Enhanced Multi-Grade Breast Cancer Classification Using Vision Transformer

1. Introduction

Breast cancer is one of the most common and deadly cancers among women globally and early and accurate detection can be crucial for improving lives , as it significantly increases the chances of successful treatment and survival Yet, the diagnostic process is often time-consuming, subject to human error, and reliant on the expertise of highly trained radiologists. Our solution: a Vision Transformer-based model, that is designed to assist healthcare professionals in detecting breast cancer with greater accuracy, speed, and efficiency. our project aims to automate the classification of breast cancer images into categories (benign, malignant, and normal), which can assist radiologists and clinicians in diagnosing cancer more efficiently and accurately as it can reduce the rate of false negatives and false positives which in turn improves the diagnostic accuracy and aides in making informed clinical decisions.

This project leverages a Vision Transformer (ViT) architecture to classify breast cancer images into three categories: benign, malignant and normal. The data is split into training and testing data/inherits from a directory and PyTorch is used for both implementing the model and training. The implementation comprises data preprocessing, patch embeddings, Transformers, and classification.

2. Code Walkthrough

**2.1 Environment Setup Dependencies**:

We use Python 3.10 together with PyTorch and proprietary torchvision for data processing and matplotlib for visualization. Device Setup: Some of the checks include checking whether a GPU is being used using a command on the torch.device(“cuda”).

**2.2 Data Preprocessing Dataset Handling:**

Data is loaded with torchvision.datasets.ImageFolder, images parsed with torchvision.transforms.Resize now at 256\*256 224 × 224 224×224 pixels. Transformations: This was done using a transforms.Compose pipeline: Resizing images. Converting images to tensors. DataLoaders: DataLoader is used to forming the batches with batch size equal to 32.

**2.3 Patch Embedding:**

The model divides each 224 × 224 such that it is in the form of 224×224 image into several patches of size 16 × 16 16×16, which are converted into a sequence of learnable embeddings: Patch Extraction: nn.conv2d describes the creation of features through considering the patches as convolution results. Flattening: A nn.Flatten layer is then used to reshape the patches into a 1D form of sequence. Output: Each image is converted into 196 196 patches with the feature size of the 768.

**2.4 Vision Transformer Architecture Embedding Preparation:**

To final embeddings of patches, a learnable class token is appended. Positional information is also incorporated by introducing learnable position embeddings. Transformer Encoder: Multi-head Self-Attention: Stores dyads of patches. MLP Block: An F³ layer using feedforward with a non-linear activation function (GELU). Residual Connections: Applied for the mechanism of the self-attention block and MLP block. Final Classification: Outputs logits for each class through a fully connected layer, which is nn.Linear.

**2.5 Training Optimizer:**

The chosen model is Adam optimizer with the learning rate equal to 3 × 1 0 − 3 3×10 −3 . Loss Function: Cross-Entropy Loss the used the loss function inmulti-class classification problem. Training Loop: Trains for 10 epochs. Records training and testing losses and accurities.

**2.6 Evaluation Metrics:**

They all measure accuracy or loss on a test set have the model. Prediction: A helper function (pred\_and\_plot\_image) allows showing the predictions on different custom test images.

3. Results Training Performance:

Decreasing loss was observed through epochs and more and more accurately. Test Performance: The first generations displayed lower test accuracy, approximately equal to 15%. The accuracy was much higher in the later epochs as we moved from the first one (~61% in epoch two).

4. Insights Strengths:

As a transformer-based initialization method, Vision Transformer achieve competitive performance on visual tasks. It also conducts a detailed treatment of position information and the interactions between patches. Challenges: The training of Vision Transformers is computationally intensive as shown above. The data may require data augmentation to improve the probabilities of generalization into the test data set.

5. Conclusion:

The work completed a Vision Transformer for classifying breast cancer. Future work could involve: Applying data augmentation into our propose system. The next step is to adjust regulizers for improved accuracy of the model. Here it applies the transfer learning with the pre-trained ViT models.