Assignment 4 Report

Introduction

Our customers are looking for a solution to automate the classification of images of 151 different animal species using machine learning techniques. The included animal dataset consists of 6270 RGB images, each with a resolution of 224 x 224 pixels, organized into 151 folders, each representing a different animal class. Despite the uniform resolution, the images exhibit poor clarity and low resolution, which can impact the performance of image classification models. On average, there are 41.52 samples per class with a standard deviation of 10.07, indicating a moderate imbalance. The number of samples per class ranges from 26 to 60, with a median of 36.0. This variability indicates that some classes may be underrepresented, leading to biased model training and poor generalization. Ultimately, the model's accuracy and efficiency are critical in real-world applications, to quickly identify and classify animal species from large volumes of images.

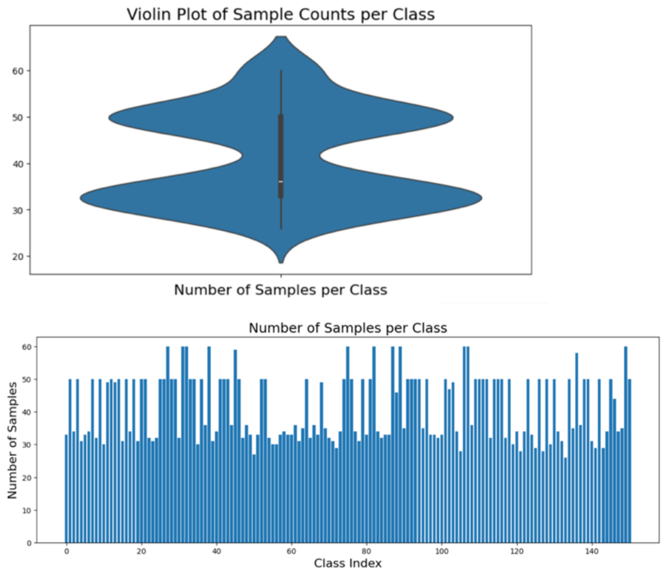

Figure 1: Dataset distribution

Baseline Model

The baseline model provided is a Convolutional Neural Network (CNN) designed specifically for image classification tasks. This model consists of four convolutional layers that are accountable for learning hierarchical features from the input images. Each convolutional layer applies a set of filters to extract spatial features, followed by a ReLU activation function to introduce non-linearity and a max-pooling layer to reduce spatial dimensions.

Then, the final fully connected layer (fc1) serves as the classifier, it takes the flattened feature maps from the last convolutional layer and maps them to 151 output classes (i.e. 151 animal categories in the dataset). The output is then processed through a log-softmax function to produce probability scores for each class.

Upon evaluation, the baseline model showed significant issues with overfitting and poor generalization. It was clear from the relatively low validation accuracy and persistently high error that the model had trouble operating effectively on unseen data. Specifically, after 10 epochs, the baseline model achieved a validation loss of 5.0514 and a validation accuracy of 36.61%. The model has 857,239 trainable parameters and required 0.69 GFLOPs, suggesting moderate-to-high computational cost (in comparison to what we can eventually achieve).

Hence, due to its overfitting and accuracy problems, deploying the baseline model in its current condition may lead to inaccurate predictions and a subpar user experience. With its moderate-to-high computational cost, the baseline model may not be suitable for deployment, further optimization for faster inference/predictions may be required depending on the target platform.

Modifications and Improvements

Increasing Batch Size	Adding Batch Normalisation	Decreasing LR	Accuracy (%)
N	N	N	37.44%
Y	N	N	41.50%
Y	Y	N	58.03%
Y	Y	Y	60.14%

The ablation study we performed revealed that optimizing the batch size, applying batch normalization, and adjusting the learning rate to 0.0005 were crucial for improving model performance. These adjustments increased validation accuracy to 60.14% and reduced validation loss, effectively addressing the initial issues of overfitting and accuracy.

- **Increasing Batch Size from 16 to 64**: This adjustment significantly improved performance, increasing validation accuracy to 41.50% and reducing validation loss to 4.5523. It’s because larger batch sizes enhance convergence and stabilise gradient updates.
- **Adding Batch Normalization**: Integrating Batch Normalization layers after each convolutional layer boosted training speed and model stability. This modification resulted in a validation accuracy of 58.03% and a validation loss of 2.9831, with a slight increase in parameters to 858,135, while maintaining FLOPs at approximately 0.69G.
- **Decreasing Learning Rate to 0.0005**: Lowering the learning rate allowed for more accurate weight adjustments, leading to better convergence and a validation accuracy of 60.14% with a validation loss of 2.7064.

Several other modifications were also explored, including different learning rate schedulers (Reduce on Plateau and Step Size), changes in constant learning rates (0.005, 0.0001, etc.), and different optimizers (SGD, Adagrad, and RMSprop). However, a constant learning rate of 0.005 and Adam (the original optimizer) proved to be the best performing. Advanced techniques such as adding residual layers and data augmentation were also explored but did not yield better results. Adding residual layers decreased the accuracy to 47.37% and significantly increased the FLOPs to 2.18G. Similarly, data augmentation reduced the accuracy to 45.68% without improving the model's performance.

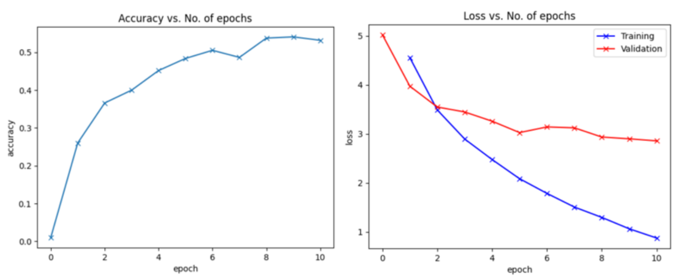

Figure 2: Baseline model best performance

Transfer Learning

To enhance performance, we employed transfer learning for this task. We evaluated pretrained models including ResNet50, EfficientNet-B0, and MobileNetV3 Large. Each of these models has unique strengths and is widely recognized for high performance in image classification tasks.

ResNet50 is a widely used model with high accuracy with moderate efficiency. EfficientNet-B0 balances accuracy and efficiency, providing a good trade-off between performance and computational cost. MobileNetV3 Large is highly efficient with low computational cost, designed for mobile and edge devices, with moderate accuracy.

We chose MobileNetV3 Large for its exceptional efficiency, which is crucial given the computational constraints. It provides a lower computational cost compared to ResNet50. Although EfficientNet-B0 is also efficient, initial trials indicated that MobileNetV3 Large performed better, achieving higher accuracy.

To tailor MobileNetV3 Large for our objective, we fine-tuned the model by freezing the initial layers to retain learned features from the ImageNet dataset and retraining the final layers on our dataset. This approach leverages the pre-trained knowledge of the model while adapting it to the specific nuances of the animal classification task. The modified architecture involved unfreezing the last six layers and replacing the final fully connected layer to match the number of animal classes. Additionally, we changed the input size from 112 to 224 to leverage the full resolution of the dataset and employed a learning rate scheduler to optimize training.

The performance of MobileNetV3 Large was significantly better than the baseline model. The baseline convolutional neural network achieved an accuracy of 60.1%, whereas MobileNetV3 Large reached an impressive 96.41% accuracy. This improvement demonstrates the effectiveness of transfer learning and the suitability of MobileNetV3 Large for this task. The accuracy and loss plots over 10 epochs further illustrate the model's learning curve, showing rapid convergence and stability, indicating effective learning and generalization.

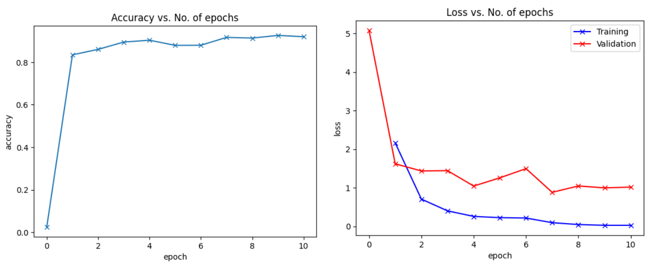

Figure 3: MobileNetV3 Large performance

Transfer Learning	Learning Rate Scheduler	Increase Input Size	Accuracy (%)
Y	N	N	82.8125
Y	Y	N	89.21875357627869
Y	Y	Y	96.4062511920929

Efficiency and Computational Cost

To further reduce computational cost, quantization-aware training (QAT) was applied to the MobileNetV3 Large model. Quantization reduces the precision of the numbers representing the model's parameters from 32-bit floating-point to 8-bit integers, thereby reducing model size and increasing throughput. QAT simulates quantization effects during training, allowing the model to learn to be robust to these effects, leading to better performance when the model is quantized.

For QAT, the quantizable version of the MobileNetV3 Large model was used, with only the last eight layers being trainable to fine-tune the model for our specific use case.

The MobileNetV3 Large model, with QAT applied, achieved an accuracy of 92.81% with a computational cost of 0.002964856 GFLOPs. While the non-quantized model achieved a higher accuracy of 96.41%, it did so at a significantly higher computational cost of 0.428555071 GFLOPs. The efficiency of the non-quantized model is significantly lower (224.95) compared to the quantized model (31,303.05). This stark difference highlights the advantage of using quantization to *achieve a better trade-off* between accuracy and computational cost.

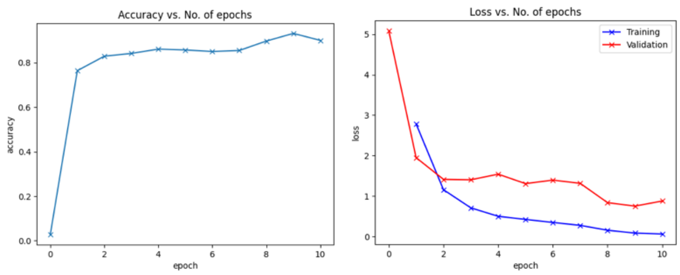

Figure 4: MobileNetV3 Large with QAT performance

	Accuracy (%)	Number of Flops (G)	Efficiency*
MobileNetV3 Large	96.4062511920929	0.42855507135391235	224.956505
MobileNetV3 Large with QAT	92.81250238418579	0.0029648561030626297	31304.2182

$^{*}\,efficiency = \,Accuracy\,(\%)/Number\,of\,Flops\,(G)$

Limitations and Conclusions

One significant limitation of using quantization-aware training (QAT) on MobileNetV3 Large is the accuracy trade-off. While QAT significantly reduce computational cost, there is a slight reduction in accuracy compared to the non-quantized model. In this study, MobileNetV3 Large with QAT achieved an accuracy of 92.81%, whereas the non-quantized MobileNetV3 Large reached 96.41%. This reduction in accuracy, although small, may be critical in applications requiring the highest possible precision. Thus, it is essential to balance the benefits of reduced computational cost and increased efficiency against the potential loss in accuracy when considering QAT.

Another important factor is the quality, quantity, and distribution of the data used for training. The effectiveness of both the baseline and the quantized models is heavily dependent on the dataset. Insufficient or low-quality training data, as well as the uneven distribution of data across classes (as shown earlier), can significantly impact the model's performance, leading to suboptimal accuracy.

In conclusion, despite the slight reduction in accuracy compared to the non-quantized model, MobileNetV3 Large with QAT applied offers a robust approach to achieving high-performance, efficient models suitable for environments with limited computational resources. Ensuring high-quality and ample training data can further enhance the applicability and effectiveness of this approach. For future improvements, expanding the dataset and employing data augmentation techniques can help address these limitations and improve overall accuracy.