ASSIGNMENT 3: CONVOLUTIONAL NEURAL NETWORK (BA_64061_001)

Prachi Agrawal

Introduction

In this report we explored the use of convolutional neural networks (CNNs) for binary image classification, focusing on distinguishing between cat and dog images. The goal was to compare the performance of models trained from scratch versus models using pretrained models, while changing the size of the training data. We began with a small training sample and progressively increased its size, evaluating how these changes influenced the model's accuracy and robustness. The objective was to find the best method for achieving high performance across different sample sizes, comparing the effectiveness of scratch-trained models versus pretrained models.

Data Overview

The data used for this assignment is the Cats and Dogs dataset, a popular benchmark for image classification, containing thousands of labeled images divided into two categories: "Cats" and "Dogs." For our experiments, we used a subset of this dataset. Each image was preprocessed into a standardized format suitable for input into convolutional neural networks (CNNs). We continuously changed the training sample size to evaluate its effect on model performance, while keeping the validation and test sets at a fixed size to ensure consistent evaluation across different configurations.

Data Preprocessing

- Accessing the image files.
- Decoding the JPEG content to retrieve RGB pixel grids.
- Normalizing the pixel values to the [0, 1] range, as neural networks perform better with smaller input values. This is done by scaling the pixel values, which initially range from 0 to 255.

Model Training

- Trained a CNN from scratch and optimized it with varying sample sizes.
- Implemented a pretrained network for comparison.

Performance Evaluation

- Analyzed accuracy and loss metrics across different experiments to assess performance.
- Monitored validation accuracy to identify and evaluate overfitting trends.

Methodology & Performance Analysis

To evaluate the impact of varying training sample sizes on model performance, we used a consistent CNN architecture across five models, with the only difference being the training dataset size. The training sample sizes were systematically set to 1000, 1400, 1600, 1800, and 2000 images, while the validation and test sets remained fixed for consistency.

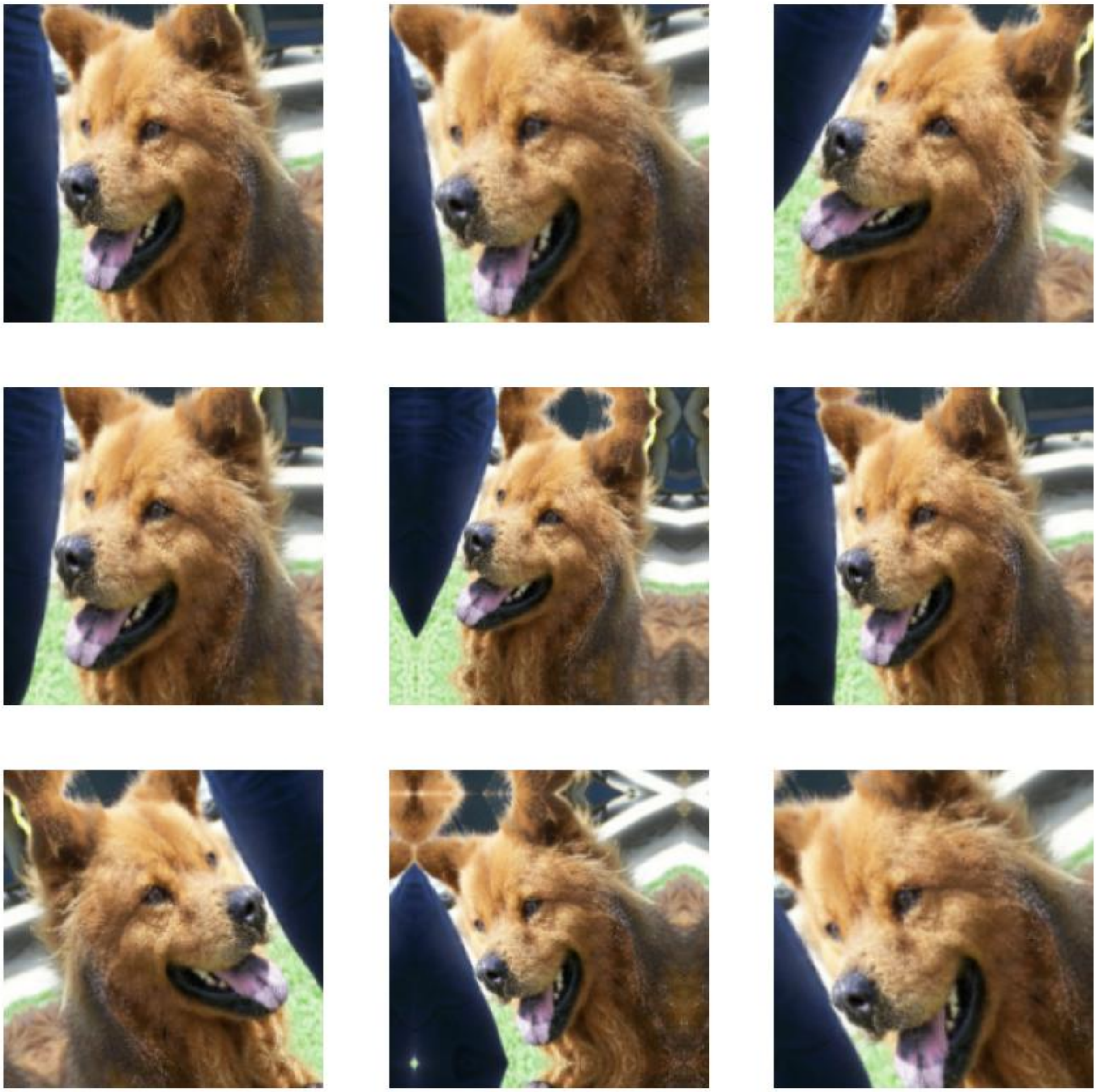
Each model consisted of five convolutional layers with ReLU activation functions, followed by a dense output layer with a sigmoid activation. The models were compiled using the binary cross-entropy loss function, the RMSprop optimizer, and accuracy as the evaluation metric. This standardized setup allowed for a clear comparison of accuracy and loss metrics, helping isolate the effects of training data variations on performance.

To enhance generalization and improve robustness, we applied data augmentation techniques, including random flipping, rotating, and zooming, to introduce variability into the dataset. Additionally, a pretrained model was used to leverage transfer learning, enabling the extraction of meaningful features without the need for extensive retraining. This approach improved performance while reducing computational complexity.

Each model was trained both with and without data augmentation, ensuring a comprehensive evaluation of the impact of these techniques on performance. All models were trained for 30 epochs, balancing optimization with computational feasibility. This framework allowed us to assess how changes in training data size, along with data augmentation and transfer learning, influenced accuracy and loss metrics, providing valuable insights into the model's behavior and generalization ability.

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.



ANALYSIS

Model and 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

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

TRAINING FROM SCRATCH & PRETRAINED MODEL

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

CONCLUSION

Model 5, trained with 2000 samples and using data augmentation, balances good accuracy with low loss, indicating effective generalization. Model 9, trained with 1800 samples, achieves higher accuracy but suffers from significant overfitting, as shown by its high loss values. While Model 9 performs better in terms of accuracy, Model 5 is more reliable for generalization and avoids overfitting.

Convolutional neural networks (ConvNets) are highly effective for computer vision tasks, achieving good results even when trained from scratch on small datasets. However, overfitting remains a key challenge with limited data. Data augmentation is a useful technique to combat overfitting by introducing variation into the dataset. Additionally, leveraging pretrained ConvNets through feature extraction enables efficient adaptation to new datasets, making it an excellent choice for smaller image datasets. Fine-tuning further boosts performance by adjusting the pretrained model's learned features for the specific task.

This study offers valuable insights into the trade-offs between training from scratch and using pretrained networks, helping guide the choice of approach based on dataset size and computational constraints.