Dataset

The ShanghaiTech dataset contains 1,198 annotated images with a total of 330,165 labeled head positions. The dataset is split into two parts: Part A (more crowded, 300 training and 182 testing images) and Part B (less crowded, 400 training and 316 testing images). To train and test models only part B was used.

Architecture

VGG16

This model contains 21 layers: thirteen convolutional layers, five Max Pooling layers and three Dense layers. It has 16 weight layers. The model has convolution layers of 3x3 filter with stride 1and maxpool layer of 2x2 filter of stride 2. Convolutional layers are divided into 5 internal classes and each class contains different number of filters:

- Conv-1 layer: 64 filters
- Conv-2 layer: 128 filters
- Conv-3 layer: 256 filters
- Conv-4 layer: 512 filters
- Conv-5 layer: 512 filters

MobileNet

This model contains 28 layers (only convolutional and fully-connected layers). It can be divided into 3 fundamental components:

- Standard convolution: 3x3 filters with stride 2, 3 input channels and 32 output channels
- Depthwise convolution: 3x3 filters with separable convolution for each channel
- Pointwise convolution: 1x1 convolution to connect channels

EfficientNetB7

This model contains 813 layers including all ingredients. It is divided into 7 blocks with MBConv with 3x3 or 5x5 filters in blocks. These blocks use depthwise separable convolutions and squeeze-and-excitation optimization. Additionally, the activation functions ReLU and ReLU6 are employed throughout the network to introduce non-linearity and improve learning capability.

Xception

This model contains 36 convolutional layers and 3 fundamental components:

- Entry Flow – 3x3 convolutional layer used with 32 different filters and stride 2x2 then followed by another 3x3 convolutional layer with 64 filters and ReLU. After that, the modified depthwise separable convolution layer is applied, along with the 1x1 convolution layer and stride 2x2. Max pooling (3x3 with stride=2) reduces the size of the feature map.
- Middle Flow – depthwise separable convolution with 728 filters and a 3x3 kernel, then ReLu activation. The block is repeated 8 times.
- Exit Flow - separable convolution with 728, 1024, 1536, and 2048 filters, all with 3x3 kernels, further extracts complex features.

All models are built using a specific base architecture derived from transfer learning, with each base model pre-trained on the ImageNet dataset. The top layer for classification is frozen. The architecture is extended as follows:

- **Base Model** (pre-trained on ImageNet, used as a feature extractor)
- **Flatten Layer**
- **Dropout Layer** (rate = 0.3)
- **Dense Layer** with 128 units and ReLU activation
- **Dropout Layer** (rate = 0.3)
- **Dense Layer** with 64 units and ReLU activation
- **Output Layer**: Dense layer with 1 unit and linear activation

Images preprocessing

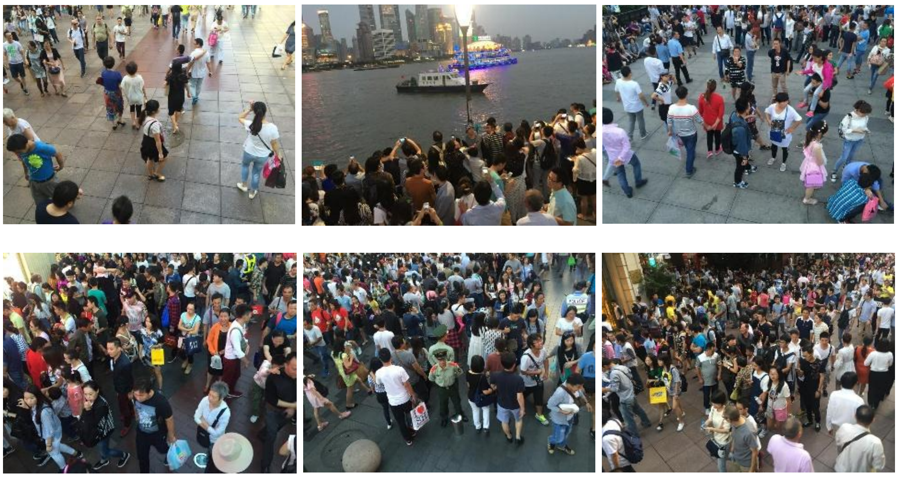

Figure 1: Some sample images.

Some image preprocessing steps were applied prior to training. All images were resized to 224×224 pixels, as this is the input size expected by the pre-trained models. The resizing was performed using the LANCZOS filter to preserve image quality. Additionally, model-specific preprocessing functions were used—such as normalization—to ensure compatibility with each base model's requirements.

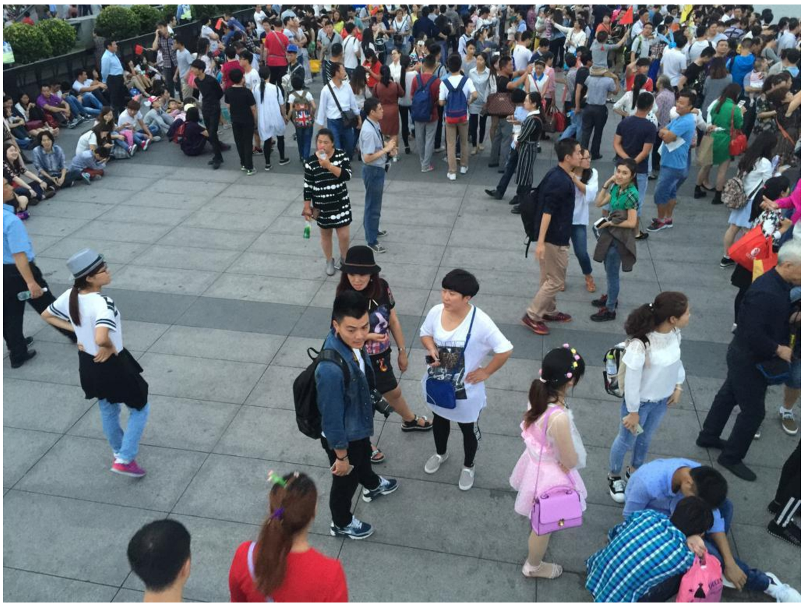

Figure 2: Image before preprocessing.

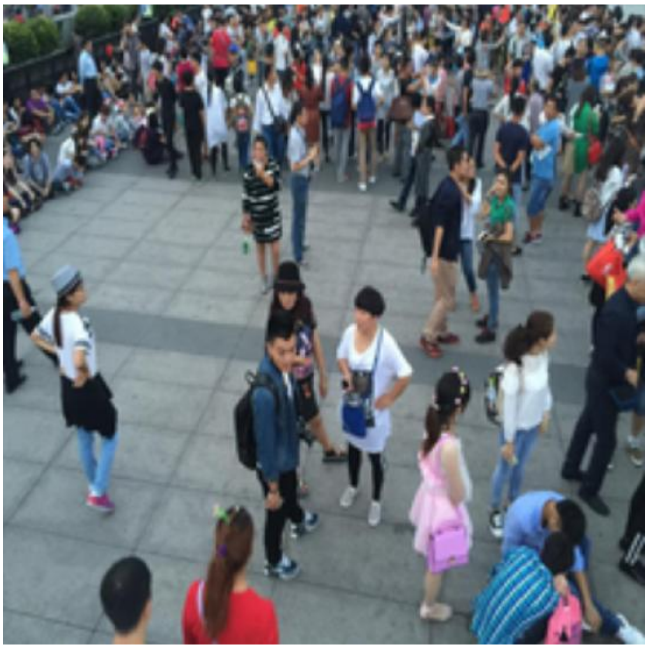

Figure 3: Image after preprocessing.

There were some experiments with adding black padding to keep the original ratio width to height. Finally, it had no influence on the results.

Model training

There were some training configurations for all models:

- **Optimizer:** Adam with a learning rate of 1e-4
- **Number of Epochs:** 20
- **Input Image Size:** 224 × 224 pixels
- **Loss:** Mean Squared Error
- **Metrics:** Mean Absolute Error
- **Batch size:** 32

Data augmentation was also applied to increase the diversity of the training data and improve model generalization. The augmentation techniques included Random Flip (in the 'horizontal' direction), as well as Random Zoom and Random Rotation, each with a 10% transformation factor. Moreover, for each epoch the training set was shuffled to protect from learning the order of images.

Evaluation results

This section presents the results of two evaluation approaches. In the first approach, all pre-trained layers of the base models were kept frozen during training. In the second approach, selective fine-tuning was applied: the last 30 layers were unfrozen for MobileNet and EfficientNetB7, while the last 3 layers were unfrozen for VGG16 and Xception. The evaluation was performed using the following metrics on the validation set: Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), and Root Mean Squared Error (RMSE).

Model	**MSE**	**MAE**	**R2**	**RMSE**
VGG16	**3637**	**38**	**0.60**	**60**
MobileNet	4441	39	0.51	67
EfficientNetB7	4426	40	0.51	67
Xception	6357	48	0.30	80

Figure 3: Evaluation: all layers frozen.

Model	**MSE**	**MAE**	**R2**	**RMSE**
VGG16	2776	38	0.69	53
MobileNet	**2167**	**29**	**0.76**	**47**
EfficientNetB7	3415	36	0.62	58
Xception	5082	42	0.44	71

Figure 4: Evaluation: some layers unfrozen.

Figure 5 shows comparison of training curves for all models.

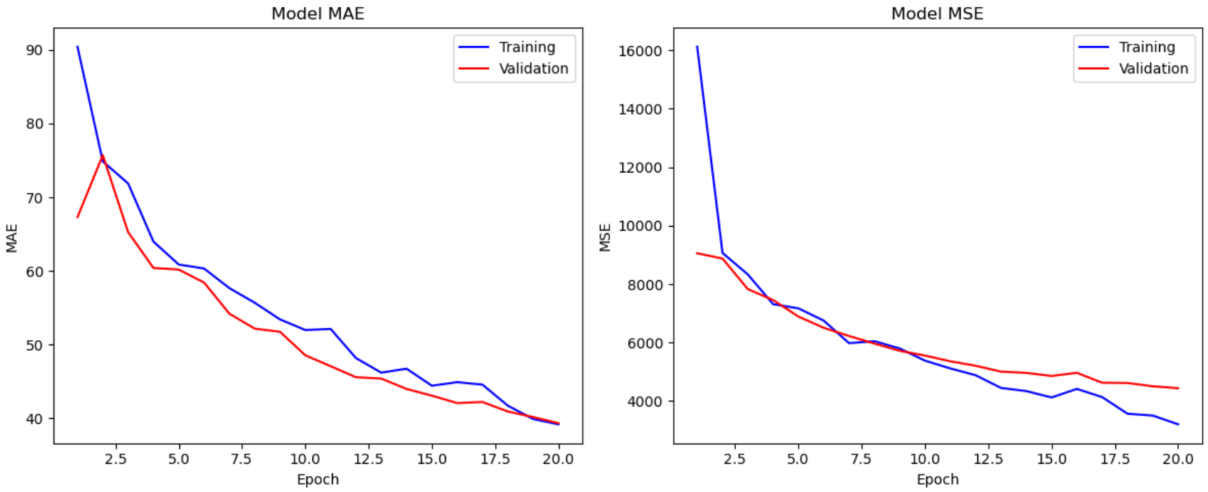

Figure 5.1.: MobileNet training curves with all pre-trained layers frozen.

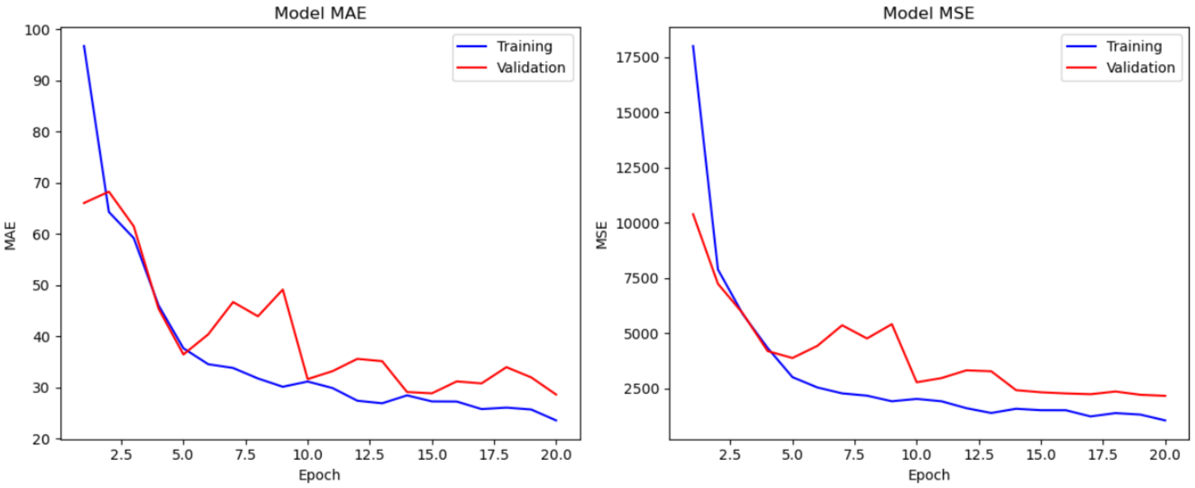

Figure 5.2.: MobileNet training curves with some pre-trained layers unfrozen.

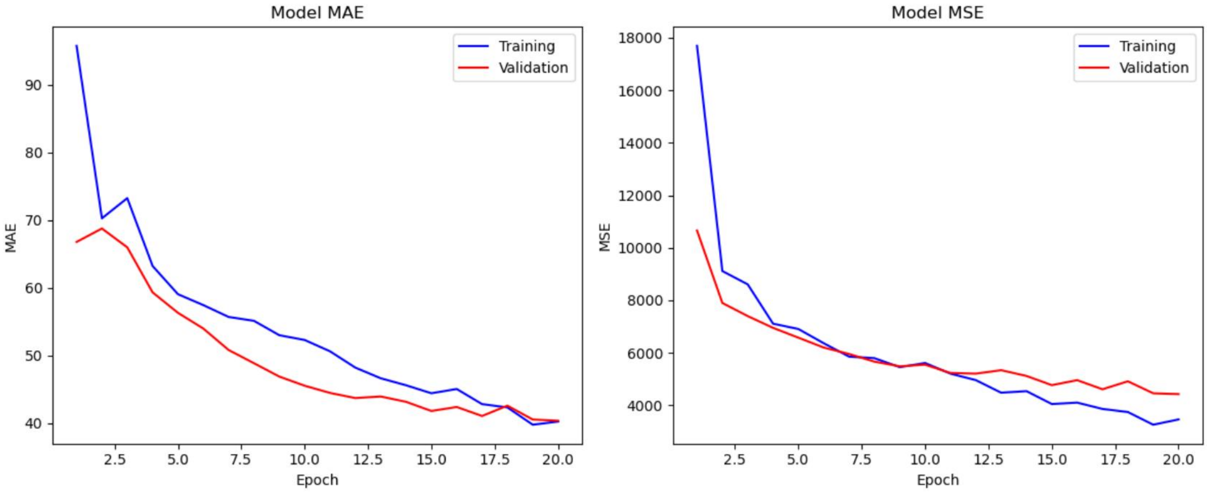

Figure 5.3.: EfficientNetB7 training curves with all pre-trained layers frozen.

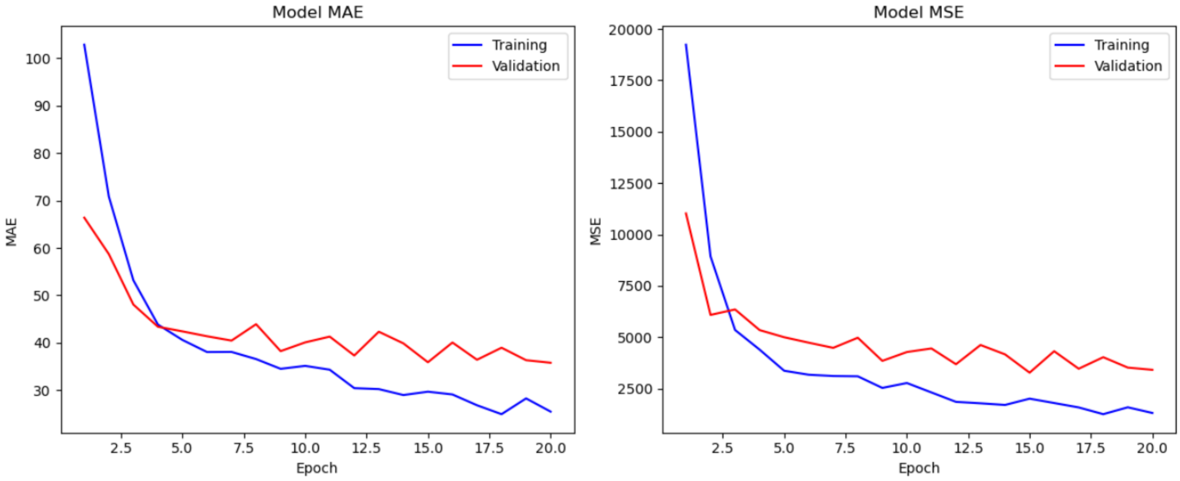

Figure 5.4.: EfficientNetB7 training curves with some pre-trained layers unfrozen.

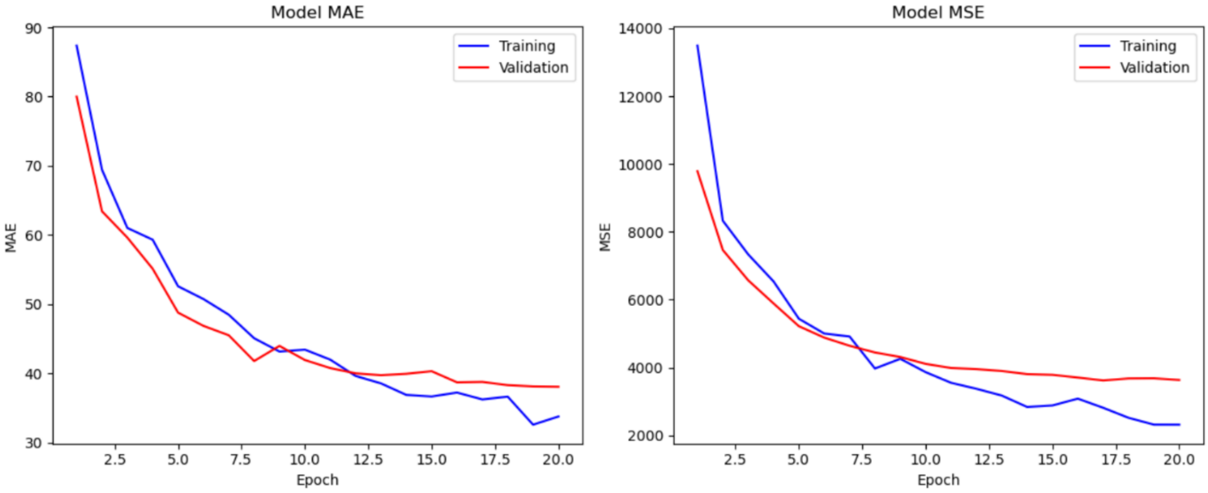

Figure 5.5.: VGG16 training curves with all pre-trained layers frozen.

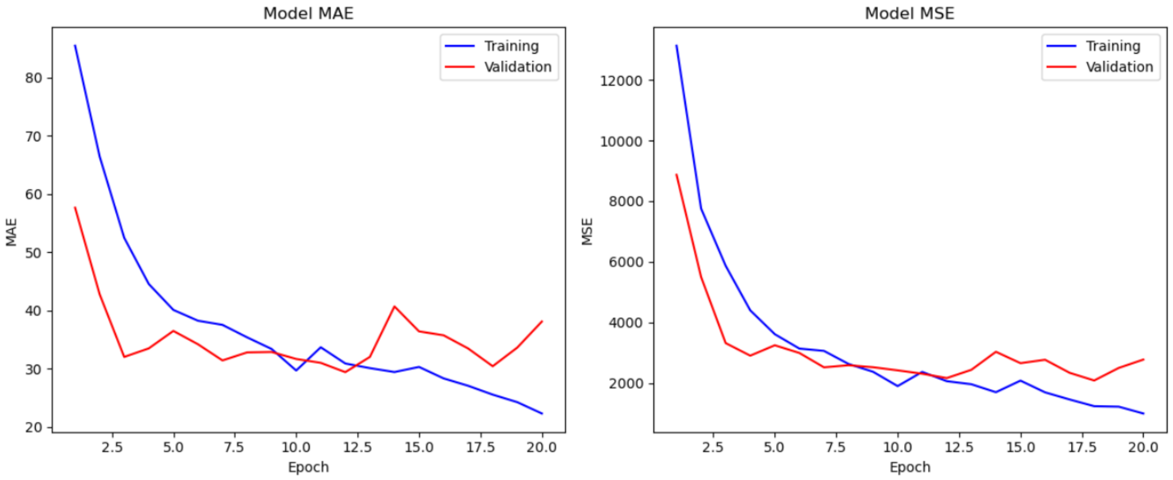

Figure 5.6.: VGG16 training curves with some pre-trained layers unfrozen.

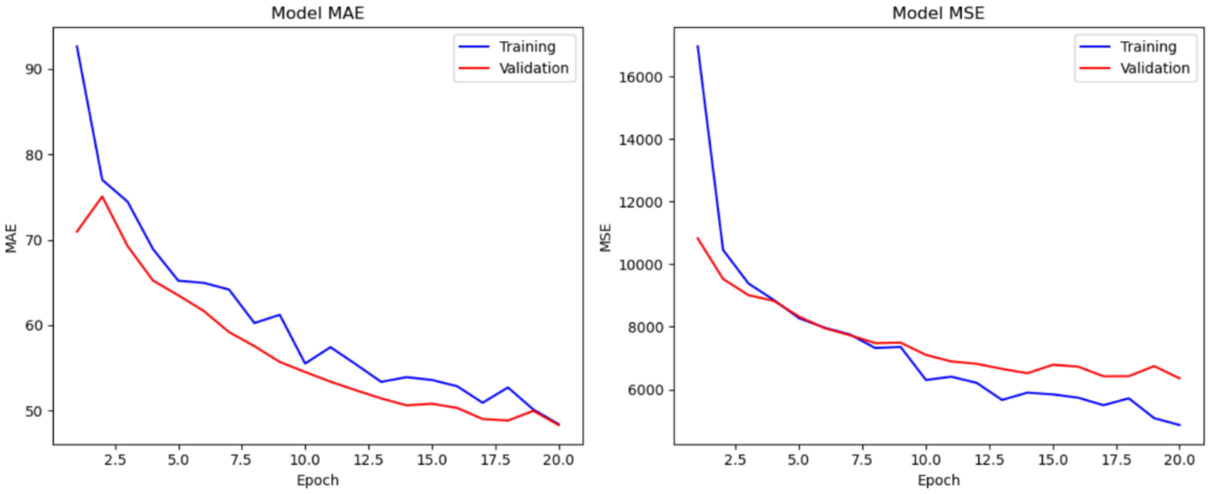

Figure 5.7.: Xception training curves with all pre-trained layers frozen.

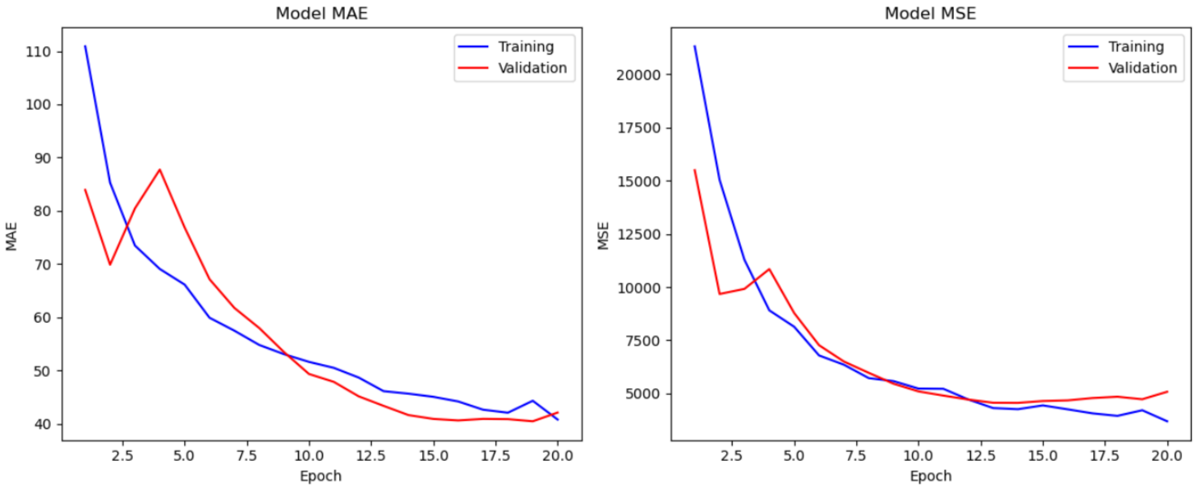

Figure 5.8.: Xception training curves with some pre-trained layers unfrozen.