Colab Link: <https://colab.research.google.com/drive/1XuHZHcjALyrVLkot3HocxVDE0PD-gw0j?usp=sharing>

I have chosen **Option 2: Image classification based on Deep Learning**

Dataset: I have used Flowers Dataset present in the provided github repo <https://github.com/lkk688/MultiModalClassifier>.

The objective of this solution is to evaluate and compare the performance, ease of use, and flexibility of PyTorch and TensorFlow frameworks in a practical machine learning task,so in this assignment I have used the ResNet50 model based on these frameworks.

To make it a Faircomparison I tried to make both the models as identical as possible in terms of architecture, training data, and training regimes. This way, any differences in performance can more accurately be attributed to the frameworks themselves rather than differences in model design or training procedure.

**PyTorch ResNet50 Implementation:**

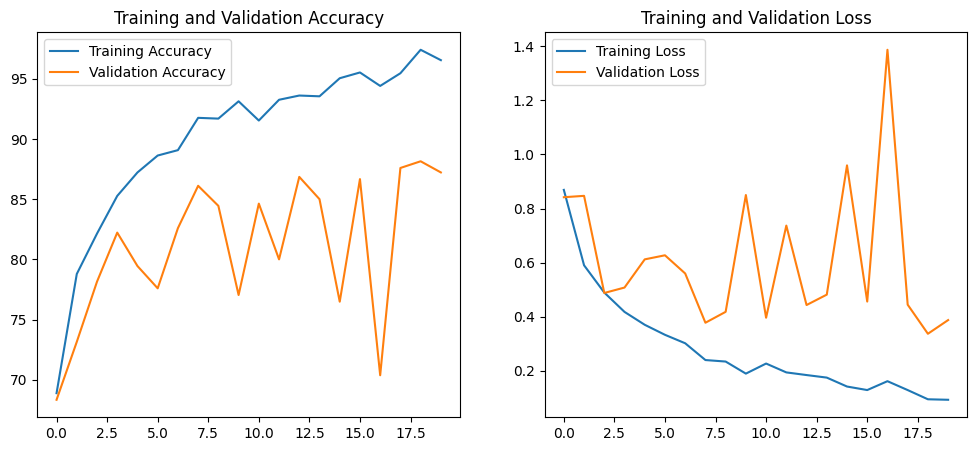
* Model Configuration: A ResNet50 model, pretrained on ImageNet, was adapted for the flower classification task. The final fully connected layer was modified to output five classes, corresponding to the flower types in the dataset.
* Data Preprocessing: The images were resized to 224x224 pixels, randomly flipped horizontally, converted to tensors, and normalized using standard mean and standard deviation values.
* Training Process:The model was trained for 20 epochs, using the Adam optimizer and CrossEntropyLoss. Training and validation losses and accuracies were recorded for analysis

**TensorFlow ResNet50 Implementation:**

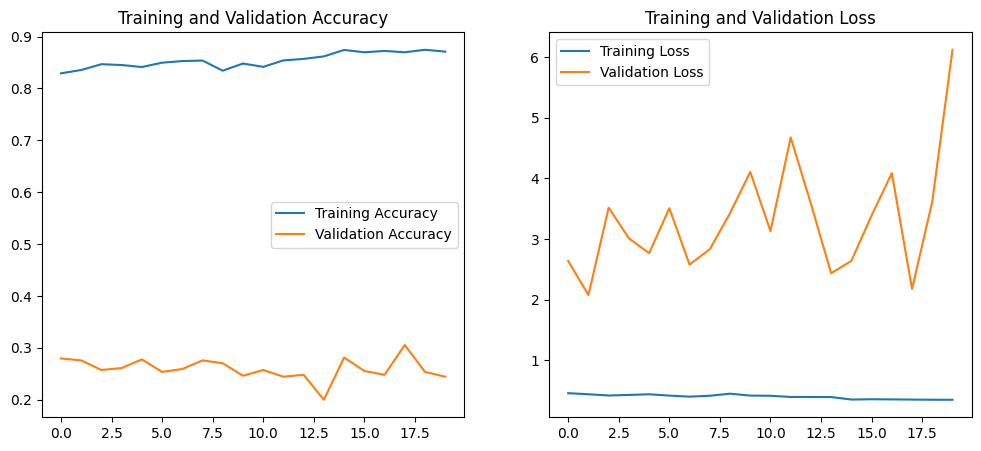
* Model Configuration: The TensorFlow implementation also used a pretrained ResNet50 model. Custom layers were added for the classification task, and the base layers were set to be non-trainable.
* Data Preprocessing: Similar preprocessing steps were applied as in the PyTorch implementation, including resizing, normalization, and data augmentation techniques like rotation, width shift, height shift, and horizontal flipping.
* Training Process: The TensorFlow model was compiled with Adam optimizer and categorical crossentropy loss, and trained for 20 epochs. Training and validation metrics were recorded.

**Results:**

Pytorch ResNet50:



Training and Validation Accuracy and Loss:The PyTorch model showed a steady increase in training accuracy and a decrease in training loss over 20 epochs. The validation accuracy and loss also improved, indicating good model performance.



Training and Validation Accuracy and Loss: The TensorFlow model demonstrated similar trends in training accuracy and loss. However, the validation accuracy and loss showed more fluctuation, suggesting potential overfitting

The comparison revealed that both PyTorch and TensorFlow are capable of effectively training a ResNet50 model on the flower dataset. PyTorch showed slightly more stable performance in validation metrics, while TensorFlow exhibited higher fluctuations. These differences could be attributed to the frameworks' distinct approaches to model training and optimization.

This study highlights the strengths and limitations of PyTorch and TensorFlow in handling a practical image classification task. Both frameworks proved to be powerful tools, with PyTorch showing a bit more stability in validation performance.