**Classification of Bleeding and Non-Bleeding WCE Frames**

**Introduction:**

Wireless Capsule Endoscopy (WCE) is a critical technology for diagnosing gastrointestinal bleeding, generating between 60,000 to 1,00,000 images during an 8–12hour session. Reviewing these images is a labour-intensive process that typically requires 2-3 hours of meticulous frame-by-frame analysis by experienced gastroenterologists. This method is not only time-consuming but also prone to human error, particularly given the limited number of specialists relative to the global patient population. To address these challenges, we propose a novel approach that harnesses advanced Artificial Intelligence (AI) models for the efficient and accurate analysis of WCE videos. Our solution integrates cutting-edge neural networks including the Swin Transformer, DaViT, and Focal Net for the initial classification of frames potentially showing bleeding. For more precise localization of bleeding within these frames, we employ the RT-DETR model. This combination of AI technologies aims to enhance the accuracy, efficiency, and generalizability of bleeding detection in WCE, potentially reducing the workload on gastroenterologists and improving diagnostic outcomes.

**Data Preprocessing and Augmentation:**

The dataset consists of 1310 frames for bleeding and 1310 frames for non-bleeding, with each frame having a resolution of 224x224. The dataset was split into three subsets: 1047 frames for training, 262 frames for validation, and 1 frame for testing. A series of image enhancement techniques were applied, including conversion of images to the Lab color space, application of Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance contrast, and use of Gaussian blur to overcome noise. Furthermore, data augmentation methods were employed, including various affine and perspective transformations, rotations, flips, and Gaussian blur. Additionally, a “mix-up” augmentation was used, which combines two random frames in a blend weighted by a random variable following a beta distribution.

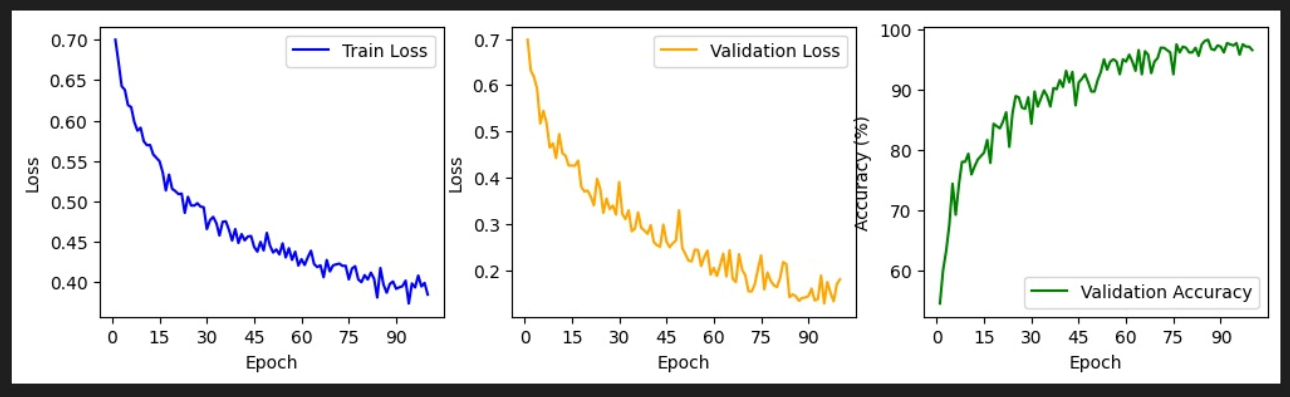
**Training and Results:**

We train our model with batch size of 8 by applying random transformations mentioned above over 100 epochs. We used AdamW optimiser and Binary Cross Entropy as loss function. We saved odd epoch models, for testing, we tested with the best model obtained by looking at accuracy and loss plots. We finetune our model parameters to get better results.

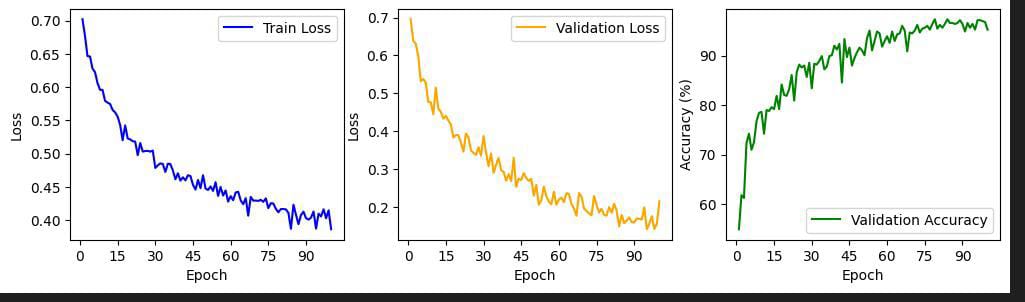
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.No** | **Parameter** | **Swin** | **DaViT** | **Focal Net** |
| 1 | Image size | 224 | 224 | 224 |
| 2 | Num classes | 2 | 2 | 2 |
| 3 | Patch size | 4 | 4 | 4 |
| 4 | In Channels | 3 | 3 | 3 |
| 5 | Embed dims | 96 | (96,192,384,768) | 96 |
| 6 | Depths | (2,2,6,2) | (1,1,3,1) | (2,2,6,2) |
| 7 | Heads | (3,6,12,24) | (3,6,12,24) | - |
| 8 | Window size | 7 | 7 | (7,5,3,1) |
| 9 | Mlp ratio | 4 | 4 | 4 |
| 10 | Stochastic depth prob | 0.2 | 0.1 | 0.1 |
| 11 | Learning rate | 7.8135e-6 | 7.8e-7 | 7.8135e-6 |
| 12 | **Accuracy** | 95.45 | 90.9 | 88 |

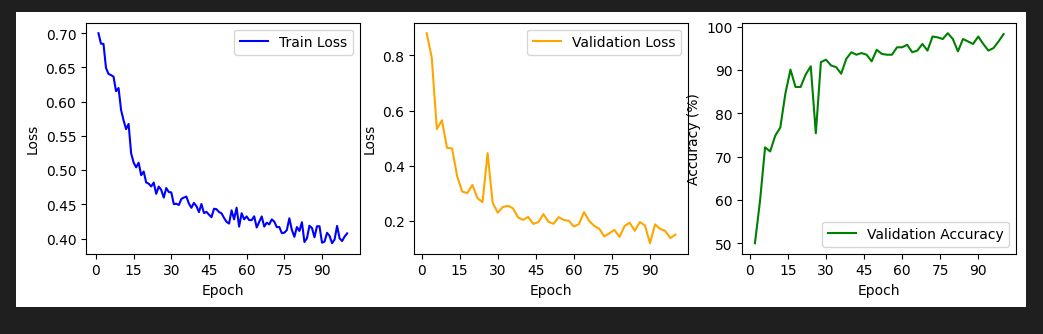
**Plots:**

Swin plots:



Davit plots:

Focal Net plots:



**Architectural Details:**

**Attention Mechanism in Transformers:**

Standard **qkv** self-attention (SA, Vaswani et al. (2017)) is a popular building block for neural architectures. For each element in an input sequence **z** ∈ R*N*×*D*, we compute a weighted sum over all values **v** in the sequence. The attention weights *Aij* are based on the pairwise similarity between two elements of the sequence and their respective query **q***i* and key **k***j* representations.

[**q***,***k***,***v**] = **ZU***qkv* **U***qkv* ∈ R­­­­­*D*×3*Dh,*

*A* = softmax *A* ∈ R*N*×*N,*

SA(**z**) = *A***v***.*

Multihead self-attention (MSA) is an extension of SA in which we run *k* self-attention operations, called “heads”, in parallel, and project their concatenated outputs. To keep compute and number of parameters constant when changing *k*, *Dh* (Eq. 5) is typically set to *D/k*.

MSA(**z**) = [SA1(*z*);SA2(*z*);··· ;SA*k*(*z*)]**U***msa* **U***msa* ∈ R*k*·*Dh*×*D*

**Swin:**

* The proposed Swin Transformer builds hierarchical feature maps by merging image patches in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window. It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks.
* In contrast, previous vision Transformers produce feature maps of a single low resolution and have quadratic computation complexity to input image size due to computation of self-attention globally.
* Swin Transformer is built by replacing the standard multi-head self-attention (MSA) module in a Transformer block by a module based on shifted windows, with other layers kept the same. A Swin Transformer block consists of a shifted window based MSA module, followed by a 2-layer MLP with GELU nonlinearity in between. A Layer Norm (LN) layer is applied before each MSA module and each MLP, and a residual connection is applied after each module
* The shifted window strategy in the Swin Transformer involves computing self-attention within local, non-overlapping windows of an image. In successive transformer blocks, the windows are shifted by half their size in both the horizontal and vertical directions. This shift enables cross-window connections that enhance the ability to capture relationships across different parts of the image.

**DaViT (Dual Attention Vision Transformers):**

* DaViT contains two transformer blocks: spatial window self-attention and channel group self-attention blocks. By alternately using the two types of attention, DaViT enjoys the benefit of capturing both local fine-grained and global image-level interactions.
* Spatial window self-attention operates by dividing an image into evenly distributed, non-overlapping windows. Within each window, the attention mechanism (computing query, key, and value matrices) is applied locally, restricting interactions to within the window. This local focus significantly reduces the computational complexity compared to global self-attention since it only processes a fraction of the image at any one time. As a result, this method manages to be computationally efficient while capturing local contextual information within each window.
* It divides the channels into multiple groups, each with a designated number of channels, and performs self-attention within these groups. This approach transforms the data such that each channel group processes global information, thus reducing computational complexity while still capturing important global features across the grouped channels. The method effectively allows the model to scale to different image sizes by maintaining consistent operations across varying spatial dimensions.

**Focal Net:**

* The Focal Modulation Network (FocalNet) architecture builds upon the concepts of self-attention, MLP architectures, and convolutions to efficiently capture both short- and long-range contextual information in visual data.
* FocalNet captures contextual information at multiple levels of granularity

**Focal Modulation (FM):**

* Focal modulation aggregates context features via shared operator (eg.depth-wise convolution) at each location and then query interacts with aggregated features

**Context Aggregation:**

* Hierarchical contextualization involves projecting the input feature map into a new space and applying a series of depth-wise convolutions to capture contexts at different levels of granularity.
* Gated aggregation condenses the contextual information from different levels into a modulator, which is then used for modulation.

### **Network Architecture:**

* FocalNet variants follow the stage layouts and hidden dimensions of Swin and Focal Transformers but replace self-attention modules with Focal Modulation modules.

**Relation to Other Architectures:**

* **Depth-wise Convolution:** Focal Modulation uses depth-wise convolution to capture the hierarchical contexts, which are then converted into modulator to modulate each query
* **PoolFormer**:Both Focal Modulation and poolformer extract the local context and enable the query-context interaction but in different ways (Pooling v.s. Convolution, Subtraction v.s. Modulation)

**The Real-Time Detection Transformer (RT-DETR):**

### **Hybrid Encoder Design:**

RT-DETR introduces an efficient hybrid encoder consisting of two modules: Attention-based Intra-scale Feature Interaction (AIFI) and CNN-based Cross-scale Feature Fusion (CCFF).

AIFI performs the intra-scale interaction only on last stage of backbone with the single-scale Transformer encoder

CCFF Inserts several fusion blocks consisting of convolutional layers into the fusion path

The fusion block contains two 1 × 1 convolutions to adjust the number of channels, N RepBlocks composed of RepConv are used for feature fusion, and the two-path outputs are fused by element-wise add,The role of the fusion block is to fuse two adjacent scale features into a new feature

**Uncertainty-minimal Query Selection:**

By explicitly modeling the uncertainty of encoder features, RT-DETR selects high-quality queries based on both classification and localization scores, enhancing the performance of the detector.

**Model architecture:**

We feed the features from the last three stages of the backbone into the encoder. The efficient hybrid encoder transforms multi-scale features into a sequence of image features through the Attention-based Intra-scale Feature Interaction (AIFI) and the CNN-based Cross-scale Feature Fusion (CCFF). Then, the uncertainty-minimal query selection selects a fixed number of encoder features to serve as initial object queries for the decoder. Finally, the decoder with auxiliary prediction heads iteratively optimizes object queries to generate categories and boxes

**Challenges faced in project:**

* The error we faced was version error.
* We also faced cuda out off memory error. We resolved it by restarting the system
* After resolving these silly errors. We observed that while testing both bleeding and non-bleeding shows only bleeding. We beat our heads for a week with this error. After that we observed that the validation loss and validation accuracy are looks fine. And the training is also good. We later found that we didn’t apply a crucial data augmentation step while testing.
* Later on accuracies were low,so we fine tune the model parameters,learning rate and weight decay valuesto get better accuracies.