**Introduction:**

Cats & Dogs is an illustration of how well Convolutional Neural Networks (convnets) perform picture categorization tasks. This abstract delves into two primary methods: using pretrained models and training convnets from scratch. Techniques like data augmentation and regularization are used to combat the problem of overfitting with tiny datasets.

**Dataset:**

Train and test are the two main folders into which the Dogs vs. Cats Kaggle dataset is divided. There are 25,000 tagged photos of cats and dogs in the 'train' section. Your image classification model is trained using these photos and the labels (such as "dog" or "cat") that go with them. 12,500 label less photos may be found in the 'test' folder. The objective for your model is to identify, for every image in the 'test' collection, the proper label (dog or cat). JPEG format is usually used for photographs.

**RESULTS:**  
  
In order to assemble the model, the models were run with the following parameters: optimizer='rmsprop, loss='binary\_crossentropy,' metrics equals "accuracy". Augmentation and drop out layers were utilized at each stage of the model to check for any increases in accuracy.

* There were 1000 base models in the training samples, 500 in the validation samples, and 500 in the testing samples. In the figure below, the loss curve is visible. The training curve begins at a high point and begins to rapidly decline throughout the first few epochs. culminating in an extremely low number around 0 at the conclusion of the training process. The model is overfitting to the training set, as seen by the validation loss, which also declines in the early epochs before beginning to rise once again around epoch 18.

A graph with blue dots

Description automatically generated

* The graph below displays a picture of accuracy curves. At the end of the epochs, the training accuracy increased from a low starting value to extremely high values exceeding 0.9. After reaching a high at epoch 19, the validation accuracy shows a similar early growing pattern before varying between 0.7. Once more, overfitting is shown by this training accuracy graph.

A graph with blue dots

Description automatically generated

* Validation loss and accuracy curves show that the model can fit the training data very well, leading to very low training loss and excellent training accuracy; yet, after a certain point, it is unable to generalize optimally to the unknown validation data. This is a well-known instance of overfitting.   
  The original model produced a test accuracy of 68% using 1000 training samples, 500 validation samples, and 500 testing samples.

63/63 [==============================] - 1s 7ms/step - loss: 0.6055 accuracy: 0.6840

Test accuracy: 0.684

* The test accuracy rose to 58% after drop out layers and model augmentation were applied to the original model.

63/63 [==============================] - 1s 7ms/step - loss: 0.6693 accuracy: 0.5800

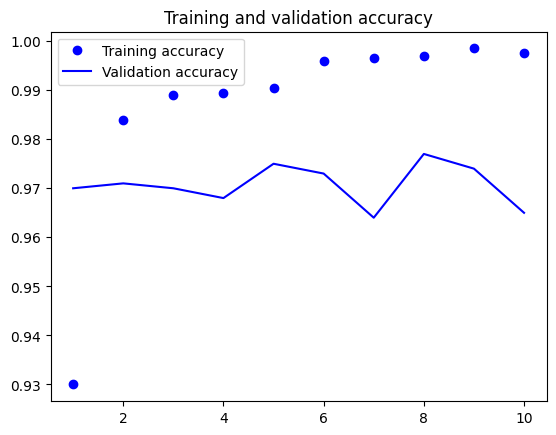
Test accuracy: 0.580

* The model of training samples has been increased to 1000, keeping the Validation samples of 1500, and testing samples of 1500 as above.By the conclusion of training, the training loss has slowly decreased and converged to a very low value near 0. But there are greater variations in the validation loss, with some rises suggesting that the training data may have been overfitted..

A graph with blue lines and white text

Description automatically generated

* The training accuracy shows that the model has fitted the training data quite well, increasing quickly to very high values over 0.99 by the final epochs. Initially, the validation accuracy similarly increases, but in subsequent epochs, it varies between 0.95 and 0.97, not continuously matching the high training accuracy.



* The low training loss and high training accuracy show that the model performs well on the training data toward the end, but the differences between the training and validation curves, particularly in the later epochs, point to possible overfitting. This indicates that even if the model fitted the training set quite well, it might not generalize as well to new data. The validation samples were kept at 1500, the testing samples at 1500, which produced a 96% test accuracy, and the training samples at 1000. As we can see in the video below.

63/63 [==============================] - 0s 2ms/step - loss: 5.7715 - accuracy: 0.9655

Test accuracy: 0.965

* When model augmentation and drop out layers were used to the base model the test accuracy increased to 97%.

63/63 [==============================] - 0s 2ms/step - loss: 8.8282 - accuracy: 0.9700

Test accuracy: 0.970

* In the training and validation accuracy curve graph. Training accuracy gradually increases from 0-10 epochs. And validation accuracy gradually decreases from the point 0.97% and later increased near to the point 0.98%.

A graph with blue dots

Description automatically generated

* In the training and validation loss curve graph. Training loss didn’t increases and it is conistant from 0-10 epochs. And validation losss gradually increases from the point 5% and later decreased near to the point 4%. And later again increased above the point of 6%.

A graph with red lines and white text

Description automatically generated

**Conclusion:**

The size of the training sample and the network selection for image classification are directly correlated. For image classification applications, deeper networks are needed to get improved performance as the training sample size grows. Larger training sample sizes further mitigate the problem of overfitting, enabling the construction of more complicated models without compromising their ability to generalize on unobserved data. Simpler networks can occasionally outperform more complex ones on image classification tasks, even with smaller training sample sizes. This may happen if there are few visual cues or patterns in the categorized photos that are simple for shallower networks to identify, or if the network design has been precisely tailored and optimized for the particular image classification issue.

We can see that when the number of training samples rose for the present dataset—which is picture categorization of dogs vs cats—the model performed better. 2000 is the ideal training sample, and the model performs at 0.97. Apart from the size of the training sample and network design, performance in image classification is determined by several additional factors. This includes the optimization process, the regularization techniques used in training, and the quality of the pretreatment of the data. To guarantee that an image classification system is well-tuned and capable of attaining high accuracy on a variety of datasets, all these factors must be taken into account while constructing the system. Additionally, we included data augmentation and pre-trained convents in this model, which greatly enhanced. Therefore, while creating an image classification system, it is essential to take all of these elements into account for reliable and accurate findings.