Convolutional Neural Network for Cats and Dogs Image Classification

1. Objective

The main goal of this exercise is to find out how the training dataset size and model architecture type affect the performance of a Convolutional Neural Network (CNN).
The selected problem is the famous Cats and Dogs image classification task.

The objective was to first train a CNN from scratch on a small dataset and then try several approaches to battle overfitting and improve the accuracy.
Later on, the same exercise was performed using a pre-trained deep learning model and compared with one that was trained from the start.
The end goal is to find out the relationship between training sample numbers and with performance of the model, and what best fits to achieve maximum accuracy.

2. Experimental Setup

The experiment was conducted using the Dogs vs Cats dataset offered by Kaggle. The dataset contains 25,000 labeled images with a balanced mix of dogs and cats.
For this task, different subsets of data were created with differing sizes of training samples but with constant validation and test samples.
All the experiments were executed in Google Colab using Python and TensorFlow with Keras.

Parameter	**Description**
Dataset	Kaggle – Dogs vs Cats
Image Size	180 × 180 pixels
Batch Size	32
Training Samples	Varied (1000, 5000, 10000)
Validation Samples	500
Test Samples	500
Optimizer	RMSProp
Loss Function	Binary Cross Entropy
Evaluation Metric	Accuracy
Techniques Used	Data Augmentation, Dropout, Early Stopping, Transfer Learr (VGG16)

3. Step 1: Model Trained from Scratch with 1000 Samples

In the first experiment, a single CNN model was constructed and trained from scratch. Data was divided into the training, validation, and test sets of 1000 images, 500 images, and 500 images, respectively.
The architecture had five convolutional layers with MaxPooling layers thereafter and a Dense layer at the final stage with sigmoid activation.

Model	Training Samples	Regularization	Training Accuracy	Validation Accuracy	Test Accura	Observation
Model 1	1000	None	98.5%	73.5%	74.1%	Strong overfitting due minimal training data

Analysis:
The model performed excellently on the training data but performed poorly on the validation and test sets.
This is because the model memorized the limited training data rather than the general features.
The accuracy plot showed that training accuracy kept increasing, while validation accuracy leveled off, indicating clear overfitting.

4. Step 2: Reducing Overfitting (Using Same 1000 Samples)

Here, the same number of training samples (1000) was employed, but strategies were used to counter overfitting and enhance generalization.
Three variations of the model were attempted:

1. **Dropout Method**: Turn off randomly 50 percent of the neurons during training to avoid co-dependency.
2. **Data Augmentation**: Added random rotations, flips, and zooms to diversify images.
3. **Dropout with Data Augmentation**: Applied both dropout and data augmentation simultaneously to get the highest effect.

Model	Training Samples	Technique	Test Accuracy	Observation
Model 1a	1000	Data Augmentation	80.9%	Improved generalization slightly
Model 1b	1000	Dropout (0.5)	74.7%	Helped to control overfitting marginally
Model 1c	1000	Data Augmentation + Dro	82.2%	Best performance among all three

Analysis:
Alongside augmentation, dropout improved model performance from 74 percent to around 82 percent.
This happened because data augmentation produced synthetic image variations, and dropout ensured that the model did not over-depend on certain neurons.
But there was not abundant training data, and that prevented the model from being very precise.

5. Step 3: Training Sample Size Increased to 5000

In the second step, the training dataset was increased to 5000 images, and validation and test sets were kept at 500 each.
The same CNN model was used, and Data Augmentation, along with Dropout (rate of 0.5) were utilized.

Model	Training Samples	Regularization	Training Accuracy	Validation Accuracy	Test Accu	Observation
Model 2	5000	Data Augmentation Dropout	97.5%	98.8%	87.6%	Major improvement; overfitting been reduced significantly

Analysis:
Since it had a greater amount of training data, the model started learning general patterns and showed clear improvement in test and validation accuracy.
The accuracy of the validation and training set was significantly brought down, and this showed that overfitting had occurred.
Overall accuracy was enhanced by nearly 15 percent with the bigger dataset.
This test clearly validated that the optimal way to improve model performance is to provide more training data.

6. Step 4: Increasing Training Sample Size to 10000
Next, the training data was increased to 10000 images, keeping the validation and test data the same.
The same network structure and regularization methods were used.

Model	Training Samples	Regularization	Training Accuracy	Validation Accuracy	Test Accuracy	Observation
Model 3	10000	Data Augmentation Dropout	96.7%	91.8%	90.0%	Stable model minimal overfitting

Analysis:
On 10000 training images, the model achieved approximately 90 percent test accuracy.
Convergence of training and validation curves was seen, indicating good generalization.
Additional data again indicated poorer improvement, which suggests approximately 10000 images was some sort of optimal dataset size for this task when training from scratch.

7. Step 5: Using Pre-Trained VGG16 Model

To get even improved performance, the pre-trained VGG16 model was utilized.
This model was pre-trained on the ImageNet dataset, which contains millions of images.
Pre-trained convolutional layers were used as a feature extractor, and new dense layers were added to finish the cat vs dog classification task.
The same data splits were employed in fine-tuning the pre-trained model.

a) Pre-Trained Model with 1000 Training Samples

Model	Training Samp	Validation Accuracy	Test Accuracy	Observation
Pre-Trained VGG16	1000	97.6%	97.2%	Sharp increase in accuracy compare training from scratch

Even with only 1000 samples, the pre-trained model gave 97 percent test accuracy because it already had rich feature representations from the ImageNet dataset.

b) Pre-Trained Model with 5000 Training Samples

Model	Training Samples	Validation Accurac	Test Accuracy	Observation
Pre-Trained VGG1	5000	98.0%	97.8%	Consistent and stable results

The model quickly converged during training and showed consistent improvement without overfitting.
It was also computationally more efficient since fewer epochs were needed to achieve high accuracy.

c) Pre-Trained Model with 10000 Training Samples

Model	Training Sampl	Validation Accura	Test Accura	Observation
Pre-Trained VGC	10000	98.3%	99.2%	Nearly perfect accuracy achieved

This was the best-performing configuration. The pre-trained model with fine-tuning and augmentation achieved almost 99 percent accuracy on the test set.

8. Summary

Model Type	Training Sampl	Regularization	Test Accuracy	Key Points
From Scratch	1000	None	74.1%	Severe overfitting
From Scratch	1000	Dropout + Augmentation	82.2%	Improved generalization
From Scratch	5000	Dropout + Augmentation	87.6%	Stable performance
From Scratch	10000	Dropout + Augmentation	90.0%	Best scratch model
Pre-Trained VGG16	1000	Fine-Tuning + Augmentation	97.2%	High accuracy with small data
Pre-Trained VGG16	5000	Fine-Tuning + Augmentation	97.8%	Consistent and fast convergence
Pre-Trained VGG16	10000	Fine-Tuning + Augmentation	99.2%	Best overall accuracy

9. Observations

1. Models trained from scratch were not able to handle small data and had excessive overfitting.
2. Using data augmentation and dropout helped the performance even with small data.
3. Tripling the training size from 1000 to 10000 samples provided a huge increase in test accuracy.
4. The pre-trained model performed significantly better even on smaller datasets.

5. Transfer learning is extremely efficient since it leverages knowledge acquired through a large dataset.
6. Precision increases extremely fast at first when data rises but diminishes after a point.
7. The VGG16 model trained on 10000 samples produced the best compromise of stability, speed, and precision, obtaining around 99 percent.

10. Conclusion

From this study, it is clear that both training data size and model selection have a direct impact on the accuracy of a CNN.
A model trained from scratch can achieve good accuracy only when enough data and regularization are provided.
However, when data is limited, techniques like Data Augmentation and Dropout help reduce overfitting and improve results.

The use of a pre-trained network such as VGG16 significantly improves performance.
Although having limited data, it could be extremely accurate as it has already learned general features common to most images.

During our project, a pre-trained VGG16 model trained on more than 10000 samples achieved almost 99 percent test accuracy, which was the highest among all experiments.
It shows that using transfer learning and data augmentation, along with dropout, is the optimum and most practical answer to real-world image classification problems.

11.References

1. Kaggle. (2013). *Dogs vs Cats Dataset.*
2. Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition.*
3. Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting.* Journal of Machine Learning Research.
4. Chollet, F. (2015). *Keras: The Python Deep Learning Library.*
5. TensorFlow Developers. (2023). *TensorFlow: Machine Learning Framework.* Google Brain.
6. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). *ImageNet: A Large-Scale Hierarchical Image Database.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).