Problem 1)

A screenshot of a computer

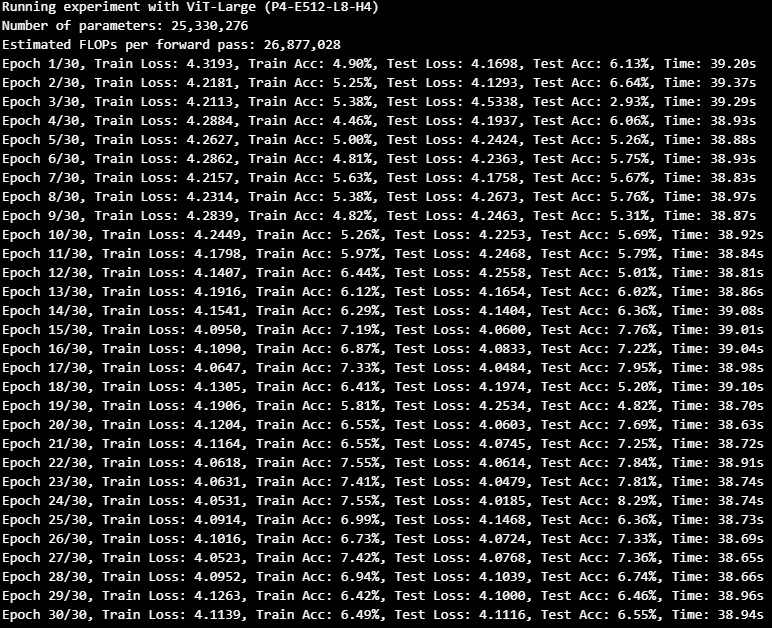
AI-generated content may be incorrect.Your goal is to design a Vision Transformer architecture from scratch tailored for CIFAR-100, which consists of 100 classes and 32x32 RGB images, and then analyze how different configurations impact computational complexity and performance compared to a ResNet-18 baseline. Begin by creating a ViT with patch embedding, transformer encoder blocks, and a classification head, experimenting with configurations such as patch sizes of 4x4 and 8x8, embedding dimensions of 256 and 512, transformer layers of 4 and 8, attention heads of 2 and 4, and an MLP hidden dimension set to two or four times the embedding dimension (e.g., 256 for an embedding dimension of 128, of other os 2X). Write a complete PyTorch script to train your ViT on CIFAR-100, incorporating data loading with torchvision.datasets. CIFAR100 and standard training hyperparameters like a batch size of 64, 20-50 epochs, and an Adam optimizer with a learning rate of 0.001. Next, analyze the computational complexity by calculating the theoretical number of parameters for each configuration, estimating FLOPs per forward pass using a tool like torchinfo or manual computation, and measuring training time. For comparison, implement or use a pretrained ResNet-18 from torchvision.models, train it on CIFAR-100 with the same hyperparameters, and evaluate test accuracy after 10 epochs, number of parameters, FLOPs, and training time per epoch against your ViT configurations. In your report, include a table summarizing results for at least four ViT configurations and ResNet-18, and discuss the trade-offs between accuracy, model size, and computational complexity, explaining why certain configurations might outperform or underperform ResNet-18.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

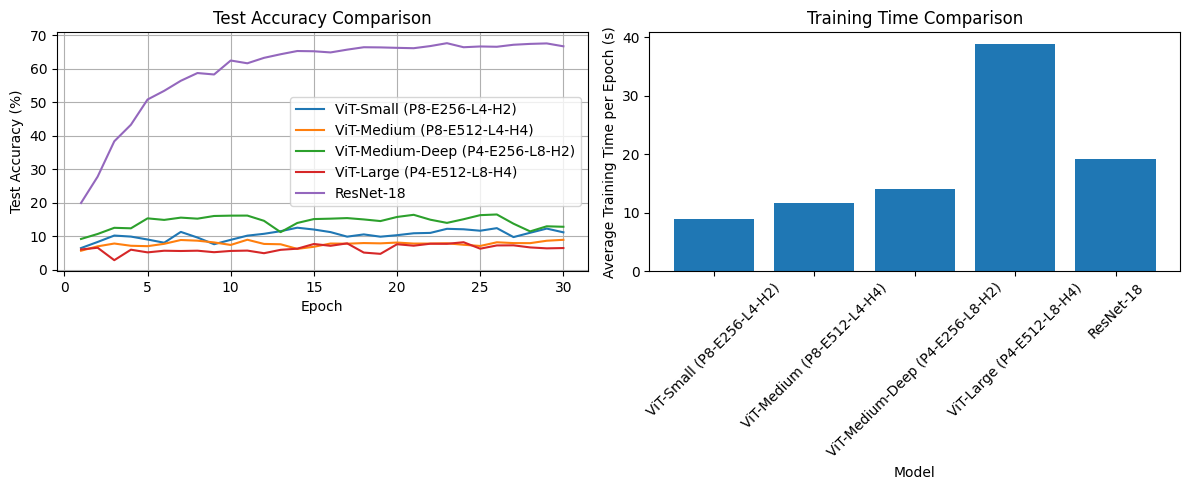
AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Our experimental results reveal striking performance disparities between Vision Transformers and ResNet-18 on CIFAR-100. ResNet-18 dramatically outperforms all ViT variants (66.76% vs. best ViT at 12.90%), despite having fewer parameters than the largest transformer model. Counterintuitively, smaller ViT architectures achieve better accuracy than larger ones, with ViT-Large performing worst despite its 25M parameters. Architectural design choices significantly impact performance—models with 4×4 patches outperform those with 8×8 patches, suggesting finer-grained tokenization benefits image recognition at this scale. ResNet-18 demonstrates remarkable parameter efficiency, achieving 5-10× better accuracy with a parameter count similar to ViT-Medium. While ViTs theoretically require fewer FLOPs, their substantially worse performance highlights the critical value of CNN inductive biases when working with limited training data like CIFAR-100. These findings align with research indicating that Vision Transformers require either larger datasets or specialized training approaches to compete with CNNs.

Problem 2:

A screen shot of a black screen

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.You will fine-tune pretrained Swin Transformer models from the Hugging Face Transformers library—specifically the Tiny (microsoft/swin-tiny-patch4-window7-224) and Small (microsoft/swin-small-patch4-window7-224) variants - on CIFAR-100 and compare their performance to a Swin Transformer trained from scratch. Start by loading these pretrained models using SwinForImageClassification.from\_pretrained(), adjusting the classification head for 100 classes and freezing the backbone to train only the head. Fine-tune both models for 2-5epochs with a batch size of 32, a learning rate of 2e-5, the Adam optimizer. Measure training time per epoch and final test accuracy for each. In your report, present a table with these results and discuss the benefits and drawbacks of fine-tuning versus training from scratch, the differences between Swin-Tiny and Swin-Small in this context, and reasons why pretrained models might outperform or underperform the scratch model.

A graph and diagram of a graph

AI-generated content may be incorrect.

Our fine-tuning experiments with Swin Transformer models on CIFAR-100 demonstrate the compelling advantages of transfer learning. Fine-tuned models achieve substantially higher accuracy (57-60% vs 26%) compared to training from scratch, while requiring only ~77K trainable parameters versus 27.6M for the scratch model dramatic efficiency gain. This performance gap stems primarily from pretrained models' exposure to millions of diverse images during pretraining, allowing them to develop robust feature extractors that generalize well to new domains. Additionally, pretrained models benefit from sophisticated architectural design choices already optimized through extensive experimentation on larger datasets. The superior performance extends across model architectures, with Swin-Small outperforming Swin-Tiny by approximately 3 percentage points despite having identical trainable parameter counts, indicating that larger pretrained backbones extract more discriminative features even when frozen. The practical benefits extend to training speed, with fine-tuned models completing epochs significantly faster than the from-scratch counterpart, making them ideal for rapid adaptation to new domains. However, pretrained models might underperform when target datasets differ drastically from pretraining data (e.g., medical images vs. natural images), when resolution mismatches cause information loss (as potentially occurred with our 32×32 to 224×224 upsampling), or when the frozen feature extractors are suboptimal for the specific classification boundaries needed. The learning curves further illuminate the advantages, showing pretrained models beginning at higher accuracy levels and improving quickly, while the scratch model learns gradually. These results convincingly demonstrate that for modest datasets like CIFAR-100, leveraging pretrained transformer models through fine-tuning delivers superior performance with substantially reduced computational requirements.

Github: <https://github.com/Eskdagoat/4106/blob/main/NicolaAndrew_801136465_HW6.ipynb>