"MICCAI 2023 Semi-supervised Tooth Segmentation" IGIP-CBCT team competition introduction document

## **Innovative ideas:**

1. We constructed a new semi-supervised denoising framework through mutual learning. Based on the characteristics of the test set and labeled data set, we adopted a coarse-to-fine segmentation strategy. However, during the collaborative training of the two networks, it is easy to learn similar information and also subject to coupled noise interference. Therefore, we adopted a strategy of dividing the dataset into two different sub-datasets to enhance the diversity of information encountered by the input network.
2. We observed that during the learning process, the model's prediction results gradually become more accurate than the initial noisy pseudo-labels. Therefore, in the training process, in addition to the initial pseudo-labels, we introduced further network predictions as more reliable pseudo-labels and gradually increased the weight of loss constraints based on these pseudo-labels.
3. We have noticed that there are errors in the labeling of some areas in the labeled dataset (a problem that is hard to avoid with manual labeling), and for unlabeled data, the pseudo-labels generated by the network often contain noise. To address this issue, we introduce a noise transfer matrix for loss correction, effectively reducing the impact of noisy labels. Given that the regions most susceptible to errors are primarily located at the edges, we apply the noise transfer matrix to these edge areas to avoid unnecessary interference with internal regions.

## **Development process and method details:**

**1. The training process is divided into three stages:**

In the first stage, we used a 12-instance labeled dataset provided by the competition and selected 103 samples from an unlabeled dataset as training data. The model was trained using the traditional semi-supervised segmentation method Mean-Teacher[1] to obtain initial full-field segmentation results. Then, we tested all unlabeled data with the trained network to generate pseudo-labels for these unlabeled datasets. However, since the labeled dataset only contained partial field-of-view regions, we observed that the quality of the pseudo-labels obtained in this phase was poor, including the issue of incorrectly segmenting large background areas into foreground regions, known as noise scattering. Therefore, we conducted the second stage of training to improve the segmentation results.

In the second phase, our goal is to achieve more accurate background region segmentation results. To accomplish this, we merged the unlabeled data from the first phase with the generated pseudo-labels into a labeled dataset, creating a new dataset for the training in the second phase. Specifically, we used 12 samples from the labeled dataset and selected 186 samples from the unlabeled dataset. It is important to note that since the pseudo-labels generated in the first phase contained numerous scattered points, this phase employed a method of finding the largest connected area to remove these points. The specific procedure was as follows: once the network obtained the largest connected area, it filtered out the pseudo-labels using bounding boxes containing the largest connected area, retaining only those pseudo-labels within the boundaries of the largest connected area. Then, we used the processed data for the training in the second phase. Next, the trained network will test all the unlabeled datasets to further update the pseudo-labels of the unlabeled data.

In the third stage, to achieve more refined segmentation of the region of interest, we merged the unlabeled data from the second stage with the generated pseudo-labels into a labeled dataset to create a new dataset for training in the third stage. We conducted data screening on the unlabeled dataset and ultimately selected 281 unlabeled samples. In this phase, to achieve even finer segmentation, we only cropped the region of interest to input into the neural network.

In addition to the traditional Mean-Teacher architecture used in the first stage, the second and third stages use the same network architecture, namely our newly designed semi-supervised noise-resistant segmentation network. The overall network architecture is shown in Figure 1. Next, we will introduce this network in detail.

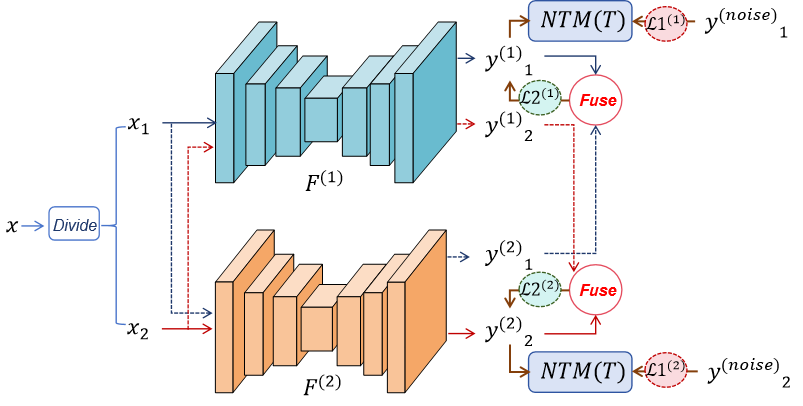


Figure 1 Network architecture

First, we adopted two parallel networks to enable them to learn from each other. However, during the collaborative training of these two networks, it is easy for similar information to be learned and for coupling noise to interfere. To address this issue, we divided the dataset into two mutually exclusive sub-datasets. These different sub-datasets were fed into two distinct networks to ensure that the networks learn more diverse features. Both networks can use noisy labels as initial supervision (here, we consider both the labels of the labeled dataset and the pseudo-labels of the unlabeled data as noisy labels). This process can be constrained by a loss function:

(1)

The first is represented as the network, and the second is represented as the network.

Then, we integrate the prediction results from both networks as further pseudo-labels and gradually increase the weight of these pseudo-labels. The purpose is that during the network learning process, the prediction results will become more accurate than the noise labels, and the integration of predictions from both networks will produce even more accurate pseudo-labels. The process of generating pseudo-labels by integrating the results from the fusion network is shown below:

(2)

And the loss function established by using this pseudo label is:

(3)

So the total loss function is summarized as:

(4)

Among them, it increases with the increase of the number of iterations. However, we use /2, which uses the common cross entropy loss function and Dice loss function, so it can be expressed as:

/2 = (5)

In addition, we noticed some errors in the labels of the labeled dataset. For unlabeled data, the pseudo-labels generated by network predictions are undoubtedly noisy. Therefore, we use a noise transfer matrix (noise transition matrix NTM) [3][4] for loss correction to reduce the interference from noisy labels. The noise transfer matrix represents the probability of clean labels of different class data points being converted into noisy labels. We assume that the generated noisy labels containing classes can be bridged to manually annotated Ground Truth through a noise transfer matrix (NTM), which indicates the probability of class flipping from Ground Truth labels to noisy labels, i.e.:

We observe that the edge regions of teeth are more prone to errors compared to the interior, making them more susceptible to noise labels. When applying the noise matrix, it is unreasonable to apply it to all areas; therefore, we design the noise transfer matrix in the edge regions to enhance its accuracy. Based on this process, formula (1) can be redesigned as:

(6)

+(1-) (7)

The mask is 1 at the edge and 0 in other areas, so the noise at the edge can be corrected for loss.

**Ii. Reasoning process:**

## In the reasoning process, we only used the network models from the second and third stages. Specifically, we first performed coarse segmentation using the second stage, then clipped the regions of interest and input them into the third stage network for more detailed segmentation. Additionally, we employed post-processing operations, including removing scattered points and filling voids along the cross-section.

## **Parameter setting and training techniques:**

1. Data processing: We compress the input data into the range of 0-1. In order to deal with the problem of overfitting, we adopt data enhancement techniques, such as random rotation and flipping.

2. During the test phase, we used test time data enhancement (TTA) to improve the prediction results, that is, the results after multiple flips were tested separately and integrated.

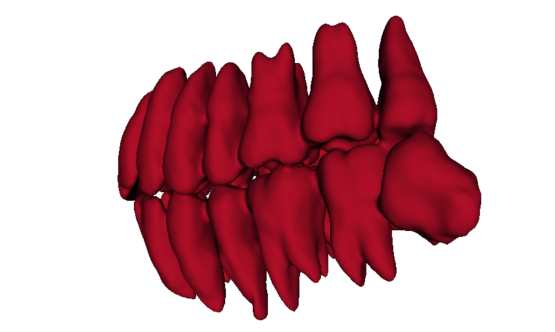
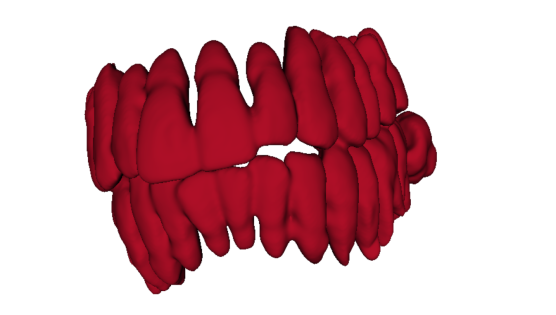
3. Our optimization method involves stochastic gradient descent (SGD), with an initial learning rate of 0.1, and the poly learning rate strategy is used for updating. The momentum is set to 0.9, and the weight decay coefficient is set to 0.0001. The network structure we use in each stage is 3DUnet[2].

4. Post-processing operations are adopted, including removing scatter points and filling voids along the cross section.

5. The loss function we use is mainly the traditional cross entropy loss function and Dice loss function, the details of which can be seen in the formula.

## **Results show:**

## **Preview of the prediction results**



## **Evaluation index results**

|  |  |  |  |
| --- | --- | --- | --- |
| Score | Dice | Iou | Hausdorff\_distance |
|  |  |  |  |

## **reference documentation:**

1. Tarvainen A, Valpola H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results[J]. Advances in neural information processing systems, 2017, 30.
2. Çiçek Ö, Abdulkadir A, Lienkamp S S, et al. 3D U-Net: learning dense volumetric segmentation from sparse annotation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19. Springer International Publishing, 2016: 424-432.
3. Shu J, Zhao Q, Xu Z, Shu J, Zhao Q, Xu Z, et al. Meta transition adaptation for robust deep learning with noisy labels[J]. arXiv preprint arXiv:2006.05697, 2020.
4. Guo X, Yang C, Li B, Guo X, Yang C, Li B, et al. Metacorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 3927-3936.