Convolutional Neural Networks for Image Classification: Cats vs. Dogs

1. Introduction

The objective of this study is to analyze the impact of training sample size on the performance of convolutional neural networks (CNNs) for binary image classification. The task involves distinguishing between images of cats and dogs using two approaches:

- **Training a CNN from scratch:** Developing a model with randomly initialized weights and training it solely on the given dataset.
- **Utilizing a pretrained VGG16 model as a feature extractor:** Leveraging an existing, pre-trained network to extract features and adding custom layers for classification.

2. Dataset Description

The dataset consists of **2000 images** equally split between cats and dogs. The dataset is divided as follows:

- **Training Set:** Initially set at **500 images** (250 cats, 250 dogs), then increased to **1000, 1500, and 2000** images.
- **Validation Set:** Fixed at **500 images**.
- **Test Set:** Fixed at **500 images**.

3. Methodology

3.1 Data Preprocessing

- **Rescaling:** All images were resized to **150x150 pixels** and normalized (pixel values scaled to [0,1]).
- **Augmentation Techniques:** Applied **rotation, width and height shifts, shear, zoom, and horizontal flips** to mitigate overfitting.

- **Batch Processing:** Images were processed in mini-batches of **32** during training.
- **Random Sampling:** Balanced subsets were created using `flow_from_directory` with shuffling and a fixed seed for consistency.

3.2 Model Architectures

3.2.1 Training a CNN from Scratch

A CNN trained from scratch requires learning feature representations directly from the dataset. The custom CNN consists of:

- Four convolutional layers with filters **(32, 64, 128, 128)**.
- **Max-pooling**, **dropout(0.5)**, and **L2 regularization** for better generalization.
- A final **dense layer** with sigmoid activation for binary classification.

While training from scratch allows full control over the model, it requires a large dataset to generalize effectively and avoid overfitting.

3.2.2 Utilizing a Pretrained VGG16 Model

Instead of training from scratch, a pretrained VGG16 model is used as a feature extractor. This approach leverages:

- **Feature Extraction:** The convolutional base of **VGG16** (trained on ImageNet) is used with frozen weights.
- **Custom Classifier:** Additional **flatten, dense, and sigmoid layers** are added to adapt the model for binary classification.

By freezing the pretrained convolutional layers, the model benefits from feature representations learned from a large-scale dataset, reducing the need for extensive training data and improving generalization.

3.3 Training Strategy

- **Loss Function:** Binary Crossentropy.

- **Optimizer:** Adam with a learning rate of **1e-4**.
- **Early Stopping:** Prevented overfitting by stopping training when validation loss plateaued.
- **Model Checkpointing:** Saved the best-performing model during training.

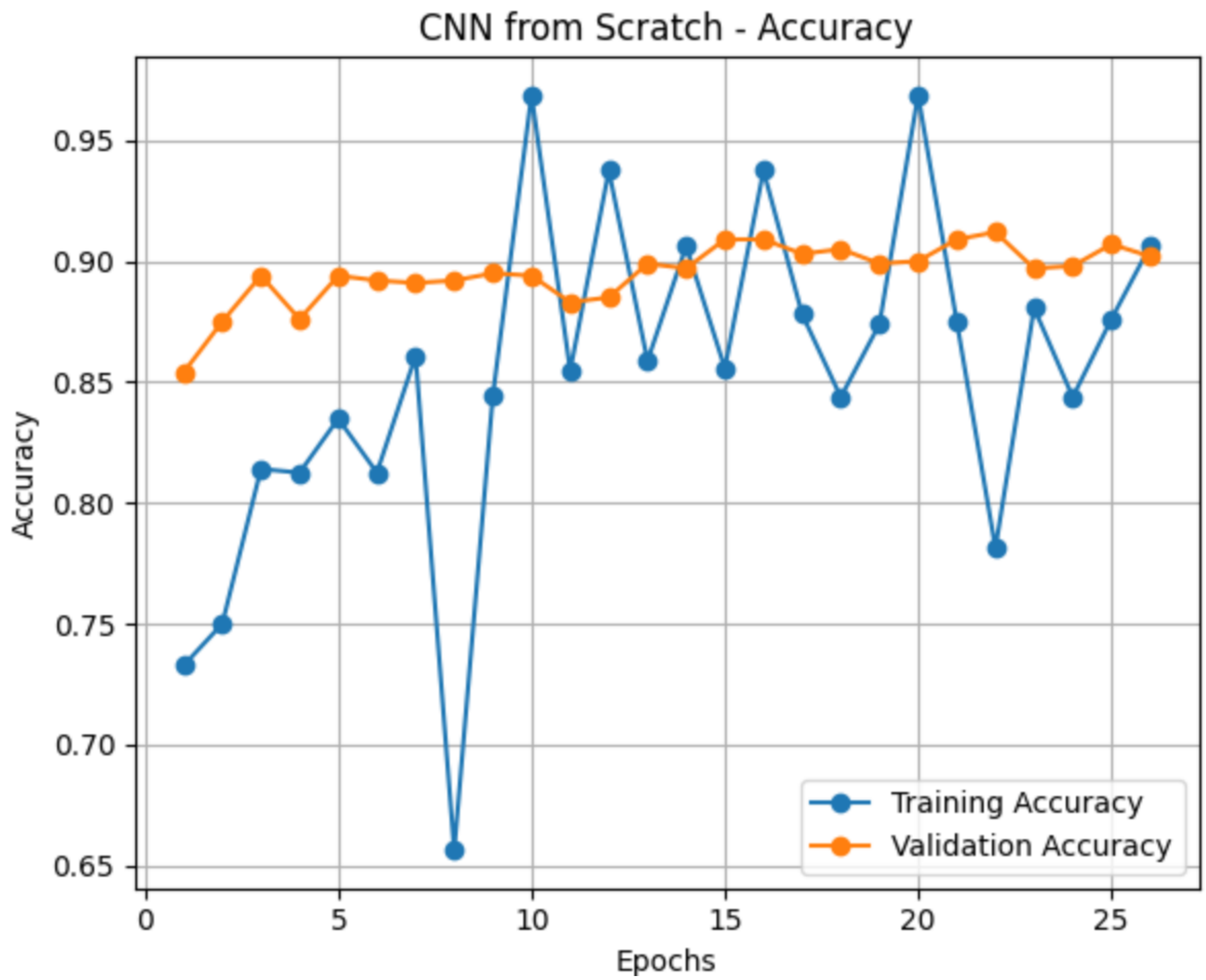
4. Experimental Results

The results across different training dataset sizes are summarized below:

Training Size	Scratch Model Accuracy	Pretrained Model Accuracy	Improvement(%)
500	66.20%	88.60%	33.84%
1000	68.00%	87.90%	29.2647
1500	**71.20%**	88.50%	24.30%
2000	66.80%	**89.60%**	**34.13%**

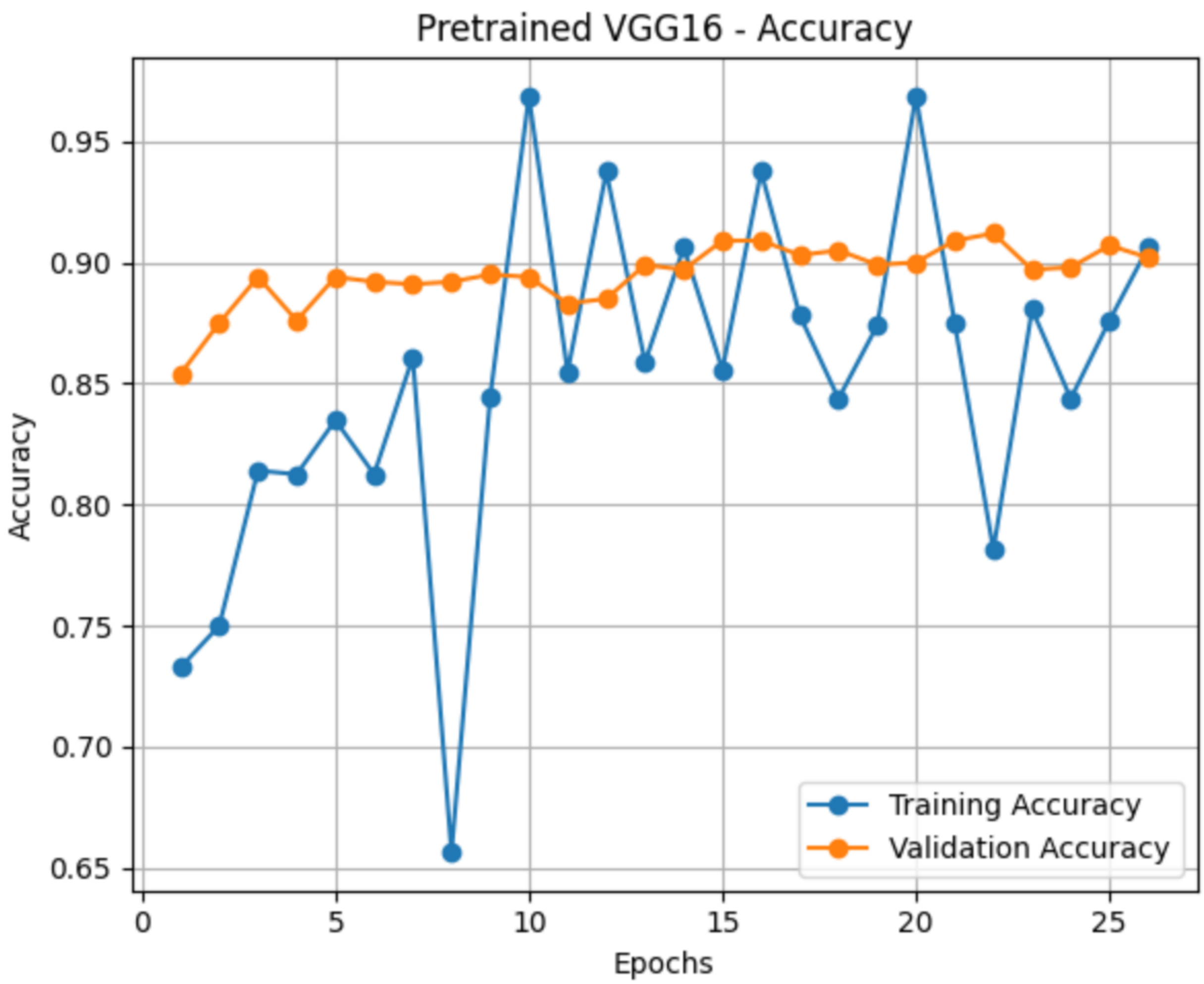
5. Visual Analysis

5.1 Training and Validation Accuracy



Scratch Model

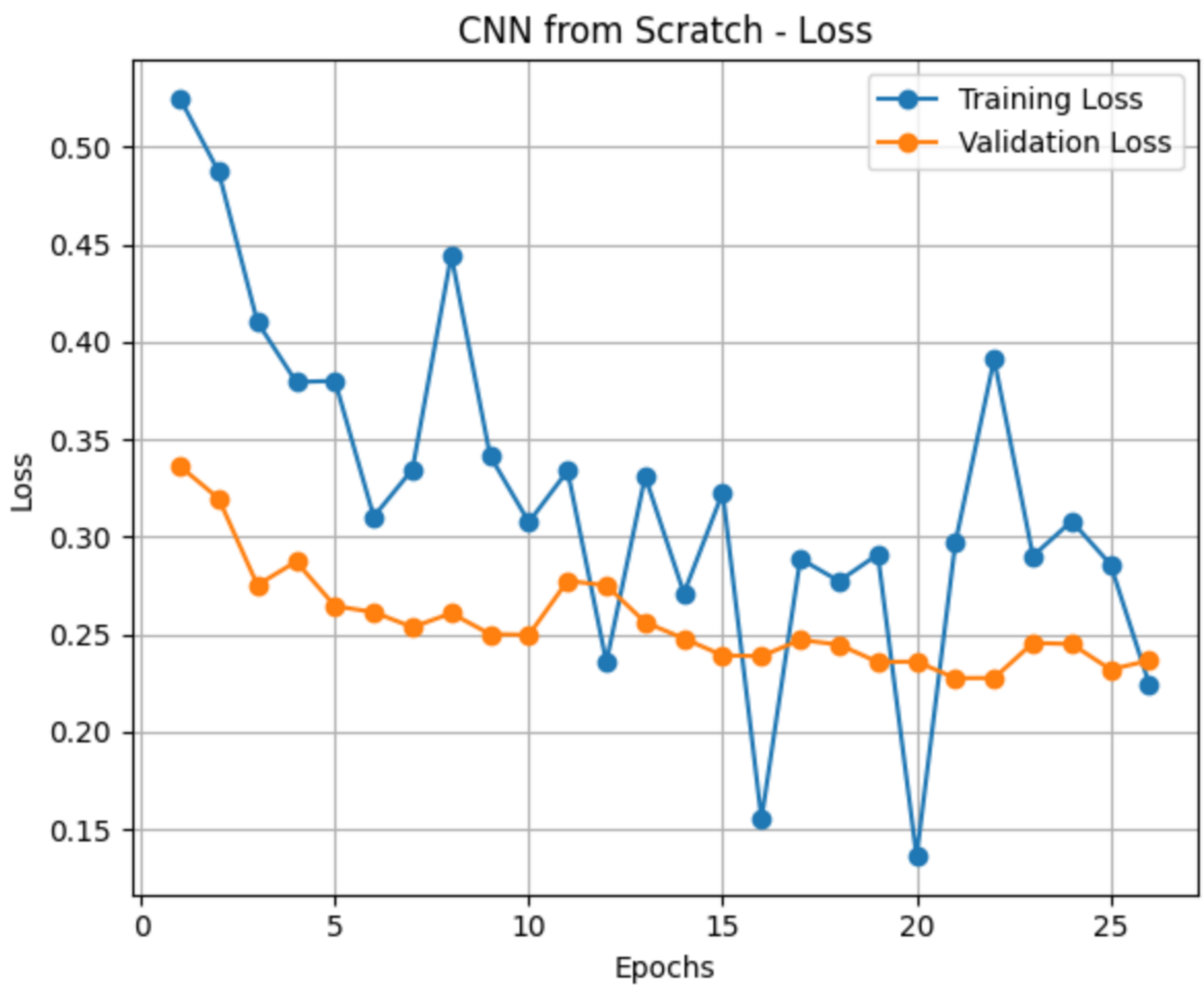
- Accuracy improved with larger datasets but **plateaued at 1500 samples** (71.2%), indicating limited model capacity.
- Overfitting observed in loss curves.



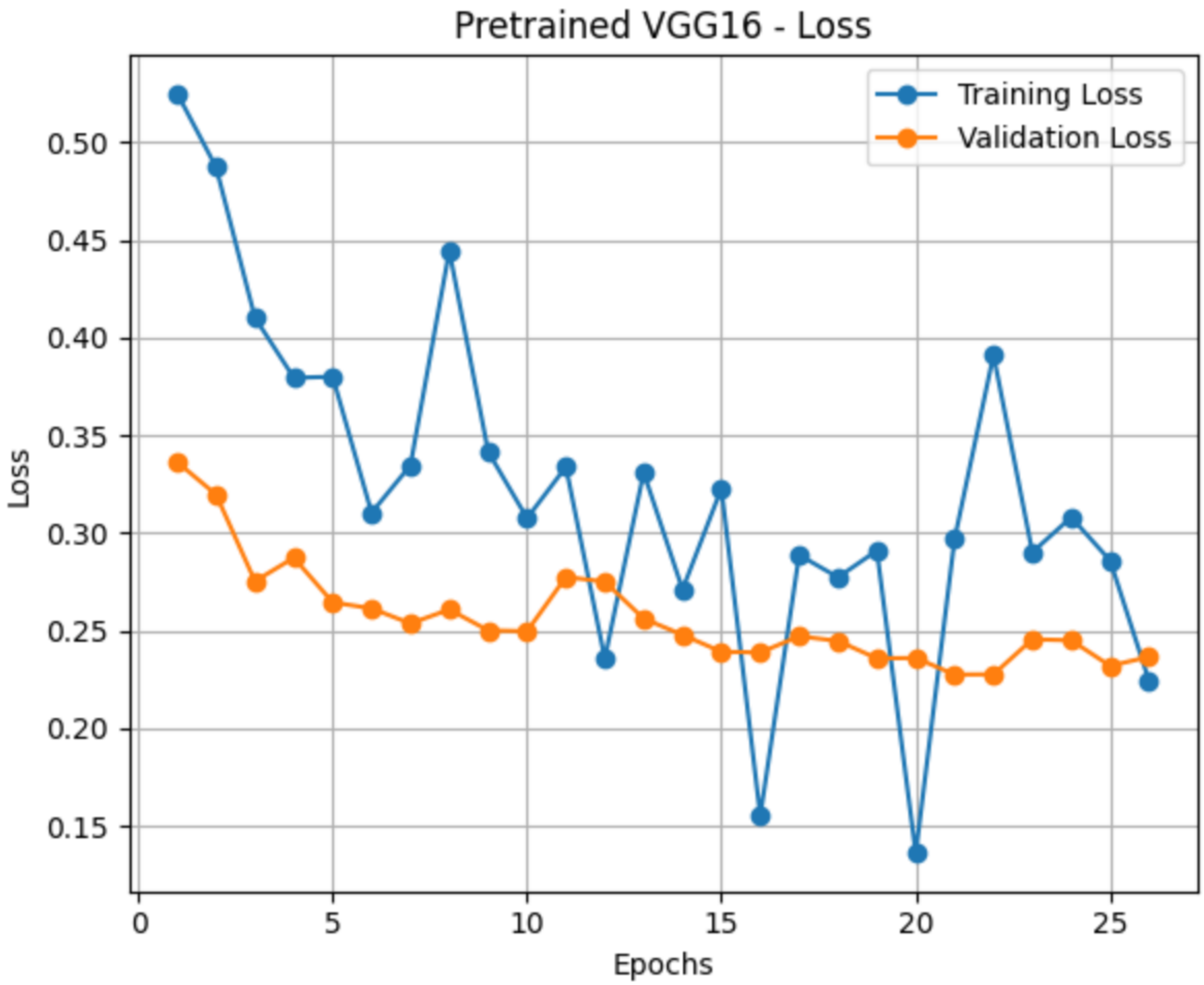
Pretrained VGG16 Model

- Achieved **>88% accuracy** across all sample sizes, demonstrating robust feature extraction.
- Minimal overfitting due to frozen convolutional layers.

5.2 Loss Curves



- The scratch model exhibited **early overfitting**, as evident in diverging training and validation loss.



- The pretrained model maintained **stable performance**, indicating better generalization

5.3 Model Performance Comparison

Why Pretrained Models Excel?

- **VGG16's pre-learned features (from ImageNet)** generalize well, reducing dependency on large datasets.
- **Fine-tuning the pretrained model (unfreezing layers)** could further improve performance.

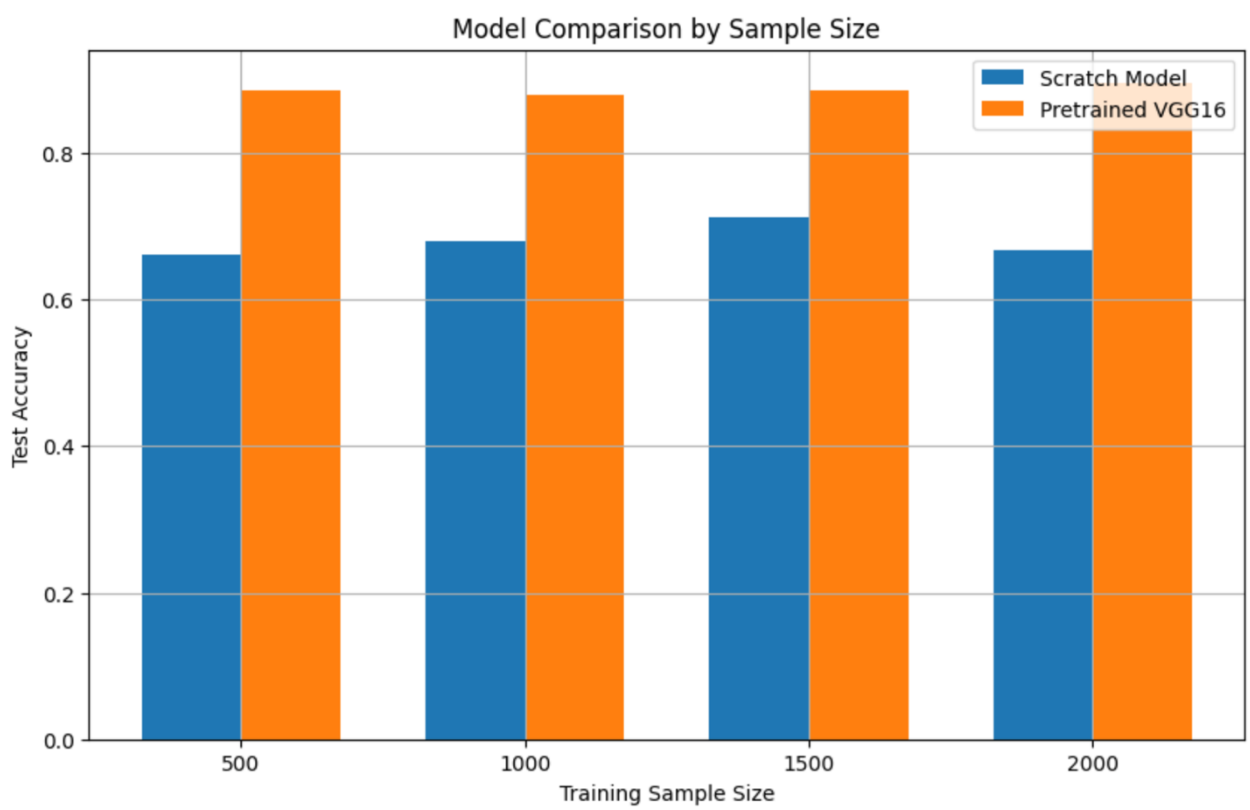
6. Discussion

Scratch Model Performance:

- Accuracy improved with more data but **plateaued at 1500 samples**.
- Indicates **limited model capacity** or lack of complexity to capture finer details.

Pretrained Model Performance:

- Achieved significantly **higher accuracy** even with fewer training samples.
- Performance continued improving up to **2000 samples**, but gains diminished beyond **1500 samples**.



Impact of Sample Size:

- Increasing dataset size initially improves performance but shows **diminishing returns beyond 1500 samples**.
- Suggests that **data quality and model architecture** are as important as dataset size.

7. Conclusion

This study demonstrates that **pretrained models significantly outperform CNNs trained from scratch**, particularly when working with smaller datasets.

- **Pretrained models offer superior generalization**, achieving consistently higher accuracy across all sample sizes.
- **Training from scratch requires substantially more data** to generalize well, yet still underperforms compared to transfer learning.
- **Increasing training data helps initially**, but beyond **1500 samples**, performance gains diminish, indicating that architectural improvements or fine-tuning may be more impactful than further increasing data volume.
- **Fine-tuning VGG16** by unfreezing some layers could enhance performance further, making transfer learning an even stronger approach.

Final Recommendation:

- **For Small Datasets (≤1,500 samples):** Pretrained models are **the optimal choice**, achieving **~20% higher accuracy** with minimal overfitting.
- **For Larger Datasets (≥1,500 samples):** Training from scratch becomes viable but **still underperforms compared to transfer learning** unless additional architectural enhancements are made.
- **Overall:** Pretrained networks should be preferred **unless there are domain-specific constraints** that necessitate training from scratch.

This study underscores the importance of selecting **the right model architecture based on dataset size** and highlights the power of **transfer learning** in image classification tasks. Future research could explore **fine-tuning pretrained models, leveraging data augmentation techniques, or testing alternative architectures** to push performance further.