***1. The transformation techniques I have used:***

```python

# Update transform thêm Augmentation: https://albumentations.ai/docs/getting\_started/mask\_augmentation/

# Nhiều hơn: https://albumentations.ai/docs/api\_reference/full\_reference/

train\_transform = A.Compose(

    [

    A.Resize(256, 256, interpolation=cv2.INTER\_LINEAR),       # kiểu numpy array,

    A.VerticalFlip(),

    A.HorizontalFlip(),  #  mặc định p=0.5

    A.OneOf([

    A.Blur(blur\_limit=3, p=0.3),   # làm mờ, độ mờ ngẫu nhiên từ 0-3

    A.GaussianBlur(blur\_limit=3, p=0.3),

    A.MedianBlur(blur\_limit=3, p=0.3),

    A.RandomBrightnessContrast()  # độ tương phản hình ảnh, mặc định p=0.2

    ], p=0.3),

    A.Rotate(limit=30, p=0.3),

    A.RGBShift (r\_shift\_limit=20, g\_shift\_limit=20, b\_shift\_limit=20, always\_apply=False, p=0.5),

    # ngẫu nhiên dịch chuyển kênh màu từ 0-20

    A.Cutout (num\_holes=5, max\_h\_size=8, max\_w\_size=8, fill\_value=0, always\_apply=False),    # mặc định p=0.5

    # cắt đi 1 khoảng model vẫn học được

    A.GaussNoise(p=0.3),

    A.Normalize (mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225), max\_pixel\_value=255.0, always\_apply=False, p=1.0),

    # Mean and std of ImageNet (tập data rất lớn) là như trên phổ biến cho các model,

    ToTensorV2(),

]

)

- I used a series of augmentation, transformation techniques to create diversity in the training data and enhance the generalization ability of the model.

1. **\*\*Resize\*\*:** Resize the image to 256x256 pixels using linear interpolation method.

2. \*\***VerticalFlip\*\*:** Flip the image vertically with default probability of 0.5.

3. \*\***HorizontalFlip**\*\*: Flip the image horizontally with default probability of 0.5.

4. \*\***OneOf (Blur, GaussianBlur, MedianBlur, RandomBrightnessContrast)\*\*:** Randomly choose one of the following transformations:

- \*\*Blur\*\*: Blur the image with random blur from 0-3.

- \*\*GaussianBlur\*\*: Blurs the image using the Gaussian method with random blur from 0-3.

- \*\*MedianBlur\*\*: Blur the image using the median method with random blur from 0-3.

- \*\*RandomBrightnessContrast\*\*: Increase or decrease contrast and brightness randomly.

5. \*\***Rotate**\*\*: Rotate the image randomly between -30 and 30 degrees with probability 0.3.

6. \***\*RGBShift**\*\*: Randomly shifts color channels (red, green, blue) from 0-20 units with probability 0.5.

7. **\*\*Cutout**\*\*: Cut some holes into the image with the number of holes being 5, the maximum hole size being 8x8 pixels, and the fill value being 0, with a default probability of 0.5.

8. \*\***GaussNoise\*\*:** Adds Gaussian noise to the image with probability 0.3.

9. **\*\*Normalize\*\*:** Normalize the image according to the mean value and standard deviation of ImageNet with probability 1.0.

10. **\*\*ToTensorV2\*\*:** Convert image into tensor format.

***### 2.What I have changed in the model architecture:I have tested many Resnet models***

- Resnet-34:

- U-Net++ with ResNet-34: 83.3M parameters. Will be faster in training and inference due to lighter encoder architecture. However, you have to train for about 300 epochs to reduce the loss to 0.1, acc > 0.7 - U-Net++ with ResNet-50: 97.8M parameters

- U-Net++ with ResNet-101: 170M para. Using ResNet-101 as an encoder provides a balance between accuracy and speed compared to ResNet-152.

- U-Net++ with ResNet-152: 232M parameters. Train for about 100 epochs will reduce loss to 0.1, acc > 0.7

- This is a model with a very deep encoder, providing the ability to learn complex representations.

- Can achieve high accuracy but will be slower in training and inference.

- Requires more GPU memory resources and can be difficult to train without powerful enough hardware.

- U-Net++ with Attention ResNet-34: ...

=> Finally: I use the U-Net++ model with ResNet-101 as the encoder, as name U-net++101Resnet to train, val, infer model.(The U-Net++ model is an improved version of the U-Net model, with dense connections between layers and a nested structure to improve segmentation accuracy. Using ResNet-101 as the encoder provides a balance between accuracy and speed compared to ResNet-34 , ResNet-152).

4. Link to my github: