

Vision Transformer Report

Darsh Jain - 240324

1. Abstract and Theoretical Framework

The transition from local spatial hierarchies to global dependency models marks a paradigm shift in computer vision. While Convolutional Neural Networks (CNNs) rely on inherent inductive biases like translation invariance and locality, the Vision Transformer (ViT) architecture minimizes these assumptions. This report explores the implementation of a ViT tailored for low-resolution datasets, specifically CIFAR-10, focusing on the mechanics of self-attention as a substitute for traditional convolutional filters.

2. Experimental Environment: CIFAR-10 Dynamics

The CIFAR-10 dataset provides a rigorous benchmark, consisting of 60,000 32×32 color images distributed across 10 balanced categories. Because the images are low-resolution, the choice of patch size is critical for capturing fine-grained structural detail.

2.1 Advanced Preprocessing and Augmentation

To mitigate the "inductive bias gap" inherent in Transformers, a sophisticated data manipulation pipeline was utilized to improve generalization and robustness.

- **Geometric Transformations:** Horizontal flipping and random cropping with 4-pixel padding were used to simulate spatial variability.
- **Standardization:** Input tensors were centered using mean and standard deviation specific to the CIFAR-10 distribution ($\mu = [0.4914, 0.4822, 0.4465]$).
- **Regularization:** Techniques such as Weight Decay (0.05) and Dropout (0.1) were integrated to prevent the model from memorizing noise in the training set.

Attribute	Value	Theoretical Impact
Batch Size	128	Balances gradient stability and memory throughput.
Initial Learning Rate	3e-4	Optimized for stable

		convergence in deep architectures.
Gradient Clipping	1.0	Prevents exploding gradients in attention layers.
Optimizer	AdamW	Efficient weight decay decoupled from the learning rate.

3. Vision Transformer Architecture

The core innovation of the ViT is "tokenization"—treating an image as a sequence of discrete patches rather than a continuous grid.

3.1 Tokenization and Patch Embedding

The input image $x \in \mathbb{R}^{H \times W \times C}$ is decomposed into N non-overlapping patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where $P = 4$ for this implementation. These patches are projected into a constant latent space of $D = 384$ dimensions. While manual flattening is possible, modern implementations often utilize a 2D convolution with a kernel and stride equal to the patch size to achieve performance gains during the projection step.

3.2 Global Context via Self-Attention

The Multi-Head Self-Attention (MHSA) module allows each patch to "attend" to every other patch, capturing long-range dependencies inaccessible to local convolutional kernels.

- **Query, Key, Value (QKV):** Tensors are computed through learnable projections: Q, K, V .
- **Scaled Dot-Product:** Attention weights are derived using the formula:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

Using einsum for these tensor contractions provides a pithy, self-documenting syntax that facilitates clearer visualization of the latent dimensions.

3.3 Positional Encoding and the CLS Token

- **CLS Token:** Prepending a learnable classification token $z_0 = x_{class}$ allows the model to

- aggregate global image features into a single vector for the final classification head.
- **Positional Embeddings:** Because transformers are permutation-invariant, learnable identifiers are added to provide the necessary spatial context to the sequence. Recent research suggests that positional embeddings play a "counterbalancing role" in deeper layers, stabilizing token values as features become more abstract.

4. Optimization Strategy: Pre-Norm and GELU

To ensure training stability in deep stacks (6 layers), the **Pre-Normalization** configuration was adopted, where Layer Normalization (LN) is applied before the MHSA and MLP modules.¹ The Feed-Forward Network (MLP) employs the **GELU** (Gaussian Error Linear Unit) activation function, which provides a smoother non-linearity compared to ReLU, aiding in the capture of complex patterns during training.

5. Performance

After 50 epochs, the model achieved a training accuracy of 95.03% and a peak test accuracy of 79.30%. This performance highlights the data-hungry nature of transformers, which lack the "spatial bias" of CNNs and therefore require higher data volumes or more aggressive augmentation (such as MixUp or Manifold Mixup) to reach competitive benchmarks on small datasets like CIFAR-10.