Project Assessment Report

Cat vs Dog Classification using CNN with Data Augmentation.

Course: Deep Learning and Reinforcement Learning

Ayush Kumar	1BG23CS020
Tanish Jain	1BG23CS167
Aditya Raj	1BG23CS008

Project Objectives and Scope

Model Development

Building a robust image classification model capable of distinguishing between cat and dog images using deep learning.

Comparative Analysis

Evaluating and comparing the performance of a custom CNN against a transfer learning approach with MobileNetV2.

Practical Deployment

Developing a functional web application for real-time image prediction, demonstrating the model's utility.

Our primary objective was to develop an accurate image classifier for cats and dogs, a fundamental task in computer vision. We aimed to not only build a functional model but also to conduct a comparative study between a baseline custom CNN and a more advanced transfer learning model. This allowed us to understand the practical advantages of leveraging pre-trained networks. Furthermore, the project extended to the development of a deployable web application, transforming our model from a theoretical concept into a tangible tool for image classification.

Dataset and Preprocessing Strategies

Dataset Overview

The project utilized the Kaggle Dogs vs. Cats dataset, a comprehensive collection of 25,000 labeled images. This balanced dataset provided ample examples for training and validating our models.

Data Split

The dataset was meticulously split into training and validation sets, with an 80/20 ratio. This division ensured that our models were evaluated on unseen data, providing an unbiased assessment of their generalization capabilities.

Preprocessing Steps

- **Resizing:** All images were uniformly resized to a consistent dimension, crucial for consistent input to the neural network.
- **Normalization:** Pixel values were normalized to a standard range (e.g., 0-1) to improve model convergence and performance.
- **Augmentation:** Techniques like random flips and rotations were applied to the training data. This artificially expanded our dataset, reducing overfitting and enhancing the model's robustness to variations in image orientation and composition.

Effective data handling is paramount in deep learning. Our approach involved a robust pipeline for data preparation, starting with the selection of a diverse and sufficiently large dataset. The strategic split into training and validation sets was critical for reliable model evaluation. Furthermore, the implementation of various preprocessing techniques, especially data augmentation, played a significant role in improving the model's ability to generalize from the training data to new, unseen images.

Custom CNN Model: The Baseline

Convolutional Layers

Multiple convolutional layers were used to extract hierarchical features, starting with basic edges and textures and progressing to more complex patterns.

Pooling Layers

Max-pooling layers were interspersed to reduce spatial dimensions, thereby decreasing computational cost and preventing overfitting by selecting the most salient features.

Fully Connected Layers

Flattened feature maps were fed into dense layers, enabling the model to learn complex relationships between extracted features and the final classification.

Activation Functions

ReLU activation was applied after each convolutional layer for non-linearity, while a sigmoid activation in the output layer provided a probability for cat or dog classification.

The custom CNN model served as our foundational baseline for this project. Its architecture was designed to progressively learn features from the input images. By employing a sequence of convolutional and pooling layers, the model could effectively capture intricate patterns relevant to distinguishing between cats and dogs. The subsequent fully connected layers then processed these learned features to make a final prediction. This baseline model provided valuable insights into the performance capabilities of a model built from scratch and highlighted the complexity involved in achieving high accuracy without leveraging pre-trained knowledge.

Transfer Learning with MobileNetV2

MobileNetV2 Base

Utilized MobileNetV2, pre-trained on ImageNet, as the foundational feature extractor. Its lightweight architecture is ideal for mobile and embedded vision applications.

Frozen Base Layers

The initial layers of MobileNetV2 were frozen, preserving the highly effective, pre-learned features for general object recognition.

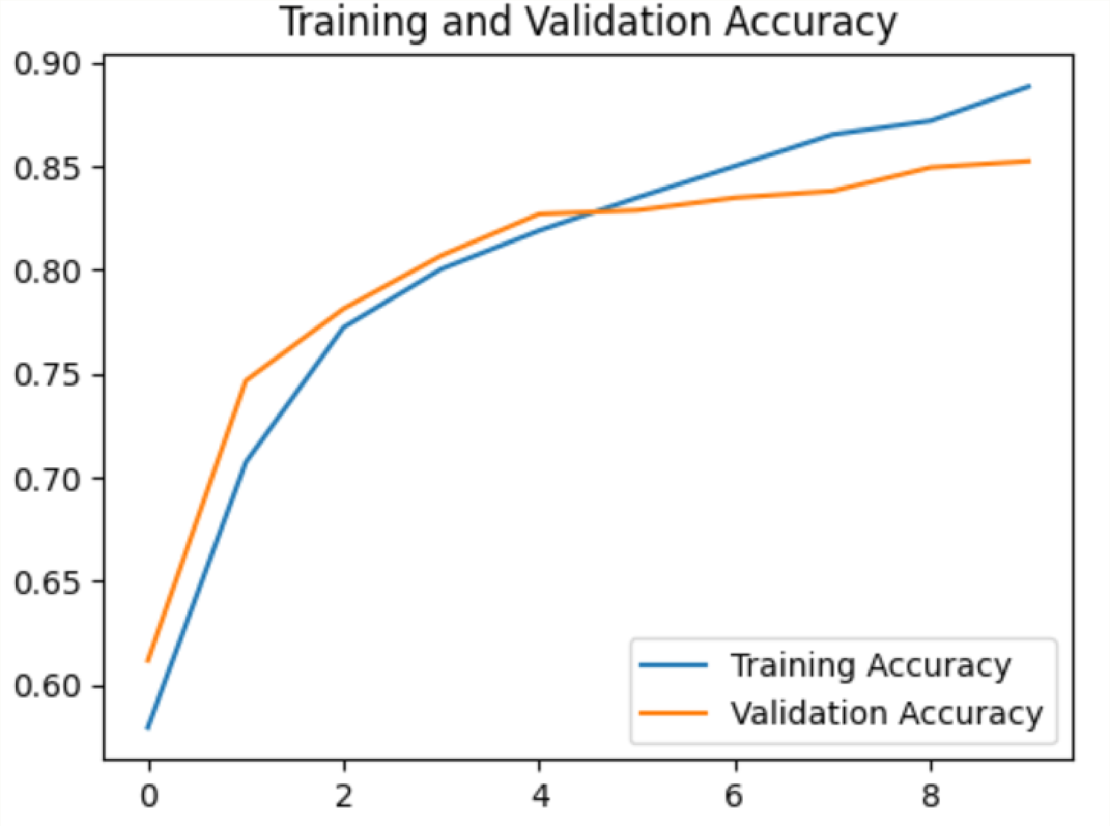
Custom Top Layers

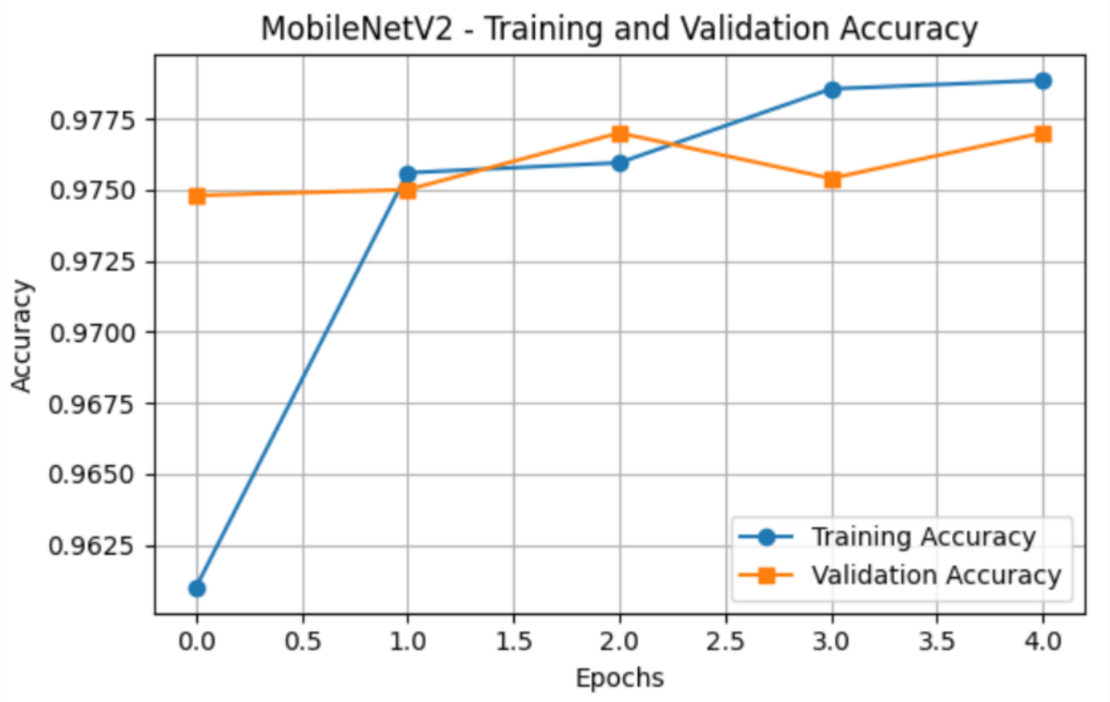
New layers, including a global average pooling layer and a dense output layer, were added on top of the frozen base. This adapted the pre-trained features to our specific cat vs. dog classification task.

Fine-Tuning (Limited)

A limited number of top layers from the pre-trained model were optionally unfrozen and fine-tuned alongside the new custom layers. This allows for slight adjustments of the pre-trained weights to better fit our dataset, balancing generalization and specificity.

Transfer learning emerged as a powerful technique to achieve superior performance with less training data and computational resources. By leveraging MobileNetV2, a model pre-trained on the vast ImageNet dataset, we could capitalize on its extensive knowledge of visual features. Freezing the base layers ensured that these well-established features were preserved, while the addition of custom top layers allowed the model to specialize in distinguishing between cats and dogs. This hybrid approach significantly expedited the training process and led to remarkable accuracy, demonstrating the efficiency and effectiveness of transfer learning in practical deep learning applications.





Performance Metrics and Results

97.70%

Validation Accuracy

Achieved by the MobileNetV2 model during training with early stopping.

97.89%

Test Accuracy

Demonstrated by the MobileNetV2 model on an unseen test set.

5–10

Epochs Trained

Number of epochs required for optimal performance before early stopping.

The performance of our MobileNetV2 transfer learning model was outstanding, achieving high accuracy metrics on both validation and unseen test datasets. The model quickly converged to optimal performance within just 5 epochs, largely thanks to the early stopping mechanism which prevented overfitting and ensured robustness. These results underscore the effectiveness of transfer learning for image classification tasks, showcasing how leveraging pre-trained models can lead to highly accurate and efficient solutions.

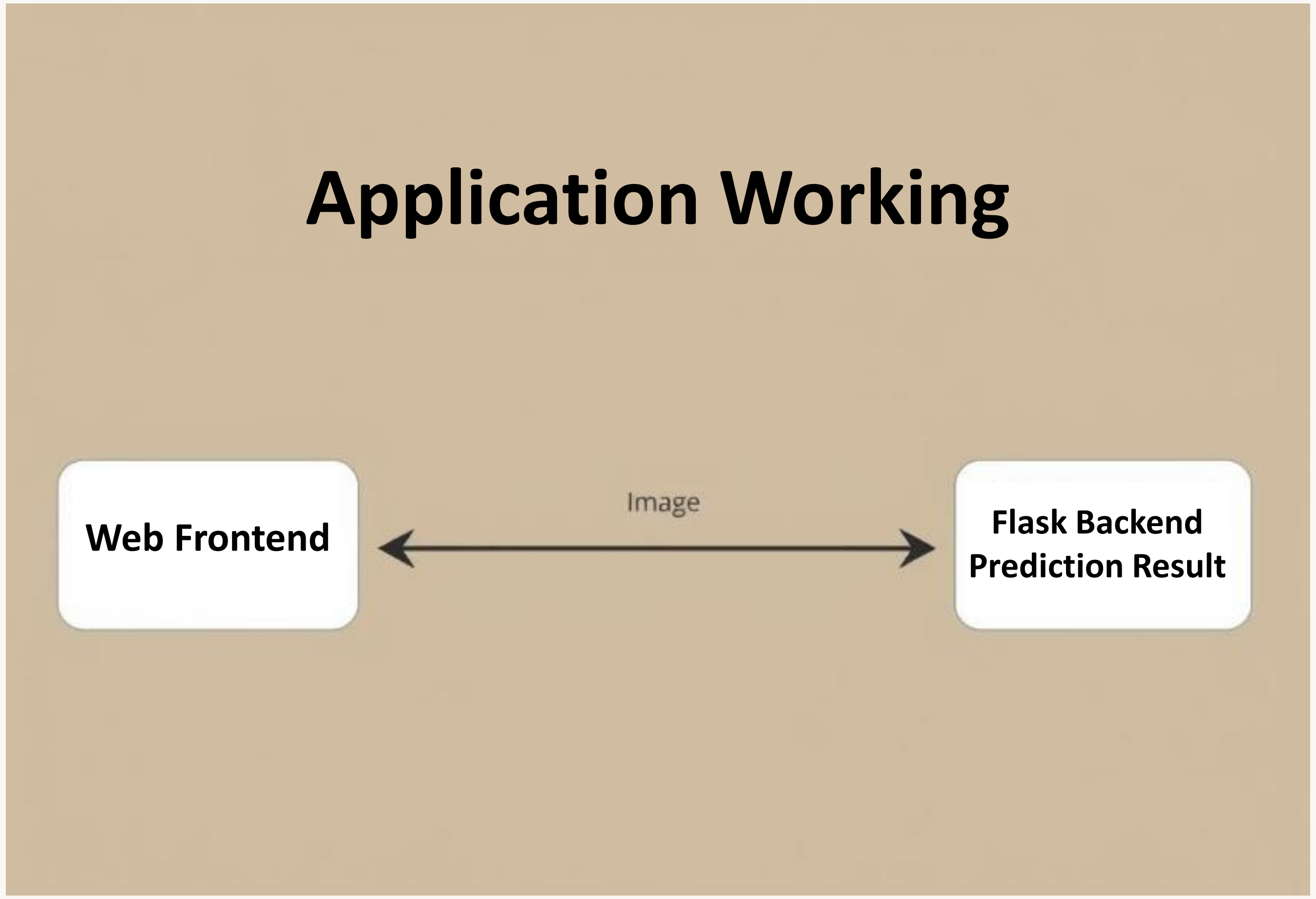
Model Deployment and Error Analysis

Deployment Architecture

A Flask-based backend was developed to host the trained model, providing an API endpoint for image uploads. A simple frontend, designed for user interaction, allowed for easy image submission and displayed prediction results with confidence scores.

Real-World Ambiguities

A few instances of misclassification were observed, primarily with ambiguous images where distinguishing features between cats and dogs were minimal or obscured. This included images with poor lighting, unusual angles, or hybrid-like appearances of animals. Such errors highlight the inherent challenges in real-world image classification and the limitations of even highly accurate models.



Deploying the model as a web application provided a tangible demonstration of its utility and allowed for real-time interaction. Users could upload images and instantly receive predictions, complete with confidence scores. While the model exhibited high accuracy, a crucial aspect of our analysis involved examining misclassifications. These errors, often stemming from ambiguous or challenging images, provided valuable insights into the model's edge cases and areas for potential future improvement. Understanding these limitations is vital for developing robust and reliable AI systems.

Future Work and Conclusion

Fine-Tuning Deeper Layers

Explore unfreezing and fine-tuning additional layers of the MobileNetV2 base to potentially capture more domain-specific features, further optimizing performance.

Advanced Model Architectures

Investigate the use of more sophisticated pre-trained models like EfficientNet or Vision Transformers for potential accuracy gains and robustness improvements.

Model Explainability (Grad-CAM)

Integrate explainability tools such as Grad-CAM to visualize which parts of the image the model focuses on, providing insights into its decision-making process.

This project successfully developed and deployed a highly accurate cat vs. dog image classifier, demonstrating proficiency in deep learning workflows from data preprocessing to model deployment. The transfer learning approach proved exceptionally effective, providing a strong foundation for future enhancements. Our next steps involve exploring more advanced fine-tuning strategies to push the accuracy boundaries further. Additionally, integrating model explainability tools will be crucial for understanding the 'why' behind the model's predictions, fostering greater trust and enabling more targeted improvements. This project not only delivered a functional product but also provided invaluable hands-on experience in the practical application of deep learning concepts.