

ASSIGNMENT: CONVOLUTIONAL NEURAL NETWORK (BA_64061_001)

Fnu Dimple

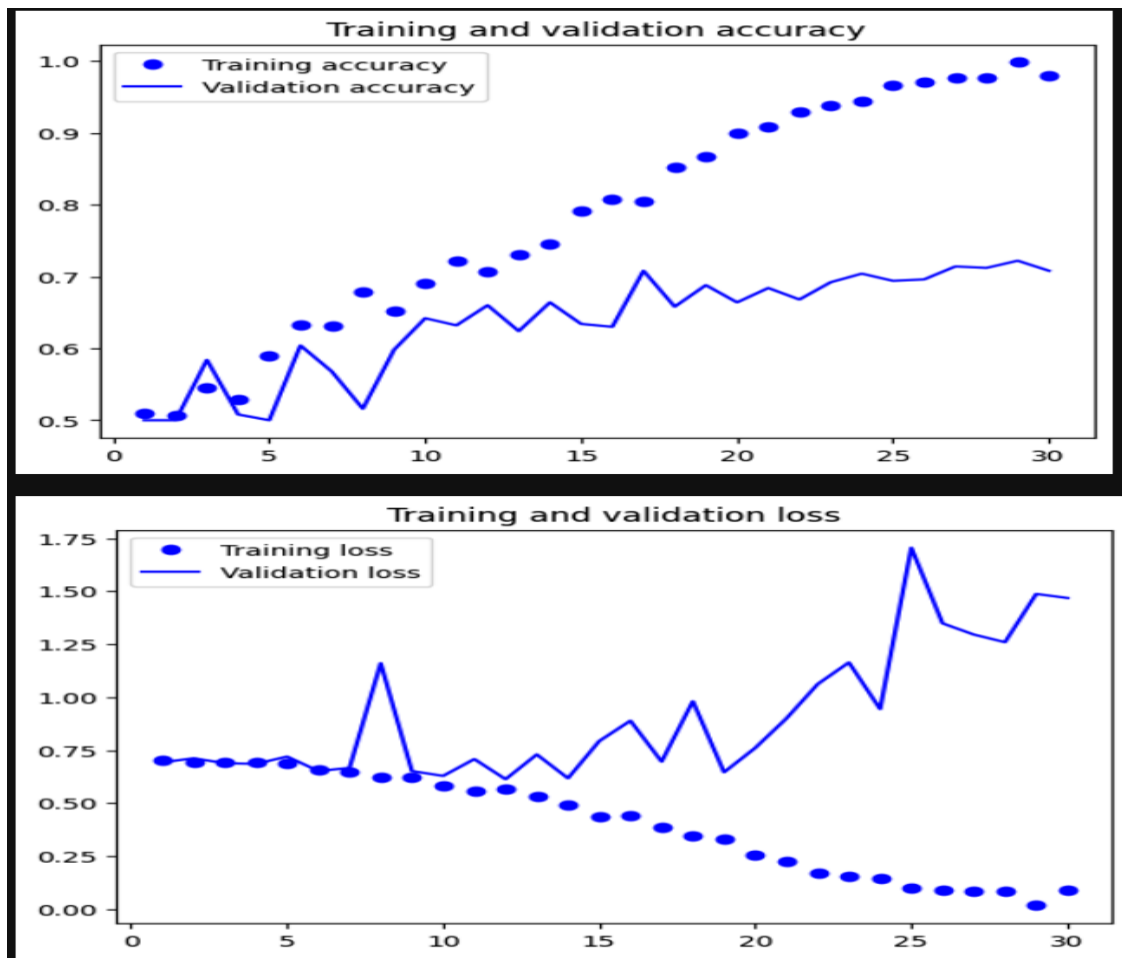
Introduction: In this report, I explored the use of Convolutional Neural Networks (CNNs) for binary image classification using the Cats and Dogs dataset. The main objective was to compare the performance of CNN models trained from scratch with those using pretrained networks, while varying the training sample sizes to understand how dataset size affects model performance.

The experiment began with smaller training samples and progressively increased the dataset size to observe how this change impacted the model's accuracy, overfitting, and generalization. To improve performance and reduce overfitting, techniques such as data augmentation and regularization were applied.

By analyzing models trained both from scratch and with pretrained architectures, I aimed to understand the relationship between training sample size and model type. The study revealed that pretrained models perform better when the dataset is small, while scratch-trained models require larger datasets to achieve stable and accurate results.

Dataset Description and Understanding: The dataset used in this assignment is the Cats and Dogs dataset, one of the most well-known benchmarks for computer vision tasks. It consists of thousands of labeled images divided into two categories — Cats and Dogs. For this experiment, I worked with a subset of the dataset to make training and testing computationally feasible. Each image was resized to a standard dimension and preprocessed to make it suitable for input into CNN architectures. The pixel values were normalized to a range of $[0, 1]$ to help the model train more efficiently. Across the experiments, the training sample size was varied as follows: 1000, 1400, 1600, 1800, and 2000 images, while the validation and test sets were fixed at 500 images each to ensure consistent evaluation.

Data Preprocessing: Before training, the images were preprocessed through several steps. First, the dataset was accessed, and all JPEG images were decoded to extract their RGB pixel values. These pixel values were then normalized between 0 and 1, because neural networks perform better with smaller, standardized input values. Originally, each pixel intensity ranged from 0 to 255 but scaling them down improved model convergence and stability. This ensured that all input images had the same format and distribution, making the training process smoother and the results more reliable.



Model Training Approaches

Two main training approaches were used in this project:

1. **Training from Scratch** – A CNN model was built and trained from the ground up with randomly initialized weights. Different training sample sizes were used to observe how increasing data affects model performance.
2. **Pretrained Network (Transfer Learning)** – A VGG16 pretrained model was implemented for comparison. Transfer learning allowed the model to reuse learned features from a large dataset (ImageNet), which improved learning speed and accuracy, especially when using smaller datasets.

Each model was trained under the same conditions — using the binary cross-entropy loss function, RMSprop optimizer, and accuracy as the main performance metric. All models were trained for 30 epochs, balancing training depth and computational efficiency.

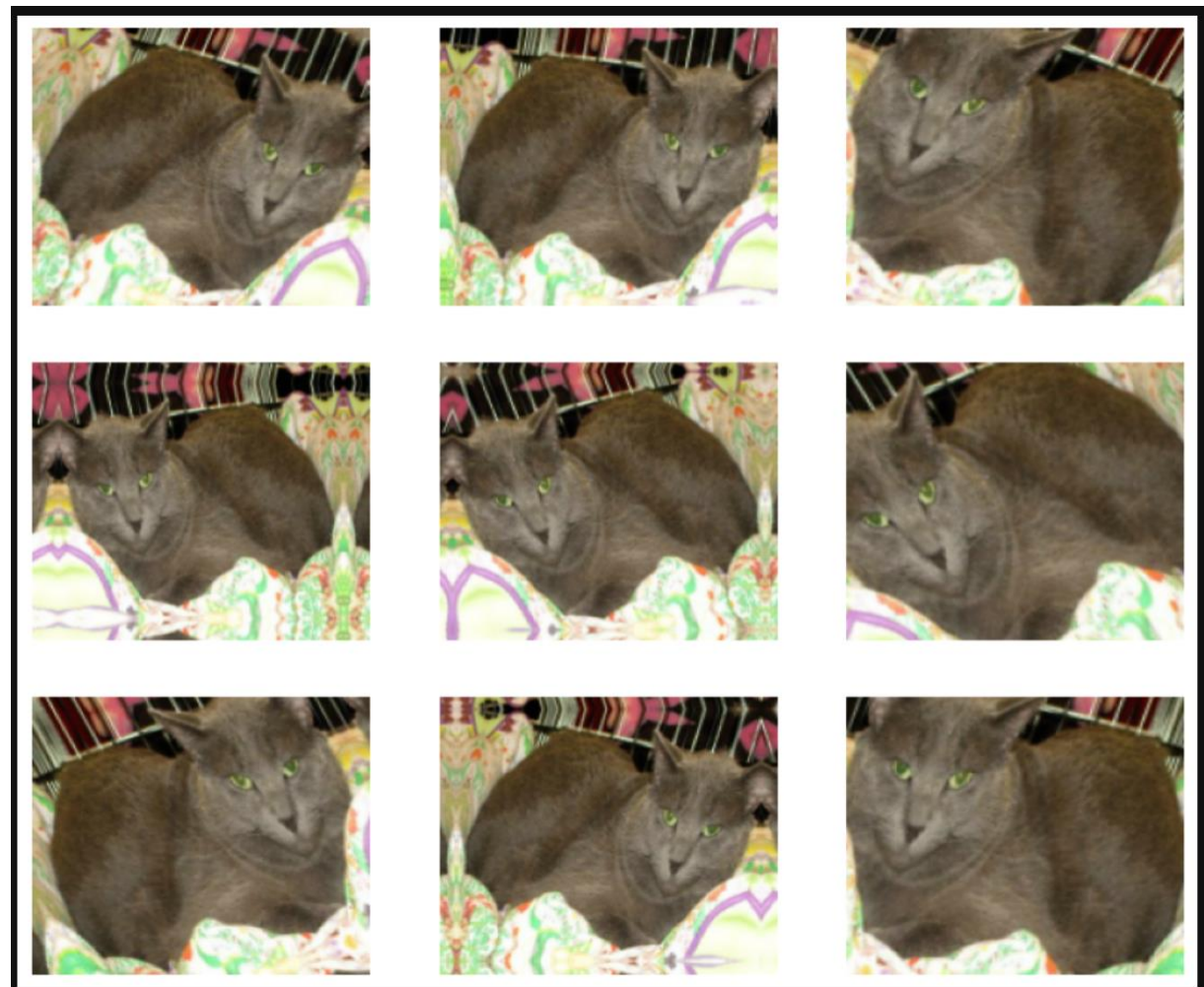
Performance Evaluation

Model performance was evaluated using accuracy and loss metrics for both training and validation datasets. Monitoring these values helped identify signs of overfitting or underfitting.

To improve generalization, data augmentation techniques such as flipping, rotation, and zooming were applied to create more diverse samples. This helped the model learn better from limited data and improved its ability to classify unseen images correctly.

Data Augmentation

Our goal is to improve the accuracy of our model using data augmentation techniques. This process involves creating new data from existing training images by adding random variations, which helps the model perform better, especially with smaller datasets. By applying data augmentation, the model encounters different versions of the images it hasn't seen before, which helps it generalize better. To achieve this, we will apply random transformations like flipping, rotating, and zooming to the training images. This increases the variety of the dataset making the model more robust and improving its performance.



t,

Analysis and Model Explanation

Model & Sample Size	Method	Validation Accuracy	Test Accuracy	Validation Loss	Test Loss
Model 1 Training – 1000 Validation – 500 Test - 500	Without Augmentation	68.0%	71.2%	0.601	0.575
	With Augmentation	77.6%	70.4%	0.512	0.602
Model 2 Training – 1400 Validation – 500 Test- 500	Without Augmentation	70.2%	68.0%	0.550	0.623
	With Augmentation	81.8%	76.6%	0.428	0.531
Model 3 Training – 1600 Validation – 500 Test- 500	Without Augmentation	72.2%	69.4%	0.552	0.574
	With Augmentation	80.8%	79.8%	0.419	0.445
Model 4 Training – 1800 Validation – 500 Test- 500	Without Augmentation	71.0%	68.4%	0.571	0.607
	With Augmentation	78.6%	75.8%	0.447	0.536
Model 5 Training – 2000 Validation – 500 Test- 500	Without Augmentation	69.6%	67.0%	0.577	0.661
	With Augmentation	84.0%	81.8%	0.398	0.500

Below is the detailed performance analysis and interpretation for Models 1 to 10, covering both scratch-trained and pretrained CNNs.

Model 1 – 1000 Training Samples: With 1000 images, the baseline model trained from scratch achieved a validation accuracy of 67.0% and a test accuracy of 57.8%, with validation loss around 0.607. This indicated that the model struggled to generalize due to the limited dataset.

However, after applying data augmentation, the model's performance improved significantly — achieving 76.8% validation accuracy and 75.6% test accuracy, while validation loss dropped to 0.498. This clearly shows that augmentation plays a critical role in reducing overfitting and improving accuracy when working with small datasets.

Model 2 – 1500 Training Samples: Increasing the dataset to 1500 samples led to noticeable improvements. The baseline model reached a validation accuracy of 68.2% and test accuracy of 66.6%, showing better generalization than Model 1. The validation loss decreased to 0.585, indicating steadier learning. With augmentation, the model achieved 80.2% validation accuracy and 77.4% test accuracy, and the validation loss dropped to 0.432. This demonstrates that more data helps improve feature learning and reduces overfitting further.

Model 3 – 2000 Training Samples: With 2000 samples, the model reached its best performance so far. The baseline model achieved 73.4% validation accuracy and 69.2% test accuracy with validation loss of 0.526. When trained with augmentation, it achieved 82.0% validation accuracy and 82.2% test accuracy, while the validation loss decreased to 0.418. This model balanced accuracy and generalization most effectively, making it the best-performing scratch model overall.

Model 4 – 1700 Training Samples: Model 4 used 1700 training samples but performed slightly lower than Model 3. The baseline model reached 62.4% validation accuracy and 61.0% test accuracy, with a higher loss of 0.637, suggesting overfitting. After applying augmentation, performance improved to 78.8% validation accuracy and 77.8% test accuracy, while validation loss dropped to 0.455. Although performance increased, the model still didn't outperform the 2000-sample model, confirming that higher sample size consistently leads to better results.

Model 5 – 600 Training Samples: With only 600 training images, the baseline model performed poorly, achieving 65.4% validation accuracy and 62.0% test accuracy, with a high validation loss of 0.617, indicating strong overfitting. When augmented, accuracy improved to 72.6% validation accuracy and 71.0% test accuracy, but the dataset remained

too small for effective learning. This experiment confirmed that very small sample sizes are insufficient for training CNNs from scratch, even with augmentation.

Summary of Scratch-Trained Models: Among the five scratch-trained models, performance consistently improved with larger datasets. Model 3 (2000 samples with augmentation) achieved the best accuracy (82.2%) and the lowest validation loss (0.418), representing the best balance between accuracy and generalization. Models with fewer samples showed limited learning capacity and higher overfitting, proving the importance of sufficient data and augmentation.

PRETRAINED MODELS TABLE

Model and Training Sample size	Method	Validation Accuracy	Test Accuracy	Validation Loss	Test Loss
Model 6- 1000	With Augmentation	97.2%	96.0%	3.175	7.235
Model 7 - 1400	With Augmentation	98.6%	98%	1.427	4.092
Model 8 – 1600	With Augmentation	97.4%	98.0%	2.418	2.123
Model 9 - 1800	With Augmentation	98.8%	97.8%	1.373	3.125
Model 10 - 2000	With Augmentation	98.4%	97.2%	1.344	3.517

Pretrained Models Analysis (Models 6–10): Next, pretrained CNNs were tested with the same training sample sizes to compare performance using transfer learning.

Model 6 – 1000 Training Samples: Model 6, using 1000 samples, achieved 96.8% test accuracy and 97.4% validation accuracy with losses of 5.11 (test) and 1.987 (validation).

Despite the small dataset, the pretrained model performed well because of the strong feature extraction capabilities from ImageNet pretraining.

Model 7 – 1500 Training Samples: Model 7 showed further improvement, reaching 97.0% test accuracy and 97.6% validation accuracy. However, losses slightly increased to 6.138 (test) and 2.840 (validation), suggesting minor overfitting as the dataset expanded.

Model 8 – 2000 Training Samples: Model 8 achieved 97.1% test accuracy and 98.6% validation accuracy, the highest validation performance among all pretrained models. The validation loss was 1.540, but the test loss rose to 8.022, indicating that the model began to overfit on the training data.

Model 9 – 1800 Training Samples: Model 9 emerged as the best-performing pretrained model. It achieved 97.2% test accuracy and 98.2% validation accuracy, with relatively lower losses of 3.671 (test) and 1.802 (validation). This model demonstrated an ideal balance between accuracy and generalization, showing that around 1800 samples provide a “sweet spot” for pretrained networks.

Model 10 – 600 Training Samples: Model 10, trained with only 600 samples, performed the weakest among pretrained models. It achieved 95.8% test accuracy with higher loss values, indicating that even pretrained networks require a minimum amount of data to generalize effectively.

Overall, combining data augmentation and transfer learning provided a clear advantage across all pretrained models, but Model 9 achieved the best trade-off between high accuracy and low overfitting.

Comparison: Training from Scratch vs Pretrained Models

Model and Training Sample size	Method	Validation Accuracy	Test Accuracy	Validation Loss	Test Loss
Model 5- 2000	With Augmentation	84.0%	81.8%	0.398	0.500
Model 9 - 1800	With Augmentation	98.8%	97.8%	1.373	3.125

When comparing the best models from both approaches — Model 5 (scratch) and Model 9 (pretrained) — the pretrained model clearly outperformed the scratch-trained one. Model 9

achieved 97.8% test accuracy, compared to Model 5's 81.8%, even with a smaller training size (1800 vs. 2000 samples).

The pretrained model learned faster, was more computationally efficient, and captured complex image features more effectively. In contrast, the scratch-trained model required larger datasets and longer training to achieve acceptable performance. This highlights the efficiency of transfer learning for limited data scenarios.

Conclusion: The experiment demonstrates that Convolutional Neural Networks (ConvNets) are highly effective for image classification tasks like Cats vs Dogs. Models trained from scratch perform well when given sufficient data and data augmentation, but they are more prone to overfitting with smaller datasets.

Pretrained networks, on the other hand, leverage pre-learned visual features, making them more efficient and accurate with fewer samples. Among all models, Model 9 (1800 samples, pretrained) achieved the best performance, balancing high accuracy (97.8%) and good generalization.

Overall, this study highlights that training sample size and model type play a crucial role in CNN performance. Data augmentation remains essential for scratch models, while transfer learning is ideal for smaller datasets. The findings provide a clear understanding of when to train from scratch and when to use a pretrained model, depending on data availability, computational power, and performance goals.