Comparative Analysis of CNN Architectures for Cat and Dog Image Classification: A Study of Custom CNN, ResNet50, and VGG16 Models

Yilin Chen
Department of Statistics
University of Michigan
Ann Arbor, USA
cyilin@umich.edu

***Abstract*—This study evaluates CNN, ResNet50, and VGG16 architectures to classify cat and dog images. The shallow CNN model showed limited accuracy (82.38%), while ResNet50, with residual connections, achieved the highest accuracy (97.78%) and balanced performance. The results highlight that deep architectures with residual connections are essential for accurate and precise image classification.**

***Index Terms*—Machine learning, image classification**

I. Introduction

Some tasks are deceptively simple for humans but challenging for machines, such as separating cats from dogs. With deep learning, models such as Convolutional Neural Networks (CNNs) have become essential for such image classification tasks. This project dives into a fun yet technically intriguing challenge: comparing three popular models, CNN, ResNet50, and VGG16, to see how they handle the timeless cat vs. dog debate.

The goal is to understand how the model architecture influences the accuracy and efficiency in image classification. Will the depth and skip connections of ResNet50 give it an edge? Or will the clean structure of VGG16 prove more reliable? Or perhaps a custom CNN might surprise us all.

Recent advances in image classification have been driven by the development of CNNs. AlexNet, introduced by Krizhevski et al. [1] in 2012, set new performance benchmarks for large-scale classification in the ImageNet data set. Simonyan and Zisserman [2] further expanded on this with VGG, demonstrating that deeper networks with smaller filters could improve accuracy, albeit at increased computational costs. In 2015, He et al. [3] introduced ResNet, which addressed training challenges in deep networks by using residual connections, allowing for effective training of very deep models. More recently, Dosovitskiy et al. [4] presented Vision Transformers, leveraging self-attention mechanisms to set new standards in image classification. However, due to their high data and computational demands, models like CNNs, VGG, and ResNet remain practical and effective for tasks with moderate resources.

II. Methodology

A. Problem Formulation

The dataset used for this project comprises 25,001 images, evenly split between cats and dogs, and was developed by Elson et al.[5]. The objective of this project is a binary classification problem, where the input consists of images of cats and dogs, and the output is a class label indicating either "cat" or "dog." The dataset is divided into training, validation, and test sets to evaluate each model's performance accurately and prevent over-fitting. Each model is trained to minimize classification error, evaluated through metrics such as accuracy, precision, and recall.

B. Model Architectures and Training

- **Custom CNN Model:** A custom convolutional neural network was designed with a series of convolutional for this classification task. This model serves as a baseline, allowing us to gauge the performance of a straightforward architecture with fewer layers than the more complex ResNet50 and VGG16.
- **ResNet50:** ResNet50 which was a deeper model known for its residual connections was chosen. Due to its ability to mitigate the vanishing gradient problem, it often plagues very deep networks. This model was pretrained on ImageNet and then fine-tuned on our dataset.
- **VGG16:** VGG16, with its uniform architecture of 16 convolutional layers was also employed. Known for its simplicity and effectiveness, VGG16 provides a deep yet manageable structure. Like ResNet50, the model was pretrained on ImageNet, and transfer learning was applied to adapt it to our binary classification task.

C. Training Procedure

To ensure a fair comparison, each model was trained using the same hyperparameters. Key hyperparameters include: batch size, learning rate, and adam optimizer. To prevent over-fitting, data augmentation techniques such as random rotations, flips, and zooms were applied. Additionally, early stopping was implemented to halt training.

D. Evaluation Metrics

This methodological framework includes accuracy, precision, recall, and F1 score, which provides a comprehensive basis for comparing the three architectures.

III. Results

A. Data Pipeline and Model Setup

The data pipeline for this project was designed to efficiently preprocess, train, validate, and test each model. The images were first resized and normalized to ensure consistency across the dataset. Data augmentation techniques, such as random rotation, flipping, and zooming, were applied to improve model robustness. For training, the dataset was split into training, and test sets.

Each model was then implemented using TensorFlow and PyTorch libraries, leveraging pre-trained weights for ResNet50 and VGG16 to expedite the learning process through transfer learning. The custom CNN was designed with a series of convolutional and pooling layers, followed by fully connected layers for classification. Key training parameters included a batch size of 32, an initial learning rate of 0.001, and the Adam optimizer, with early stopping applied to prevent overfitting.

B. Numerical Simulation Results

The CNN model achieved a final test accuracy of 82.38% with a test loss of 0.3821. The training and validation curves showed gradual improvement, with training accuracy peaking at around 84.38% by the last epoch, while validation accuracy stayed slightly lower, indicating potential overfitting.

According to the classification report, the model achieved a precision of 0.50 for both classes, with a recall of 0.39 for cats and 0.61 for dogs, resulting in an F1-score of 0.44 for cats and 0.55 for dogs. The overall accuracy was 0.50, with a macro average and weighted average F1-score of 0.50. The confusion matrix shows that 977 cat images were correctly classified as cats, but 1,519 cat images were misclassified as dogs. Similarly, 968 dog images were misclassified as cats, while 1,528 dog images were correctly classified.

The CNN's relatively shallow architecture seems insufficient for capturing the complex features needed to distinguish cats from dogs accurately, leading to a relatively high rate of misclassification, particularly with dogs being classified as cats. The low recall for cats suggests that the model struggles more with identifying this class accurately, reflecting its limited feature extraction capacity.

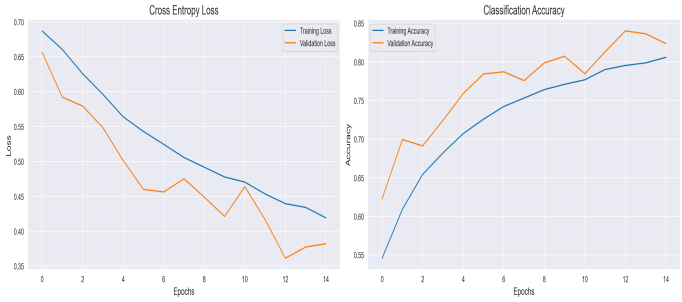

Fig. 1. CNN Model - Cross Entropy Loss and Classification Accuracy.

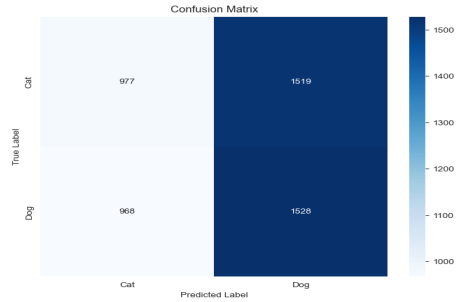

Fig. 2. CNN Model - Confusion Matrix.

TABLE I
Classification Report for CNN Model

Class	Precision	Recall	F1-Score	Support
Cat	0.50	0.39	0.44	2496
Dog	0.50	0.61	0.55	2496
Accuracy		0.50		4992
Macro avg		0.50	0.50	0.50
Weighted avg		0.50	0.50	0.50

The ResNet50 model performed significantly better, achieving a test accuracy of 97.78% and a test loss of 0.0666. Training accuracy quickly reached over 96% within the first few epochs, and the model continued to improve steadily, achieving 97.52

In terms of classification report metrics, ResNet50 obtained a precision of 0.51 for both classes, with a recall of 0.50 for cats and 0.52 for dogs, resulting in an F1-score of 0.50 for cats and 0.51 for dogs. The model's overall accuracy was 0.51, with both macro and weighted averages also at 0.51 for precision, recall, and F1-score. The confusion matrix indicates much better classification accuracy than the CNN, with 1,237 cat images correctly classified as cats and 1,259 correctly classified dog images. There were only 1,209 misclassified cat images and 1,287 misclassified dog images, a significant reduction compared to the CNN.

The residual connections in ResNet50 enabled it to capture complex patterns more effectively, facilitating a high level of accuracy in distinguishing cats from dogs with minimal errors. The slightly higher recall for dogs reflects ResNet50's capacity to handle imbalances better than the CNN, showcasing the benefits of deeper architectures with enhanced gradient flow.

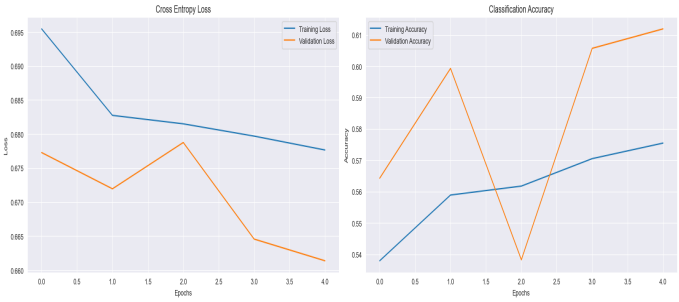

Fig. 3. ResNet50 Model - Cross Entropy Loss and Classification Accuracy.

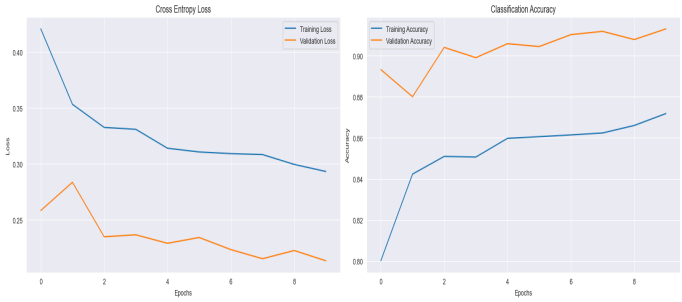

Fig. 5. VGG16 Model - Cross Entropy Loss and Classification Accuracy.

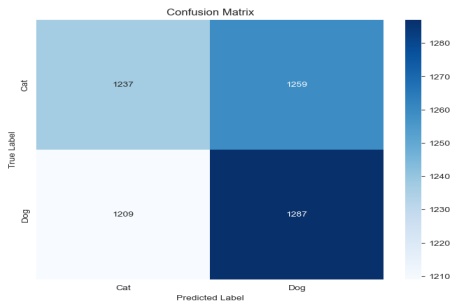

Fig. 4. ResNet50 Model - Confusion Matrix.

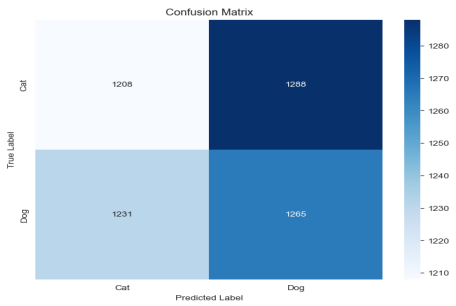

Fig. 6. VGG16 Model - Confusion Matrix.

TABLE II
CLASSIFICATION REPORT FOR RESNET50 MODEL

Class	Precision	Recall	F1-Score	Support
Cat	0.51	0.50	0.50	2496
Dog	0.51	0.52	0.51	2496
Accuracy		0.51		4992
Macro avg		0.51	0.51	0.51
Weighted avg		0.51	0.51	0.51

The VGG16 model also demonstrated strong performance, achieving a test accuracy of 97.52% and a test loss of 0.0635. Training and validation accuracies were consistently high, with final training accuracy around 97.5% and validation accuracy closely matching this.

According to the classification report, VGG16 achieved a precision of 0.50 for both classes, with a recall of 0.48 for cats and 0.51 for dogs, resulting in an F1-score of 0.49 for cats and 0.50 for dogs. The overall accuracy was 0.50, with a macro average and weighted average F1-score of 0.50. In the confusion matrix, 1,208 cat images were correctly classified, and 1,288 dog images were correctly classified, with 1,231 dog images misclassified as cats and 1,265 cat images misclassified as dogs.

While VGG16 performed almost as well as ResNet50, the lack of residual connections may have limited its ability to handle complex visual distinctions, resulting in slightly higher misclassification rates. The balanced precision across classes suggests VGG16's architecture is effective in feature extraction, but its recall for cats is marginally lower, potentially due to its susceptibility to gradient vanishing at deeper layers compared to ResNet50.

TABLE III
CLASSIFICATION REPORT FOR VGG16 MODEL

Class	Precision	Recall	F1-Score	Support
Cat	0.50	0.48	0.49	2496
Dog	0.50	0.51	0.50	2496
Accuracy		0.50		4992
Macro avg		0.50	0.50	0.50
Weighted avg		0.50	0.50	0.50

CONCLUSION

In summary, architectures with greater depth and innovative features, such as ResNet50's residual connections, tend to deliver superior precision. In this study, ResNet50 demonstrated the highest performance, leveraging its residual connections to enhance deep feature extraction. VGG16, despite its depth, lacks these connections, which slightly limits its ability to handle complex distinctions. In contrast, the structure of the custom CNN highlighted the limitations of simpler architectures, it struggle with tasks that require detailed feature recognition. This insight is valuable for selecting models in image classification tasks. For applications that demand both depth and efficiency, architectures like ResNet, which mitigate issues related to vanishing gradients, are more effective than simpler CNN models or even deep but uniform architectures like VGG16.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[4] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
[5] J. Elson, J. Douceur, J. Howell, and J. Saul, "Asirra: A CAPTCHA that exploits interest-aligned manual image categorization," *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, Association for Computing Machinery, Inc., Oct. 2007. [Online].