Pretraining and self-supervised finetuning for 3D light-sheet microscopy image semantic segmentation

Abstract

Light-sheet microscopy (LSM) is a valuable method in modern biological research fields such as neuroscience, immunology, oncology and cardiology. However，so far there is still no effective segmentation tool to perform automatic segmentation for analyzing large-scale or whole-organ LSM images. Here, we report on the implementation details of the methods submitted in the SELMA3D competition.

Method

Pretraining

Firstly, we use the unannotated dataset for pretraining. We use a classic network architecture Swin UNETR which is widely used in 3D medical image analysis. Before training the model, we implemented several preprocessing steps. (1) We scaled each 2D plane image down to 1/3 of its original width and height to optimize memory usage and converted the ‘tif’ images to ‘png’ format for compatibility with the *MONAI* Python package. (2) We created a sample dictionary, allowing for random sampling of continuous sets of plane images along the z-dimension. (3) A 3D patch was randomly selected from the 3D image, followed by normalization, cropped and padding (if the image was smaller than the required input size).

The model input size is configured to 64x64x64. Instead of the conventional inpainting transformation, we adopt a masked self-supervised learning strategy. In contrast to the original inpainting method, our approach directly sets certain regions to zero, elevating the pre-training challenge and encouraging the model to learn more robust features. The batch size was set

as 4 and the epoch was 100.

Finetuning

For fine-tuning, we utilized the annotated data, applying several preprocessing techniques: normalization, random cropping, padding, random flipping, random rotation, and random intensity variation. The batch size was set to 4, with training conducted 2000 epochs. We employed the DiceMetric from the MONAI Python package to assess performance on validation images, and applied a threshold to convert the model outputs into binary format. The best performing and final checkpoints on the validation set during training was saved for submission.