SUMMARY OF FINDINGS

I begin by training a convolutional neural network (ConvNet) using a sample of 1000 photos in the training set, with 500 each for validation and testing. Initially, this model exhibits overfitting after approximately 16 epochs, achieving an accuracy of **63.8%** on the test set. To mitigate overfitting, I apply data augmentation, a technique that helps reduce overfitting by introducing variations in the training data. As a result, the validation loss now starts to degrade around 50 epochs, and the regularized model achieves an improved accuracy of **73.4%.**

Next, I double the training sample size to 2000 images and observe that the larger sample still achieves the same **73.4%** accuracy on the test set as the smaller sample with augmentation. However, the larger model overfits more quickly, reaching this accuracy at around 13 epochs.

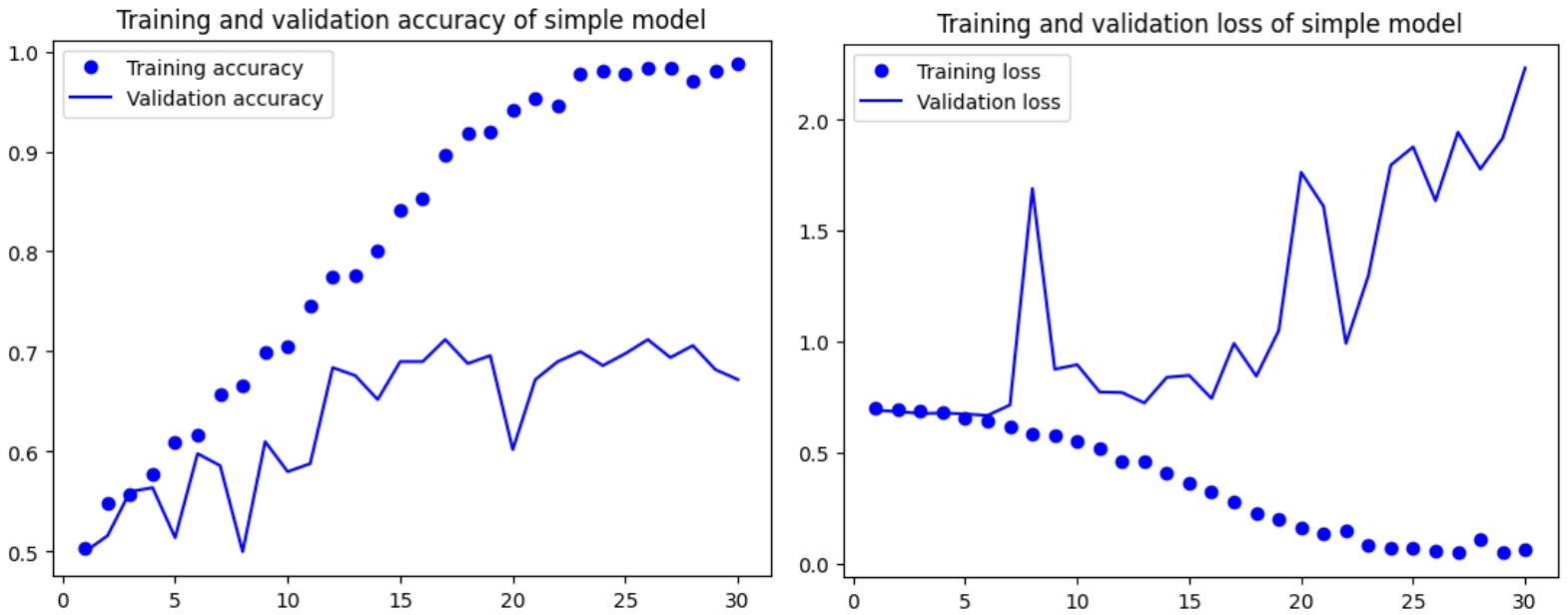
To further enhance performance, I continue using data augmentation on the larger dataset. With this approach, the accuracy on the test set improves significantly, reaching **87.8%,** and the model experiences less trouble with overfitting. The validation loss only stops improving as late as 80 epochs.

From these examples, I conclude that when training a ConvNet from scratch, a larger sample with data augmentation is preferable.

Moving forward, I explore using a pretrained model on the initial small training set of 1000 elements. This pretrained model achieves an impressive accuracy of around **97%** on the test set. However, the validation loss stops improving after only 2 epochs. Surprisingly, when I increase the training set to 2000 elements, the accuracy on the test set drops slightly to **96.6%,** suggesting that the pretrained model tends to overfit as the training data size increases.

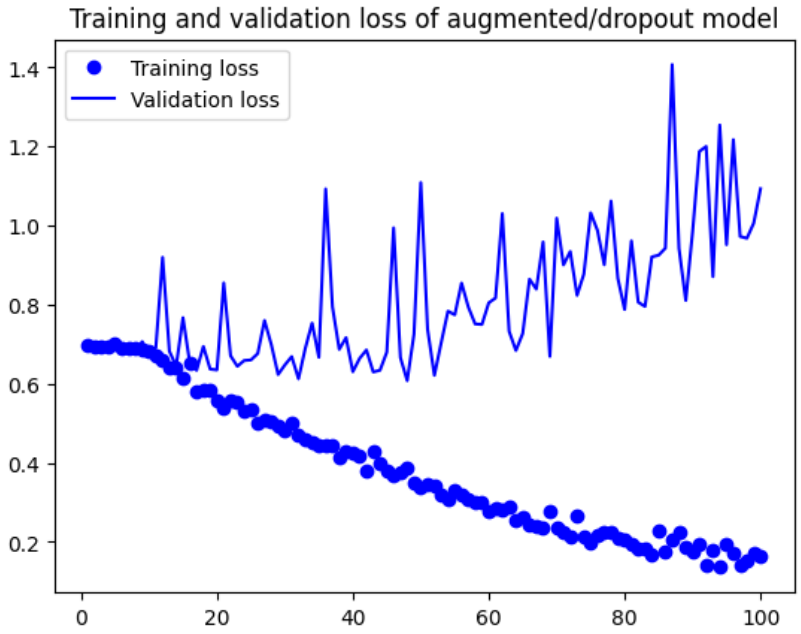
I attempted to improve by applying augmentation to the pretrained model. The training could not be completed due to connectivity issues. As the literature seems to suggest that augmentation on a pretrained model is computationally expensive in exchange for only marginal improvement in accuracy, I decided to end the exercise.

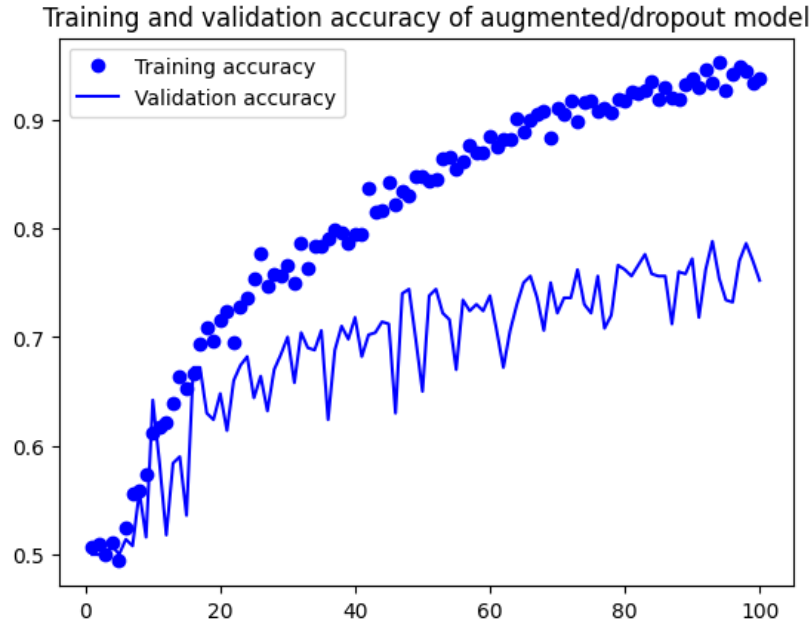
Convnet with small data



The model with little data is overfitted. Training accuracy improves through epochs whereas validation accuracy maxes out around 16 epochs. The model achieves accuracy of 63.8% over the test set.

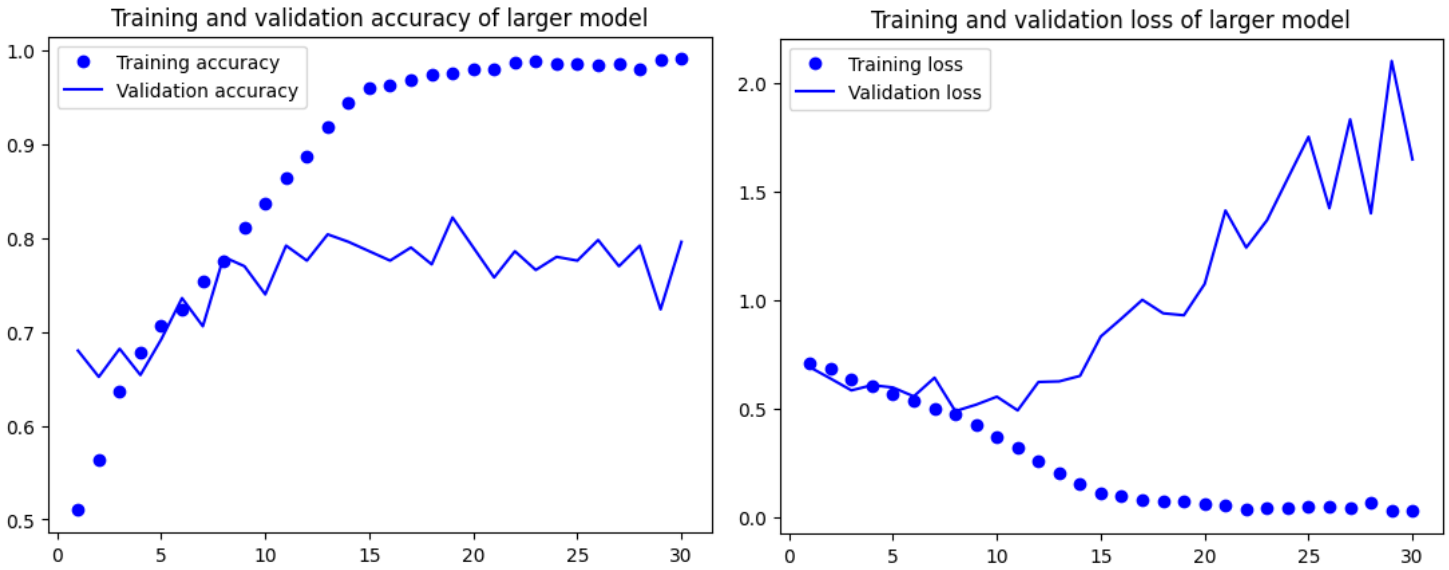
Convnet with small data and augmentation





Validation loss stops improving around 50 epochs. The regularized model obtains 73.4% accuracy on the test set.

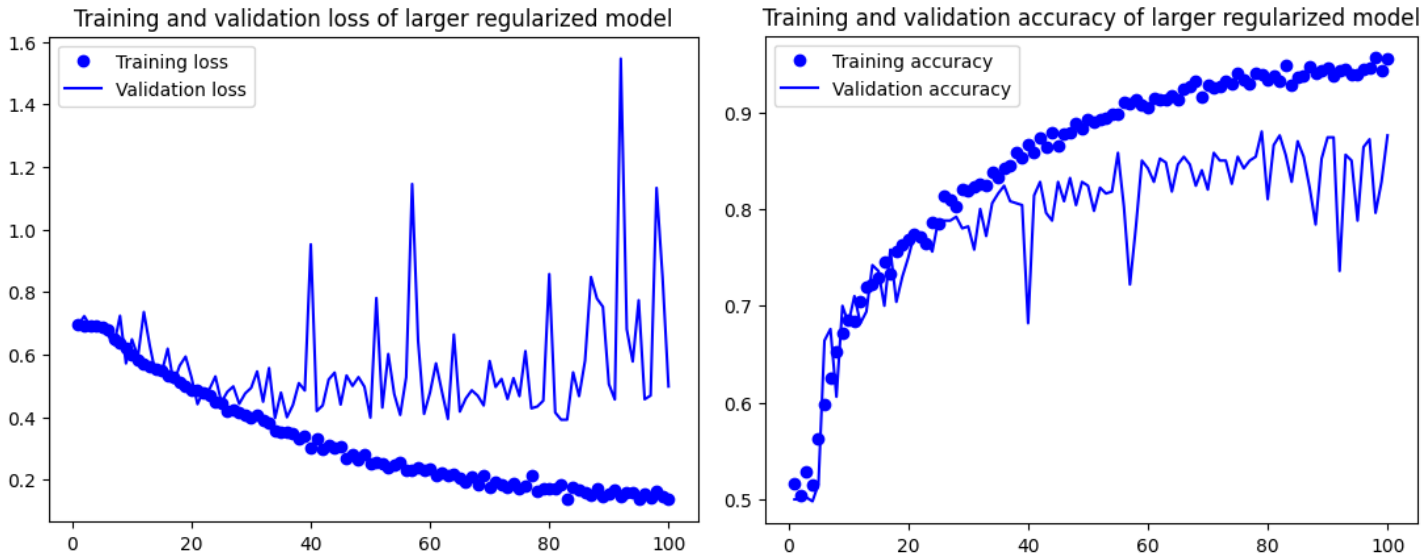
Convnet with increased data



Over fitting only starts around 13 epochs. That is where validation loss stops improving.

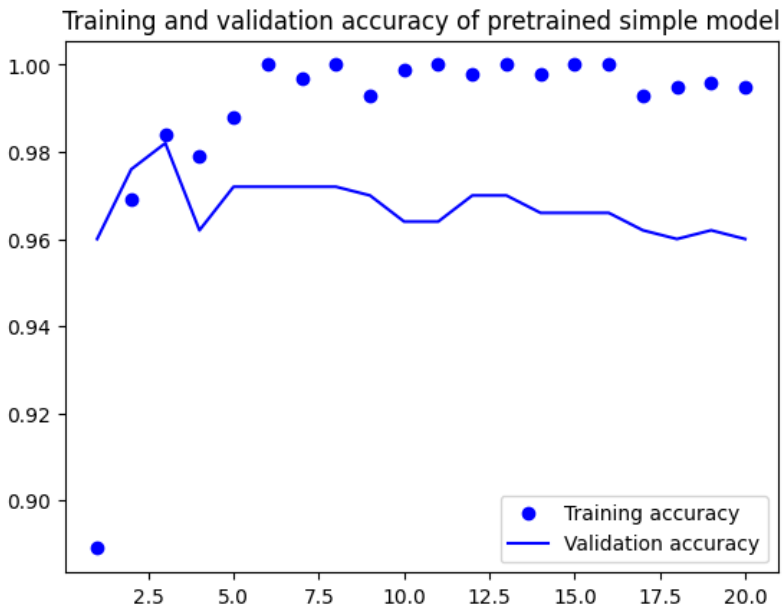
The larger model achieves the same 73.4% accuracy over the test set as the small, regularized model.

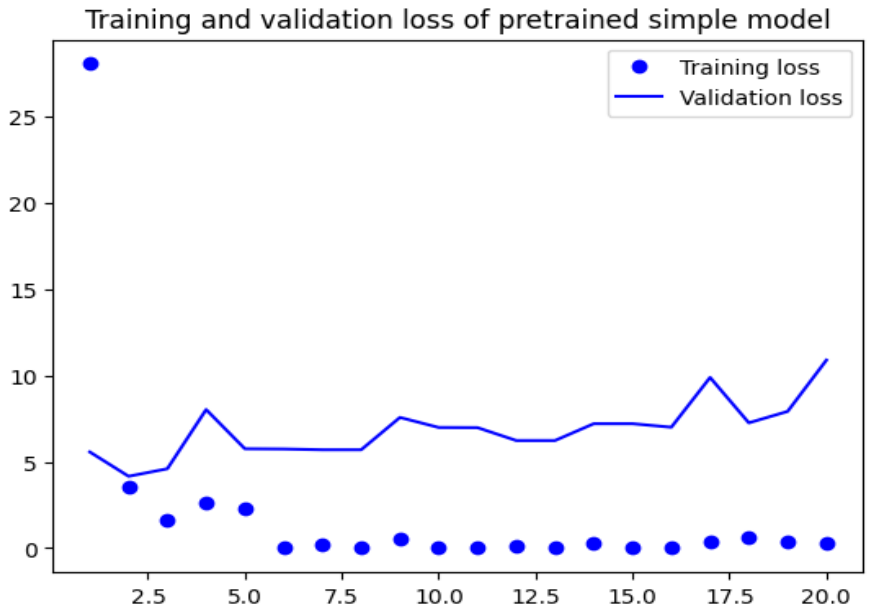
Convnet with increased data and augmentation



The larger regularized model achieved accuracy of 87.8% on test set. Validation accuracy improves with training accuracy over epochs

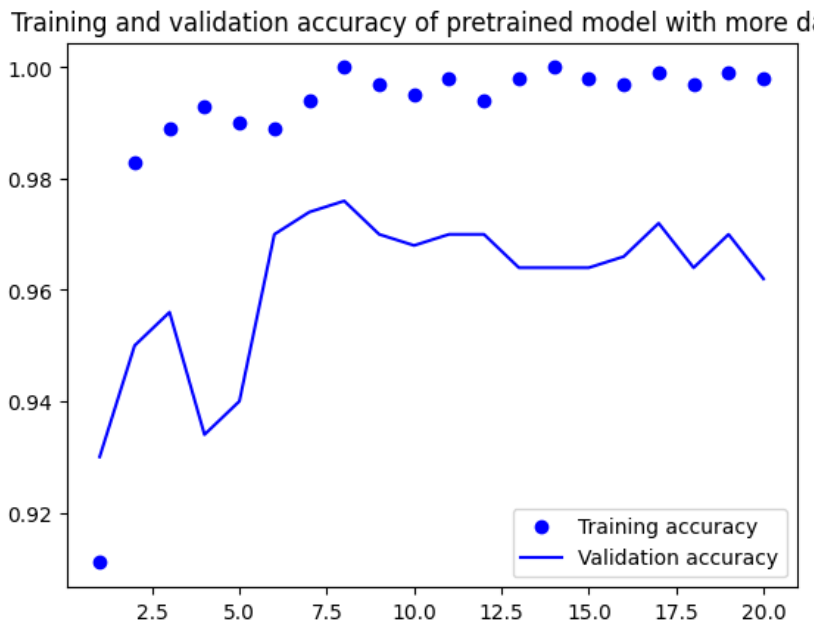
Pretrained ConvNet with small data

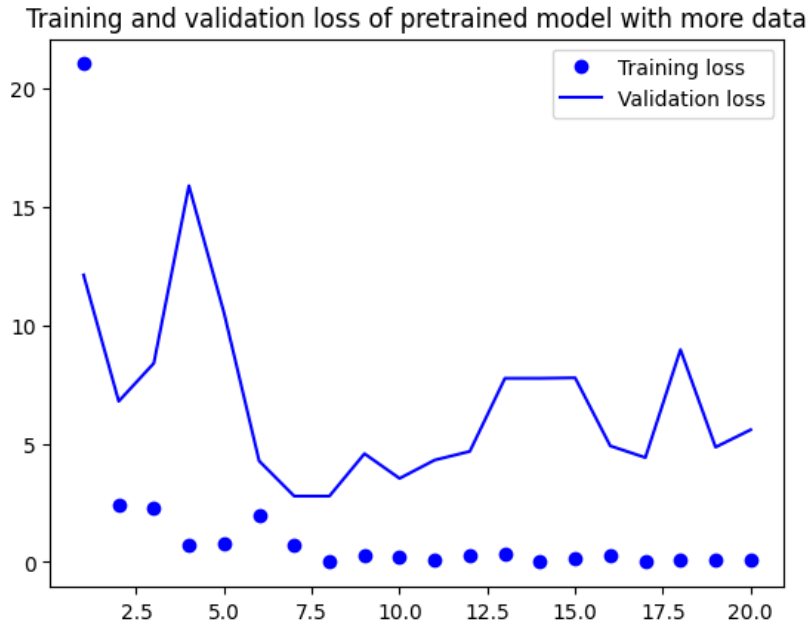




A pretrained model attain validation accuracy of around 98%. Validation loss ceases to improve around the second epoch.

Pretrained ConvNet with more data





Pretrained model with more data achieves 96.6% accuracy. Validation loss stops improving at about 7 epochs