Choo Han Ye 1002439

This code was written with much reference to
https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html.

Model used: resnet 18

Loss used: Cross Entropy Loss

Data augmentation used: transform = transforms.Compose([

transforms.Resize(224),

transforms.CenterCrop(224),

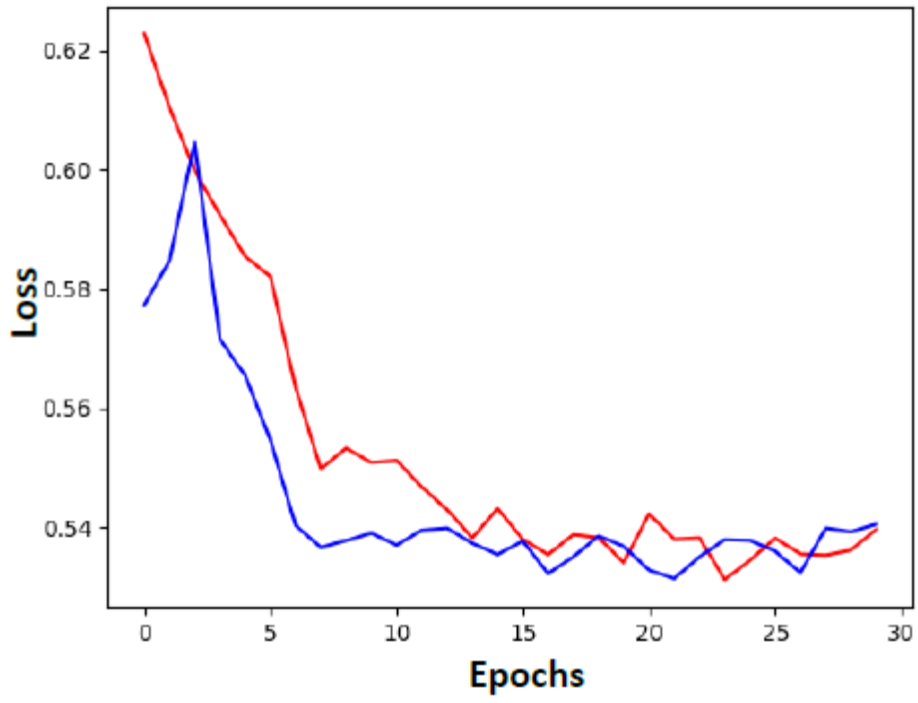
transforms.ToTensor(),

transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])])

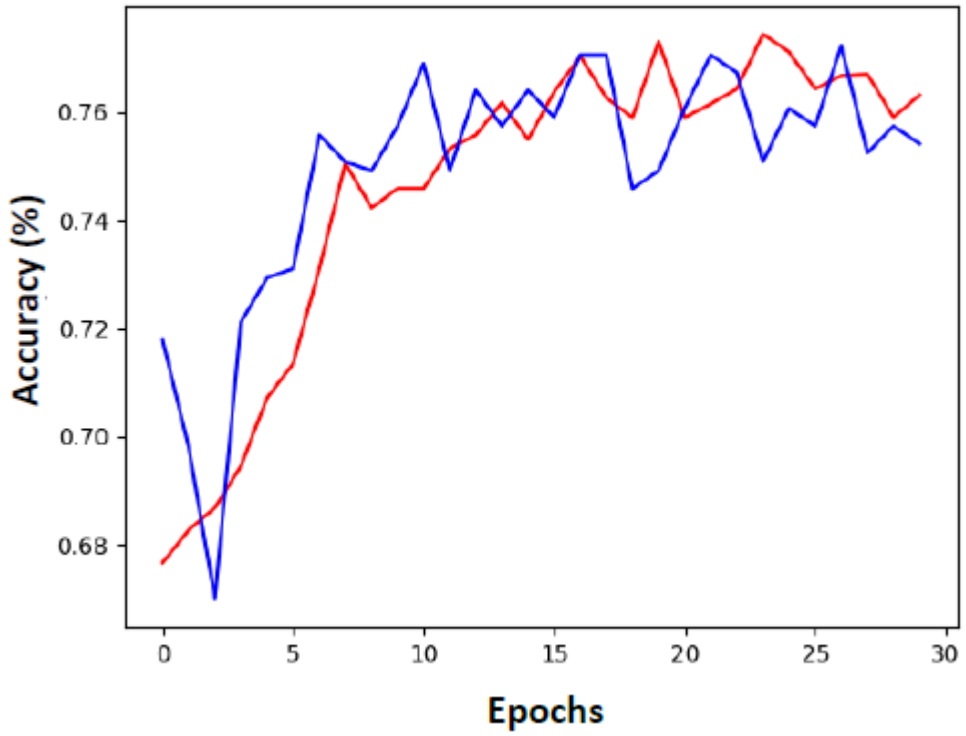
learning rate used: 0.001

Red line is training, Blue line is validation

Model A (without weights, train all layers)



Choo Han Ye 1002439

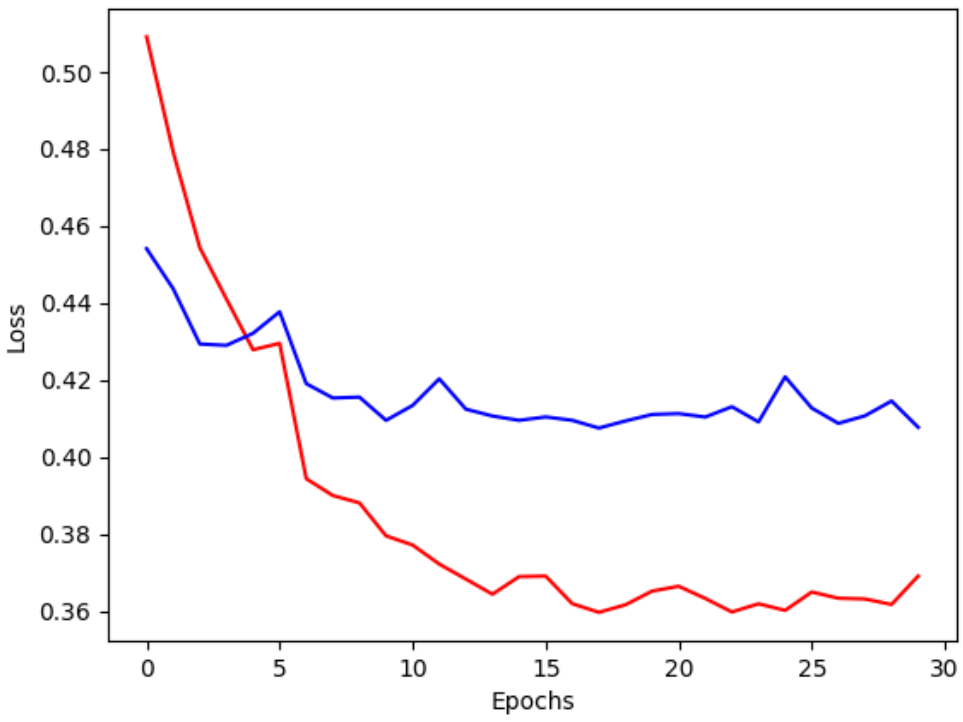


Best Validation Accuracy: 77%, at epoch 25

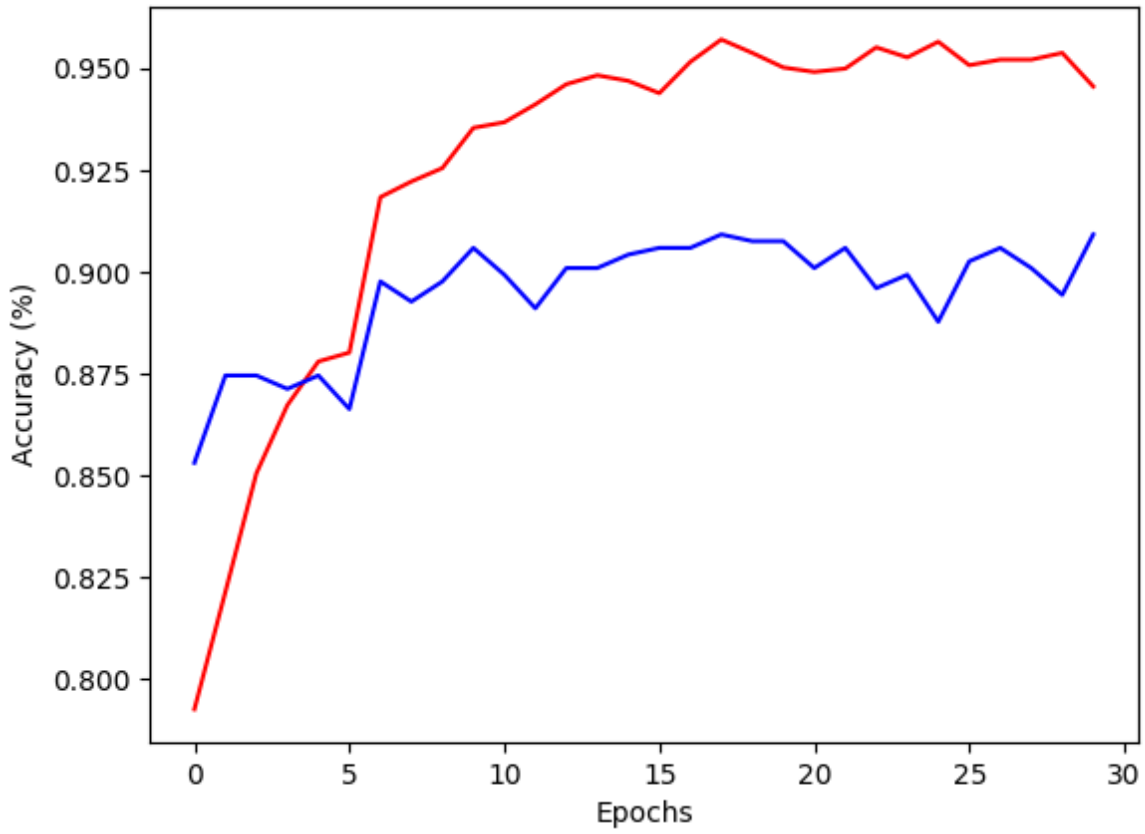
Corresponding Test Accuracy: 75%

Training and Validation Accuracy are more or less consistent.

Model B (loading weights, train all layers)



Choo Han Ye 1002439

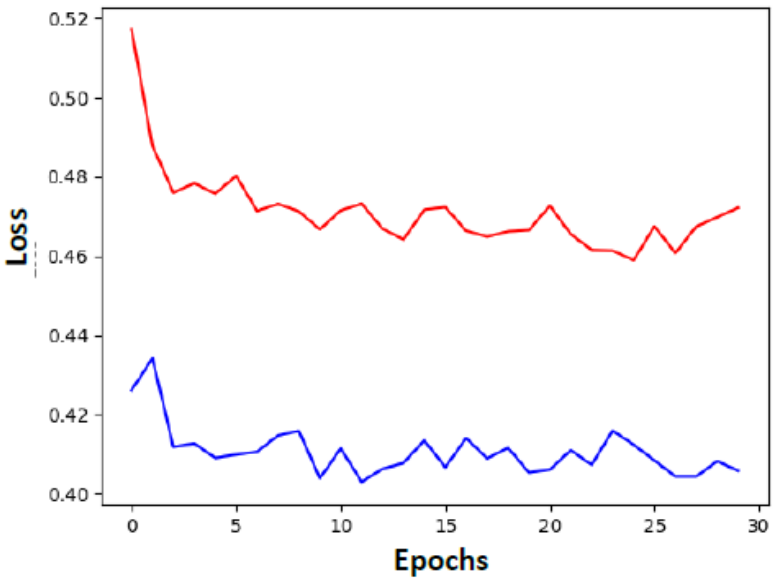


Best Validation Accuracy: 90%, at epoch 25

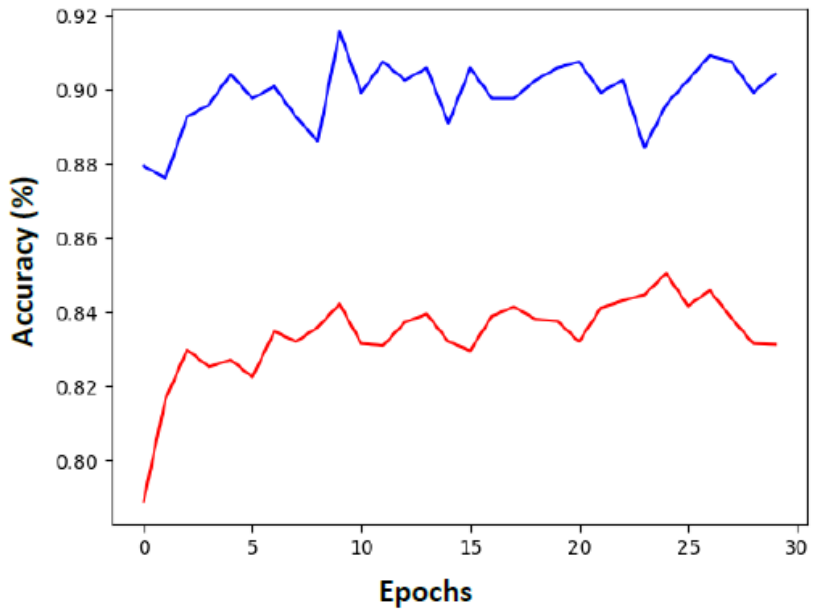
Corresponding Test Accuracy: 88%

The training accuracy is much higher than the validation accuracy. I suspect that there is some kind of overfitting here. Validation accuracy more or less stable across the epochs with minor improvement towards the end.

Model C (loading weights, freeze all except last layer):



Choo Han Ye 1002439



Best Validation Accuracy: 91%, at epoch 9

Corresponding Test Accuracy: 87%

The Validation accuracy is consistently higher than the training accuracy here. It also peaks faster than the previous two models, although it has similar performance to Model B. This may be the most ideal model in this given setting.

Test Set Frequency:

	Indoor	Outdoor	Total
Frequency	591	1230	1821

Given that outdoor is significantly more frequent than indoor, a naïve way to obtain baseline accuracy would be to predict “Outdoor” every single time. This would give us a baseline accuracy of 1230/1821 = 67.5%. It may be noted that all of the model performed better than this baseline accuracy.