Optimizing ResNet50 for Flower Classification : Performance and Efficiency Improvements

Emily Carey (A1851030) , Nethmi Ranathunga(A1895261)

Final Accuracy:89.00% **GFLOPs: 4.1105** **Efficiency Score: 21.85**

The Flower Classification Challenge involved categorising flower images into one of five categories: daisy, dandelion, rose, sunflower, and tulip. The provided dataset consisted of 4,317 training images and 864 test images, each varying in size, lighting, and composition. The baseline model ,a simple CNN with three convolutional layers followed by fully connected layers, achieved a test accuracy of approximately 60%. It used the Adam optimiser with a learning rate of 0.001 and was trained using the CrossEntropyLoss function over 5 epochs, tracking both training and test performance. Basic augmentations were applied, including Resize(224), CentreCrop(224), ToTensor(), and normalisation.

The goal of this project was to significantly improve classification accuracy while maintaining computational efficiency. To achieve this, we implemented transfer learning using a ResNet50 architecture pre trained on ImageNet, leveraging its robust feature extraction capabilities. Alongside architectural changes, we enhanced data augmentation and experimented with various training hyperparameters, loss functions, and optimisers. Our final model achieved a test accuracy of 89.00%, reflecting a significant improvement over the baseline while remaining efficient in terms of computational cost (GFLOPs).

Architecture: Resnet50 and Training Parameters

We replaced the baseline CNN with a pretrained ResNet50, which uses residual connections to improve gradient flow and feature extraction (Wang et al., 2024; He et al., 2015). To preserve pretrained knowledge, all base layers were frozen. A custom classification head with ReLU, Dropout (0.2–0.5), and a five-node output layer was added. This allowed efficient fine-tuning while adapting the model to the flower classification task.

We further tuned the training configuration to improve performance. A learning rate of 0.001 was selected after lower rates showed more stable convergence than 0.01. A smaller learning rate avoids large parameter updates that could destroy pre-trained weights when fine tuning a pretrained network (Matani 2023). Among the optimisers tested: SGD, Adam, RMSprop, AdamW, and Adadelta. SGD with momentum consistently yielded the highest final accuracy. This could be due to SGD being superior when it comes to image classification tasks as adaptive gradient methods like Adam converge fast but generalise worse (Wilson et al. 2017). A batch size of 8 was chosen after comparison with larger sizes, which showed diminished performance. This is consistent with prior findings that smaller batch sizes often improve generalisation by introducing beneficial noise in gradient updates, helping the model avoid sharp minima (keskar et al. 2017). Given dropout rates between 0.2 and 0.5 are typically effective for fully connected layers in deep neural networks (Srivastava et al. 2014, p.1940) the dropout rates between 0.2 and 0.5 were explored in the final classification layer. The dropout rate of 0.3 provided the best balance between regularisation and generalisation.

To address class ambiguity caused by visual similarities between flower types, we used Focal Loss, which helps the model focus on harder-to-classify samples by down-weighting easier ones (Lin et al., 2017).While we didn't manually define residual blocks, our pretrained ResNet50 inherently leveraged them. These skip connections enable learning of residual functions, improving gradient flow and preventing vanishing gradients (He et al., 2015).We froze the pretrained layers and fine-tuned only the final classification head. Despite this, the residual backbone extracted strong high-level features, allowing our model to generalise well while training fewer parameters.

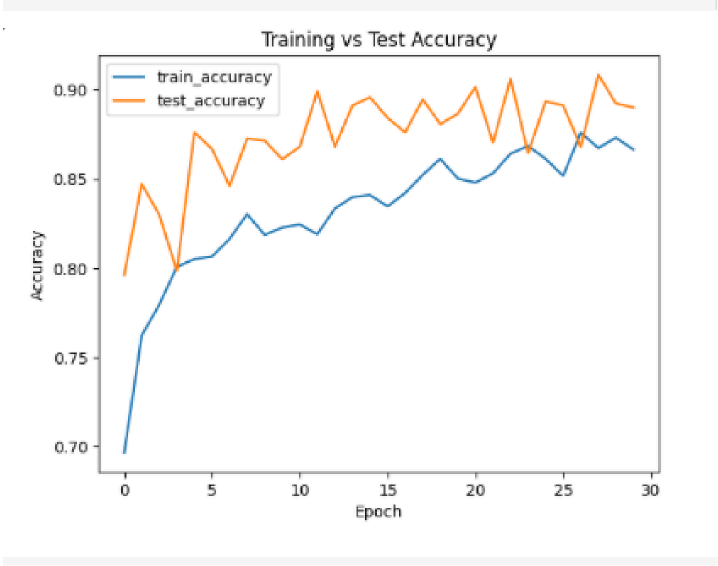

Figure 1: Training Vs Test Accuracy

The training and validation curves confirm that the final configuration converged steadily without signs of overfitting, as shown in Figure 1. Training accuracy and loss improved consistently across epochs, and the gap between training and validation performance remained small indicating strong generalisation.

Data Preprocessing

We retained the core preprocessing pipeline from the baseline model to maintain consistency and ensure compatibility with ResNet50. This included:

- Resize(224) and CenterCrop(224) to standardise image dimensions,
- ToTensor() for conversion to PyTorch tensors,
- Normalisation using standard ImageNet mean and standard deviation values.

To improve generalisation, we added RandomRotation(15) and RandomHorizontalFlip() to introduce variability in orientation and symmetry. ColorJitter was initially tested but removed after it reduced accuracy.Visual inspection using inverse normalisation confirmed that transformed images remained interpretable. The impact of different preprocessing configurations is reflected in Appendix Table 1.

Computational Efficiency

The final model used a fixed ResNet50 architecture with an estimated 4.1105 GFLOPs. With a test accuracy of 89.81%, the resulting efficiency score was 21.85.While all configurations had similar FLOPs, the optimiser choice affected performance. SGD with momentum converged more slowly but achieved the highest accuracy. In contrast, Adam and RMSprop trained faster but generalised worse. This reflects a key trade-off: adaptive optimisers offer speed, while SGD supports better generalisation (Wilson et al., 2017).

SGD's tendency to find flatter minima is especially useful in transfer learning, helping preserve pretrained features while adapting to new data (Keskar et al., 2017).

Limitations and Conclusions

Future work would involve further exploration of data augmentation techniques with the exploration of colour jitter, blur and random rotations, as this could improve the model's robustness to variations in lighting, composition and background. Another limitation is that the provided dataset is relatively small (4300 training images) and it is limited to 5 classes, so the model's ability to generalise to a wider variety of flower species is unknown. Additionally, with more time, exploration of larger models such as EfficientNet could be explored, as it could result in an increased gain in accuracy. Another limitation pertains to the limited number of training epochs the model was trained over; longer training could potentially improve the accuracy.

An important observation was that by utilising a pretrained architecture (ResNet50), the model significantly improved accuracy, demonstrating that pretrained models are highly efficient when working with small datasets. Furthermore, while SGD converged more slowly compared to the other investigated optimisers, it produced the best test accuracy, emphasising its superior generalisation. In terms of balanced efficiency, the model achieved a good balance between computational cost and performance, making it suitable for lightweight applications. Overall, the model produced really good results and provides a solid baseline for future iterations. Future improvements could be achieved through the use of deeper models and more advanced augmentation strategies.

Appendix

Table 1: Ablation Table: Resnet50 - Optimizer, Focal Loss and Training Parameters

Focal Loss	ResNet50 (Residual Blocks)	Learning Rate	Dropout	Batch Size	Optimiser	Accuracy
□	□	0.001	0.5	10	Adam	64.47
□	□	0.001	0.3	30	SGD	89.00

References

Wang, R, Zhou, X, Liu, Y & Su, M 2024, ‘Identification of thee Surface Cracks of Concrete Baes on ResNet-18 Depth Residual Network’, Applied Sciences, vol.14, no.8, pp.1-18

Matani, D, 2023, A Practical Guide to Transfer Learning using PyTorch, Data skills for decision making, viewed 7 June 2025,
<https://www.kdnuggets.com/2023/06/practical-guide-transfer-learning-pytorch.html>

Wilson, A, Roelofs, R, Stern, M , Srebro, N & Recht,B 2017, ‘The Marginal Value of Apaptive Gradient Methods in Machine Learning’ in 31st Conference on Neural Information Processing System, Long Beach, CA, USA.

Keskar, NS, Mudigere, D, Nocedal, J, Smelyannskiy, M, Tang, PTP 2017, ‘Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima’, in conference at ICLR, Intel Corporation

Srivastavva, N, Hinton, G, Sutskever, I & Salakhuutdinov, R 2014, ‘Dropout: A simple way to prevent neural networks from overfitting’, Journal of Machine learning Research, no.15, pp.1929-1958

He, K, Zhang, X, Ren, S & Sun, J 2015, ‘Deep Residual Learning for Image Recognition’, Computer Vision and Pattern Recognition

Lin, TY, Goyal, P, Girshick, R, Kaiming, H & Dollar, P, ‘Focal Loss for Dense Object Detection’, Computer Vision and Pattern Recognition