Summary

Cats and Dogs is a data set that contains 25000 images of cats and dogs that can be used for training. I have used a subset of this data for training the model to predict cats and dogs using convolution networks. I used 1000 images to train, 500 images to validate, and 500 to test for the first model. The following image shows the architecture used to build the model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 150, 150, 3)]     0

 rescaling (Rescaling)       (None, 150, 150, 3)       0

 conv2d (Conv2D)             (None, 148, 148, 32)      896

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 conv2d_3 (Conv2D)           (None, 15, 15, 256)       295168

 max_pooling2d_3 (MaxPooling  (None, 7, 7, 256)        0
 2D)

 conv2d_4 (Conv2D)           (None, 5, 5, 256)         590080

 flatten (Flatten)           (None, 6400)              0

 dense (Dense)               (None, 1)                 6401

=================================================================
Total params: 984,897
Trainable params: 984,897
Non-trainable params: 0
```

After building the model, I used the binary cross entropy loss function, Adam optimizer, and accuracy as metrics while compiling the model. Upon running the model, I got a training accuracy of 0.9940, a validation accuracy of 0.7520, and a test accuracy of 0.711.

From the above results it is evident that the model is overfitting as training accuracy is much higher than test accuracy. In order to reduce overfitting I have tried various methods. Augmentation is the method that has produced better results compared to others. It has produced a training accuracy of 0.9415, a validation accuracy of 0.7670, and a test accuracy of 0.743

For the second model, I increased the training data to 3000 from 1000 and kept all others in the model the same as the first model. This has given a training accuracy of 0.9953, a validation accuracy of 0.8210, and a test accuracy of 0.778.

As an attempt to improve the accuracy I used augmentation for this model and it has produced a training accuracy of 0.9538, a validation accuracy of 0.8500, and a test accuracy of 0.819.

From the above results, it can be seen that increasing the number of images in the training set has led to increase in test accuracy.

Considering the above finding, for the third model I have increased the training data to 5000 images. It has produced a training accuracy of 0.9955, a validation accuracy of 0.8450, and a test accuracy of 0.835. The same model with augmentation has produced a training accuracy of 0.9445, a validation accuracy of 0.8960, and a test accuracy of 0.895.

Building a model using a pre-trained network:

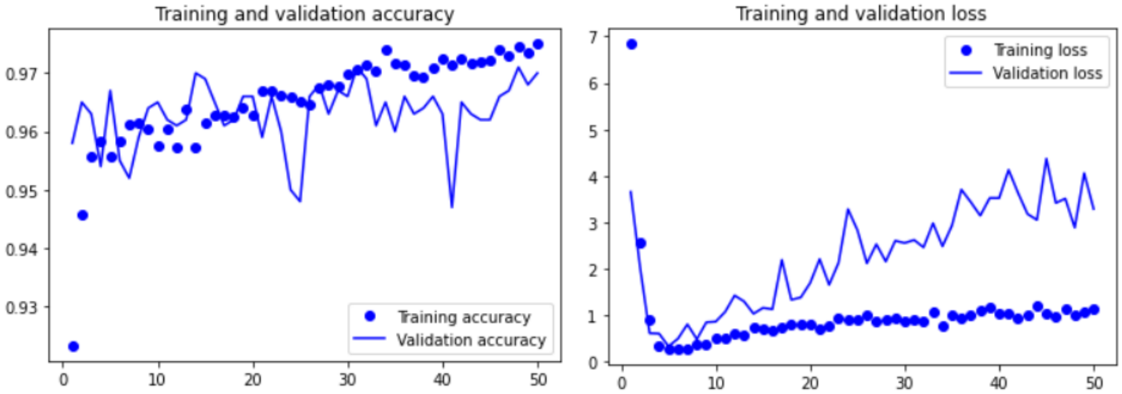
I have used VGG16 pre-trained network to build the model. At the end, I used dense layers, flattener and dropouts, and binary cross entropy loss function, Adam optimizer, and accuracy as metrics while compiling the model.

In for the first model, I used 1000 images to train the model. This has produced the following results:

Training Accuracy	Validation Accuracy	Test Accuracy
0.9900	0.9650	0.959

In the next model, I used 5000 images to train the model and it produced the following results

Training Accuracy	Validation Accuracy	Test Accuracy
0.9750	0.9700	0.972



From the above results, it can be seen that the model with a pre-trained network and the training sample of 5000 has performed better than other models with an accuracy of 97.2. This is because VGG 16 network is trained on very large datasets with millions of images. This model is very likely to perform better than the models that are built from scratch as it has already learned many features from the large data set.

What is the relationship between training sample size and choice of the network?

It is very important to choose the correct sample size before training the model. This is because a small sample size will lead to overfitting the model as the model tries to closely capture the patterns of the training set and it does not generalize well. Such a model doesn't perform well on unseen data. So normally increasing the sample size will improve the performance of the model as it tries to capture more generalized features of the data.

When the training sample size is small it is better to use a simple network with few layers, doing so will help us to avoid overfitting. On the other hand, if the training sample size is large we need to use a complex network with many layers. This is because with the large data set a model can learn many features and that requires a large network with many layers.

Conclusion:

On comparing all the models, it can be seen that with the increase in training sample size, the performance of the model on the test is improved. So, it is better to use large data set to train the model in order to avoid overfitting. The performance of the model has drastically improved when a pre-trained network is used because that network has already learned many features from millions of images. Overall it is better to use large training sets and pre-trained networks to have better performance on the test set.