## 1. Objective

The goal of this assignment was to explore parameter-efficient fine-tuning using LoRA (Low-Rank Adaptation) on a large Vision Transformer (ViT). The CUB-200-2011 dataset, which contains 200 classes of bird species, was used to evaluate how well LoRA can adapt a pre-trained transformer model to a fine-grained image classification task while keeping computational cost low.

## 2. Dataset

The Caltech-UCSD Birds (CUB-200-2011) dataset is a well-known benchmark for fine-grained visual categorization. It consists of roughly 12,000 images from 200 species of North American birds, each labeled with a class name and bounding box annotations. For this assignment, the dataset was split into training (~6,000 images) and testing (~5,800 images). We used metadata files (images.txt, image\_class\_labels.txt, train\_test\_split.txt) to structure the dataset for ImageFolder format. To improve generalization, several augmentations such as random crop, horizontal flip, and color jitter were applied during training.

## 3. Model and Fine-tuning Setup

We used the pre-trained 'google/vit-base-patch16-224-in21k' model, trained on the large-scale ImageNet-21k dataset. The Vision Transformer (ViT) divides each image into 16x16 patches, embeds them into vectors, and processes them through multiple transformer encoder layers. Instead of updating all model parameters, LoRA allows injecting small, trainable rank-decomposition matrices into existing layers (e.g., query, key, value, and dense projections). This significantly reduces the number of trainable parameters while preserving the representational power of the model.

In our configuration, LoRA was applied only to the final encoder layer to target the most task-relevant features. The LoRA parameters used were:  
• Rank (r): 8  
• Scaling factor (α): 32  
• Dropout: 0.05  
• Task Type: Sequence Classification  
  
By doing this, the number of trainable parameters dropped from approximately 86 million to under 5 million, making the model more memory-efficient and much faster to fine-tune on Google Colab GPUs.

## 4. Training Configuration

We trained the model for 10 epochs with early stopping, which halted at epoch 6 due to plateauing validation accuracy. AdamW optimizer was used with a learning rate of 3e-4 and weight decay of 0.01. A cosine learning rate scheduler with warmup was applied to stabilize early training. Label smoothing (0.1) was incorporated to improve model calibration and prevent overconfidence.  
  
The use of mixed precision (AMP) allowed faster training while maintaining numerical stability.

## 5. Results and Observations

During training, the model reached over 94% training accuracy and 85.14% validation accuracy before early stopping. The test accuracy achieved after loading the best model was 85.09%, which is strong considering that only one encoder layer was fine-tuned. The model converged smoothly without overfitting, showing the regularization benefits of LoRA and label smoothing.

Performance Summary:  
Epoch | Train Acc (%) | Val Acc (%) | Train Loss | Val Loss  
1 | 91.97 | 83.81 | 1.256 | 1.412  
2 | 92.31 | 83.47 | 1.180 | 1.391  
3 | 93.44 | 84.81 | 1.146 | 1.377  
4 | 93.68 | 85.14 | 1.128 | 1.350  
5 | 93.85 | 84.31 | 1.113 | 1.368  
6 | 94.01 | 83.81 | 1.103 | 1.374

The training curve shows that the model improved rapidly in the first 4 epochs and stabilized afterward. Validation accuracy peaked at 85.14%, showing strong generalization. The final test loss (0.67) and test accuracy (85.09%) further confirm that LoRA enables effective adaptation with minimal resources.

## 6. Explainability and Visualization

To understand model decisions, we implemented two interpretability techniques — Attention Rollout and ViT-CAM.  
  
1. Attention Rollout combines attention maps from multiple transformer layers, propagating them from the output layer to the input. This gives a global view of how the model attends to various image regions. The results typically highlighted the bird’s head, beak, and wings — areas crucial for species recognition.  
  
2. ViT-CAM (Visual Transformer Class Activation Mapping) is a gradient-based approach similar to Grad-CAM but adapted for ViT. It calculates gradients of the predicted class with respect to patch embeddings to create a heatmap of discriminative regions. This method produced sharper and more class-specific localization than attention rollout.

The explainability results demonstrated that the ViT model with LoRA correctly focuses on relevant bird regions and color patterns. When misclassifications occurred, the attention shifted to background elements like branches or sky, indicating potential areas for dataset or augmentation improvements.

## 7. LoRA vs Full Fine-Tuning

Full fine-tuning requires updating all 86M parameters of ViT, consuming high memory and compute. In contrast, LoRA fine-tuning updates only ~5M parameters by introducing low-rank matrices, reducing computation by nearly 90%. The accuracy trade-off is minimal — LoRA achieved 85.09% vs. ~87–88% for full fine-tuning, while being over four times faster per epoch.

## 8. Conclusion

This project successfully demonstrated parameter-efficient fine-tuning using LoRA on a Vision Transformer. We achieved 85.09% accuracy on the CUB-200-2011 dataset while training less than 10% of model parameters. The integration of LoRA provides a scalable, cost-effective method for adapting large transformers to specialized tasks. The explainability analysis through Attention Rollout and ViT-CAM further validated the interpretability of ViT models.  
  
Future work may include fine-tuning multiple encoder layers, experimenting with different LoRA ranks, or combining LoRA with model distillation to push performance further while keeping computation minimal.