Method

We use Unet model for the image segmentation task, and the Unet model can be divided into two parts: encoder and decoder. Our goal is to find out the best backbone structure for the encoder. We decide to use image classification models for the backbone structures, because the model share the same process of convolutional steps for increasing receptive field, followed by a downsample step which reduces the feature map in each stage. Here we choose three models for our experiment: ResNet, DenseNet and EfficientNet. For each backbone model, we will use both randomly generated weights and pretrained weights to further find out whether pretrianed weights are beneficial. So, there will be total 6 training processes (3 models \* use pretrained weights or not) We use the same setting for each training process. The training settings is set as the following:

1. Epoch: 20
2. Data argumentation: GridDistortion>ElasticTransform>Affine>GaussNoise>Blur>Downscale>RandomBrightnessContrast'
3. Input Image preprocessing: CLAHE
4. Loss function: binary cross entropy loss \* 0.2 + dice loss \* 1
5. Optimizer: Adam, learning rate: 0.001
6. Schedule (two stage):
   1. Warmup stage: multiplier range from 0 to 1 linearly in the first epoch.
   2. Reduce stage: after the warmup stage, the multiplier is 1/(epoch count)\*\*2

Models

1. ResNet-50 (23M) is the most commonly used model and it’s our baseline.
2. DenseNet-201 (18M) can extract dense features from previous layers and this property maybe helpful for out medical image.
3. EfficientNet-b4 (17M) has high accuracy and relatively small numbers of parameters.
4. The subtype of the three models are chosen by their similar numbers of parameters.

Result

We compare the six training results through dice accuracy and visualization of the prediction. We find out that backbones with randomly generated weights tend to give a prediction of blur edge, but the dice accuracy is almost as high as the pretrained ones. Dice accuracy of backbones with pretrained weights ranking from high to low is EfficientNet >= ResNet > DenseNet.

Conclusion

Although the dice loss in both pretrained or randomly generated weights are similar, the blur edge of prediction from randomly generated weights is unreasonable. So, we think pretrained weights is necessary in the task. And the dice accuracy of EfficientNet and ResNet is close but EfficientNet has fewer parameters. As a result, EfficientNet-b4 with pretrained weights is the most suitable for the task among the three models.