# POLYP DETECTION EXERCISE REPORT

Ngo Duy Dat – 20225480

## Overview

The highest score I achieved on this contest is about 0.73 with the model used here is Unet++ from library pytorch.segmentation\_models.

Here my github repository: <https://github.com/nzd1001/Polyp-Detection.git>

## Image preprocessing and augmentation

All training images are resized to (256,256) to ensure consistency. Furthermore, here are some augmentation techniques that I use:

* + **Horizontal and vertical flip**: randomly flip the image horizontally and vertically
  + **RandomGamma**: randomly adjust the brightness of the image
  + **RBGShift**: randomly adjust values of RGB channels by a small amount
  + **Normalize**: normalize the pixels based on ImageNet standards

## Model Training

Here I use holdout technique to train and evaluate the model. The loss function used here is DiceLoss and the optimizer is Adam. The learning rate at the beginning is 0.001 and a scheduler is used to automatically decrease it if the train loss does not decreased after 5 epochs.

### 3.1. Unet++

Unet++ is an enhanced version of traditional Unet model. In Unet, at each resolution level, feature map from encoder will be directly passed to decoder (skip connection). On the other hand, Unet++ introduces additional intermediate convolutional layers to refine the feature maps from the encoder before they goes to the decoder. This helps capture finer details and improves segmentation of small or complex structures.

Here, I use resnet34 as the architecture of the encoder with pre-trained weight from ImageNet dataset .

### 3.2. Result

A graph with blue lines

Description automatically generated

Figure 1. Train loss

A graph with blue lines

Description automatically generated

Figure 2. Validation loss

## Copiable scipts

git clone <https://github.com/nzd1001/Polyp-Detection.git>

cd Polyp-Detection

python3 infer.py --image\_path image.jpeg