Animals-10 Image Classification Report

By: Ginosca Alejandro and Natanael Santiago
Ironhack Data Science and Machine Learning Bootcamp
January 24, 2025

Introduction

The purpose of this project is to classify images from the Animals-10 dataset using Convolutional Neural Networks (CNNs). Two models were implemented: a custom CNN model designed from scratch and a transfer learning model leveraging the MobileNetV2 architecture pretrained on ImageNet. The project aims to evaluate the performance of both models and identify the best approach for accurate image classification.

Approach

1. CNN Architectures

Custom CNN Model:

- **Architecture**:
    - Convolutional and MaxPooling layers were stacked to extract features from the input images.
    - Fully connected dense layers were used for classification.
    - A final softmax layer outputs predictions for the 10 classes.
- **Key Design Choices**:
    - Simplicity in design for computational efficiency.
    - Moderate depth to avoid overfitting.

Transfer Learning Model:

- **Architecture**:
    - MobileNetV2 pretrained on ImageNet as the base model for feature extraction.
    - Custom dense layers added for classification, with Dropout layers for regularization.
    - Fine-tuning was conducted for improved adaptation to the dataset.

---

2. Preprocessing Steps

- **Data Augmentation**:
  - Horizontal flips, rotations, and brightness adjustments were applied to increase dataset diversity and prevent overfitting.
- **Normalization**:
  - Pixel values were scaled between 0 and 1 for faster convergence.
- **Resizing**:
  - All images resized to 128x128 pixels to match the input requirements of the MobileNetV2 model.
- **Splitting**:
  - Dataset divided into training and validation sets (80:20 ratio).

---

3. Training Process

Custom CNN Model:

- **Learning Rate**: 0.001 (with ReduceLROnPlateau for dynamic adjustment).
- **Batch Size**: 32.
- **Number of Epochs**: 30.
- **Callbacks**:
  - EarlyStopping to avoid overfitting.
  - ReduceLROnPlateau for learning rate adjustment.

Transfer Learning Model:

- **Base Model**: MobileNetV2 pretrained on ImageNet.
- **Learning Rate**: 0.001 (adjusted dynamically).
- **Batch Size**: 32.
- **Number of Epochs**: 20.
- **Callbacks**:
  - EarlyStopping for early termination.
  - ReduceLROnPlateau to fine-tune learning rate.
  - ModelCheckpoint to save the best model during training.

Results and Analysis

Performance Metrics

Metric	Custom Model	Transfer Learning Model
Validation Loss	1.0980	0.2086
Validation Accuracy	72.32%	93.71%
Validation Precision	78.12%	95.29%
Validation Recall	67.62%	92.78%

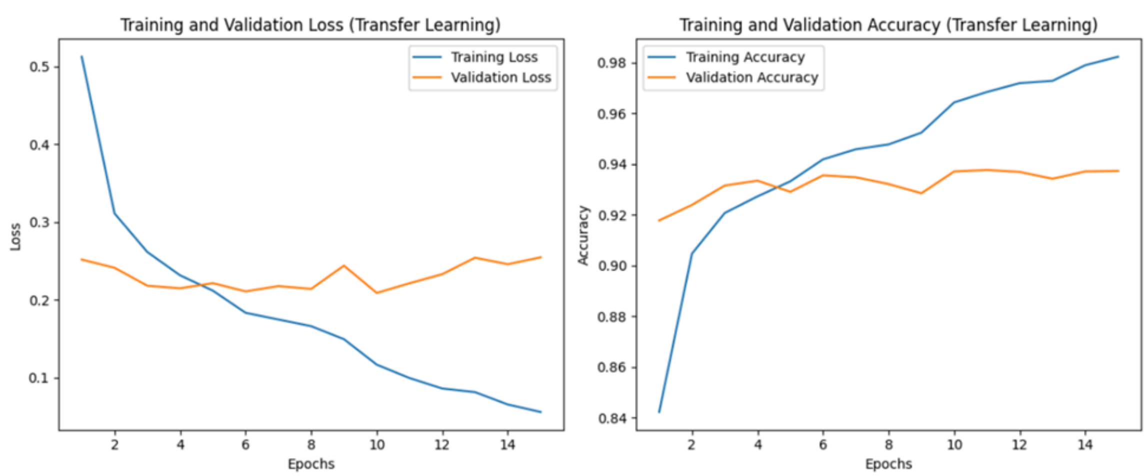
- **Custom Model**:
  - Achieved moderate performance with decent precision but struggled with recall, indicating difficulty in correctly identifying certain classes. It faced challenges in class-specific differentiation and generalization, particularly for complex or visually similar categories.
- **Transfer Learning Model**:
  - Significantly outperformed the custom model across all metrics. The model demonstrated strong feature extraction due to the pre-trained MobileNetV2, leading to better precision and recall. It showed robust generalization and higher consistency in correctly identifying classes, even for challenging categories.

Visualizations

Training Dynamics

1. **Training vs. Validation Loss and Accuracy**:
    - Loss and accuracy for both models.
        - Custom Model

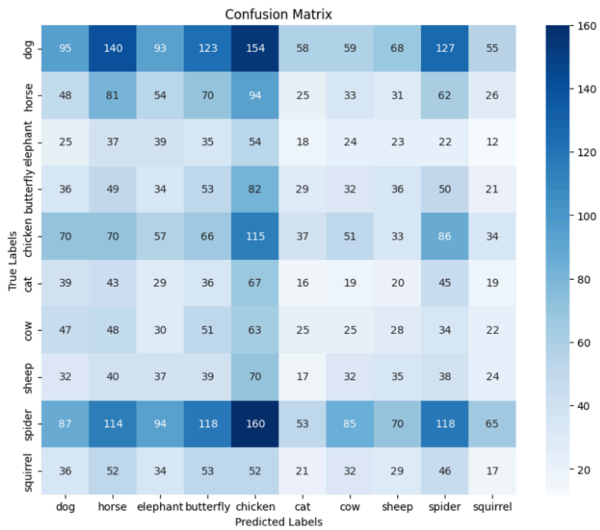
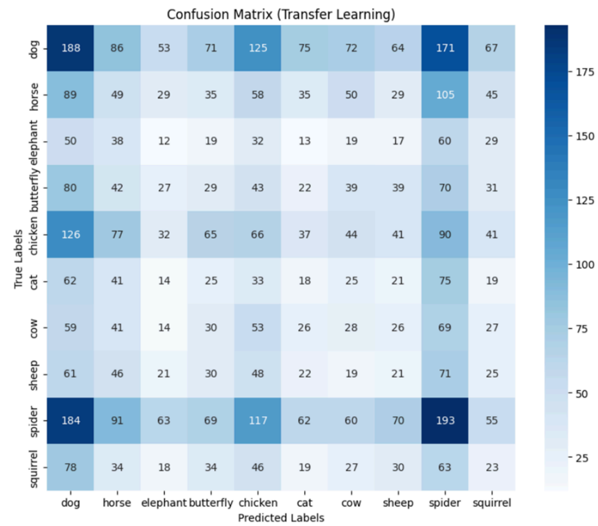
        - Transfer Learning Model



    - Insights:
        - The transfer learning model exhibited faster and more stable convergence compared to the custom model.
        - Validation metrics for transfer learning stabilized earlier, reflecting better generalization and reduced overfitting.
        - The custom model struggled with fluctuations in validation metrics, indicating less robust feature learning.

Confusion Matrices

2. **Confusion Matrix for Transfer Learning**:
   - Confusion matrix from the evaluation results.

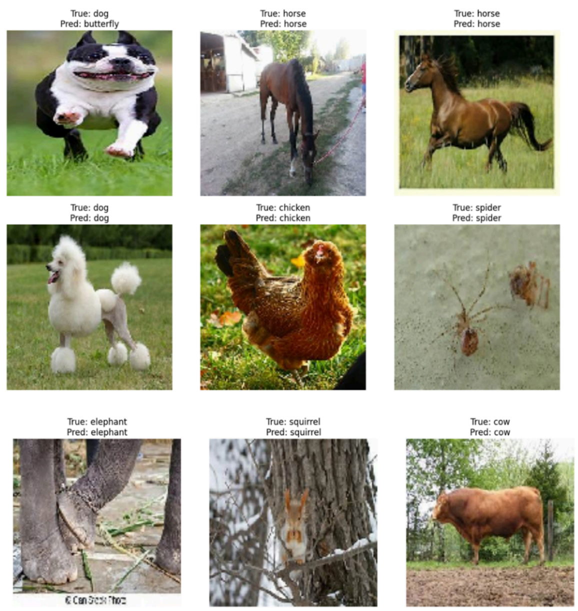
Custom Model	Transfer Learning Model
	

- Insights:
  - Custom Model:
    - Significant misclassifications across multiple categories.
    - Struggles to differentiate visually similar classes (e.g., "dog" frequently misclassified as "chicken").
    - Limited focus on specific class predictions, indicating weaker feature extraction.
  - Transfer Learning Model:
    - Improved accuracy for most classes (e.g., "dog" predictions are more accurate).
    - Fewer severe misclassifications compared to the custom model.
    - Better generalization across categories, reflecting stronger feature learning from MobileNetV2.
- Comparison:
  - The transfer learning model outperforms the custom model in minimizing misclassifications and shows a more balanced prediction distribution.

- While both models face challenges with certain classes, the transfer learning model demonstrates stronger class differentiation and generalization.

Sample Predictions

3. **Sample Predictions**:
    - Images showcasing correct and incorrect predictions for both models.

Custom Model	Transfer Learning Model
	

- Insights.
    - The transfer learning model demonstrates flawless accuracy and consistency in the sample predictions, further highlighting its superior generalization compared to the custom model.

	Custom Model	**Transfer Learning Model**
Accuracy in Predictions	Displays one misclassification, predicting a "dog" as a "butterfly." Other predictions were accurate.	All predictions were accurate, showing a strong grasp of the dataset.
Consistency	Generally consistent but has occasional misclassifications, as seen in the "dog" misclassification.	Perfect consistency in the provided sample predictions.
Generalization	Handles most cases well but demonstrates occasional issues with feature differentiation.	Robust generalization with no errors in the sample predictions.

Best Model

The **Transfer Learning Model** using MobileNetV2 is the best-performing model based on:

- **Highest validation accuracy (93.71%)**.
- **Better precision and recall**, indicating robust feature extraction.
- **Consistent training dynamics**, with reduced overfitting and smooth convergence.

Despite its strengths, further improvements can enhance the model:

- **Fine-tuning** the MobileNetV2 base layers.
- **Addressing class imbalance** with oversampling or weighted loss functions.
- **Enhanced data augmentation** targeting challenging categories.

Insights and Learnings

1. **Impact of Transfer Learning**:
    - Leveraging pretrained models significantly improved performance compared to building a model from scratch.
2. **Importance of Preprocessing**:
    - Data augmentation and normalization played a crucial role in achieving generalization.
3. **Challenges with Specific Classes**:
    - Categories like "dog" and "butterfly" require more refined feature extraction due to visual similarities.
4. **Role of Callbacks**:
    - EarlyStopping and ReduceLROnPlateau contributed to efficient training and prevented overfitting.

Next Steps

1. **Fine-Tuning**:
    - Unfreeze additional layers in MobileNetV2 for task-specific learning.
2. **Dataset Enhancements**:
    - Collect more diverse examples for underperforming classes.
3. **Exploring Alternate Architectures**:
    - Experiment with other lightweight architectures like EfficientNet or ResNet.

4. **Deploying the Model**:
    - Integrate the model into an application for real-time image classification.