Deep learning report

Student Vu Tuan Truong
ID 20225535

1 Introduction

The assignment involves participating in the BKAI-IGH-NeoPolyp Kaggle competition with a task of image segmentation in medical field. The main goal is to develop and train a U-Net model to achieve a minimum score of 0.7, with the work consistently attaining a score of approximately 0.74.

2 Transformation Techniques

To improve the model's robustness and generalization, I applied various image transformation and augmentation techniques. These were implemented using the *Albumentations* library, which ensures consistent augmentation of both images and their corresponding masks, preserving the integrity of the training dataset.

The transformations and augmentations used during the training phase are outlined below:

- **Resize**: Resize the image to 256 x 256, ensuring uniformity for model input.
- **Horizontal flipping, Vertical flipping and 90° rotate**: Randomly flips images horizontally and vertically and rotates them by 90°, each with a probability of 0.5, adding variability to object orientations in the dataset.
- **Random gamma**: Applying gamma correction to the images with a limit of 70% to 120% of the original image 20% of the times, simulating different lighting condition.
- **RGB shift**: Randomly shift the RGB value of the images by 10, with a probability of 0.3, introducing color variation to improve model robustness.
- **Normalization**: Scales pixel values to have a mean of (0.485, 0.456, 0.406) and standard deviation of (0.229, 0.224, 0.225), consistent with the pretrained ResNet34.
- **ToTensor**: Convert the image to tensor, the correct input format of the model.

3 Model Architecture

The segmentation model is built on the U-Net architecture, enhanced with ResNet34 serving as the encoder for feature extraction. This combination takes advantage of ResNet34's powerful ability to capture deep features, improving the model's performance in segmenting complex patterns.

3.1 U-net with ResNet34 Encoder

The two primary components of U-net are as below:

- **Encoder (ResNet34)**: ResNet34, a 34-layer residual network pretrained on ImageNet, serves as the feature extractor in the U-Net. It efficiently captures multi-scale features through its residual connections, enabling deep representation learning while avoiding gradient vanishing issues. The encoder's outputs at various stages are passed to the decoder via skip connections, preserving spatial details for precise segmentation.

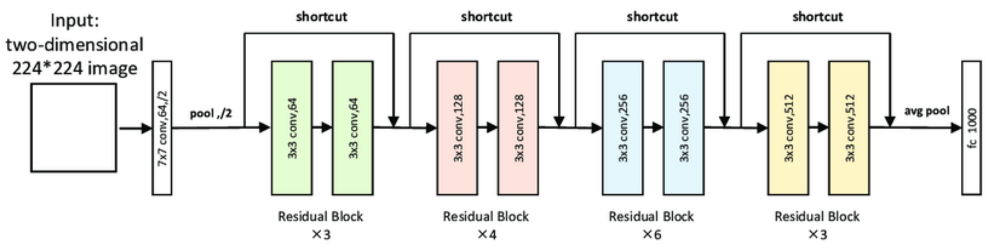


Figure 1: ResNet34 architecture

- **Decoder**: The decoder is responsible for reconstructing the segmented output from the features extracted by the encoder. It do this by applying upsampling operations, specifically transpose convolution and encorporate skip connections from the corresponding encoder block. These techniques helps the model produce accurate and detailed segmentation maps.

3.2 Loss function

To optimize the model, I used a combination of the **Dice Loss** and **Cross-Entropy Loss**, designed to handle class imbalance and segmentation accuracy:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2 \cdot \sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2},$$

where p_i and g_i are the predicted and ground truth values for pixel i, and N is the total number of pixels.

The total loss is computed as:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Dice}} + \mathcal{L}_{\text{CE}},$$

where $\mathcal{L}_{\text{CE}}$ is the Cross-Entropy Loss.

3.3 Model Configuration

The following configurations were used for the model:

- **Input image size**: All images and masks were resized to 256 x 256
- **Input channels**: 3 input channels corresponding to RGB images
- **Output classes**: 3 output classes corresponding to the colors in mask. 0 (black) for background, 1 (red) for neoplastic polyp and 2 (green) for non-neoplastic polyp.
- **Optimizer**: Adam optimizer was used with a learning rate of 0.0001.

4 Epirical results

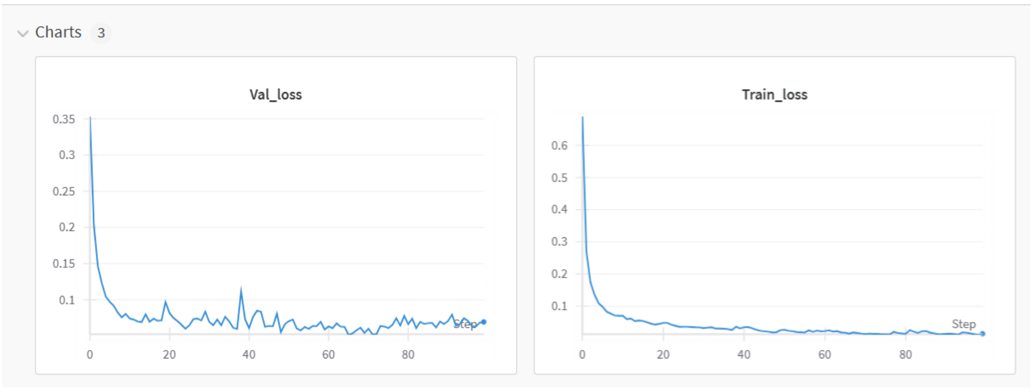
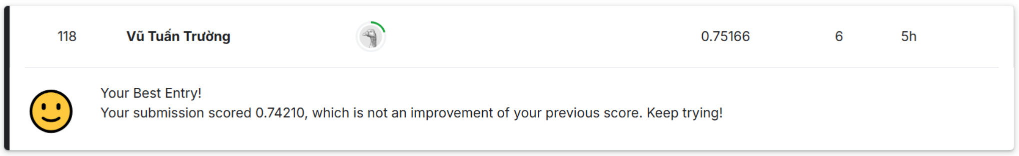


Figure 2: Training losses

The results were recorded in wandb during training, above is a run that achieve the minimum validation loss of 0.052 and 0.74 point in the Kaggle competition



118 Vũ Tuấn Trường 0.75166 6 5h

Your Best Entry!
Your submission scored 0.74210, which is not an improvement of your previous score. Keep trying!

Figure 3: Kaggle result

github link: `https://github.com/Frostedpilot/BKAI-IGH-Competition`