trained model.

Figure 3: Learning curves observed training Model B

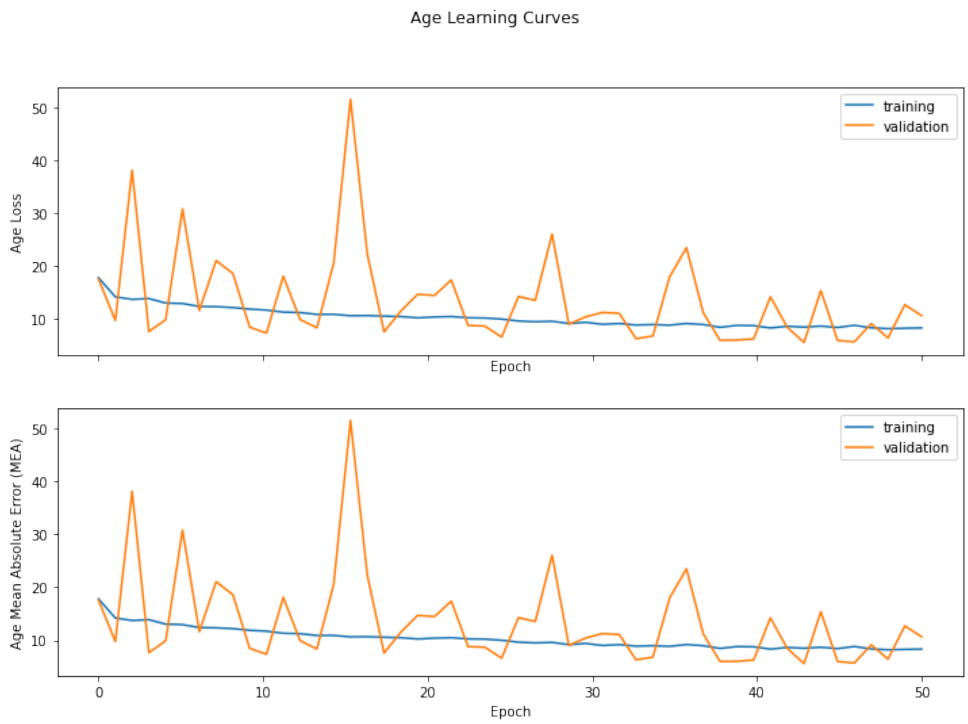
# 4 Summary and Conclusions

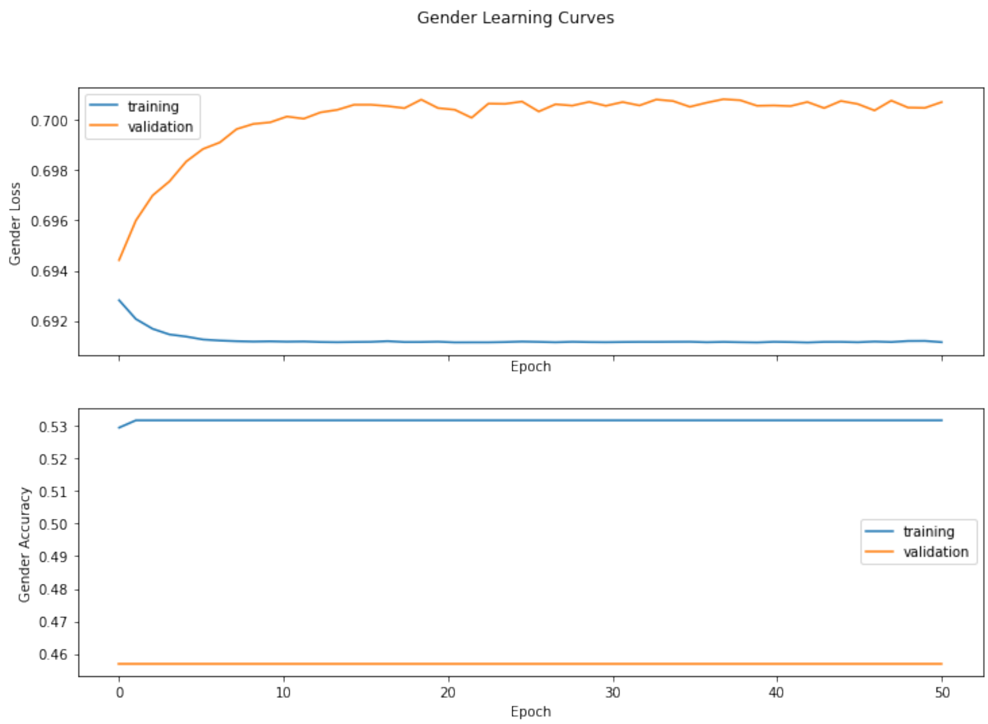
Both model perform well, achieving good accuracy and low mean absolute error scores on our validation data. Both models over-fit significantly despite measures to counteract it including L2 regularisation, dropout and data augmentation. This is likely due to the relatively small amount of training data.

Model B performs slightly better than Model A. Model B suffered from worse over-fitting on gender but almost none on age. Better performance from model B was expected because it is able to utilise the feature extraction capabilities of ResNet50, a significantly more advanced model than our own. However the perfomance improvement is small. We have several hypotheses to explain this. The first is that our model is specifically designed for age and gender prediction, having a dedicated branch for each task allowing it to extract features more specific to the tasks. ImageNet does not contain classes related to human faces so the features it extracts may be too generic to provide significant performance improvements over our dedicated model. Additionally it is possible that there is simply not enough information in our training data to achieve higher perfomance. The size of the training set is relatively small and age and gender can be difficult to predict in some cases even for humans.

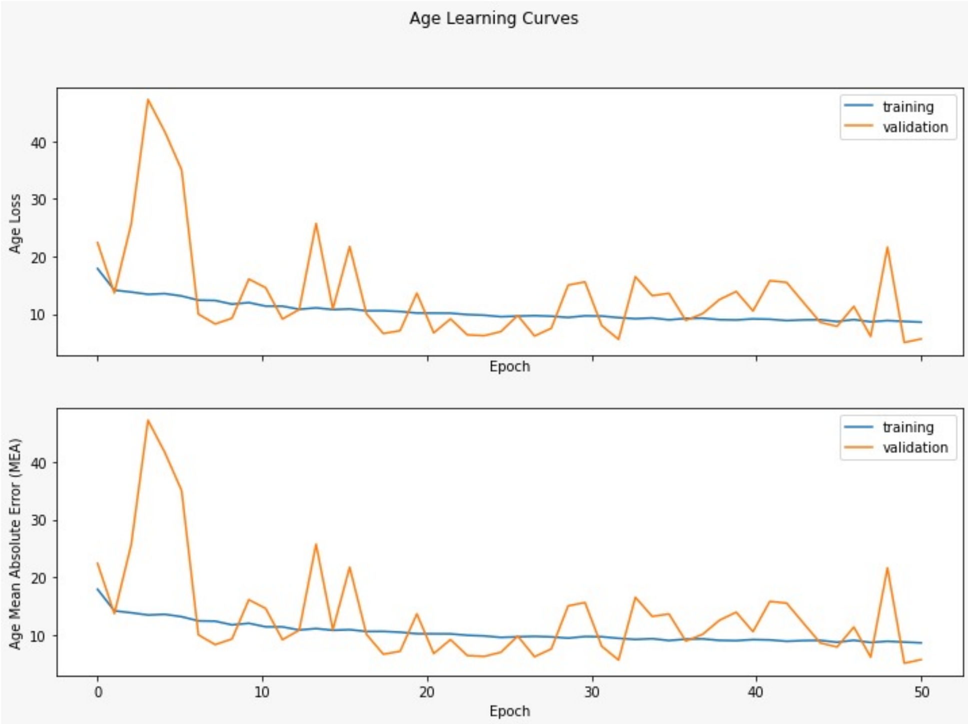
# A Greyscale Comparison - Old Model A Learning Curves

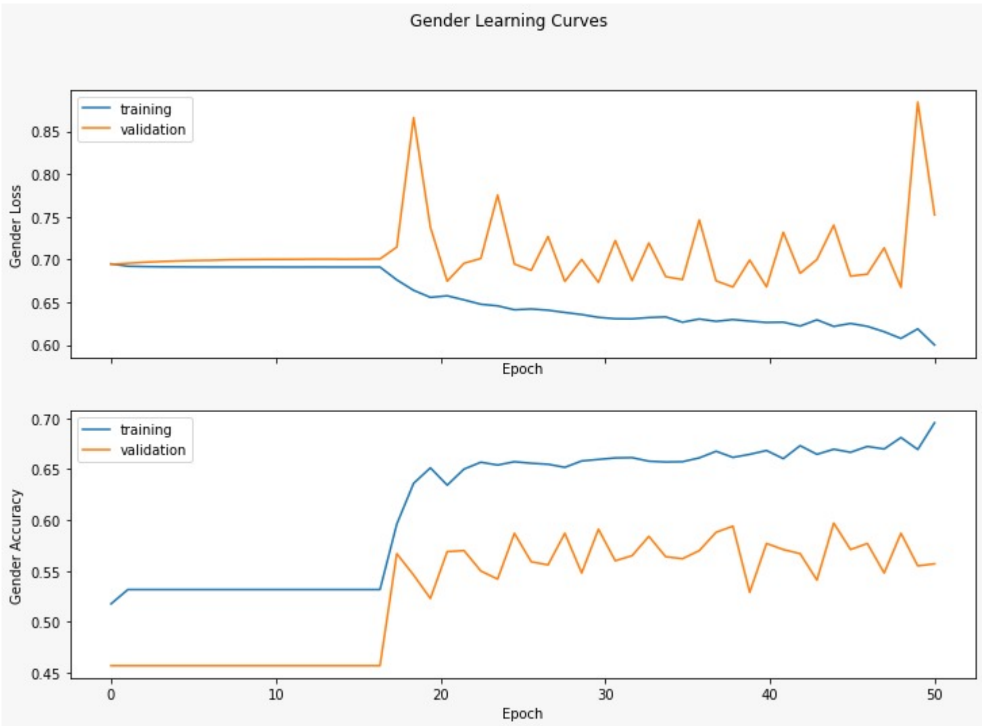
## A.1 Greyscale Applied To Gender & Age Branches



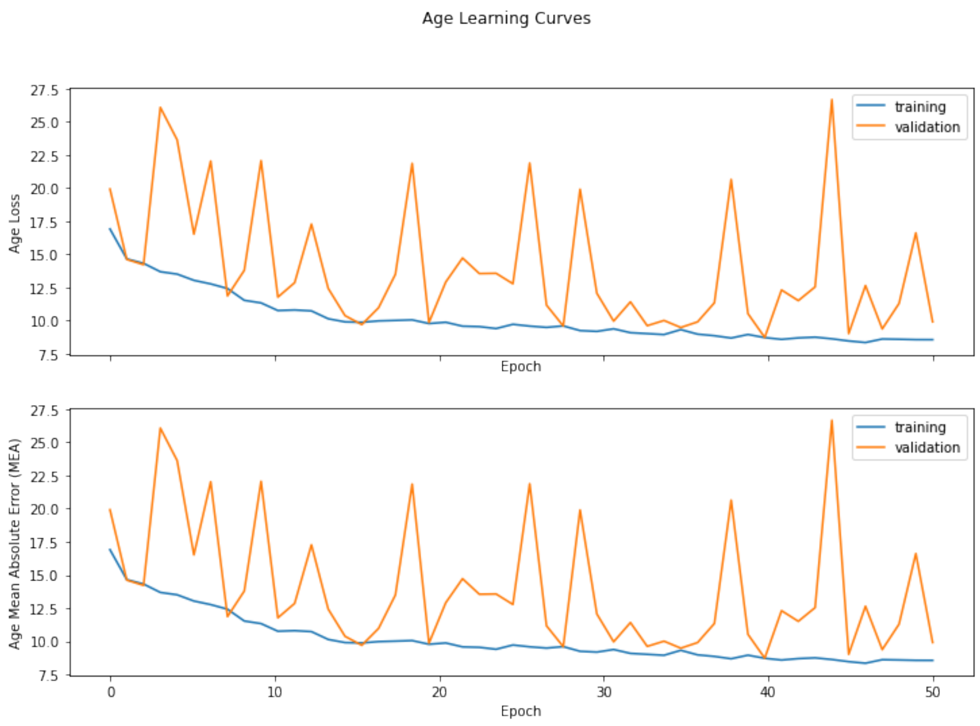



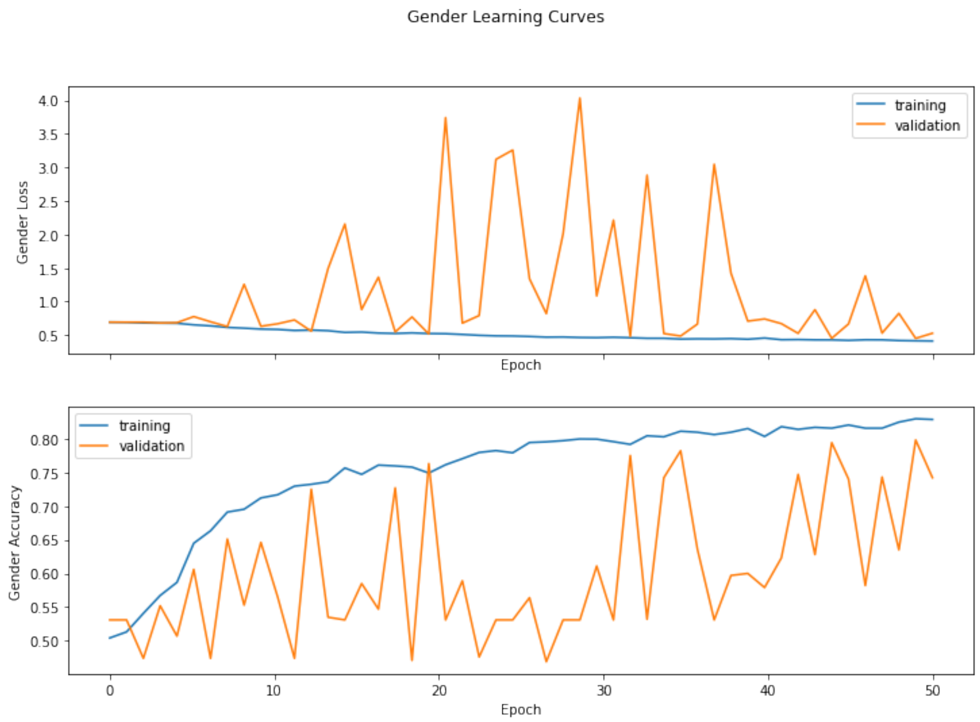
### A.2 Greyscale Applied To Neither Branch



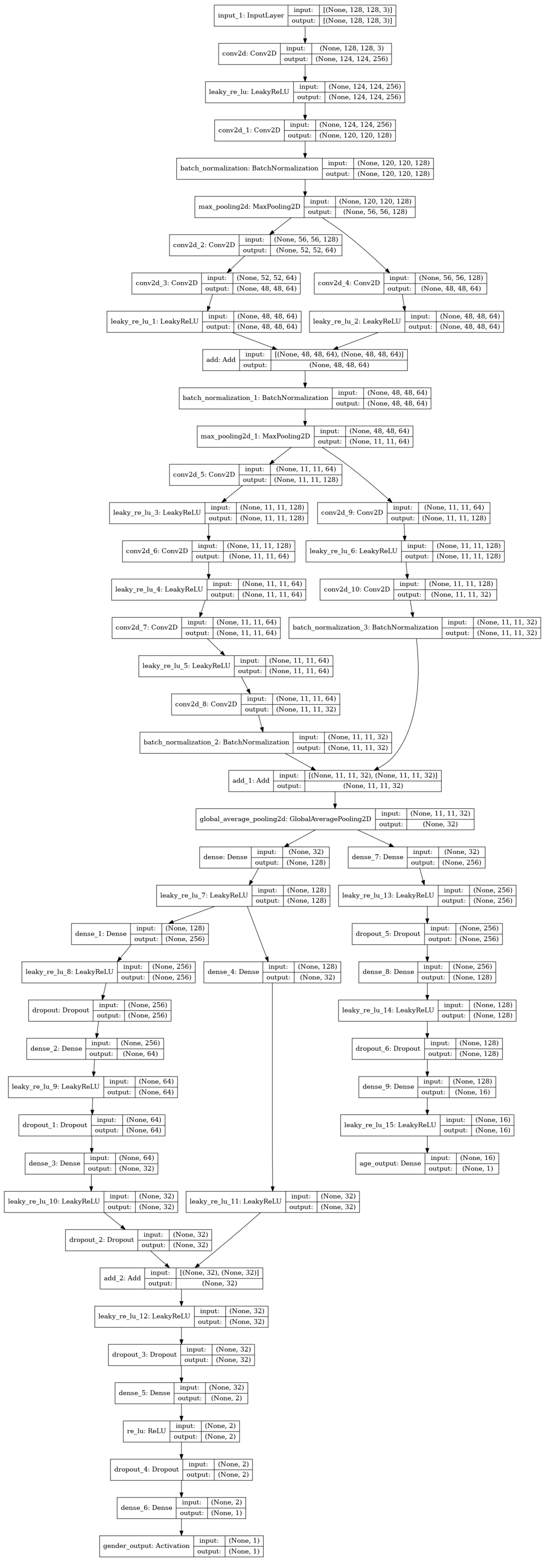



## A.3 Greyscale Applied To Gender Branch Only



# B Old Model A Graph



# C Model A Hyper-Parameter Tuning

```
Trial 12 Complete [00h 02m 56s]
val_loss: 1256.30908203125

Best val_loss So Far: 129.11569213867188
Total elapsed time: 00h 06m 10s

Search: Running Trial #13

Hyperparameter     |Value              |Best Value So Fa
gender_sigmoid     |False              |True
use_greyscale      |2                  |0
gender_cnn_feat ...|64                 |64
age_dropout_rate   |0.2                |0.1
age_cnn_feature ...|128                |32
age_loss_weight    |0.25               |0.25
gender_loss_weight |0.25               |1
learning_rate      |0.0001             |0.01
gender_dropout_ ...|0.1                |None
gender_regulari ...|0.02               |None
age_regularisation |0.05               |None
tuner/epochs       |2                  |2
tuner/initial_e ...|0                  |0
tuner/bracket      |2                  |2
tuner/round        |0                  |0

Epoch 1/2
125/125 [==============================] - 91s 703ms/s
Epoch 2/2
100/125 [========================>......] - ETA: 14s -
```