**nnU-Net with Multiple Loss Ensembles for Aneurysm Segmentation**

Maysam Orouskhani, Chengcheng Zhu
Department of Radiology University of Washington, Seattle, USA

# Method

A 3D full resolution of U-Net [1] was proposed for brain Aneurysm segmentation. In the literature, working with imbalanced datasets is one of the most challenging issues. Since lesions often occupy a very smaller volume relative to the background, the model's prediction is biased towards low sensitivity. Therefore, to address the highly imbalanced dataset, three different loss functions were analyzed as follows: 1. Dice loss + Cross entropy, 2. Dice loss + TopK loss, and 3. Dice loss + Cross entropy + TopK loss. We employed the nnU-Net architecture, a self-configuring method for deep learning-based biomedical image segmentation. The model uses an encoding and decoding path where each path includes 5 convolution blocks, and each block comprises of 3*3*3 convolution layer. We also used the instance normalization layer and leaky rectified linear unit. Long skip connections are also used in the same resolution between down-sampling and up-sampling paths.

To train the nnU-Net model, we used the ADAM dataset which includes preprocessed and original images for each patient. However, only preprocessed images of patients are given to the model as input. nnU-Net also utilizes the cropping and Z-Score normalization for processing the images before feeding the model and employs data augmentation methods such as rotation and scaling. The ADAM dataset includes 113 cases. However, there is no aneurysm in some cases. As a result, we just used the cases with aneurysm included. So, we used 65 cases as training set and 24 cases for test data.

The nnU-Net uses SGD as the default optimization algorithm with an initial learning rate of 0.01. We run the model for a maximum number of 250 epochs. We also apply five-fold cross validation for the three different loss functions. We train all the models 3* RTX 3090 GPU with patch size of 256 × 224 × 56 and a batch size of 2, and CUDA version 11.6. Each fold took about 18 hours. Table 1 indicates the results of three loss functions for Dice, Sensitivity, and Precision. The results show that none of the loss functions achieve the best metrics among all five folds. Thus, we select the best-fold models for ensembles.

Table 1. Comparison of performance metrics including Dice, Sensitivity, and Precision for three loss functions

| Fold | Metric | Dice + CE | Dice + top k | Dice + CE + top k |
|---|---|---|---|---|
| **Fold_0** | Dice | **0.5205** | 0.4944 | 0.5033 |
| | Sensitivity | **0.5152** | 0.4715 | 0.5007 |
| | Precision | **0.6584** | 0.6694 | 0.6214 |
| **Fold_1** | Dice | 0.4672 | 0.4157 | **0.4684** |
| | Sensitivity | **0.4758** | 0.3871 | 0.4460 |
| | Precision | 0.6134 | 0.6850 | **0.6459** |
| **Fold_2** | Dice | **0.5407** | 0.5155 | 0.4710 |
| | Sensitivity | **0.5284** | 0.4831 | 0.4526 |
| | Precision | **0.6720** | 0.6530 | 0.6217 |
| **Fold_3** | Dice | 0.5052 | 0.5392 | **0.5291** |
| | Sensitivity | 0.5447 | 0.5438 | **0.5584** |
| | Precision | 0.6054 | 0.6747 | **0.7153** |
| **Fold_4** | Dice | **0.5661** | 0.5508 | 0.5109 |
| | Sensitivity | **0.55904** | 0.5267 | 0.4626 |
| | Precision | 0.7037 | 0.6928 | **0.7208** |