Vision Transformer Based Image Classification on CIFAR-10

1. Introduction

Traditionally, Convolutional Neural Networks (CNNs) were used for image classification because they efficiently capture spatial information. However, Vision Transformers process images differently — they treat images as sequences of patches, similar to how words are treated in a sentence. This project helped in understanding how attention mechanisms can replace convolution operations.

We trained a Vision Transformer model on the CIFAR-10 dataset using Google Colab. During training, we experimented with different regularization techniques and hyperparameters to improve performance and prevent overfitting.

2. Dataset Description – CIFAR-10

CIFAR-10 is a well-known benchmark dataset for image classification.

It consists of:

- 60,000 color images
- 10 different classes
- 32 × 32 pixel images
- 50,000 training images
- 10,000 test images

The 10 classes are:

- Airplane
- Automobile

- Bird
- Cat
- Deer
- Dog
- Frog
- Horse
- Ship
- Truck

Since Vision Transformers usually work with larger images (like 224 × 224), we resized CIFAR-10 images before feeding them to the model.

3. Preprocessing and Regularization Techniques

Proper preprocessing and regularization are very important to ensure good performance and generalization.

3.1 Image Preprocessing

We applied the following preprocessing steps:

1. Resizing
   CIFAR-10 images are 32 × 32, but ViT expects 224 × 224 images. So, we resized all images to 224 × 224.

2. Normalization
   We normalized images using ImageNet mean and standard deviation values:

   - Mean = [0.485, 0.456, 0.406]
   - Std = [0.229, 0.224, 0.225]

3. Normalization helps stabilize training and improves convergence speed.

3.2 Data Augmentation (Regularization)

To reduce overfitting, we used several augmentation techniques:

1. Random Horizontal Flip
   Randomly flips images left-right.
   This helps the model become invariant to orientation.

2. Random Crop with Padding
   Adds padding and randomly crops the image.
   This introduces translation invariance.

3. Color Jitter
   Randomly changes brightness, contrast, and saturation.
   This improves robustness to lighting variations.

3.3 Model-Level Regularization

We also applied regularization inside the model:

1. Dropout (0.1)
   Randomly drops neurons during training.
   This prevents the model from depending too much on specific neurons.

2. Weight Decay (0.05)
   Applied using AdamW optimizer.
   It penalizes large weights and reduces overfitting.

3. Label Smoothing (0.1)
   Instead of giving full probability to the correct class, we slightly smooth the labels.
   This improves generalization and reduces overconfidence.

Effect of Regularization

Without regularization:

- Training accuracy was high

- Test accuracy was much lower

- Model overfitted

With regularization:

- Gap between train and test accuracy reduced
- Training became more stable
- Final test accuracy improved significantly

4. Mechanism of Vision Transformer

Vision Transformer works differently from CNNs. It follows these main steps:

4.1 Patch Embedding

Instead of feeding the whole image, the image is divided into small patches.

For example:

- Image size = 224 × 224
- Patch size = 16 × 16
- Total patches = $(224 / 16)^2$ = 196 patches

Each patch is flattened and projected into a vector of fixed dimension (embedding size = 768).

So the image becomes a sequence of 196 vectors.

4.2 Positional Encoding

Since transformers do not understand spatial positions automatically, positional embeddings are added to each patch embedding.

This tells the model where each patch is located in the image.

4.3 Class Token

A special learnable token called [CLS] token is added at the beginning of the sequence.

After passing through transformer layers, the output corresponding to this token is used for classification.

4.4 Multi-Head Self-Attention

This is the most important part of ViT.

Each patch attends to every other patch in the image.

Self-attention calculates:

$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^T / \sqrt{d})\, V$$

Where:

- Q = Query
- K = Key
- V = Value
- d = dimension scaling factor

This allows the model to capture global relationships between patches.

4.5 Transformer Encoder Block

Each encoder block consists of:

1. Layer Normalization
2. Multi-head Self-Attention
3. Residual Connection
4. Layer Normalization

5. MLP (Feed Forward Network)
6. Residual Connection

We stacked multiple such layers (depth = 6).

4.6 Final Classification

After passing through all encoder layers:

- We take the CLS token output
- Apply Layer Normalization
- Pass through a Linear layer
- Output 10 class logits

5. Hyperparameter Tuning

We experimented with multiple hyperparameters.

5.1 Learning Rate

Tried:

- 1e-3
- 3e-4
- 1e-4

Best performance was obtained with:
Learning Rate = 3e-4

Too high learning rate caused unstable training.
Too low learning rate made training slow.

5.2 Depth (Number of Layers)

Tried:

- 4 layers
- 6 layers
- 8 layers

6 layers gave best balance between:

- Accuracy
- Training time
- Overfitting

5.3 Number of Attention Heads

Tried:

- 4 heads
- 8 heads
- 12 heads

8 heads worked well for CIFAR-10.

5.4 Optimizer

Used AdamW because:

- It decouples weight decay
- Works better for transformers
- Provides stable training

5.5 Learning Rate Scheduler

Used Cosine Annealing Scheduler.

This gradually reduces learning rate in a cosine pattern, helping the model converge smoothly.

6. Results

After 20 epochs of training:

- Training Accuracy ≈ 92–94%
- Test Accuracy ≈ 85–88%

Regularization significantly improved generalization.

Without augmentation:

- Test accuracy dropped below 80%

With augmentation + weight decay:

- Test accuracy improved by ~7–8%

This shows that Vision Transformers require strong regularization, especially on smaller datasets like CIFAR-10.

7. Observations

1. ViT requires more data compared to CNNs.
2. Without augmentation, model overfits quickly.
3. Attention mechanism captures global context effectively.
4. Computational cost is higher than CNN.

5. AdamW optimizer performs better than SGD for transformers.

8. Conclusion

In this project, we successfully implemented a Vision Transformer model from scratch and trained it on the CIFAR-10 dataset.

We studied:

- Patch embeddings
- Self-attention mechanism
- Transformer encoder layers
- Role of CLS token
- Importance of positional embeddings

We also experimented with:

- Data augmentation
- Dropout
- Weight decay
- Label smoothing
- Learning rate scheduling

Through hyperparameter tuning and regularization, we achieved strong classification performance.

This project helped in understanding how transformer architectures can be applied to computer vision tasks and how attention mechanisms enable global feature learning.