Deeplearning report

Nguyen Nhat Minh- MSSV: 20225510

November 2024

1 Introduction

This project resolves task of image segmentation in medical field, with dataset from BKAI-IGH NeoPolyp, sourced from Kaggle. The primary objective of this project is to practice how to build U-net in Deep learning course and fine-tune technique to get at least 0.7 score in leaderboard of Kaggle, specifically, this work can reach 0.7780.

2 Transformation Techniques

To enhance the robustness and generalization of the model, I use a series of preprocessing transformations applied to the training datasets. The transformations were implemented using the `Albumentations` library.

The following augmentations were used during the training phase:

- **Horizontal and Vertical Flipping**: Randomly flips the images horizontally and vertically with a probability of 0.5 each, introducing variability in the orientation of objects in the dataset.
- **Random Brightness and Contrast Adjustment**: Modifies the brightness and contrast of images randomly with a probability of 0.2, helping the model adapt to varying lighting conditions.
- **Random Cropping**: Crops the images to a size of 256×256 pixels with a probability of 0.5, introducing spatial variations and increasing robustness.
- **Gaussian Noise**: Adds random Gaussian noise to the images with a probability of 0.2, simulating noisy input data.
- **Random Rotation**: Rotates the images randomly within a range of $[-30°, 30°]$ with a probability of 0.3, enhancing the model's ability to recognize objects in rotated forms.
- **RGB Shifting**: Applies a random shift to the red, green, and blue channels in the range of $[-10, 10]$ with a probability of 0.3, mimicking lighting variations.
- **Normalization**: The pixel values were normalized to have a mean of $(0.485, 0.456, 0.406)$ and a standard deviation of $(0.229, 0.224, 0.225)$, consistent with the pretrained EfficientNet-B7 encoder.

3 Model Architecture

The segmentation model is based on the **U-Net** architecture, enhanced with **EfficientNet-B7** as the encoder for feature extraction. This combination leverages the strengths of U-Net's skip connections and EfficientNet-B7's efficient feature extraction.

3.1 U-Net with EfficientNet-B7 Encoder

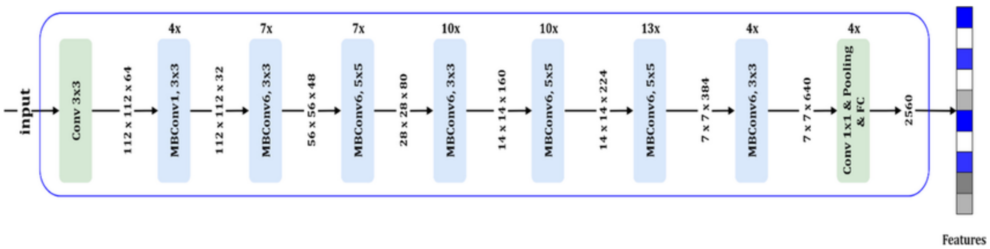

Figure 1 . Architecture of Efficient B7

The U-Net architecture consists of two primary components:

- **Encoder (EfficientNet-B7):** Pretrained on ImageNet, this encoder extracts hierarchical feature maps with varying spatial resolutions. EfficientNet-B7 utilizes compound scaling to optimize depth (d), width (w), and resolution (r) based on:

$$d = \alpha^{\phi}, \quad w = \beta^{\phi}, \quad r = \gamma^{\phi},$$

  where ϕ is the scaling factor, and α, β, γ are constants ensuring balanced scaling.

- **Decoder:** The decoder reconstructs the segmentation map by progressively upsampling feature maps and incorporating skip connections. The skip connections integrate encoder features F_i with decoder upsampled features U_{i+1}:

$$F_i' = \mathcal{U}(F_{i+1}') + F_i,$$

  where $\mathcal{U}$ represents an upsampling operation, ensuring spatial details are preserved.

3.2 Loss Function

To optimize the model, we use 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 configuration was used for the model:

- **Input Image Size**: All input images were resized to 256×256 pixels.
- **Input Channels**: The model accepts 3 input channels, corresponding to RGB images.
- **Number of Classes**: The output consists of 3 channels, each representing a segmentation class.
- **Optimizer:** Adam with a learning rate of 10^{-4}.

4 Epirical results

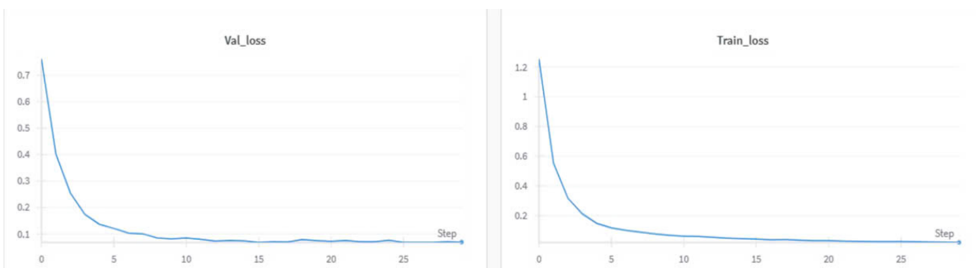

Figure 2 . Training loss

The result in Fig 2 is recorded in wandb training, with the minimum validation loss of 0.071, and reach top 75 of leaderboard in Kaggle competition.3

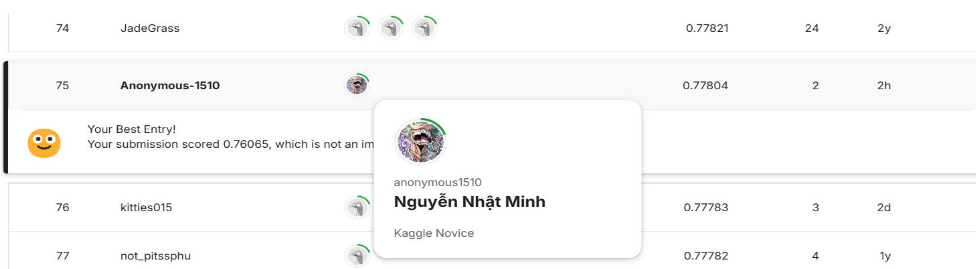

Figure 3 . Result in leaderboard

github link: github/Helooeverybody