Vision Transformers (ViT)

Applying Vision Transformer to Brain Tumor MRI Classification

Ayobami Adebesin

Introduction

What is a Vision Transformer?

- It’s a deep learning model that uses the Transformer architecture for vision problems.
- Introduced in 2020 by Google Researchers in the paper “*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*”.
- Competes or outperforms traditional CNNs (like ResNet) on many benchmarks.
- ViT uses self attention for global context modelling.
- Traditionally data-hungry, but fine-tuning on smaller datasets shows promise.

How does it work?

- Images are split into small patches (like 16x16 pixels).
- Each patch is flattened and treated like a word token in a sentence.
- These tokens are fed into a transformer encoder, just like in NLP.
- It uses learned positional encoding as opposed to fixed or sinusoidal positional encodings commonly used in traditional transformers.
- The model learns spatial patterns and features from the sequence of patches.

Image Patching

Assuming we have an image of size, $\mathbb{R}^{H \times W \times C}$, where

H - Height

W - Width

C - Number of Channels (3 for RGB)

We divide this image into a grid of non-overlapping square patches of size P x P, so that the number of patches N is

$$N = \frac{H}{P} \times \frac{W}{P}$$

Each patch will be of size $P \times P \times C$

Linear Projection

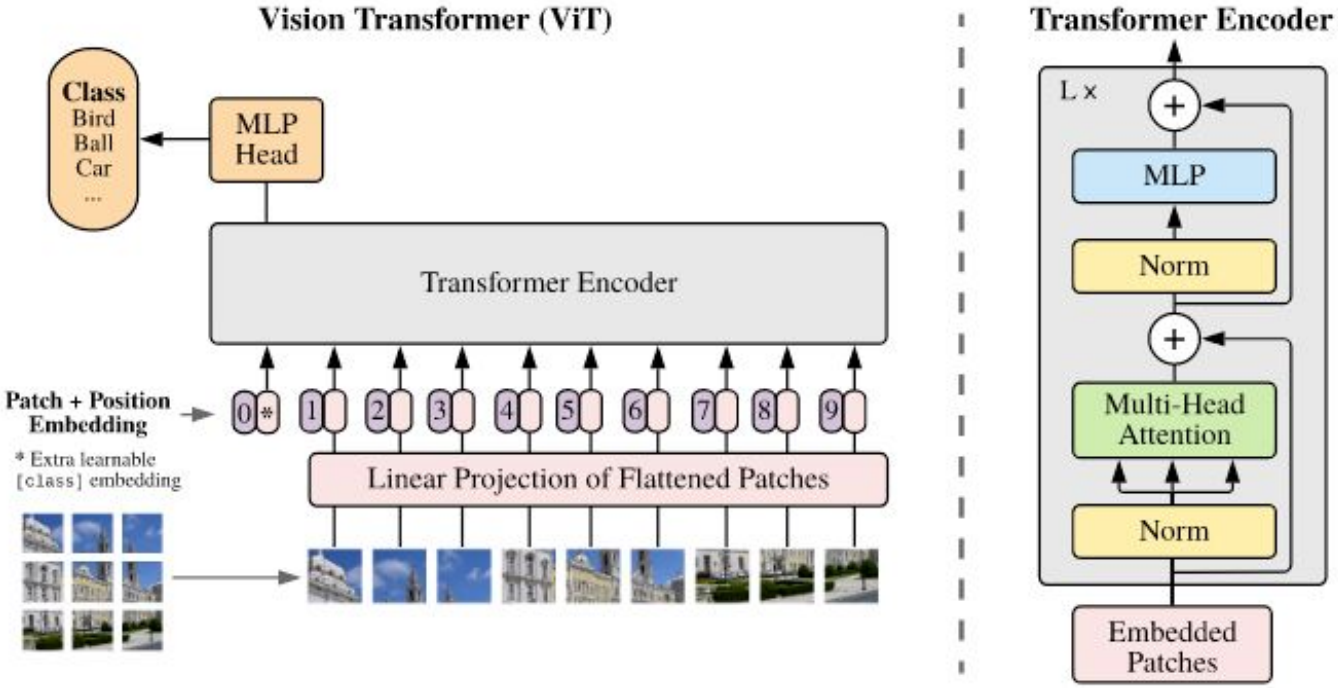
Each patch is then flatten into a vector of size $P^2 \cdot C$

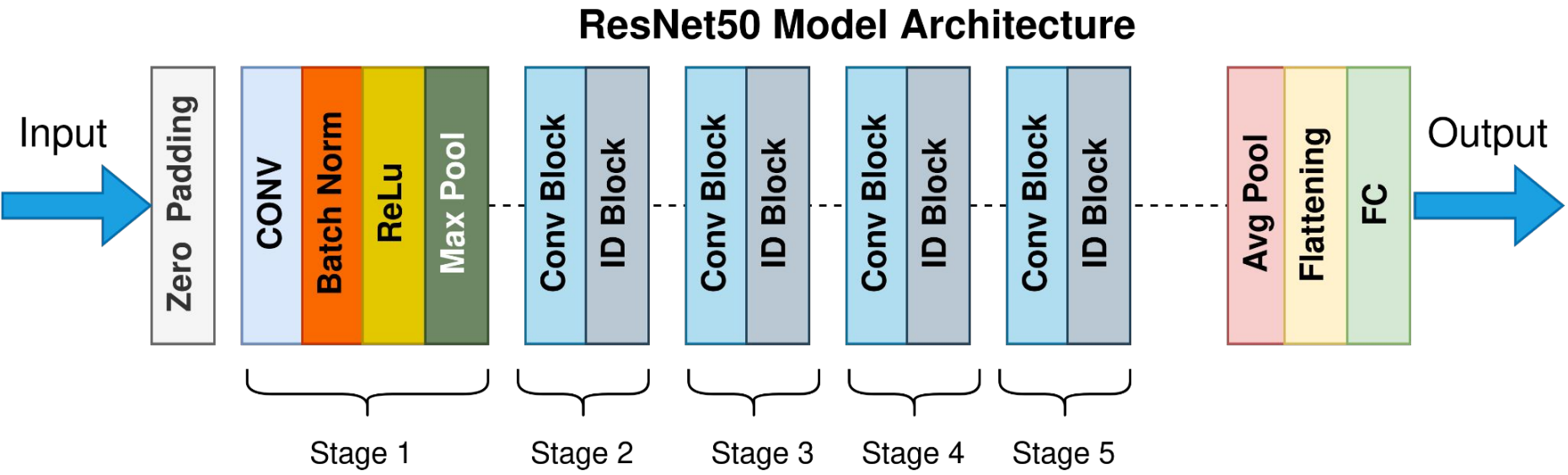
This vector would then be projected linearly using a trainable matrix

$$E \in \mathbb{R}^{(P^2 \cdot C) \times D}$$

where D is the transformer's embedding dimension.

A learned positional embedding $E_{pos} \in \mathbb{R}^{N \times D}$ is added to the projection to inject spatial information.


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP
Head
Transformer Encoder
Patch + Position
Embedding
* Extra learnable
[class] embedding
0*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head
Attention
Norm
Embedded
Patches


ResNet50 Model Architecture
Input
Zero Padding
CONV
Batch Norm
ReLu
Max Pool
Conv Block
ID Block
Conv Block
ID Block
Conv Block
ID Block
Conv Block
ID Block
Avg Pool
Flattening
FC
Output
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5

Objectives

- Implement and finetune a Vision Transformer (ViT) model for a medical image classification.
- Compare its performance on a limited medical imaging dataset to a baseline CNN architecture; ResNet50.

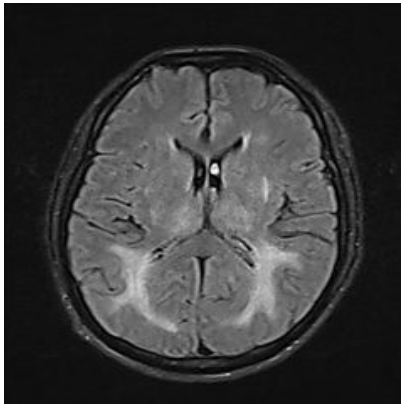
Problem Statement

Can ViT outperform traditional CNN's on small medical datasets with proper fine-tuning?

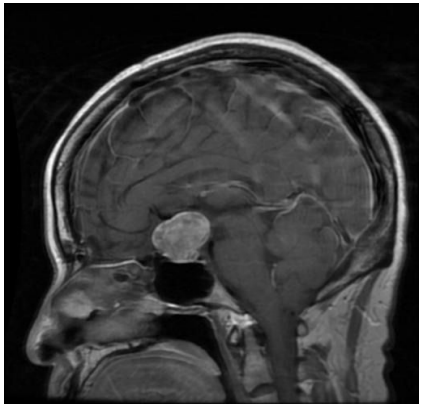
Dataset Overview

- Source - Hugging Face (Hemg/Brain-Tumor-MRI-Dataset)
- 7000+ MRI Images containing 4 classes of brain tumors:
  - Glioma - 23.1%
  - Meningioma - 23.4%
  - Pituitary - 25%
  - No Tumor - 28.5%
- Preprocessing:
  - Resize to 224 x 224 pixels
  - Normalization using ViT/ResNet-specific parameters
- Split the dataset into 90% training set and 10% validation set

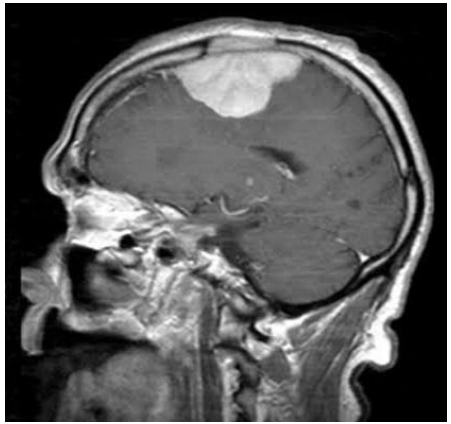
Dataset Overview



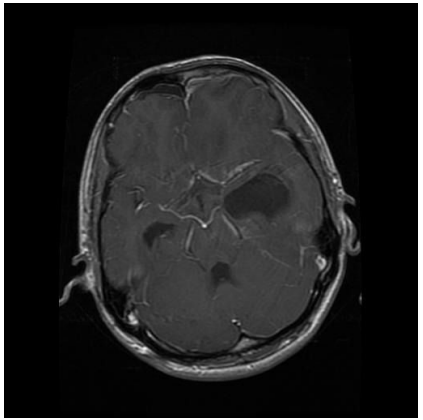
No Tumor



Pituitary



Meningioma



Glioma

Methodology

Vision Transformer Workflow

- Architecture
  - Pretrained **google/vit-base-patch15-224-in21k** model
  - Patch Embeddings ⇨ Transformer Encoder ⇨ Classification head
  -
- Fine-Tuning
  - Replace classification head for 4 classes
  - Training - AdamW optimizer (a variant of Adam optimizer with weight decay), 2e-5 learning rate, 10 epochs
  - Batch size - 16

ResNet50 Workflow

- Architecture
  - Pretrained **microsoft/resnet50** model
  - Convolution layers ⟹ Global Pooling ⟹ Classification head
- Fine-Tuning
  - Replace classification head for 4 classes
  - Training - 3e-4 learning rate, 15 epochs
  - Batch size - 16

Training Setup and resources

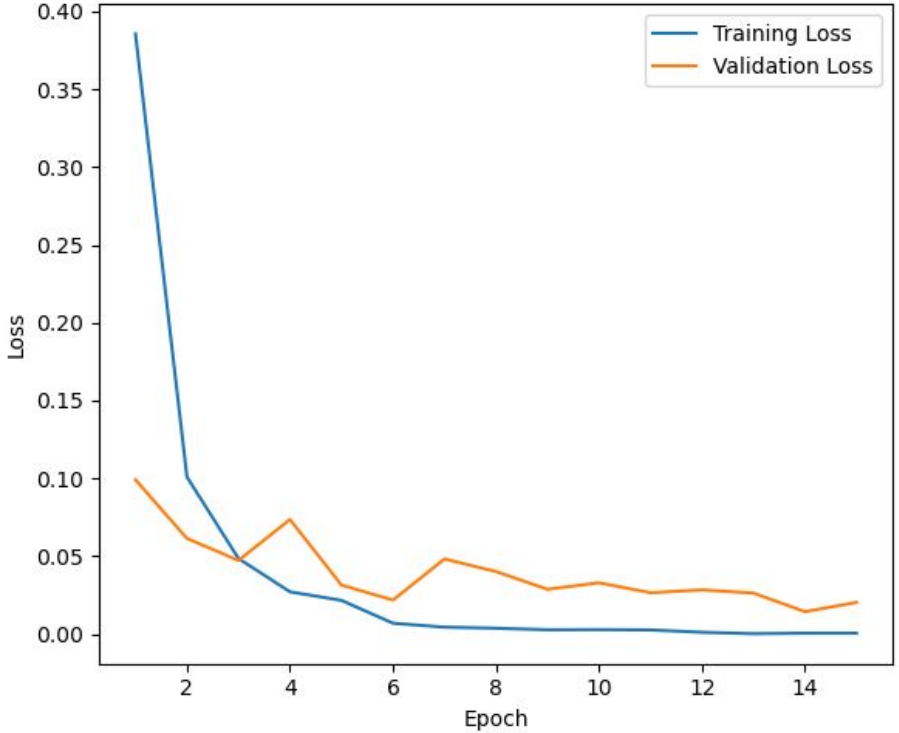
- Libraries: Hugging Face Transformers, PyTorch, Datasets
- Augmentation:
  - ViT: Random resized crops, horizontal flips
  - ResNet: Standard CNN augmentations
- Hardware: 16 Core GPU machine with MPS acceleration

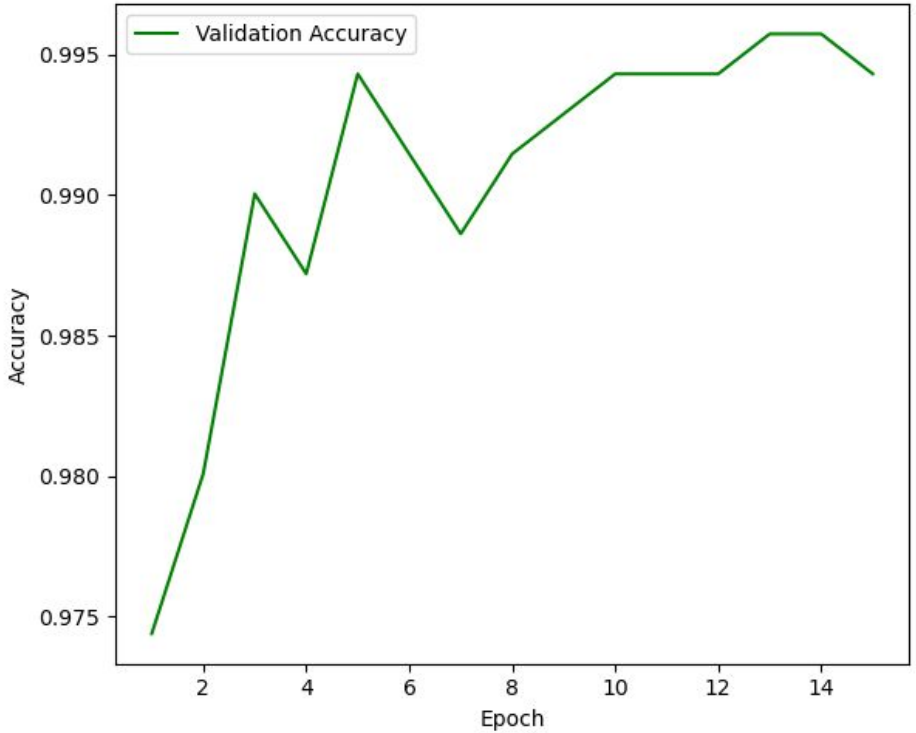
Results

Metrics

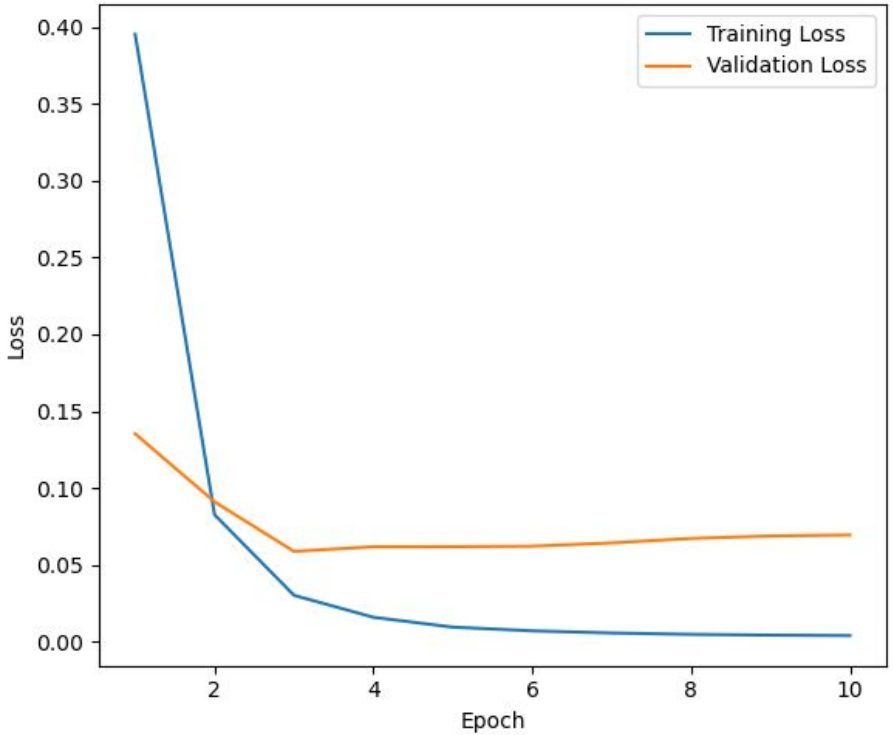
Model	Val Accuracy	Training Time	Avg. inference time (10 runs)
ViT	98.5%	35 mins	0.02 secs
ResNet50	99.4%	31 mins	0.006 secs

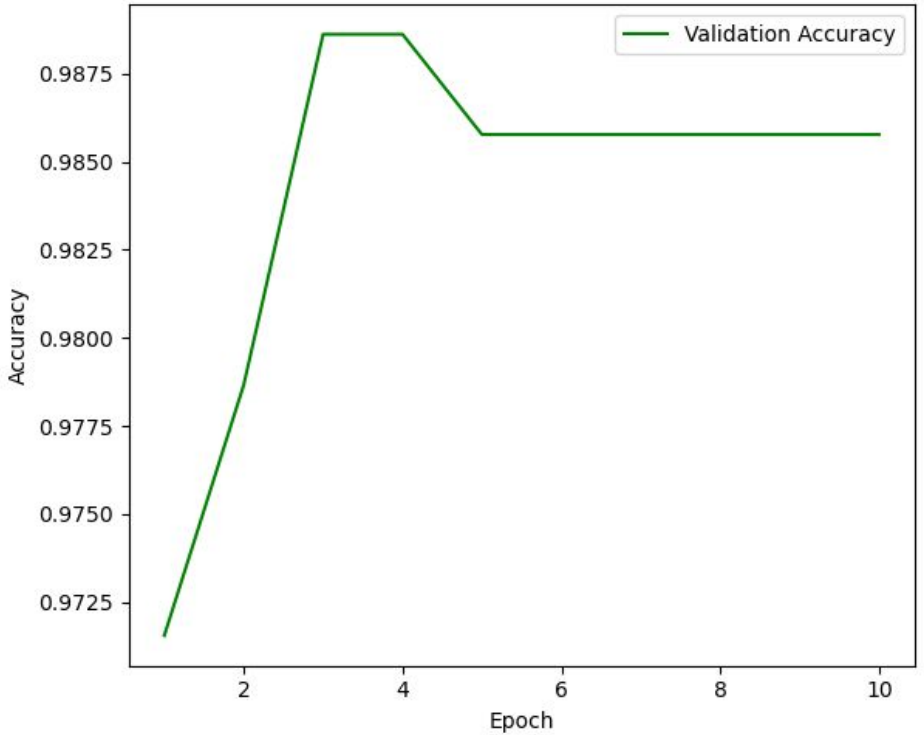
Training history - ResNet50





Training history - ViT





Summary

When to choose ResNet vs ViT

Criteria	**ResNet**	**ViT**
Small datasets	Better	Needs careful tuning
Training speed	Faster on small images	Slower
Memory usage	More efficient	Higher requirements
Transfer learning	Better for smaller domains	Better for diverse domains
Positional awareness	Built-in via conv layers	Requires explicit positional encoding

Conclusion

- ViT achieves a 98.5% accuracy on medical data without convolution layers.
- Accuracy can be improved with smaller batch size.
- ViT’s self-attention provides interpretability for clinical use.

QUESTIONS?
COMMENTS?

THANK YOU!