Report on Fine-Grained Pet Classification Challenge

Introduction

In this report, I will illustrate the methodology used in the fine-grained pet classification challenge using PyTorch. The data comes from The Oxford-IIIT Pet Dataset, which is a Kaggle dataset. The project is structured into three main tasks: binary classification (distinguishing between dogs and cats), breed classification (identifying specific breeds among 37 classes), and transfer learning using a pre-trained model. Each section will explore the rationale behind the choices made, the challenges encountered, and potential improvements to the models.

Binary Classification

Methodology

The binary classification task aims to differentiate between two classes: dogs and cats. The dataset is constructed using images from the Oxford Pets dataset, where images are labeled based on the filename's capitalization—capitalized names indicate cats, while lowercase names indicate dogs.

The `PetDataset` class is defined to handle the loading of images and their associated labels. The `__getitem__` method opens each image, applies any specified transformations and returns the image along with its label. This approach ensures that the dataset can be easily manipulated and extended, which is a way to improve model performance.

Data augmentation techniques are employed to enhance the model's robustness. The training images are resized to a standard size of 224x224 pixels, and various transformations such as random horizontal flips, rotations, and color jittering are applied. These augmentations help the model generalize better by exposing it to a wider variety of image conditions.

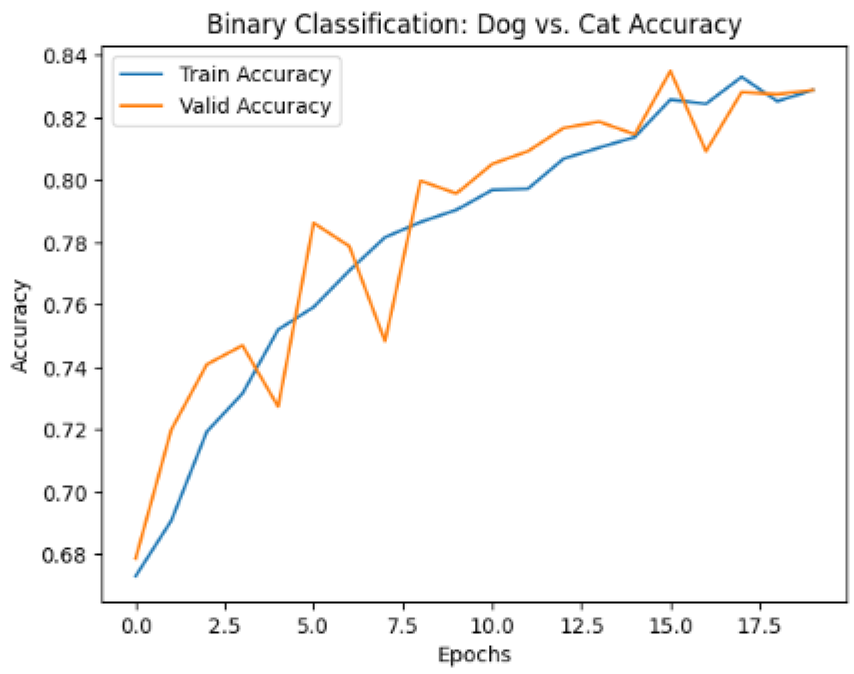
The model architecture is a simple convolutional neural network (CNN) consisting of several convolutional layers followed by batch normalization and dropout layers. The use of batch normalization helps stabilize the learning process, while dropout layers mitigate overfitting by randomly dropping units during training. The final layer employs a sigmoid activation function to output probabilities for binary classification.

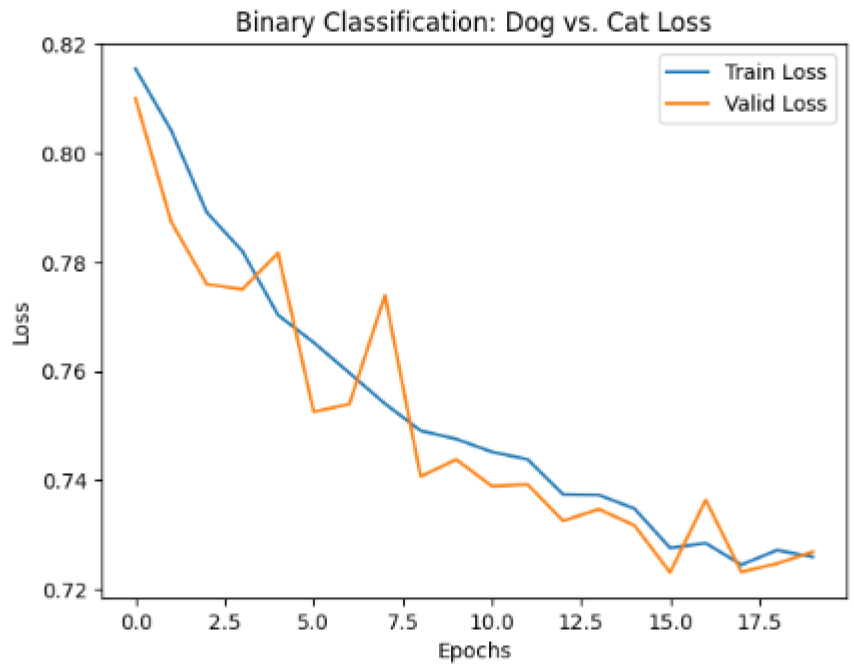
Results

The training results for the binary classification task over 20 epochs are as follows:

- **Final Train Loss:** 0.7260
- **Final Train Accuracy:** 0.8288
- **Final Valid Loss:** 0.7269
- **Final Valid Accuracy:** 0.8288
- **Training Time:** 1263.10 seconds

The model showed a steady improvement in both training and validation accuracy throughout the epochs, indicating effective learning and generalization.





The two plots above illustrate the training and the validation accuracy and loss over the 20 epochs.

The accuracy plot indicates that the model is learning effectively, as both training and validation accuracies improve over time. The validation accuracy closely follows the training accuracy, suggesting that the model is generalizing well without significant overfitting.

The loss plot demonstrates that both training and validation losses decrease over the epochs, which is a positive sign of effective learning. The gap between the training and validation loss is relatively small, indicating that the model is not overfitting and is performing consistently across both datasets.

Challenges and Improvements

One of the primary challenges faced during this task was the class imbalance, as the dataset may contain more images of one class than the other. To address this, class weights are calculated using the `compute_class_weight` function from `sklearn`, which adjusts the loss function to give more importance to the minority class. This adjustment improves the ability of the model to correctly classify the under-represented class.

In terms of improvements, experimenting with more complex architectures, such as deeper CNNs or incorporating residual connections, could enhance performance. Additionally, implementing techniques like early stopping based on validation loss could prevent overfitting and improve generalization.

Breed Classification

Methodology

The breed classification task extends the binary classification to a multi-class problem, where the model must identify one of 37 different breeds. The `PetDataset` class is reused, but the label extraction logic is modified to accommodate multiple breeds. The breed names are derived from the filenames, which are split and mapped to unique integer labels.

Similar to the binary classification task, data augmentation is applied to the training images to improve the model's robustness. The same transformations are utilized, ensuring consistency in preprocessing across tasks.

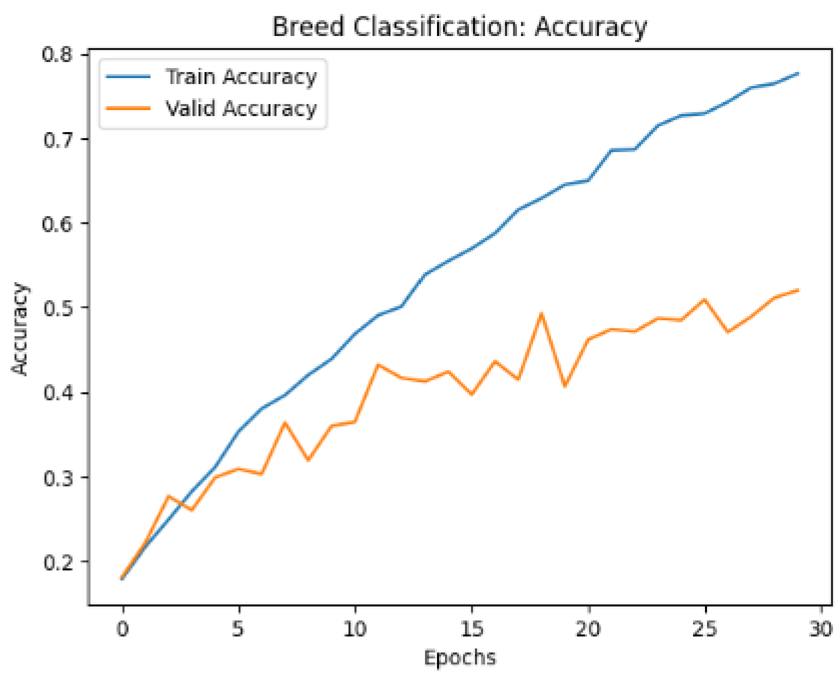
The architecture of the CNN remains largely unchanged, but the final fully connected layer is adjusted to output 37 classes instead of just one. The loss function is switched to `CrossEntropyLoss`, which is suitable for multi-class classification tasks.

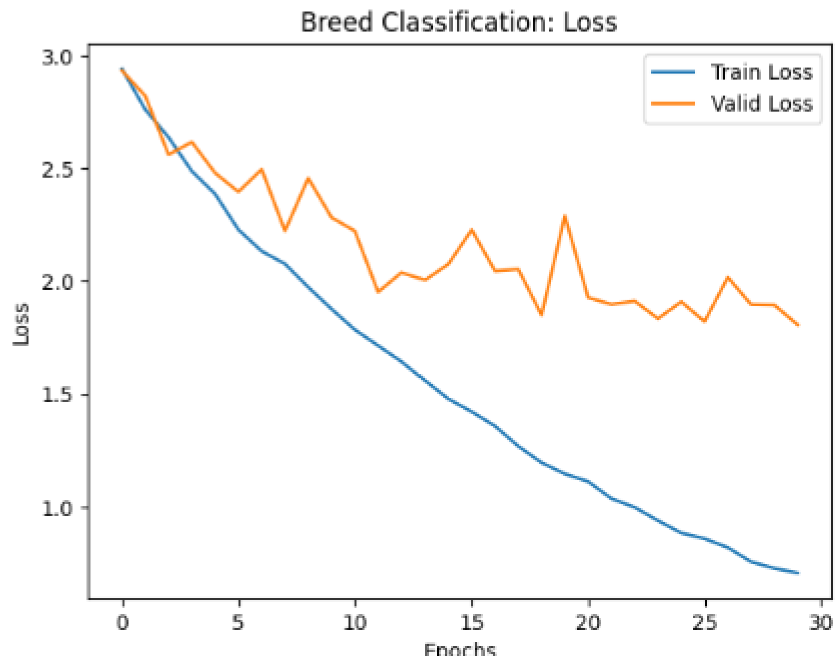
Results

The training results for the breed classification task over 30 epochs are as follows:

- **Final Train Loss:** 0.7079
- **Final Train Accuracy:** 0.7765
- **Final Valid Loss:** 1.8077
- **Final Valid Accuracy:** 0.5203
- **Training Time:** 1833.73 seconds

The model showed gradual improvement in both training and validation accuracy throughout the epochs, indicating effective learning, although the validation accuracy remains lower than the training accuracy.

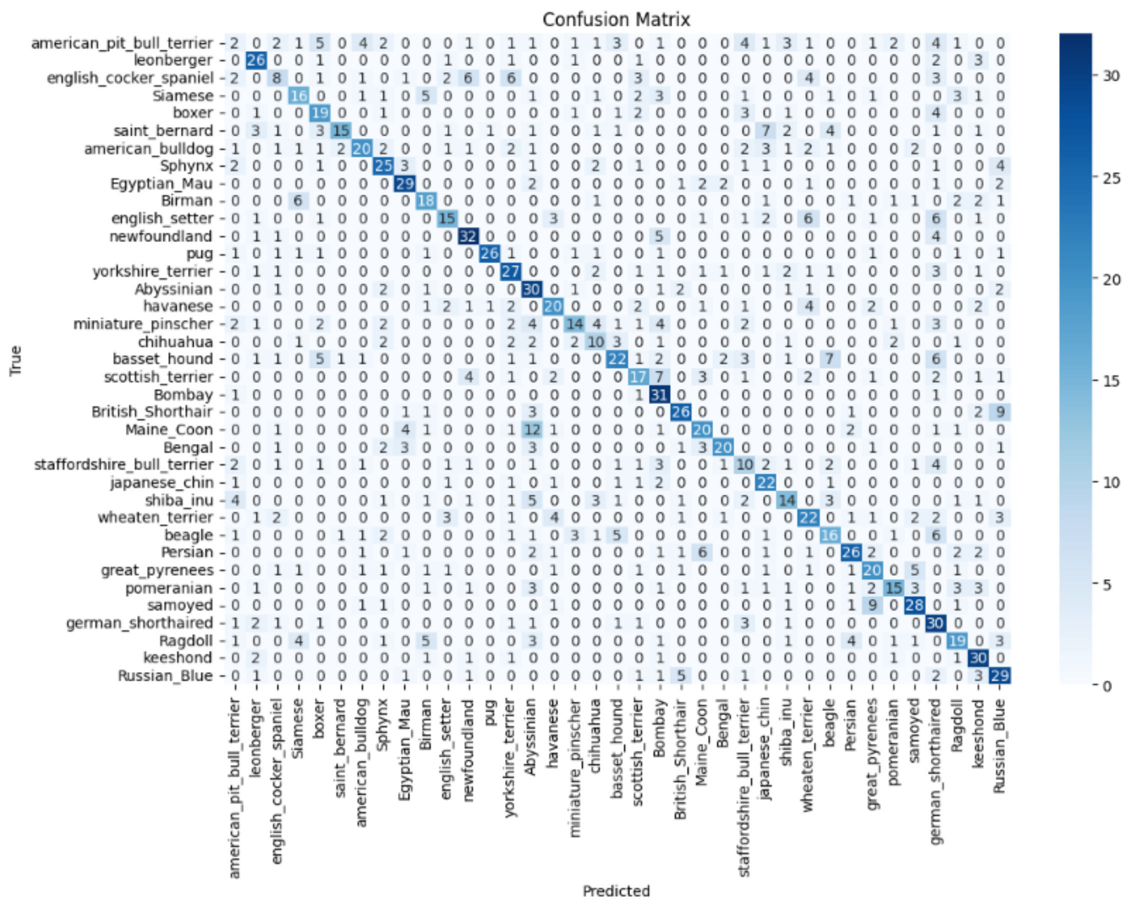




The two plots above illustrate the training and validation accuracy and loss over the 30 epochs.

The accuracy plot indicates that while the model is improving in training accuracy, the validation accuracy does not follow the same trend, suggesting potential overfitting or challenges in generalization to unseen data.

The loss plot demonstrates that both training and validation losses decrease over the epochs, indicating effective learning. However, the validation loss remains higher than the training loss, which may suggest that the model is not generalizing well to the validation dataset.



The confusion matrix provides a detailed breakdown of the model's performance in classifying the different breeds of pets. Each row of the matrix represents the true class (actual breed), while each column represents the predicted class (the breed predicted by the model). The values in the matrix indicate the number of instances classified into each category. The diagonal elements (from the top left to the bottom right) represent the number of correct predictions for each breed. The off-diagonal elements indicate misclassifications.

By examining the off-diagonal values, you can identify specific breeds that are often confused with one another. For example, if "Maine Coon" is frequently misclassified as "Abyssinian," it indicates that these breeds may share similar features, making them harder to distinguish.

Challenges and Improvements

A significant challenge in this task is the increased complexity of the classification problem. With 37 classes, the model must learn to distinguish between subtle differences in breeds, which can be visually similar. This necessitates a more sophisticated feature extraction process.

To improve the model's performance, one could consider using more advanced architectures, such as VGG16 or ResNet, which have been shown to perform well on image classification tasks.

The model certainly needed more than 30 epochs: my choice would be 50 epochs.

Transfer Learning

Methodology

The transfer learning task leverages a pre-trained ResNet-18 model, which has been trained on a large dataset (ImageNet). This approach allows the model to benefit from previously learned features, significantly reducing the training time and improving performance on the breed classification task.

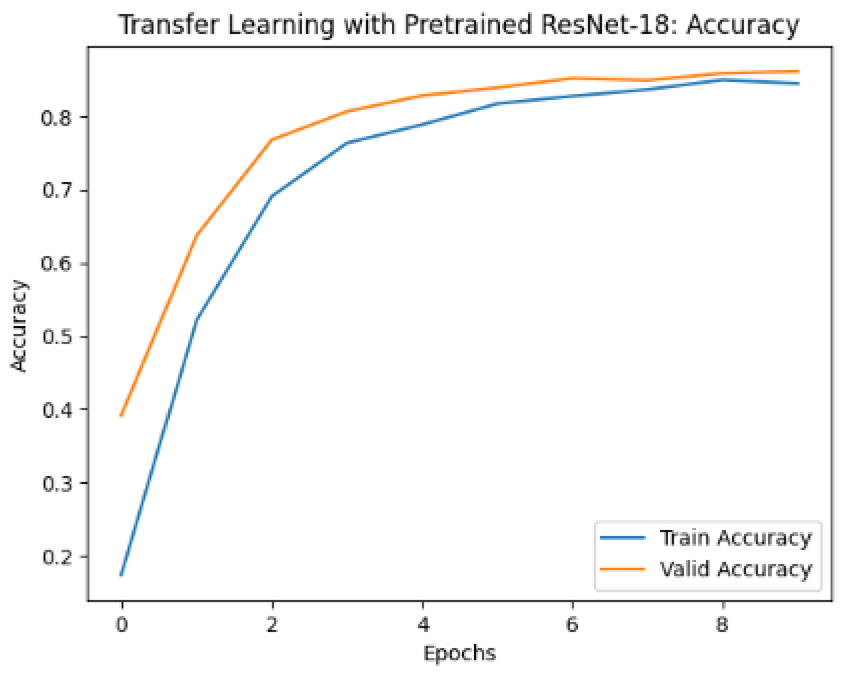
The initial layers of the ResNet model are frozen to prevent their weights from being updated during training, focusing the learning process on the final fully connected layer, which is modified to match the number of classes in our dataset. The `CrossEntropyLoss` is again used as the loss function, and the Adam optimizer is employed for fine-tuning the model.

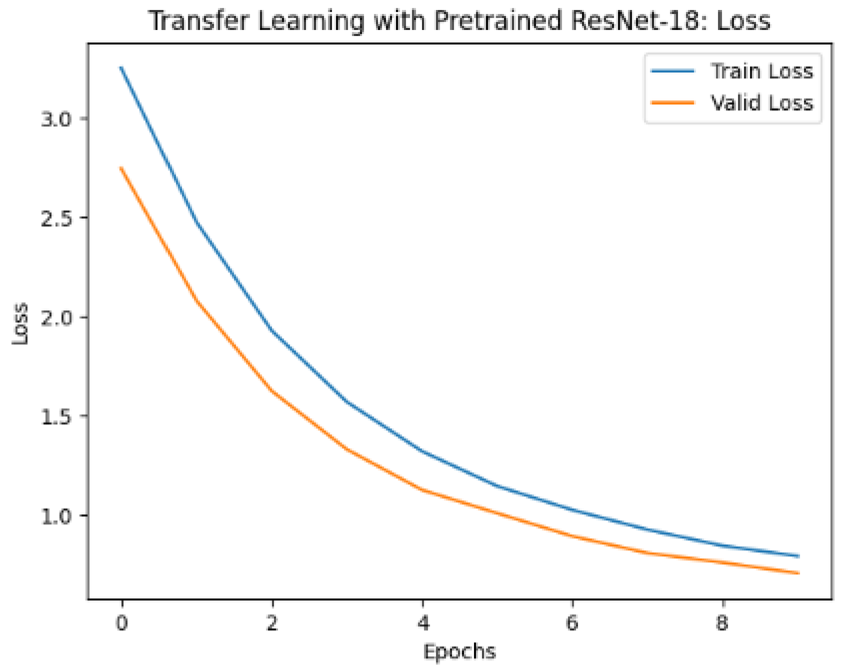
Results

The training results for the transfer learning task over 10 epochs are as follows:

- **Final Train Loss:** 0.7949
- **Final Train Accuracy:** 0.8447
- **Final Valid Loss:** 0.7089
- **Final Valid Accuracy:** 0.8613
- **Training Time:** 549.75 seconds
- **Total Images:** 7389 images with 37 unique breeds.

The model demonstrated significant improvement in both training and validation accuracy throughout the epochs, indicating effective learning and generalization.





```
Training time for Transfer Learning with Pretrained ResNet-18: 549.75 seconds
Found 7389 images in total with 37 unique breeds.
```

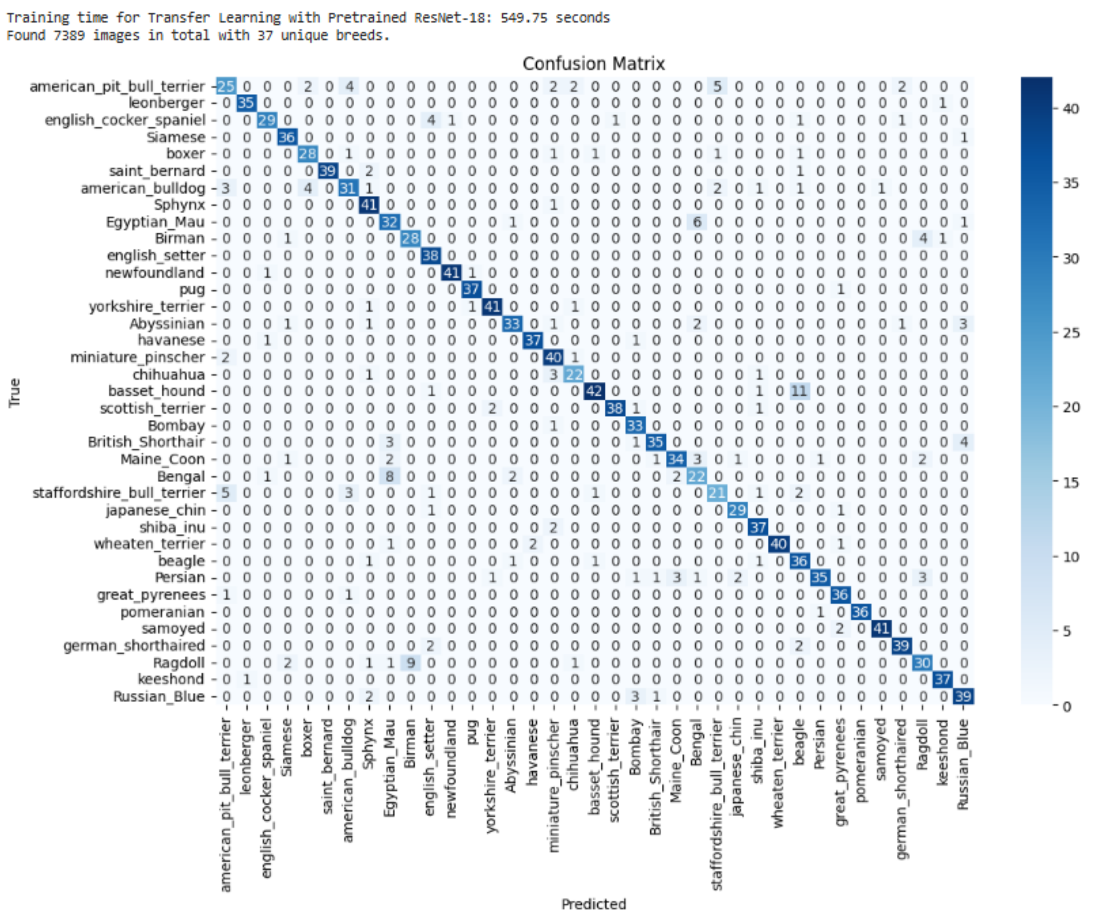
Confusion Matrix

The two plots above illustrate the training and validation accuracy and loss over the 10 epochs.

The accuracy plot indicates that the model is learning effectively, with both training and validation accuracies improving significantly over the epochs. The validation accuracy closely follows the training accuracy, suggesting that the model is generalizing well to unseen data.

The loss plot demonstrates that both training and validation losses decrease over the epochs, indicating effective learning. The validation loss is lower than the initial loss, suggesting that the model is effectively learning to classify the breeds.

As it can be seen in the confution matrix below, some breeds have significantly higher counts of correct predictions, indicating that the model performs well on those breeds.



```
Training time for Transfer Learning with Pretrained ResNet-18: 549.75 seconds
Found 7389 images in total with 37 unique breeds.
```

Challenges and Improvements

One of the challenges faced during transfer learning is ensuring that the model does not overfit to the training data, especially when fine-tuning the last layer. Regularization techniques, such as dropout and weight decay, can be beneficial in this context.

To further enhance the model's performance, one could explore fine-tuning more layers of the pre-trained model, rather than just the final layer. This approach allows the model to adapt more effectively to the specific features of the pet dataset. Additionally, experimenting with different learning rates and optimization strategies could yield better results.

Conclusions

This project explored three approaches to pet classification using the Oxford-IIIT Pet Dataset, demonstrating the progression from binary classification to fine-grained breed identification. The binary classification model achieved a validation accuracy of 82.88%, providing a solid foundation for distinguishing between cats and dogs with minimal complexity. The multi-class breed classification presented greater challenges, reaching 52.03% validation accuracy across 37 breeds, with evidence of overfitting as indicated by the divergence between training and validation metrics.

Transfer learning using a pre-trained ResNet-18 model proved significantly more effective, achieving 86.13% validation accuracy in just 10 epochs—a substantial improvement over the custom CNN while requiring only a third of the training time. This highlights the considerable advantages of leveraging pre-trained models for complex image classification tasks, particularly when working with limited datasets containing subtle class distinctions.

The confusion matrices revealed specific breed pairs that were frequently misclassified, suggesting opportunities for targeted data augmentation or feature engineering. Future work could explore fine-tuning additional layers of the pre-trained model, implementing more sophisticated regularization techniques, and extending the training duration for the breed classification model, which would likely benefit from more than the 30 epochs allocated in this study.