|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Training Set** | **Validation Set** | **Test Set** | **Data augmentation** | **Pretrained Model** | **Loss and Accuracy on Test** |
| 1. | 1000 | 500 | 500 | None | None | loss: 0.6202 - accuracy: 0.6420 |
| 2. | 3000 | 500 | 500 | None | None | loss: 0.5253 - accuracy: 0.7300 |
| 3. | 16000 | 500 | 500 | None | None | loss: 0.4193 - accuracy: 0.9020 |
| 4. | 1000 | 500 | 500 | Yes | None | loss: 0.6442 - accuracy: 0.6680 |
| 5. | 3000 | 500 | 500 | Yes | None | loss: 0.4832 - accuracy: 0.7600 |
| 6. | 16000 | 500 | 500 | Yes | None | loss: 0.2034 - accuracy: 0.9040 |
| 7. | 1000 | 500 | 500 | Yes | Yes | loss: 1.9545 - accuracy: 0.9820 |
| 8. | 3000 | 500 | 500 | Yes | Yes | loss: 1.9789 - accuracy: 0.9820 |
| 9. | 16000 | 500 | 500 | Yes | Yes | loss: 0.0871 - accuracy: 0.9840 |

**FINDINGS:**

* All the above models have been run with the following parameters to compile the model: optimizer**=**'adam', loss**=**'binary\_crossentropy,’ metrics**=**['accuracy']).
* The model has no additional parameters to find the optimal training sample size for the current dataset. The base models in training samples of 1000, 3000, and 16000 showed accuracies being increased as the training samples increased.
* We observed that increase in accuracy, even though the validation and test samples were set to 500 for every iteration. This could be because the model was learning more features of the images in the training samples.
* The base training sample size of 1000 showed loss: 0.6202 - Accuracy: 0.6420. The below results show that for 20 epochs, the training accuracy increased for every epoch, but the validation accuracy peaked at the 4th epoch at 0.61.

**Chart, scatter chart

Description automatically generated Chart, line chart

Description automatically generated**

* For the training samples of 3000, the loss was 0.5253 – accuracy was 0.7300. As the samples increased from 1000 to 3000, the test accuracy increased by 13%. The below results show that the validation accuracy peaked at the 10th epoch at 0.7560.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

* The maximum accuracy was achieved in training samples of 16000 at “0.9020”. For every other training sample, i.e., above and below 16000 samples, the model’s accuracy varied between 0.85 to 0.89. This shows the model extracted maximum features with training samples of 16000 with no additional parameters applied, even though the validation and test sets had a constant sample size of 500 each.

Shape, line chart

Description automatically generatedChart, line chart

Description automatically generated

* To reduce overfitting, various methods can be applied to the dataset. Two ways are used for the current dataset to reduce overfitting and increase the model's performance. They are Data augmentation and pre-trained convents.
* Data augmentation generates more samples from the existing training samples by applying various random transformations. This ensures that the model never sees the same picture twice, which helps to learn more data features.
* A pre-trained network is trained on a vast dataset, usually an extensive image classification dataset. When the original dataset is sizable and diverse, the knowledge acquired by the network can be applied to other image classification tasks, even if these tasks are distinct from the original dataset. A pre-trained network can serve as a foundation for various image classification challenges.
* For the current dataset, the following data augmentation parameters have been used for all the models: layers.RandomFlip("vertical"), layers.RandomRotation(0.1),

layers.RandomZoom(0.2).

* For all three base models, data augmentation was applied to improve the model’s base performance. The results are as follows:
  + The validation accuracy of training samples of 1000 peaked at 0.5140 at the 7th epoch, and the test accuracy increased by only 4%, i.e., 0.6420 to 0.6680.
  + The validation accuracy of training samples of 3000 peaked at 0.6780 at the 7th epoch, and the test accuracy increased by only 4%, i.e., 0.7300 to 0.7600.
  + The validation accuracy of training samples of 16000 peaked at 0.8640 at the 8th epoch, and the test accuracy increased by only 0.2%, i.e., 0.9020 to 0.9040.
* As per the above results, data augmentation wasn’t as effective for the current dataset as the base model’s performance, which showed no significant improvement.
* To improve the model performance further, we apply pre-trained convents with data augmentation and fine-tune the model for maximum performance. We use the imageNet dataset trained on the VGG16 architecture for the current dataset, consisting of 1.4 million labeled images and 1000 different classes.
* In a pre-trained network, we use two ways to improve performance: 1. Feature extraction 2. Fine-tuning. Feature extraction consists of using representations learned by a previous network to extract interesting features from new samples. These features are then run through a new classifier trained from scratch.
* Fine-tuning consists of unfreezing a few top layers of a frozen model base.  
  for feature extraction and jointly training both the newly added part of the model (in  
  this case, the fully connected classifier) and these top layers.
* Applying both feature extraction with data augmentation and fine-tuning for all three models, we achieved the following results:
  + The validation accuracy of training samples of 1000 peaked at 0.9740 at the 5th epoch, and the test accuracy increased by 47%, i.e., 0.6680 to 0.9820.
  + The validation accuracy of training samples of 3000 peaked at 0.5140 at the 7th epoch, and the test accuracy increased by only 29%, i.e., 0.7600 to 0.9820.
  + The validation accuracy of training samples of 16000 peaked at 0.5140 at the 7th epoch, and the test accuracy increased by only 8%, i.e., 0.9040 to 0.9840.

**Conclusion:**

There is a direct relationship between the training sample size and the network selection for image classification. As the training sample size increases, deeper networks are required for image classification tasks to achieve better performance. Additionally, overfitting becomes less of an issue with larger training sample sizes, allowing more complex models to be trained without sacrificing generalization performance on unseen data. Sometimes, even with smaller training sample sizes, simpler networks can perform well on image classification tasks. This can occur when the classified images contain limited visual features or patterns easily detected by shallower networks or if the network architecture has been carefully optimized and tuned for the specific image classification problem.

For the current dataset, which is image classification of Dogs vs. Cats, we can observe that the model performed better as the training samples increased. The optimal training sample is 16000, with a model performance of 0.9840. Performance in image classification depends on several other criteria besides the training sample size and network architecture. The standard of data pretreatment, the regularization methods employed during training, and the optimization methodology all fall under this category. All these criteria must be considered when building the system to ensure that an image classification system is well-tuned and capable of achieving high accuracy on various datasets. We also used pre-trained convents and data augmentation in this model, which helped significantly improve. Therefore, it is crucial to consider all these factors for accurate and trustworthy results while developing an image classification system.