MICCAI 2023 Challenges: STS-A technical report on tooth segmentation based on 3D CBCT

Wang Chunshi 1, Wang Yifan 1, Zhao Bin 1,2*

1 School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

2 Key Laboratory of Artificial Intelligence Algorithm Engineering, Guangxi University, Guilin, 541004, China

The technical report provides a detailed introduction to the development process of both the preliminary and final rounds, summarizing their respective innovations. Overall, this task is challenging, but through reviewing relevant literature and the team's prior accumulation, we have addressed it to some extent. We are grateful for the opportunity provided by the organizing committee and the responsible teachers.

1 preliminary contest

1.1 related work

For the semi-supervised semantic segmentation problem, the team mainly investigated and studied the work related to the Match family, including: MixMatch[1], ReMixMatch[2], FixMatch[3], FreeMatch[4] and UniMatch[5]. These works fall under the Self-Training category, integrating pseudo-labeling and consistency regularization, which aligns with the implementation approach for this task. By reproducing and studying these papers, the team ultimately selected UniMatch as the Baseline for this 3D dental segmentation project. They also drew inspiration from the adaptive adjustment of confidence thresholds in FreeMatch to optimize the parameter learning process in the segmentation task.

1.2 development process

1.2.1 Improve feature perturbation in UniMatch

(The code for this section is located at: model/my_smp/base/model.py line 37)

Semi-supervised learning (Semi-supervised Learning, SSL) has seen many related works based on the Mean-Teacher (MT) model. The essence

of this approach is to obtain differences between models through exponential moving average (Exponential Moving Average, EMA), thereby achieving slightly different prediction results for the same image and enhancing the stability of model learning. In UniMatch, the MT model was not used. By reflecting on the learning process of the MT model, the team proposed the following viewpoint: the EMA of network weights is actually also a form of perturbation.

The above idea can be understood as follows: in the UniMatch model, it proposes feature perturbations in the Bottleneck part. The effect is actually similar to that of the MT model; both essentially achieve different outputs by adding different perturbations. However, the MT model uses "perturbations" across the entire network weights, whereas the feature perturbations in UniMatch only apply nn.Dropout (0.5) to the high-dimensional features in the Bottleneck part.

UniMatch The purpose of the proposed method is to obtain a wider disturbance space. The essential purpose of the disturbance is to change the distribution of data and the richness of the information represented by it. The role of EMA is the weight of the network, and the change of the same data in the forward propagation process is obvious, so obviously its equivalent disturbance is more severe.

The above analysis can be reflected in the observations during team training UniMatch: when calculating pre and p!p loss terms L!p, they tend to become 0 in the later stages of training. This indicates that the common nn.Dropout (0.5) has limited perturbation strength in tooth segmentation tasks, requiring further enhanced perturbation to enable better consistency learning by the model.

Based on the above analysis, the team first proposed to use the following two methods to improve the feature perturbation in UniMatch:

1) Increase the probability of loss of nn.Dropout (). For example, let it be nn.Dropout (0.75).
2) Replace nn.Dropout (0.5) with other feature perturbation methods, such as random noise, or other types of Dropout methods.

Due to the Bottleneck part in UniMatch storing a large amount of high-dimensional semantic features, losing too much information can lead to a decrease in model accuracy. In this competition, the team adopted the second method, which is to replace nn.Dropout (0.5) with other types of Dropout (adding random noise during network training would severely impact training performance, hence it was abandoned). Specifically, in the code implementation, nn.Dropout (0.5) was replaced with nn.AlphaDropout (0.5). Alpha Dropout [6] is a type of Dropout that maintains normalization properties. For inputs with mean and unit standard deviation of zero, Alpha Dropout's output retains the original mean and standard deviation of the input. Compared to traditional Dropout, which only sets a certain pixel to 0, Alpha Dropout adds normalization operations on top of this, making the data smoother during training, reducing signal jumps, and allowing modifications to all values at Bottleneck within an Batch, thus achieving

stronger feature perturbation.

1.2.2 Modify the fixed threshold of UniMatch

(The code for this section is located at: util/thresh_helper.py, unimatch_fourier.py Line 145)

The threshold is crucial in the SSL training process, determining the network's confidence level. Both excessively high and low thresholds can affect the learning of model parameters. UniMatch uses a fixed threshold to output prediction results, which inevitably leads to the accumulation of incorrect annotations during training, impacting segmentation accuracy. To address this, the team introduced the adaptive adjustment of confidence threshold from FreeMatch into the current segmentation task, enabling UniMatch to adaptively adjust the threshold during training, thereby improving segmentation accuracy [15].

1.2.3 UniMatch's Encoder-Decoder selection

Based on the current task, the team experimented with the encoder (Encoder) and decoder (Decoder) in UniMatch.

For Encoder, the size of the model is not positively correlated with segmentation performance. By comparing horizontally under the same Decoder, different Encoder perform better; ResNeSt-14d[7] has better segmentation results than ResNeSt-50d; compared to ResNet-18 (with 11M parameters), ResNeSt-14d (with 8M parameters) performs better.

For Decoder, DeepLabV3+[13] can obtain better edge information but has weaker decoding capabilities for semantic information; UNet[14] is more balanced, capable of performing well in semantic information decoding, but its edge segmentation is poorer. In terms of segmentation metrics: the network using DeepLabV3+ can achieve better Hausdorff distance (Hausdorff Distance, HD), while UNet provides better IoU and Dice scores. Given that IoU and Dice scores account for a significant portion of the evaluation metrics for this task, the team ultimately chose UNet as Decoder. Although UNet++[12] can offer better semantic information decoding, its model parameters are too large, leading to higher training costs, and it was not adopted in the task.

Through the above experiments, Encoder and Decoder in UniMatch were selected as ResNeSt-14d and UNet, respectively.

1.2.4 A new plug and play attention module, SCA, is proposed

(The code for this section is located at: `model/my_smp/encoders/resnesc.py`)

According to the experimental discussion in Section 1.2.5, UNet, when used as a decoder, performed poorly compared to the other two indicators in terms of HD. To improve the decoder's ability to decode spatial position information and achieve better HD, the team proposed a new plug-and-play (Split Coordinate Attention) attention module, embedding it into the ResNeSt-14d of the Encoder selected in Section 1.2.5 to obtain SCANet, thereby enabling the encoder to retain stronger spatial information. The structure of SCA is shown in Figure

1(b).

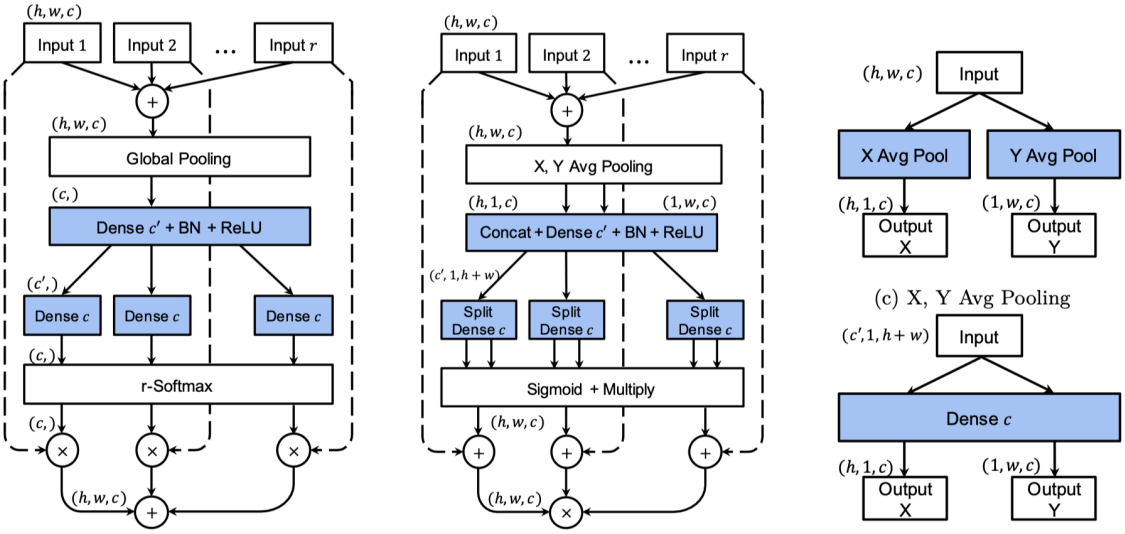


Figure 1 Comparison of the structure of SCA and SA. (a) SA (b) SCA (c) X, Y average pooling module (d) Split Dense module.

To better illustrate the proposal process of SCA, attention Split Attention (SA)[7], is introduced here, with its structure shown in Figure 1(a). SA is a novel type of attention proposed in ResNeSt, drawing inspiration from ideas in works such as GoogleNet[8], ResNeXt [9] and SE-Net[10]. It integrates Feature-map Attention, Multi Path and Channel Attention to achieve relatively good performance.

By observing the aforementioned structure, it is found that there is actually room for improvement in Split Attention within ResNeSt [7]: The premise of Split Attention is that standard convolution struggles to model channel relationships, so it employs global average pooling across spatial dimensions to collect contextual information with embedded channel statistics. However, global average pooling clearly disrupts the spatial semantic features of the network, directly compressing the spatial dimension to 1 × 1, sacrificing part of the positional information, which is crucial for capturing spatial structures in semantic segmentation.

Based on the shortcomings of SA, the team considered modeling feature maps in different dimensional directions, that is, using average pooling in the X and Y directions instead of the original pooling operation, and splitting them into different feedforward streams to achieve cross-dimensional extraction of semantic information [11]. Finally, a matrix multiplication operation is performed

before output to form an attention Map with the same size as the original input. This improvement not only effectively models cross-channel information but also maximizes the retention of spatial information.

The following subsections provide a detailed description of SCA.

1.2.4.1 Define Featuremap Group

Similar to ResNeXt Block and ResNeSt Block, two hyperparameters K and R are introduced to control the number of feature group sets and the number of splits within the base set (Cardinal Group), respectively. Therefore, the total number of feature groups is G = KR. Then, a series of transformations {F1, F2,..., FG} are applied to each individual feature group, where each

The middle of the group is represented as Ui = Fi (X), where i { 1,2,..., G}.

1.2.1.2 Problems in Split Attention

The representation of the K-th basis set is : $\hat{U}^k = \sum_{j=R(k-1)+1}^{Rk} U_j$,

among $\hat{U}^k \in \mathbb{R}^{H\times W\times C/K}, k \in \{1,2,\ldots, K\}$. Then the c-th component in Split Attention is calculated as follows:

$$s_c^k = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} \hat{U}_c^k(i,j). \quad (1)$$

However, global average pooling compresses the global spatial information into channel descriptors, so it is difficult to retain the location information, which needs to be improved.

1.2.1.3 Split Coordinate Attention

To encourage the use of precise location information in spatially capturing remote interactions, and to optimize the issue of retaining this location information, the team decomposed the global pooling defined in the equation, using two spatially defined pooling Kernels, (H, 1) and (1, W), encoding along the horizontal and vertical coordinates respectively. Therefore, the Cth component in the h, w direction can be represented as:

$$s_c^{h,k} = \frac{1}{W}\sum_{0\le i<W} \hat{U}_c^k(h,i) \quad (2)$$

$$s_c{}^{w,k} = \frac{1}{H}\sum_{0\le j<H} \hat{U}_c^k(j,w) \quad (3)$$

Next, the above quantities are combined. First, the Concat operation is performed on the channel degree, and then a convolution and nonlinear layer are passed through, which can be obtained:

$$s_c^k = \mathcal{G}^c([s_c^{h,k}, s_c^{w,k}]). \quad (4)$$

Next, the sck is split into tch, k and tcw, k, and the following is obtained:

$$\begin{cases} u_c^{h,k} = \sigma(\mathcal{S}_c^h(t_c^{h,k})) \\ u_c^{w,k} = \sigma(\mathcal{S}_c^w(t_c^{w,k})) \end{cases}, \tag{5}$$

Where σ represents Sigmoid. Then the attention of the base array needs to be aggregated, and the feature map calculation formula for the Cth channel is:

$$V_c^k = \hat{U}_c^k \sum_{i=1}^{R} u_c^{h,k} \times u_c^{w,k} \tag{6}$$

1.2.1.4 SCANet

The proposed SCA module is embedded into ResNeSt-14d to obtain SCANet, and the number of parameters does not change.

1.2.5 Total number of parameters in the model

In addition to using the parameter-minimized version of SCANet, the team further restricted the parameters of the UNet decoder to better facilitate model parameter learning. Specifically, the number of Channel output by each Stage of UNet was changed from 256,128,64,32, and 16 to 64,32,16,8, and 4, respectively. The total number of parameters in the entire model is: 11.1M

1.2.6 data preprocessing

To utilize pixel-level pseudo-annotations generated from unlabeled data in subsequent model learning, the data is processed through slicing. To obtain rich dental morphology, the data is sliced along three axes: axial plane, coronal plane, and sagittal plane. For example, a 640 × 640 × 400 CT image, when sliced along these three axes, results in 640 + 640 + 400 = 1680 slice images. The file suffix is used to divide the slicing axes; for simplicity, _x, _y, and _z are directly used as the suffixes for the slice files[16].

In addition, the slices are normalized uniformly and the image values are mapped to the [0,1] interval [5]. Since there are a large number of slices, 10% of the slices in the training set (Train) is used as the validation set (val).

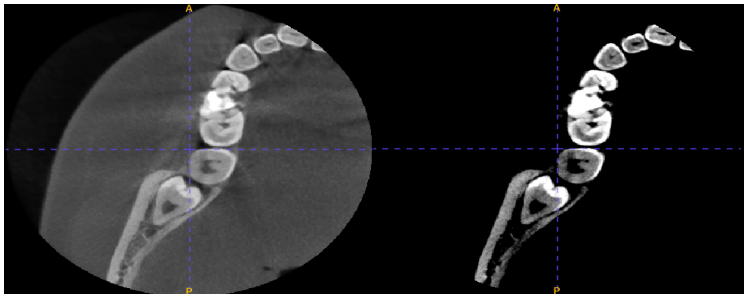
In order to explore the influence of different threshold values on the performance of the network model, the team conducted a threshold experiment, and the experimental results are shown in Table 1.

Table 1 Segmentation results of input images with different thresholds on the verification set

Bottom	Top	*Val1*	*Val2*	*Val3*
0	1500	0.7738	0.8179	0.8621
500	1500	0.8167	0.8258	0.8762
500	1800	0.8386	0.8036	0.8849
550	1950	0.7253	0.8503	0.8732
200	2000	0.7577	0.883	\
500	2000	0.8162	0.848	0.8844
750	2000	0.8269	0.8321	\
300	2000	0.8131	0.854	0.8796
450	2050	0.829	0.8201	0.8671
400	2100	0.747	0.8128	0.8454
0	2500	0.8247	0.8301	0.8603
500	2500	0.8597	0.8602	0.8783

From Table 1, it can be seen that when the threshold is set to a combination of 500 and 2000, the model's segmentation results on the validation set are generally the best, and it achieves faster convergence. Therefore, this combination is chosen as the threshold in the experiment. Assuming the image x is the original data image after CT reading, with a value range of [1000,3095], then the processing procedure for each pixel point xx, y, z of x is:

$$x_{x,y,z} = \begin{cases} 500, & x_{x,y,z} \le 500 \\ 2000, & x_{x,y,z} \ge 2000 \\ x_{x,y,z}, & \text{Others} \end{cases}$$



(a) Original image slices (b) Image slices after threshold processing

Figure 2 Slices of the original input image and slices after threshold [500,2000] processing

As shown in Figure 2, the original CT image is thresholded at 500 and 2000, which can significantly enhance the target information, which is conducive to the model's better learning of semantic information and improve the segmentation accuracy.

1.2.7 Model training and prediction | (the prediction code is in `infer`. py)

The sliced data processed in 1.2.6 is fed into the improved Improved-UniMatch for training, and trained three times along different axes, resulting in three models. The functions are defined as fx (wx; x), fy (wy; y), and fz (wz; z), where wx, wy, and wz are the parameters corresponding to the three models; x, y, and z are the inputs corresponding to the three models.

For a set of three different slices of an input image Ix, Iy and Iz, three prediction results are obtained respectively, which are px = fx (wx; Ix), py = fy (wy; Iy) and pz = fz (wz; Iz). Then, the final prediction result is obtained by using argmax:

$$Pre = argmax(\mathrm{p}_x + \mathrm{p}_y + \mathrm{p}_z) \tag{7}$$

1.3 Training tips

1.3.1 Training hyperparameter Settings

The optimizer of the model training uses AdamW, the initial learning rate lr is selected as 0.0001, and weight_ decay is 0.0001.

The learning rate is updated using the following formula:

$$lr = lr_b \times (1 - \frac{i}{N})^{0.9} \quad (8)$$

Lrb refers to the basic learning rate, which is 0.0001; i refers to the current iteration times, and N refers to the total iteration times.

When using the threshold adaptive method, the momentum parameter is set to 0.9998 for better effect. If the momentum is too small, the threshold may be 1.00, which will seriously affect the convergence of the model.

1.3.2 data augmentation

The team used the data enhancement paradigm built in UniMatch, but made some changes to better adapt to the task. The specific data enhancement, parameters, and application enhancement probability are shown in Table 2.

Table 2 Data enhancement information table

Enhance type	Parameter	Application probability
ColorJitter	0.75, 0.75, 0.75, 0.35	0.8
CutMix	\	0.5
GaussianBlur	$\sigma \in (0.1,0.5)$	0.5
RandomFlip	\	0.75
RandomRotate	\	0.5

1.3.3 Loss, function setting

The Loss function team uses the ordinary CrossEntropy Loss and Dice Loss. For different loss items, different Losses and calculation methods are used.

Specifically, each unmarked image xu is simultaneously perturbed by two operators, namely the weak perturbation w (such as cropping) and the strong perturbation s (such as color jitter). Then, the overall objective function is a combination of supervised loss s and unsupervised loss u:

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_s + \mathcal{L}_u) \tag{9}$$

Usually, the supervised loss s is the CrossEntropy and Dice losses between model predictions and true labels. The unsupervised

loss u regularizes the prediction of samples under strong perturbations to be the same as under weak perturbations, which can be expressed as:

$$\mathcal{L}_u = \frac{1}{B_u}\sum \mathbb{1}(max(p^w) \geq \tau)\mathrm{H}(p^w, p^s)\,, \tag{10}$$

Where Bu is the batch size of unmarked data, τ is the predefined confidence threshold for filtering out noisy labels, and H minimizes the entropy between the two probability distributions:

$$p^w = \hat{F}(\text{Gongw (xu)});\ \text{ps} = \text{F(Gongs (Gow (xu)))} \tag{11}$$

For simplicity, we follow the FixMatch$\hat{F}$ Set it to be exactly the same as F. Let the feature disturbance prediction be pfp, then the unsupervised loss u is:

$$\mathcal{L}_u = \frac{1}{B_u}\sum \mathbb{1}(max(p^w) \geq \tau)(\mathrm{H}(p^w, p^s) + \mathrm{H}(p^w, p^{fp})) \tag{12}$$

Due to the existence of dual-stream disturbance, the idea of contrast learning is adopted. Compared with FixMatch, two are maintained

An auxiliary feedforward current, then the final unsupervised loss u is:

$$\mathcal{L}_u = \frac{1}{B_u}\sum \mathbb{1}(max(p^w) \geq \tau)(\lambda \mathrm{H}(p^w, p^{fp}) + \frac{\mu}{2}(\mathrm{H}(p^w, p^{s_1}) + \mathrm{H}(p^w, p^{s_2}))) \quad (13)$$

1.4 Summary of innovation points in the preliminary competition

1) The feature perturbation in UniMatch is improved to make the model obtain a wider range of perturbation effects.

2) The threshold for generating pseudo-labels in UniMatch is improved to make it adaptive, which significantly improves the performance of the model segmentation edge.

3) A new plug-and-play attention module SCA is proposed to be embedded in the encoder to obtain SCANet, which can improve the Hausdorf distance index of segmentation without increasing the number of parameters and obtain better segmentation edges.

4) The proposed tooth segmentation algorithm is simple to implement, low training cost and model size is only 11.1M.

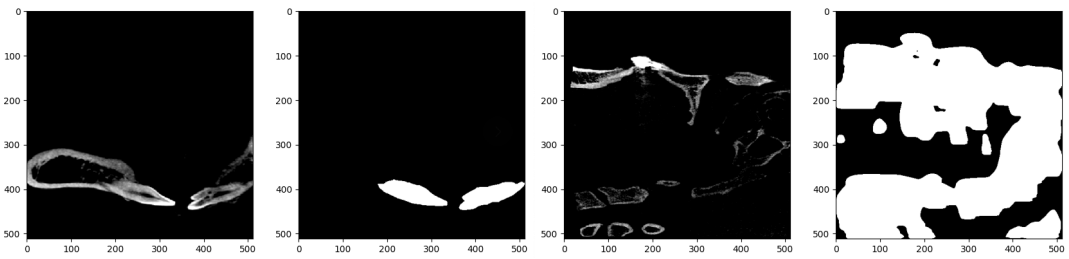
5) The influence of threshold on segmentation performance is discussed, and the most appropriate threshold is selected for data preprocessing.

2 intermediary heat

2.1 Analysis of difficulties

2.1.1 Domain offset problem

After the start of the semi-final, the team tried to use the initial model for semi-supervised training. After the training, they observed the prediction results on the test set (Test) and found that the effect was very poor, as shown in Figure 3. It is speculated that there is a serious domain shift (Domain shift) problem between labeled data and unlabeled data.



(a) Verification set image slices (b) Verification set image slice segmentation results

(c) Test set image slices (d) Test set image slice segmentation results

Figure 3 Validation set and test set segmentation results

To better address the domain shift issue, the team conducted data statistical analysis on the Labeled set, Unlabeled set, and Test set. The statistical results show that the image resolution of the Labeled set is significantly lower than that of the Unlabeled set and Test set. Moreover, the pixel intensity range of CT images in the Labeled set ranges from [-1000,3095], while the pixel intensity range for the other two sets ranges from-1000 to 10000, varying widely

As a result, it is difficult for Labeled set to guide Unlabeled set to conduct semi-supervised learning, which further hinders the generalization performance of the model on Test set.

In order to eliminate the influence of domain shift as much as possible, the team designed a 2-Stage algorithm, which enables the model to learn the features of the target domain through two Stage, and uses the Fourier Transform Augment (FTA) module to enable the model to learn the differences between the two domains.

2.1.2 The problem of artifacts

There are various kinds of artifacts in CT images, and the impact of artifacts on segmentation models is huge, as shown in Figure 4.

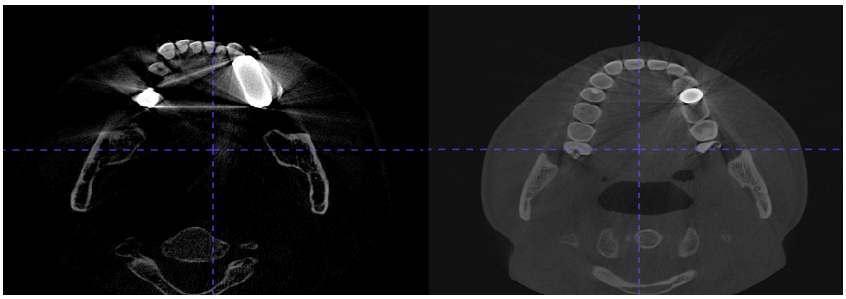


Figure 4 Example of image slice artifacts

In the semi-supervised learning process, the dental artifacts will reduce the quality of the pseudo-labels and ultimately affect the segmentation accuracy of the model. In the subsequent training stage 1 of generating pseudo-annotations, compared with 3D nnUNet, the pseudo-annotations generated by 2D nnUNet are more accurate and less affected by the artifacts.

2.2 development process

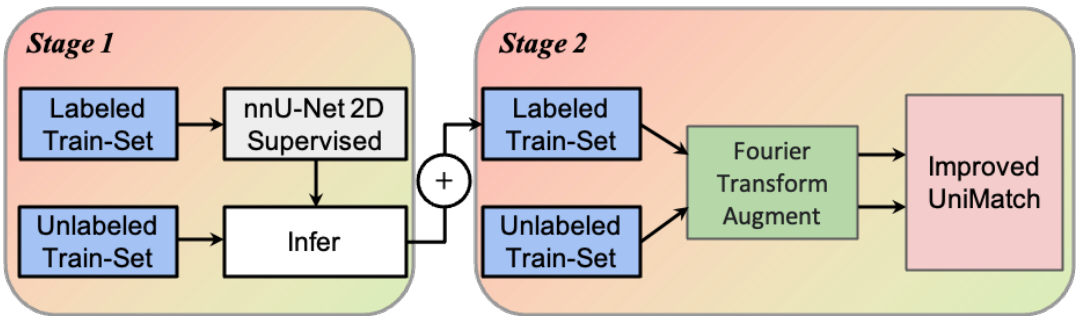


Figure 5 Overall process diagram of 3D tooth segmentation in the second competition

Based on the Improved-UniMatch model proposed in the preliminary round, during the semi-final, the team adopted a multi-stage training strategy to reduce domain shift issues between the Labeled set and

the Unlabeled set, thereby enhancing the model's generalization performance on the Test set. Figure 5 shows the overall workflow for the 3D dental segmentation in the semi-final. In the first stage, a 2D nnU-Net model is used for supervised learning to generate pixel-level pseudo-labels from a small number of randomly sampled Unlabeled samples in the training set. These pseudo-labeled samples are then merged with the labeled samples in the training set to form the labeled samples in the training set. The unlabeled samples in the training set are also processed for domain adaptation before being fed into the Improved-UniMatch model

Parameter learning.

Next, the training process of the relevant stages will be introduced in detail.

2.2.1 stage

During the training of Stage1, a 2D nnU-Net is used as the segmentation network. First, labeled data is used to train 2D nnU-Net 20 Epochs, and then the trained 2D nnU-Net network generates low-quality pixel-level pseudo-labels for 10 randomly selected unlabeled data from the Unlabeled dataset. These low-quality pseudo-labels can locate the position of teeth but will also segment out many irrelevant areas, such as the bones of the upper jaw and the base of the CT, as shown in Figure 6.

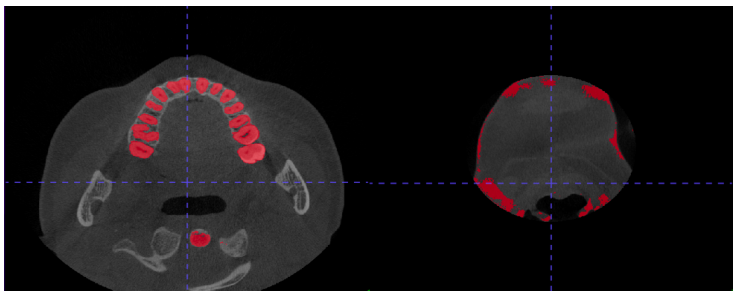


Figure 6 Low quality pixel level pseudo-annotations

Then the 10 samples of pseudo-annotations were merged with the Labeled set to form the labeled sample set in the Stage 2 training set.

2.2.2 stage

(The code for this section is located at: dataset/micca3d. py, line 198)

The training of Stage 2 is based on the data processed in the first stage. First, the dataset (Train, Val) is divided according to the same criteria as the preliminary competition. Then, the data is augmented through the FTA module, enabling the model to learn information from the target domain and use the Improved-UniMatch model proposed in the preliminary competition for training.

2.2.2.1 FTA module

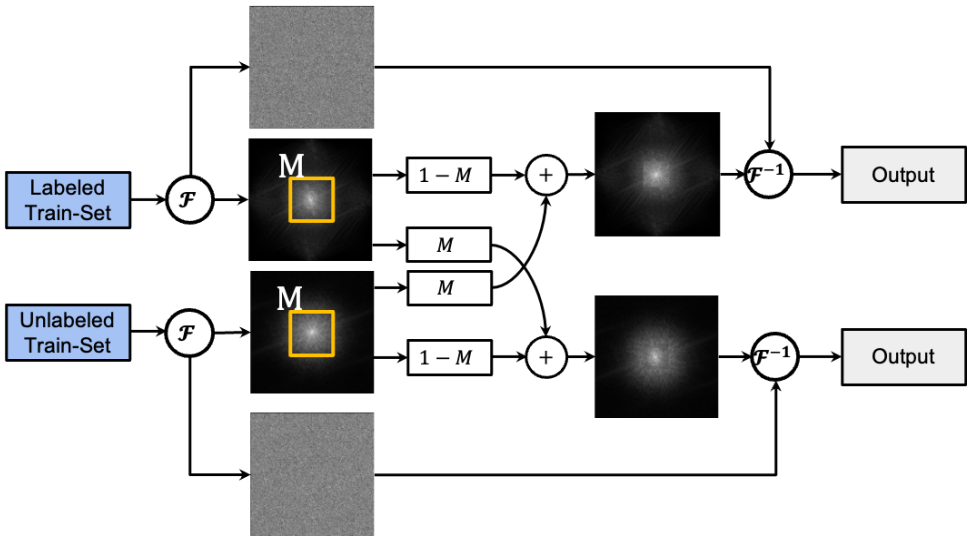


Figure 7 FTA, flow chart

During the training process, one Labeled image xw and one Unlabeled image xu are randomly selected. Then, Fourier transform is performed to transfer the images into the frequency domain, obtaining the amplitude spectrum Aw, Au and phase images Pw, Pu. The amplitude spectrum contains low-level statistics, while the phase image includes high-level semantics of the original signal [17].

Then, the amplitude information of image xu is merged to enhance image x w:

$$Awnew = (1- \lambda)Aw * (1-M) + \lambda Au * M \quad (9)$$

Among them, Awnew is the newly generated phase map; λ is the parameter for adjusting the ratio of phase information between xw and xu; M is used to control the spatial range of the amplitude spectrum to be exchanged, here M is set to include the central region of the amplitude spectrum containing low-frequency information. Then, the merged sample is transformed from the frequency domain to the image domain through -1, resulting in an image sample zw that has been enhanced by Fourier transform and fused with the low-level information of another sample:

$$Z^w = F^{-1}(A^w_{\text{new}}, P^w) \quad (10)$$

By the same token, exchanging x w and xu gives us zu:

$$Z^u = F^{-1}(A^u_{\text{new}}, P^u) \quad (11)$$

It is worth mentioning that zw and zu can be combined for calculation, which will greatly reduce the operation cost. That is,

in the same enhancement, xu and x w need to be enhanced simultaneously, and the results of mutual enhancement are sent to the subsequent training process.

2.3 Summary of innovation points in the second round

1) A multi-stage training framework is proposed to reduce the performance degradation caused by domain shift.

2) Use 2D nnU-Net to generate low-quality pixel-level pseudo-annotations to improve the performance of the model in the target domain.

3) An FTA data enhancement module is adopted to realize the dual-stream data enhancement and improve the generalization performance of the model on the test set.

2.4 The results of the second round

2.4.1 Quantitative results

Table 3 Quantitative results of the model on the validation set and test set

Data set	Dice	IoU	HD	Score
Validation set	0.8786	0.8413	0.0648	0.8844
Test set	0.8058	0.8376	0.1599	0.8256

2.4.2 Qualitative results

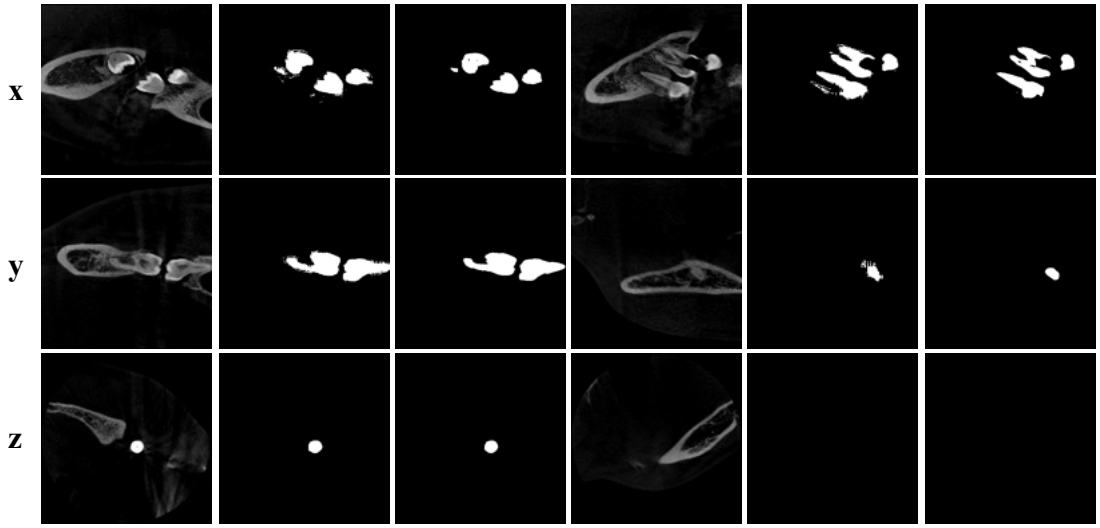


Figure 8 Example of data segmentation results from the source domain of the validation set

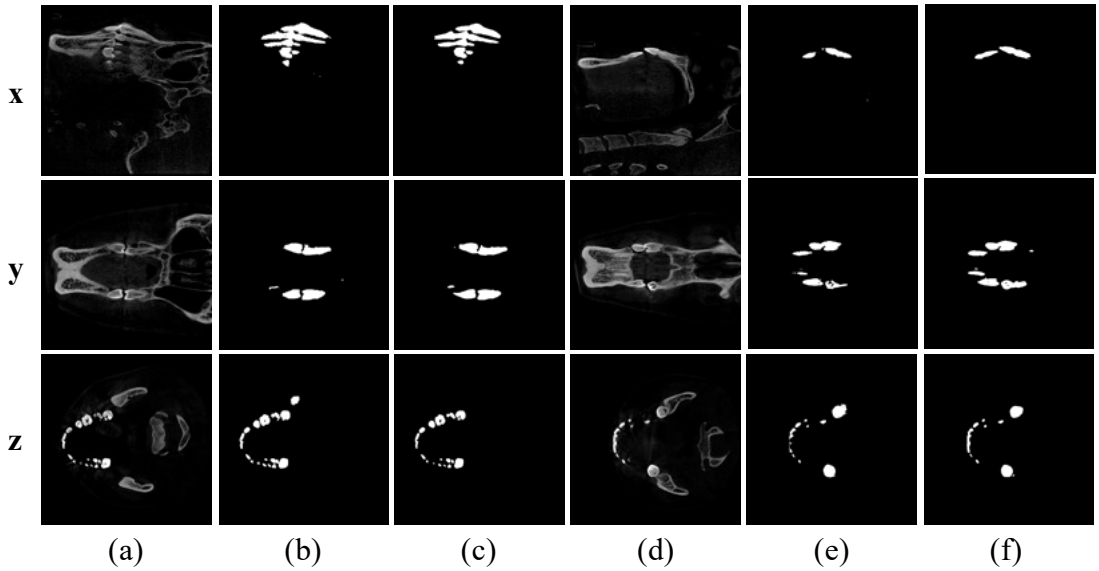


Figure 8 Example of data segmentation results of target domain in the validation set (Label is generated by nnUNet)

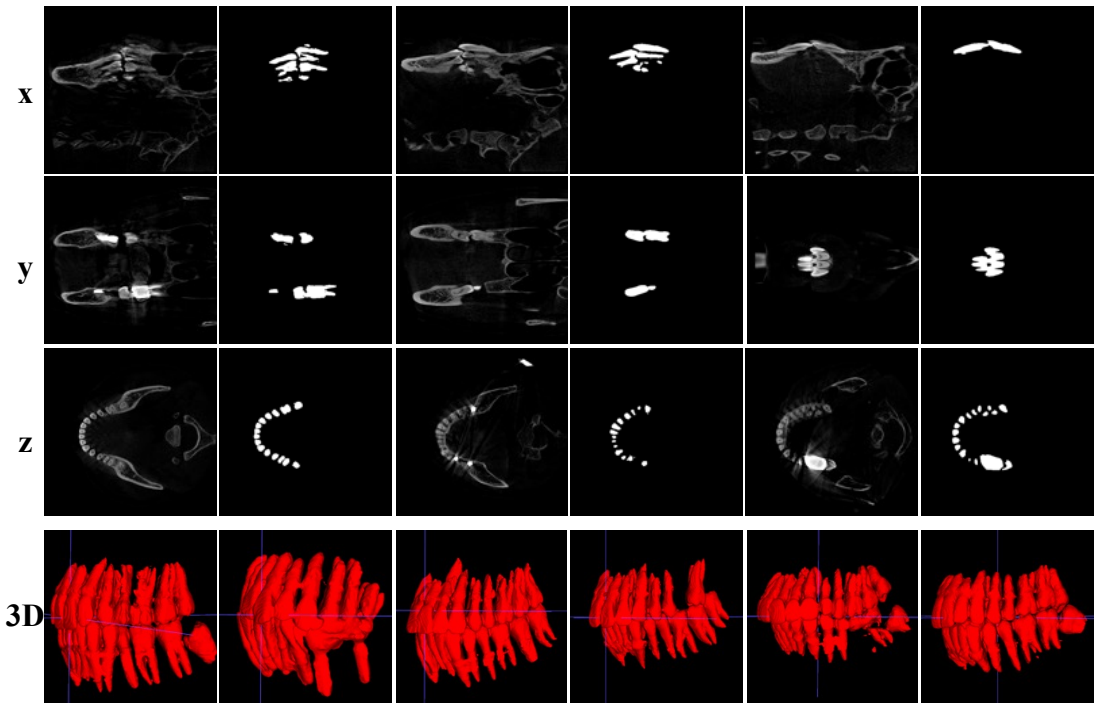


Figure 9 Example of target domain data segmentation results of the test set

reference documentation

[1] Berthelot D, Carlini N, Goodfellow I, et al. Mixmatch: A holistic approach to semi-supervised learning[J]. Advances in neural information processing systems, 2019, 32.

[2] Berthelot D, Carlini N, Cubuk E D, et al. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring[J]. arXiv preprint arXiv:1911.09785, 2019.

[3] Sohn K, Berthelot D, Carlini N, et al. Fixmatch: Simplifying semi-supervised learning with consistency and confidence[J]. Advances in neural information processing systems, 2020, 33: 596-608.

[4] Wang Y, Chen H, Heng Q, et al. Freematch: Self-adaptive thresholding for semi-supervised learning[J]. arXiv preprint arXiv:2205.07246, 2022.

[5] Yang L, Qi L, Feng L, et al. Revisiting weak-to-strong consistency in semi-supervised semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 7236-7246.

[6] Klambauer G, Unterthiner T, Mayr A, [6] Klambauer G, Unterthiner T, Mayr A, et al. Self-normalizing neural networks[J] . Advances in neural information processing systems, 2017, 30.

[7] Zhang H, Wu C, Zhang Z, [7] Zhang H, Wu C, Zhang Z, et al. Resnest: Split-attention networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 2736-2746.

[8] Szegedy C, Liu W, Jia Y, [8] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[9] Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1492-1500.

[10] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[11] Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13713-13722.

[12] Zhou Z, Siddiquee M M R, Tajbakhsh N, [12] Zhou Z, Siddiquee M M R, Tajbakhsh N, et al. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation[J] . IEEE transactions on medical imaging, 2019, 39(6): 1856-1867.

[13] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 801-818.

[14] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9,

[15] Sun B, Yang Y, Zhang L, [15] Sun B, Yang Y, Zhang L, et al. CorrMatch: Label Propagation via Correlation Matching for Semi-Supervised Semantic Segmentation[J] . arXiv preprint arXiv:2306.04300, 2023.

[16] Liu Z, Cao C, Ding S, et al. Towards clinical diagnosis: Automated stroke lesion segmentation on multi-spectral MR image using convolutional neural network[J]. IEEE Access, 2018, 6: 57006-57016.

[17] Yao H, Hu X, Li X. Enhancing pseudo label quality for semi-supervised domain-generalized medical image segmentation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2022, 36(3): 3099-3107.