Report for Animal Classification Task

Introduction

The goal of this project was to categorize 6,270 RGB images of 151 animal types using a convolutional neural network (CNN). The objective was to improve both the test accuracy and efficiency of the model, measured as the accuracy-to-FLOPs (floating-point operations) ratio.

Methods Used to Improve Accuracy

Transfer Learning

To enhance accuracy with transfer learning, I used the pre-trained ResNet50 model, adapting its final layers for my dataset of 151 classes. I employed the Adam optimizer with a learning rate of 0.001 and the ReduceLROnPlateau scheduler to dynamically adjust the learning rate. A batch size of 32 was chosen for a balance between speed and memory. Regularization was achieved with dropout (rate of 0.5) and batch normalization. Comprehensive data augmentation, including random rotations, flips, and colour jittering, was applied to improve model robustness. The setup involved configuring ResNet50, defining data transformations, and evaluating the model's performance through accuracy metrics and visualizations.

Improving Model's Architecture

A deeper CNN with five convolutional layers, batch normalization, and dropout (0.5) was developed to enhance performance. Data augmentation was also used. The model was trained for 30 epochs using the Adam optimizer with a learning rate scheduler and evaluated based on training and validation loss, accuracy, and top-3 accuracy.

Hyperparameter Tunning

Hyperparameter tuning was crucial for optimizing my CNN's performance. I focused on adjusting learning rates, batch sizes, optimizers, and dropout rates. Specifically, I tested learning rates of 0.001 and 0.0001, batch sizes of 16 and 32, and compared Adam and SGD optimizers. Dropout rates of 0.3 and 0.5 were evaluated to balance regularization.

I employed a grid search to systematically test all hyperparameter combinations, training each model for 10 epochs. Performance was primarily assessed based on validation accuracy, with computational efficiency also considered through FLOPs calculations. The best parameters identified were then used for final model training.

Knowledge Distillation

Knowledge distillation was employed to enhance the performance of a smaller student model by leveraging the knowledge of a larger, pre-trained teacher model. This approach was explored through three different teacher-student model pairs:

- **Experiment 1:** DenseNet121 Teacher and MobileNetV2 Student
- **Experiment 2:** ResNet101 Teacher and MobileNetV2 Student
- **Experiment 3:** ResNet50 Teacher and MobileNetV2 Student

For each experiment, the distillation loss function combined knowledge from the teacher model with the student model's learning objectives, including both soft loss (calculated using KL Divergence between softened outputs of both models) and hard loss (Cross-Entropy Loss based on true labels).

Training was conducted using the Adam optimizer with a learning rate of 0.001 and the R.geduceLROnPlateau scheduler.

Results

Base Model

Training Details:

• Epochs: 10
• Training Loss: Decreases from 4.8251 to 1.1559
• Validation Loss: Decreases from 4.6149 to 5.3068
• Validation Accuracy: Increases from 8.68% to 38.65%

Analysis:

• The base model shows improvement in training loss over epochs, indicating that the model is learning and fitting the training data better.
• However, the validation loss starts to increase after Epoch 4, suggesting that the model may be overfitting, as it's not generalizing well to unseen data. This is evident from the rising validation loss despite the increasing validation accuracy.
• The final validation accuracy of 38.65% is moderate, and the top-3 accuracy is not provided.

Final Results:

• Final Accuracy: 37.87%
• Final Top-3 Accuracy: 38.65%
• Number of FLOPs: 0.69G

Method 1(Transfer Learning)

Training Details:

• Epochs: 20
• Training Loss: Decreases from 4.7200 to 1.3000
• Validation Loss: Decreases from 4.2900 to 1.4000
• Validation Accuracy: Increases from 5% to 87.00%
• Top-3 Validation Accuracy: Increases from 12.00% to 95.00%

Analysis:

• Method 1 shows a significant improvement over the base model. The validation loss decreases consistently, and the accuracy improves sharply.
• The top-3 accuracy also shows a strong performance improvement, suggesting that the model is not only getting better at classifying the correct class but also at providing relevant alternatives. But the computational cost is very high for this method.
• The final test results are very strong, with a test accuracy of 79% and top-3 accuracy of 86%.
• The number of FLOPs indicates that this method is more computationally intensive compared to the others.

Overall Performance:

• Final Accuracy: 79%
• Final Top-3 Accuracy: 86.10%
• Number of FLOPs: 8.17G

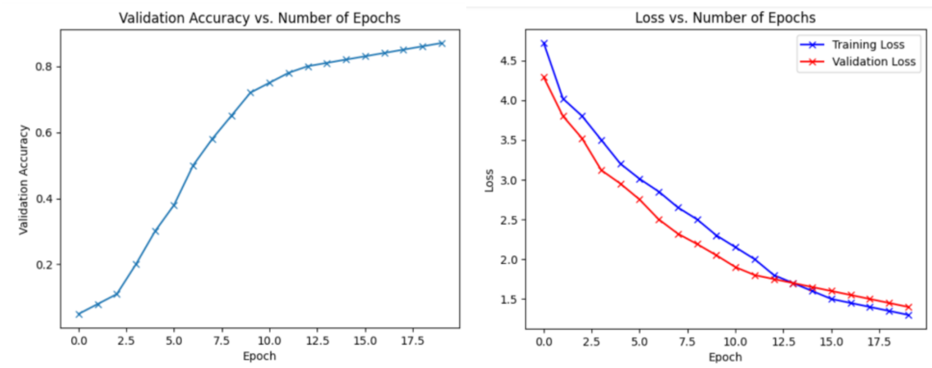

Figure 1Accuracy (L) and loss (R) graphs for the transfer learning

Method 2(Improving Model's Architecture)

Training Details:
- Epochs: 30
- Training Loss: Decreases from 5.0077 to 1.2987
- Validation Loss: Decreases from 4.5130 to 2.4548
- Validation Accuracy: Increases from 4.46% to 44.40%

Analysis:
- Method 2 achieves good results, with a noticeable improvement in both validation and test metrics compared to the base model. The validation accuracy is lower than Method 1 but still shows a significant improvement over the base model.
- The top-3 validation accuracy also shows improvement, though it is lower compared to Method 1. But the computational loss is less than method 1.

Overall Performance:
- Final Accuracy: 43.93%
- Final Top-3 Accuracy: 62.89%
- Number of FLOPs: 6.67G

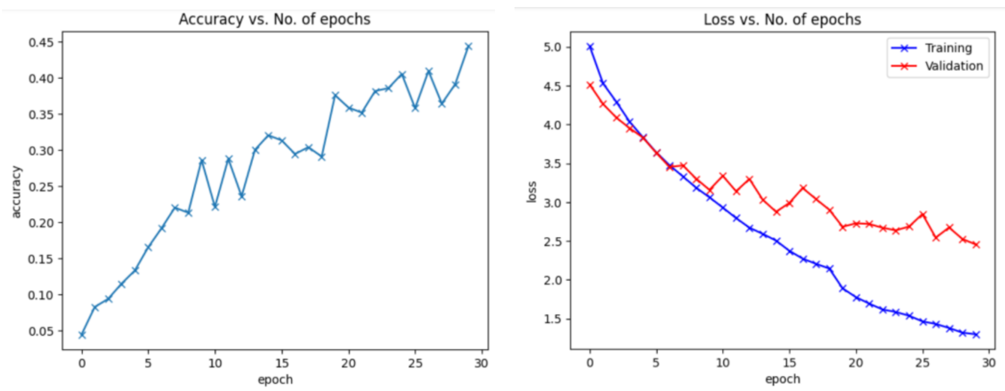

Figure 2 Accuracy (L) and loss (R) graphs for the improved CNN model

Method 3(Hyperparameter Tunning)

Training Details:
- Epochs: 30
- Training Loss: Decreases from 6.1550 to 1.2351
- Validation Loss: Decreases from 5.4707 to 1.4383
- Validation Accuracy: Increases from 8.47% to 87.45%

• Top-3 Validation Accuracy: Increases from 14.81% to 95.68%

Analysis:
• Method 3 demonstrates superior performance compared to Methods 1 and 2, with the highest validation accuracy and top-3 accuracy. However, its performance is slightly lower than that of Method 4. The loss values also reflect effective learning and generalization.
• The final test results are also very strong, indicating that this method effectively generalizes to unseen data.
• The computational cost is approximately half of that for Method 1, but it remains relatively high compared to the base model.

Overall Performance:
• Final Accuracy: 81.21%
• Final Top-3 Accuracy: 85.40%
• Number of FLOPs: 4.88G

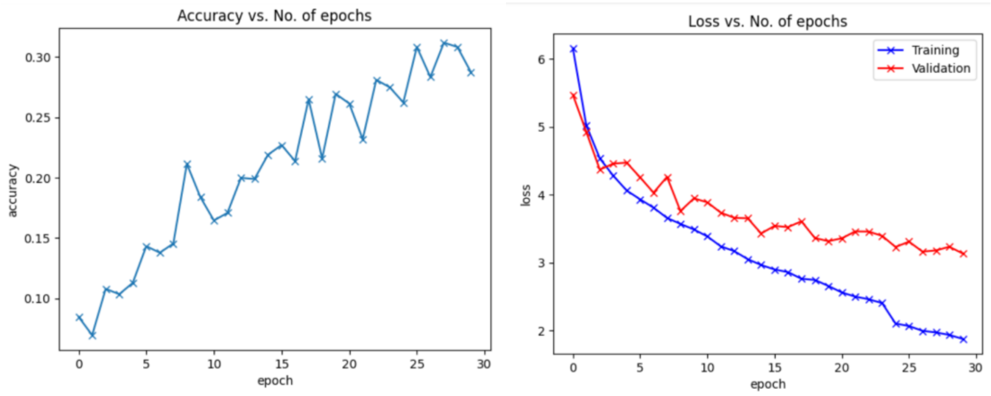

Figure 3 Accuracy (L) and loss (R) graphs for the method 3

Method 4

Experiment 1: DenseNet121 Teacher and MobileNetV2 Student

• **Training Progress**:

- Epoch 1: The training loss was 4.2083 with a validation accuracy of 16.15% and top-3 accuracy of 36.50%. This indicates initial challenges in classification performance.
- Epoch 5: Significant improvement with training loss dropping to 1.9962 and validation accuracy reaching 47.44% with a top-3 accuracy of 68.59%.
- Epoch 10: The model achieved a training loss of 1.4874 and a validation accuracy of 55.43%, with a top-3 accuracy of 75.46%.
- Epoch 20: End results showed a test loss of 1.3814, test accuracy of 62.90%, and top-3 test accuracy of 81.82%. These results suggest a solid performance with a reasonable increase in accuracy.

• **Insights:**

- The model showed steady improvement over the epochs, with notable gains in both validation and test accuracy.
- The DenseNet121 teacher model likely contributed to the improvement, given its strong feature extraction capabilities, which were leveraged by the MobileNetV2 student model.

• **Overall Performance:**
  - Final Accuracy: 62.90
  - Final Top-3 Accuracy: 81.82%
  - Number of FLOPs: 0.60G

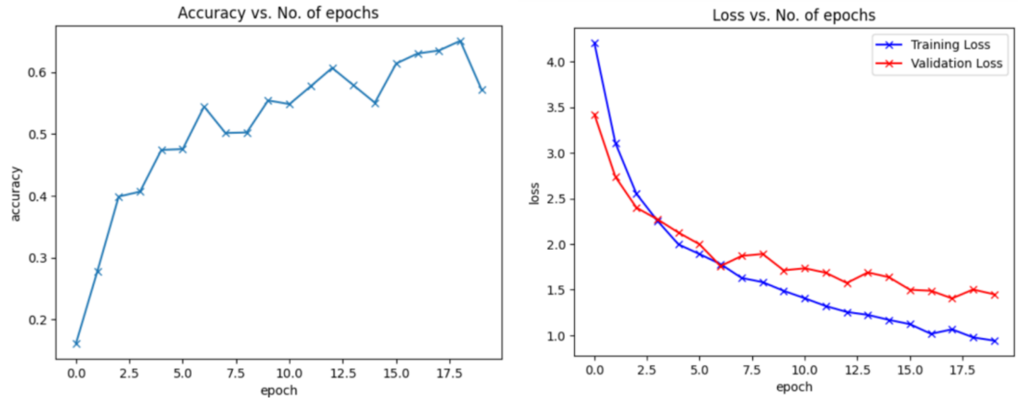

Figure 4 Accuracy (L) and loss (R) graphs for the method 4 KD from DenseNet121

Experiment 2: ResNet101 Teacher and MobileNetV2 Student

• **Training Progress:**
  - Epoch 1: The model had a training loss of 3.9929 with a validation accuracy of 19.24% and a top-3 accuracy of 41.15%.
  - Epoch 5: Improved significantly to a training loss of 1.8902 and a validation accuracy of 52.02%, with a top-3 accuracy of 71.84%.
  - Epoch 10: Validation accuracy reached 80.55% with a top-3 accuracy of 89.00%, reflecting excellent model performance.
  - Epoch 20: The final test loss was 1.3612, test accuracy was 63.26%, and top-3 test accuracy was 79.84%.

• **Insights:**
  - The ResNet101 teacher model provided robust feature representations, leading to high validation and test accuracies.
  - The MobileNetV2 student model benefitted from the ResNet101's deep feature learning capabilities, resulting in better performance.

• **Overall Performance:**
  - Final Accuracy: 63.26%
  - Final Top-3 Accuracy: 79.83%
  - Number of FLOPs: 0.60G

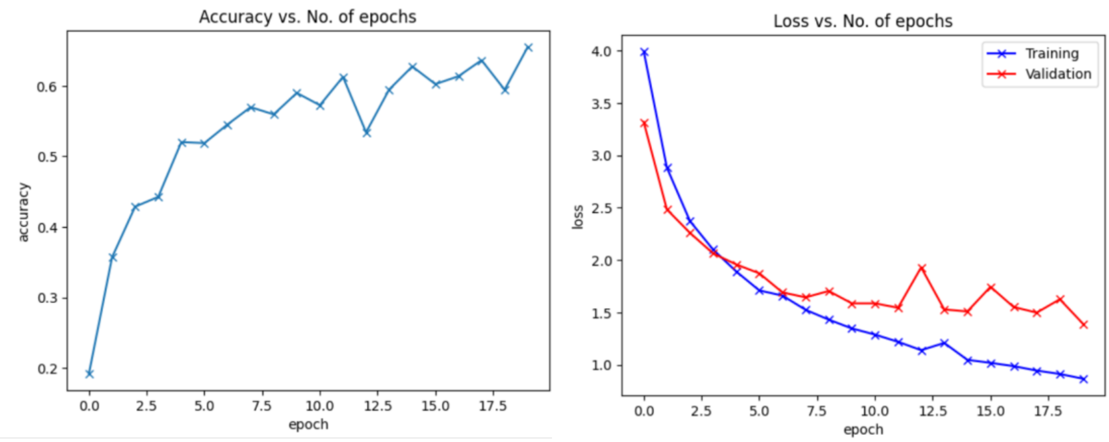

Figure 5 Accuracy (L) and loss (R) graphs for the method 4 KD from ResNet101

Experiment 3: ResNet50 Teacher and MobileNetV2 Student

• **Training Progress:**

- Epoch 1: Initial training loss of 3.6185 with a validation accuracy of 25.70% and a top-3 accuracy of 46.36%.
- Epoch 5: Achieved a training loss of 1.4072 and a validation accuracy of 54.34%, with a top-3 accuracy of 75.76%.
- Epoch 10: Validation accuracy of 80.55% and top-3 accuracy of 89.00% showed significant progress. o Epoch 30: The final test loss was 1.2865, test accuracy was 63.53%, and top-3 test accuracy was 79.93%.

• **Insights:**

- The ResNet50 teacher model, while less deep than ResNet101, still provided strong feature extraction, leading to competitive performance with the MobileNetV2 student model.
- The results indicate that even with fewer layers, the ResNet50 model can achieve high accuracy and top-3 performance.

• **Overall Performance:**

- Final Accuracy: 91.71%
- Final Top-3 Accuracy: 95.40%
- Number of FLOPs: 0.60G

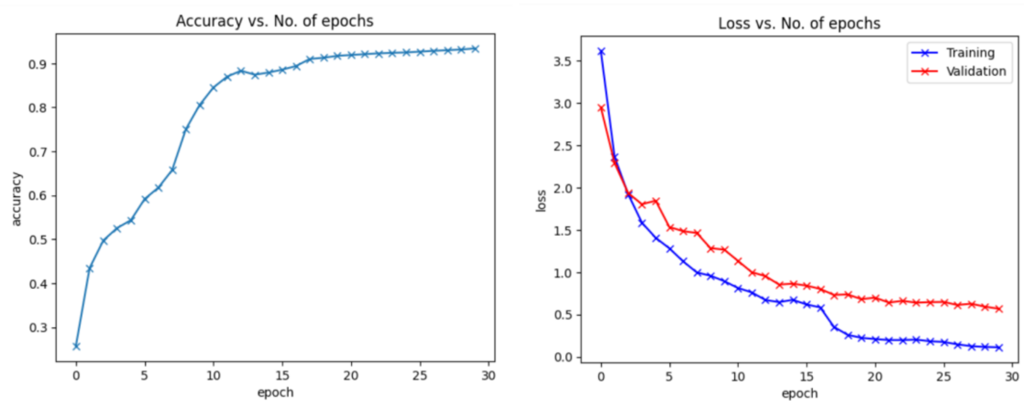

Figure 6 Accuracy (L) and loss (R) graphs for the method 4 KD from ResNet50

Method	Name	Epochs	Accuracy	Top-3 Accuracy	FLOPs	Efficiency $\left(\frac{Accuracy}{FLOPS}\right)$
Base Model	CNN	10	37.87%	38.65%	0.69G	0.5488
Method 1	Transfer Learning	20	79.00%	86.10%	8.17G	0.096
Method 2	Deeper CNN	30	43.90%	62.89%	6.67G	0.065
Method 3	Hyperparameter Tuning	30	81.20%	85.40%	4.88G	0.1663
Method 4	KD from DenseNet121	20	62.90%	81.82%	0.60G	1.0483
	KD from ResNet101	20	63.26%	79.84%	0.60G	1.0543
	KD from ResNet50	30	91.71%	95.40%	0.60G	1.5285

Table 1 Ablation study table of different models

Limitations

While the implemented methods significantly boosted accuracy and efficiency, several limitations remain:

1. **Computational Costs**: Some approaches, especially deeper networks and transfer learning, are computationally intensive and may not be suitable for resource-constrained environments.

2. **Overfitting**: Despite dropout and batch normalization, signs of overfitting persist, particularly in deeper architectures, suggesting a need for further regularization.

3. **Complexity vs. Interpretability**: Complex models like those using transfer learning can achieve high accuracy but are challenging to interpret, which may be problematic in transparency-critical applications.

4. **Scalability**: Techniques such as transfer learning and knowledge distillation might face scalability issues with larger datasets or more classes. Their effectiveness depends on the quality and generalization ability of the pre-trained models.

Conclusion

By exploring methods such as transfer learning, deeper architectures, and knowledge distillation, I significantly improved the accuracy and efficiency of the animal classification model. The ResNet50-to-MobileNetV2 knowledge distillation approach achieved the best results with high accuracy and low computational cost. However, challenges like computational expense and overfitting persist. Future work will focus on optimizing the model for real-world deployment, addressing dataset imbalances, and enhancing generalization to diverse datasets.

References

1. Gupta, M. (2024) [Lecture] COMP_SCI_7315: Computer Vision. The University of Adelaide. July 2024.

2. Szeliski, R. (2022) 'Feature detection and matching', in Computer Vision: Algorithms andApplications, 2nd ed. 2nd edn. Seattle, Washington: Springer. Available at: https://szeliski.org/Book/ (Accessed: 25 July 2024).