CONVOLUTION

INTRODUCTION:

In this assignment, we explore the application of convolutional neural networks (convnets) to image classification tasks, with a focus on the Cats & Dogs dataset. Convnets are a powerful tool for visual recognition tasks, especially when dealing with image data. The primary goal is to analyse the relationship between the training sample size and the use of network architectures, specifically comparing models trained from scratch versus those utilizing a pretrained network.

We will evaluate the performance of different models on various training sample sizes and optimize them using techniques like data augmentation and regularization to reduce overfitting. The assignment also examines how varying sample sizes influence the overall performance of networks trained from scratch compared to pretrained networks. Through this, we aim to determine the ideal conditions for optimal model performance and gain insights into the advantages of each approach.

This study will provide a comprehensive understanding of how sample sizes and network choices impact the performance of convnets, particularly in the context of image classification tasks. The findings will be summarized in a report, supported by graphs or tables to clearly present the conclusions.

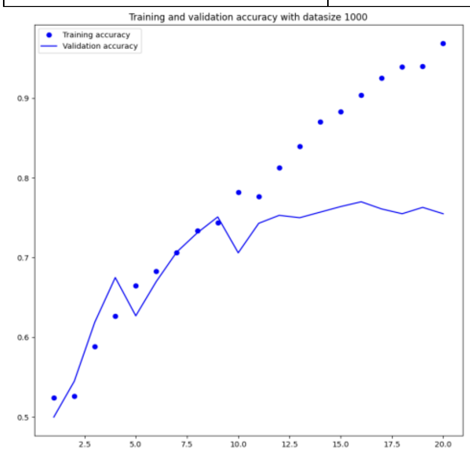
Summary

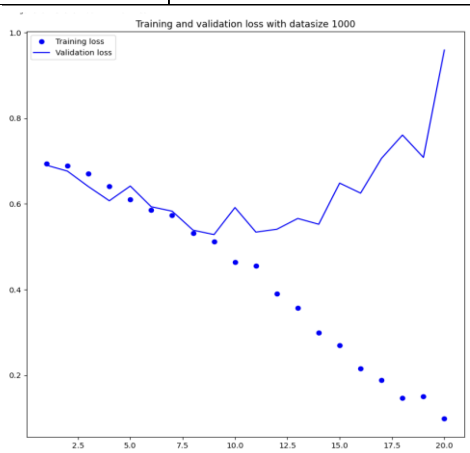
1. **Consider the Cats & Dogs example. Start initially with a training sample of 1000, a validation sample of 500, and a test sample of 500 (like in the text). Use any technique to reduce overfitting and improve performance in developing a network that you train from scratch. What performance did you achieve?**

We have trained a model from scratch with a training sample size of 1000, a validation sample size of 500, and a test sample size of 500. And to reduce overfitting we have used “dropout”.

With dropout:

Model with dropout	***Training***	***validation***	***test***
Accuracy	***96.8%***	***75.5%***	***73.9%***
loss	***0.098***	***0.95***	***0.55***





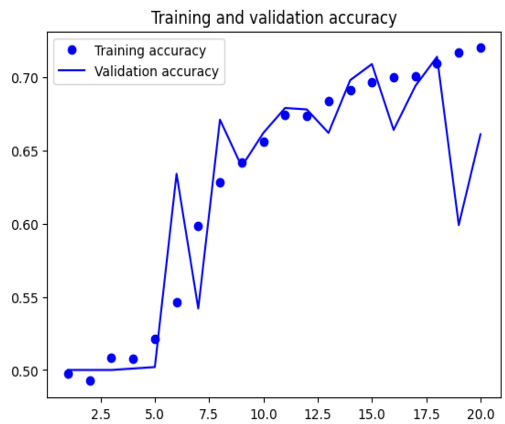
With dropout, the model achieved high accuracy on the training data (96.8%), but validation accuracy plateaued at 75.5%, and test accuracy was 73.9%. The gap between training and validation results, along with a higher validation loss (0.95), indicates overfitting. This occurs because the model is learning the training data well but fails to generalize to new, unseen data. Dropout helped to reduce this overfitting, but with limited data, it could not fully eliminate it.

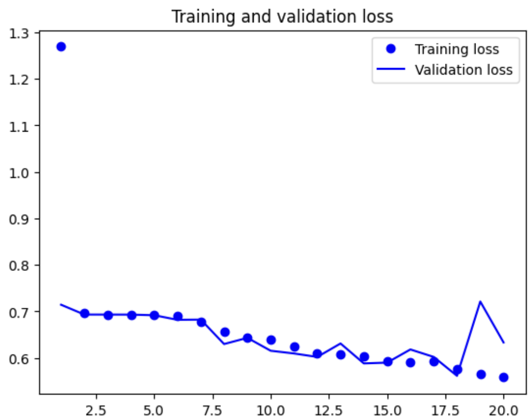
2. Increase your training sample size. You may pick any amount. Keep the validation and test samples the same as above. Optimize your network (again training from scratch). What performance did you achieve?

We have again re-trained the model from scratch with the increase in training sample size to 1500 and by keeping the remaining two sets as same with dropout.

With dropout:

Model with dropout	***Training***	***validation***	***test***
Accuracy	***72.0%***	***66.1%***	***72.5%***
loss	***0.56***	***0.63***	***0.56***





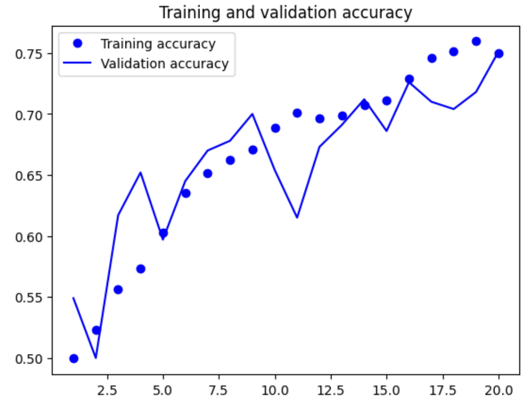
We re-trained the model from scratch, increasing the training sample size to 1500 while keeping the validation and test sets at 500 each. Dropout was used again to mitigate overfitting. The model achieved a training accuracy of 72.0% with a training loss of 0.56, while the validation accuracy was lower at 66.1% with a validation loss of 0.63. On the test set, the model reached an accuracy of 72.5% and a test loss of 0.56. Although the test accuracy improved slightly compared to the validation set, the model still shows some difficulty in generalizing, as reflected in the performance gap between training and validation accuracy.

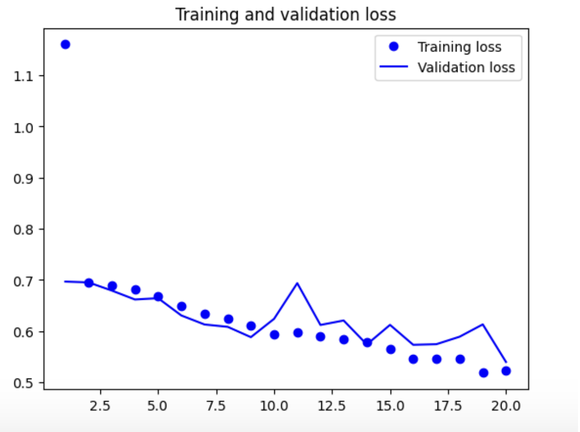
3. Now change your training sample so that you achieve better performance than those from Steps 1 and 2. This sample size may be larger, or smaller than those in the previous steps. The objective is to find the ideal training sample size to get best prediction results?

After comparing the test accuracies of both models as that shows with increase in training sample size the test accuracy increases, we have tried increasing the training sample size to 2000. The results are as follows:

With dropout:

Model with dropout	***Training***	***validation***	***test***
Accuracy	***74.9%***	***75.1%***	***74.7%***
Loss	***0.522***	***0.53***	***0.53***





After comparing the test accuracies of both models, we observed that increasing the training sample size resulted in an improvement in test accuracy. Based on this trend, we further increased the training sample size to 2000, while keeping the validation and test sets at 500 each. The model, trained with dropout to reduce overfitting, achieved a training accuracy of 74.9% with a loss of 0.522, a validation accuracy of 75.1% with a loss of 0.53, and a test accuracy of 74.7% with a test loss of 0.53. This improvement in test accuracy suggests that the model generalizes better with a larger training set.

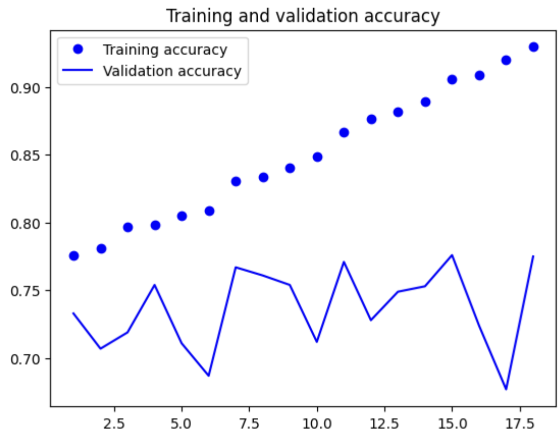
4. Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use any and all optimization techniques to get best performance.

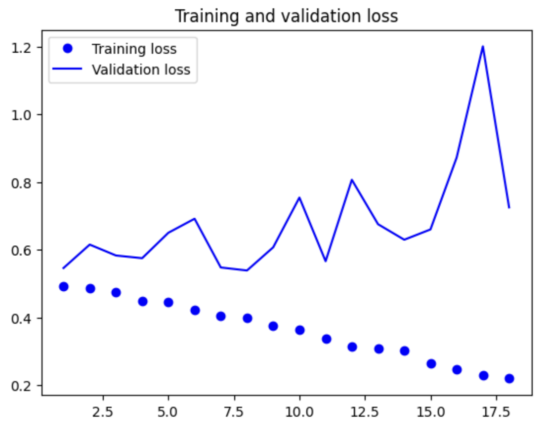
We have used VGG16 architecture as a pretrained network wit three different training data samples sizes. We have used the VGG16 convolution base and classifier with a training sample size of 1000, a validation sample size of 500, and a test sample size of 500. And to reduce overfitting we have used "dropout".

Model Type	***Training Accuracy***	***Validation Accuracy***	***Test Accuracy***
Initial CNN Model	***98.85%***	***72.2%***	***66.6%***
CNN Model with Increased Training	***96.42%***	***86.6%***	***85%***
Optimized CNN Model	***94.97%***	***89.3%***	***89.1%***
Pretrained Model 1	***95.41%***	***91.5%***	***91.5%***
Pretrained Model 2	***94.16%***	***58.6%***	***57.2%***
Pretrained Model 3	***96%***	***98.5%***	***98.6%***

Pretrained Model 1: VGG16 Pretrained Convnet Network

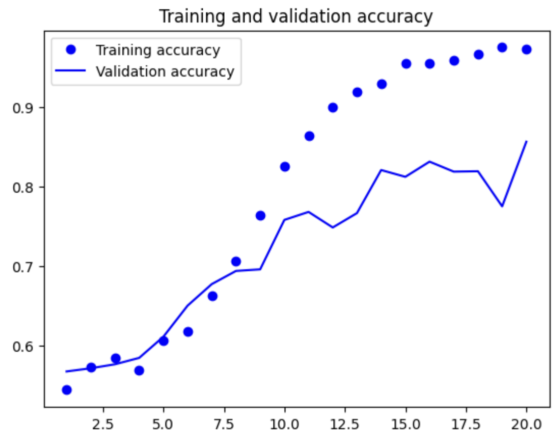
Using the VGG16 convolutional base and transfer learning, the model was trained. The convolutional basis was adjusted for the new dataset following pretraining on the ImageNet dataset. The model architecture includes a classifier and a data augmentation stage. Early halting was done in order to prevent overfitting. A batch size of 32 was used, and the training procedure lasted for 11 epochs. The test accuracy was roughly 91.50%, the validation accuracy was roughly 91.50%, and the training accuracy was roughly 95.41%. The model's great generalization to unknown data is demonstrated by its ability to balance training and validation performance. This demonstrates the efficency of transfer learning for picture categorization problems.

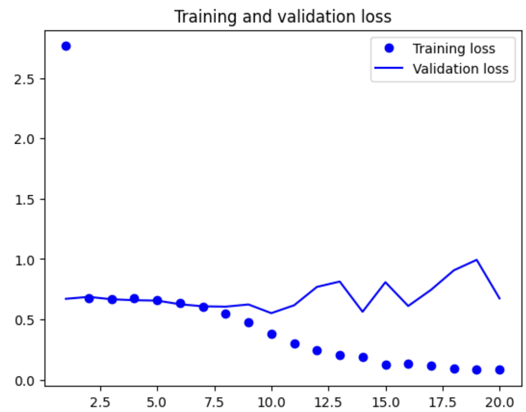




Pretrained Model 2: ResNet50V2 Convolutional Base

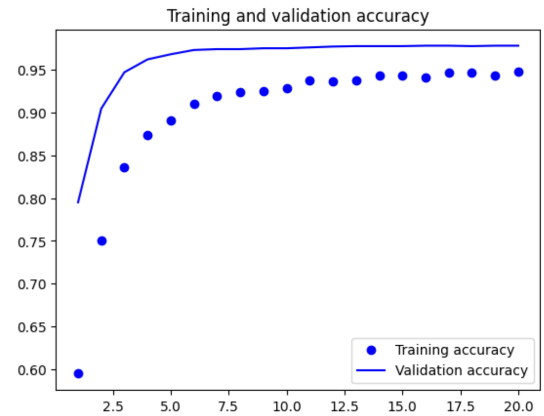
This classified images of cats and dogs using a convolutional neural network (CNN) integrated into TensorFlow's Keras API. Training, validation, and test sets are created from the 5000, 1000, and 1000 picture datasets. Following a number of convolutional layers with max pooling and ReLU activation, the model architecture is composed of fully linked layers. The last layer employs a sigmoid activation function for binary classification. The model is trained using the Adam optimizer and the binary crossentropy loss function. Overfitting may have occurred because the validation accuracy, at 58.60%, is significantly lower than the training accuracy of 94.16%.

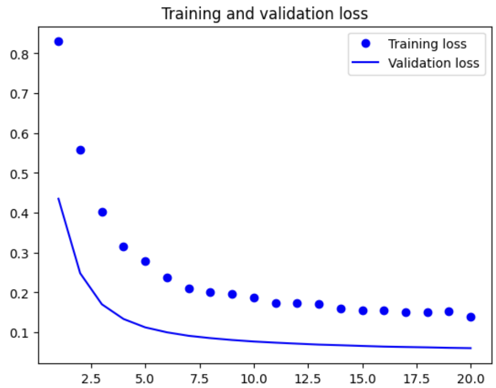




Pretrained Model 3: MobileNetV2 Convolutional Base

The final layers of the model are based on the MobileNetV2 convolutional base and are optimized for binary classification. Zooms, rotations, and random flips were used to enhance the dataset. The model was trained for 50 epochs with early stopping to prevent overfitting. The training accuracy rose across the remaining epochs, from 56.95% to 96.00% at the conclusion. Additionally, the validation accuracy increased from 77.10% to 98.50%. The model performed exceptionally well on the test set, with an accuracy of 98.60%. Transfer learning and data augmentation have been used to achieve high classification accuracy.





Conclusion:

This assignment demonstrated that increasing training sample sizes improves the performance of models trained from scratch, but overfitting can still occur with limited data. Using pretrained networks, like VGG16 and MobileNetV2, significantly enhanced performance, with MobileNetV2 achieving a high test accuracy of 98.6%. This highlights the effectiveness of transfer learning and data augmentation in image classification tasks, showing that pretrained models generalize better and perform more efficiently than models trained from scratch, especially with smaller datasets.