# **1. Divide the overall process of tasks**

## 1.1. Tagged data training

### 1.1.1 Data preparation

Provide a TaskID (e.g. Task001) for the tagged data and organize it according to the requirements of the nnUNet. The file structure is as follows:

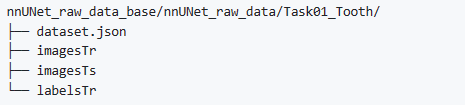


Figure 1: File structure of training data

imagesTr The folder stores the original image data of training, imagesTs the folder stores the original image data of testing, and labelsTr the folder stores the label of the image data of training.

### 1.1.2. Use nnUNet for preprocessing

Here we use the default setting, and the command is as follows:

nnUNet\_plan\_and\_preprocess -t 1 --verify\_dataset\_integrity

### 1.1.3. Training is carried out through 5-fold cross validation

for FOLD in 0 1 2 3 4

do

CUDA\_VISIBLE\_DEVICES=0,1 nnUNet\_train\_DP 3d\_fullres nnUNetTrainerV2\_DP 1 $FOLD -gpus 2 -c --npz

done

## 1.2. Generate pseudo labels for unlabeled data

### 1.2.1 Generate pseudo labels

nnUNet\_predict -i $INPUTS\_FOLDER -o $OUTPUTS\_FOLDER -t 2 -m 3d\_fullres --save\_npz

### 1.2.2. Iterative training model and generate pseudo labels

* Give a new TaskID (e.g. Task002) and organize the tagged data and pseudo-tagged data as described above.
* Use nnUNet for automatic preprocessing:

nnUNet\_plan\_and\_preprocess -t 2 --verify\_dataset\_integrity

* Train a new nnUNet model with all the training data

for FOLD in 0 1 2 3 4

do

CUDA\_VISIBLE\_DEVICES=0,1 nnUNet\_train\_DP 3d\_fullres nnUNetTrainerV2\_DP 2 $FOLD -gpus 2 -c --npz

done

* Generate new pseudo-labels for unlabeled data.

### 1.2.3. Filter low quality pseudo tags.

Using the select\_pseudo\_label.ipynb script, we compare the pseudo-labels in different rounds and filter out the labels with high variants.

## 1.3. Train nnUNet-att through 5-fold cross validation

### 1.3.1. Copy the following files from this repo to the nnUNet environment.

In order to improve the performance of the model, we modified the generic\_UNet.py of nnunet source code and constructed the nnUNet-att model.

./nnunet/network\_architecture/generic\_UNet.py

### 1.3.2. Train nnUNet-att through 5-fold cross validation

for FOLD in 0 1 2 3 4

do

CUDA\_VISIBLE\_DEVICES=0,1 nnUNet\_train\_DP 3d\_fullres nnUNetTrainerV2\_DP 3 $FOLD -gpus 2 -c --npz

done

### 1.3.3. Add the suffix "\_000" to the file name of the test set data

pyhton add\_suffix.py

### 1.3.4. Use nnUNet-att for efficient reasoning

nnUNet\_predict -i $INPUTS\_FOLDER -o $OUTPUTS\_FOLDER -t 3 -m 3d\_fullres --save\_npz

## 1.4. The abnormal segmentation points are corrected through the position correction module

python position\_correction.py

# **2. Details of the deployment experiment**

## 2.1 Model Architecture

### 2.1.1 Overall framework



Figure 1: Our overall framework. The overall architecture follows the U-Net architecture.

In our research, we carefully selected nnUNet[1] as our backbone because nnUNet has demonstrated outstanding performance in medical image segmentation tasks. However, we were not content with simply adopting it; instead, we made deep improvements to better suit our specific task. Our focus of improvement was primarily on the second-level decoder of nnUNet, as we fully understood its critical importance for segmentation performance. We decided to introduce a 3D axial attention mechanism, [2], into the second-level decoder, which is a significant innovation. The core motivation behind this decision is that we wanted the model to focus more on key details of dental structures. To achieve this goal, we utilized a 3D axial attention mechanism, which offers excellent computational and storage efficiency when processing high-dimensional data and can capture both local and global dependencies. This means our model can more accurately identify and locate complex dental structures, such as roots and crowns, thereby achieving higher precision segmentation results. Figure 1 illustrates the overall structure of our framework.

### 2.1.2 backbone:

nnUNet is a neural network architecture specifically designed for medical image segmentation tasks. The model inherits the basic structure of U-Net, including a down-sampling encoder and an up-sampling decoder, but introduces several innovations and optimizations to enhance performance and adaptability. Most notably, its adaptability: nnUNet can automatically adjust its network architecture and hyperparameters according to different datasets and task requirements. This feature significantly reduces the complexity of model tuning, enabling it to achieve state-of-the-art performance across multiple medical image segmentation challenges. In addition to adaptability, nnUNet employs a series of carefully designed training strategies, including data augmentation, loss function selection, and optimization algorithms, further enhancing the model's robustness and accuracy. Due to these advantages, nnUNet has become a key benchmark in the field of medical image segmentation, providing researchers with an efficient and reliable solution.

### 2.1.3 3D axial attention mechanism

The 3D axial attention mechanism is a key component of our innovative approach. This mechanism allows the model to better focus on horizontal and vertical structures in images during segmentation, which is particularly important for 3D dental segmentation [3]. The 3D axial attention mechanism adjusts feature maps in the decoder by considering features from different directions. This helps enhance the model's ability to capture fine details, thereby improving the accuracy of segmentation.

We add a 3D axial attention component to the upsampled part of the decoder, as shown in Figure 2. The principle of the 3D axial attention mechanism is to reduce computational complexity by decomposing 3D space into multiple single dimensions or "axes." Therefore, the model can perform attention calculations on each axis individually, rather than across the entire 3D space. For an input image with a specific shape, the iterative process of the algorithm is as follows:

1. Axis selection and slice extraction: First, we select an axis (such as the depth axis). Then, we extract all slices on that axis, each of which is a shape 2D tensor.
2. Attention weight calculation: For each 2D tensor slice, we use the standard multi-head attention mechanism [4] to calculate the query (Query), key (Key), and value (Value) matrices. The attention weight is obtained by the following formula:
3. Feature map update: Use the calculated attention weights to update the representation of each 2D tensor slice.
4. Axis merging: After completing the attention calculation of one axis, the updated slices are recombined to form a new 3D tensor.
5. Repeat the 1-3 operation for other axes.

Please note, we have attempted to enhance the model's representational capabilities using Channel-Space Attention (Convolutional Block Attention Module, CBAM) [4]. However, the channel-space attention mechanism calculates attention weights across the entire space and channels, which can lead to higher computational and storage requirements when processing high-dimensional data. Additionally, CBAM primarily focuses on global dependencies between channels and space, potentially overlooking local dependencies between different axes. Given that dental regions are concentrated and irregular in shape, relying solely on global dependencies cannot finely handle areas such as edges and depressions. Therefore, we need to pay attention to more extensive axis dimension information.



Figure 2: Schematic diagram of up-sampling. We use 3D axial attention and residual connection to enhance the representation ability and gradient flow.

### 2.1.3 Position correction module

The special nature of the D CBCT image scanning range is a critical issue that requires careful handling in our research. These images include bony structures near the teeth, some of which have morphological and density characteristics very similar to those of the teeth, such as the uncinate process [6]. This similarity can lead to segmentation models incorrectly misidentifying these non-tooth structures as teeth and segmenting them, resulting in erroneous segmentation outcomes. Figure 3 illustrates a real scenario where the uncinate process was also incorrectly segmented.



Figure 3: Incorrect segmentation of the model of the pivot conical tooth (blue box)

To address this issue, we conducted a dataset analysis and consulted relevant dental experts. We learned that during CT scans, the subject's head position is fixed, and non-tooth bony structures are typically located far from the teeth. Therefore, accurately detecting the range of teeth is crucial for excluding incorrect segmentation items. To ensure accuracy, we performed statistical analysis on the segmented ranges of teeth in the labeled data in the training set. In the training set, we found that the maximum width (i.e., length along the x-axis) of the teeth was 305 units. Through consultation with dentists, we learned that there are individual differences in tooth size. For safety, we set the detection width w of the teeth to 320 units, and the length l (i.e., length along the y-axis) to 0.8 times the width, which is 256 units.

To solve the problem of incorrect segmentation, we specially designed a position correction module, which is carefully constructed and includes the following four key steps to ensure that our segmentation results are more accurate and reliable:

1. Determination of Point A and Point B: First, in the cross-sectional image, we need to accurately locate the position of the segmented tooth structure. We find the leftmost point A (coordinates: (x0, y0, z)) and the topmost point B (coordinates: (x1, y1, z)) in the image, as shown in Figure 4.
2. Calculation of Point C: Next, we calculate point C to accurately correct the position of the tooth. Point C is located at the intersection of the line connecting points A and B, where the line from point A extends upwards and the line from point B extends to the left. The coordinates of point C can be calculated using the coordinates of points A and B, typically denoted as (x0, y1, z).
3. Construction of the tooth area frame: Based on our predefined tooth width and length parameters, we construct the tooth area frame along the x-axis and y-axis from point C, as shown in Figure 4. This frame will accurately define the range of the tooth.
4. Data filtering: The final step is to retain the segmented data located in the frame of the tooth area and discard the data outside the frame. This step ensures that only the data truly belonging to the tooth structure is retained, thus effectively solving the problem of incorrect segmentation.

The design and operation process of this detailed location correction module fully considers the particularity of the tooth structure, ensures that our segmentation model can accurately locate and segment the teeth, and excludes the possibility of missegmenting non-tooth structures.

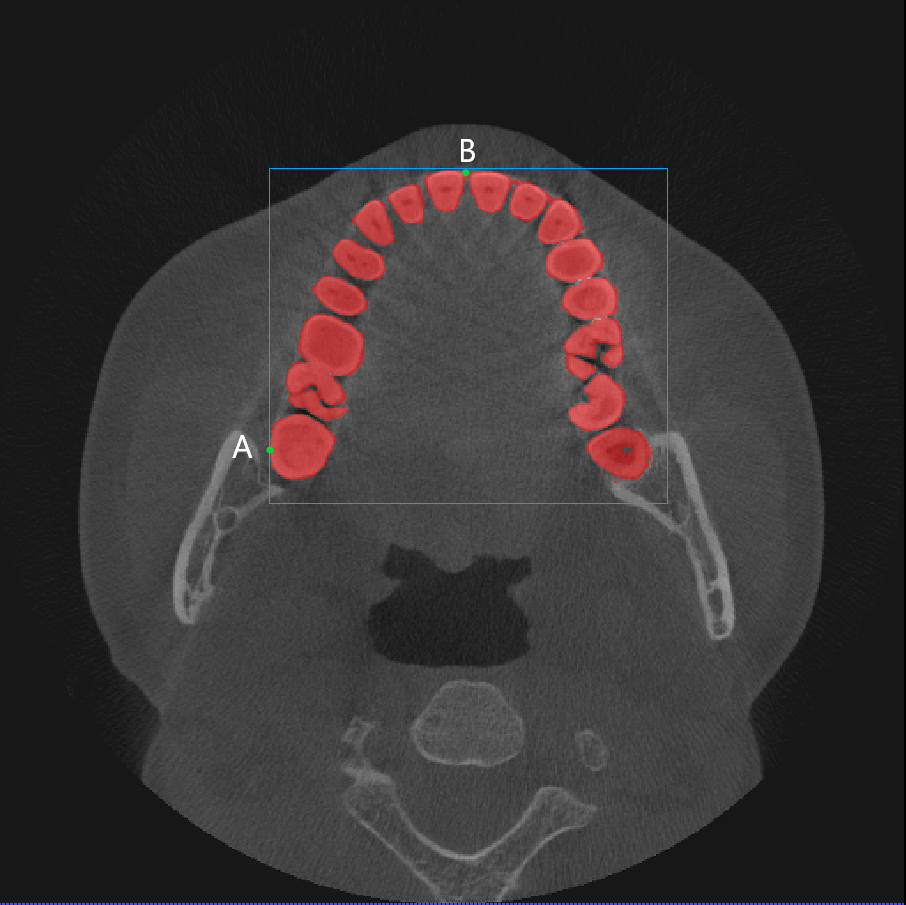


Figure 4: The constructed dental area frame

## 2.2 Pseudo-labeling method

To maximize the use of unlabeled data for model training, we adopted a complex yet efficient pseudo-labeling strategy [7][8] to ensure the quality of pseudo-labels and improve model performance. The following are the detailed steps of our pseudo-labeling strategy, which have been carefully designed to fully leverage unlabeled data to enhance model performance:

1. Training based on 5-fold cross validation: First, we conduct 5-fold cross validation on the labeled data to train 5 independent nnU-Net models to ensure that the models are diverse.
2. Generation of high-quality pseudo-labels: In the pseudo-label generation phase, we use our carefully designed 5-fold nnU-Net set for precise reasoning Settings to predict a pseudo-label on unmarked data. This step ensures the quality and accuracy of the pseudo-labels.
3. Iterative Training and Pseudo Label Update: Next, we iteratively train an nnU-Net model on the union of labeled data with pseudo labels and unlabeled data, generating a new pseudo label after each round of training. This process helps continuously improve the quality of pseudo labels and model performance.
4. Stability selection of pseudo-labels: In the process of pseudo-label generation, we select pseudo-labels according to their stability in different training rounds.
5. Final Model Training: Ultimately, we trained on the union of labeled data and selected unlabeled data with pseudo-labels, employing nnUNet (nnUNet-Att) with 3D axial attention. The purpose of introducing 3D axial attention is to better distinguish the interference from pseudo-labels, ensuring that the model can accurately segment dental structures in the final evaluation.

This complex pseudo-label generation strategy combines several key steps aimed at enhancing model performance and ensuring the quality of pseudo-labels. By carefully designing and implementing these steps, we can more effectively utilize unlabeled data to improve the accuracy and robustness of medical image segmentation models.

## 2.3. Implementation details

Our development environment is detailed in Table 1 to ensure the reproducibility and verifiability of the work. To train our models, we conducted cross-validation with different folds on four different servers. In terms of model building, we strictly followed the standard training methods of nnU-Net, with each model undergoing training through 5-fold cross-validation. Throughout the training process, we also employed dynamic data augmentation strategies to enhance the model's generalization performance. These data augmentation techniques include random rotation and scaling, elastic deformation, additive brightness enhancement, and gamma scaling.

Table 1 Development environment

|  |  |
| --- | --- |
| System version | Ubuntu 20.04.2 LTS | Ubuntu 20.04.2 LTS | Ubuntu 18.04.6 LTS | Ubuntu 18.04.6 LTS |
| GPU (number and type) | 1NVIDIA GTX 3090Ti 24GB|2NVIDIA V100 24GB|2NVIDIA GTX 3090Ti 24GB |1NVIDIA 3090 24GB |
| CUDA edition | 11.4|11.4|11.4|11.3 |
| programming language | Python 3.9.16 |
| Deep learning framework | Pytorch (Torch 1.11.0) |
| Specific dependencies | nnU-Net 1.7.1 |
| code | https://github.com/qpuchen/nnUNet\_att\_position\_correction/ |

We chose to combine Dice loss and cross-entropy loss as the overall loss function because composite loss functions have been proven robust in various medical image segmentation tasks. Unlike sample-level Dice loss, we adopted batch-level Dice loss, treating the entire batch as a single sample rather than calculating the average of dice scores for each sample within each minibatch. This batch processing Dice loss helps stabilize training by reducing errors on samples with few annotations. In terms of optimization, we trained the network using stochastic gradient descent with Nesterov momentum set to 0.99. The initial learning rate was set to 0.01 and decayed according to polynomial scheduling. Each training session lasted 3000 epochs, with each epoch containing 250 mini-batches. We used the dice score from the current fold cross-validation set to monitor the training progress in real-time, ensuring that the model achieved the desired performance during training. Three different models were developed in our study, which are:

* nnUNet-base: This is the original nnUNet model with no special modifications or enhancements.
* nnUNet-Att: Based on nnUNet-base, pseudo-labels are generated and a 3D axial attention mechanism is added after the second layer of the decoder in nnUNet. The model is trained on the union of labeled data and selected unlabeled data with pseudo-labels.
* nnUNet-Att-p: On the basis of nnUNet-Att, a position correction module is further added to correct the predicted segmentation anomaly points.

# **3. Results and discussion**

Our method has achieved remarkable results in the 3D tooth segmentation task of MICCAI Workshop. Our model performs well on the test set and has good segmentation accuracy and robustness compared with other competing methods.

Table 2 Comparison of prediction performance of different models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Dice** | **IOU** | ***Hausdorff distance*** | **Score** |
| nnUNet-base | 0.6669 | 0.7497 | 0.2921 | 0.7041 |
| nnUNet-Att | 0.7556 | 0.8045 | 0.2521 | 0.7680 |
| nnUNet-Att-p | **0.7959** | **0.8307** | **0.1922** | **0.8099** |

Table 2 shows the prediction performance of three models for 3D dental segmentation. Model nnUNet-base was trained on the entire training dataset (1-fold). As can be seen from Table 2, compared to nnUNet-base, nnUNet-Att achieved significant improvements across all metrics. This is partly due to nnUNet-base being trained only once, but more importantly, the addition of pseudo-labels and the 3D axial attention mechanism played a crucial role. As shown in Figure 5, since the unlabeled data provided by the competition were manually cropped to include only partial teeth, generating pseudo-labels for these data significantly expanded the sample size, which is essential for extracting the contours, textures, and shapes of the teeth. However, pseudo-labels are a simple and conventional semi-supervised learning method, and even after selecting pseudo-labels based on uncertainty, they may still contain noise. Therefore, we added a 3D axial attention mechanism to make the model focus more on specific structures of the teeth when processing vertical and horizontal structures, while ignoring noise information. The combination of pseudo-labels and 3D axial attention significantly improved the model's performance.

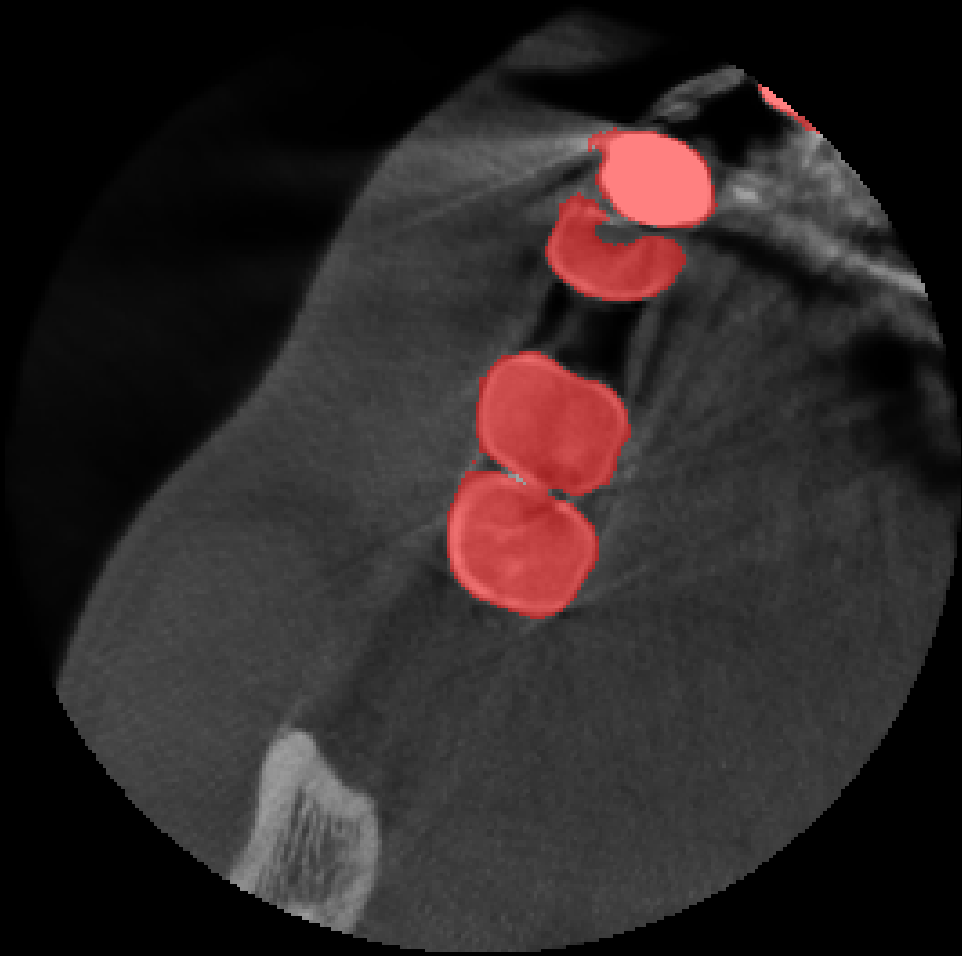


Figure 5: Segmentation sample of unlabeled data

The nnUNet-Att-p model shows significant performance improvements over the nnUNet-Att model across all four evaluation metrics: Dice Score improves by about 4.03%, IOU by about 2.62%, Hausdorff distance by about 5.99%, and overall score by about 4.19%. As analyzed in Section 2.1.3, the shape and density of the foramen magnum and similar areas to teeth lead the model to incorrectly segment these regions as teeth. The addition of a position correction module has corrected these erroneous segmentations, with results before and after using the position correction module shown in Figure 6. Experimental results indicate that the nnUNet-Att-p model demonstrates higher accuracy and precision in 3D tooth segmentation compared to the nnUNet-Att model, providing a superior solution for segmentation tasks.



A) Segmentation example before using the position correction module b) Segmentation example after using the position correction module A) Segmentation example before using the position correction module b) Segmentation example after using the position correction module

Figure 6: Comparison before and after using the position correction module

We designed a framework based on nnU-Net, using unlabeled data for training and effective inference. By modifying the second layer decoder of nnUNet and introducing a 3D axial attention mechanism, we improved segmentation performance. Our training techniques and innovative ideas have achieved significant success in this task and are expected to be applied in other areas of medical image segmentation. We believe that the framework we propose can serve as a good baseline for semi-supervised learning and efficient inference in medical image segmentation. We will continue to improve and optimize our method to further enhance the performance and efficiency of medical image segmentation.

The authors of this paper state that the segmentation method implemented for the MICCAI Workshop 2023 challenge did not use any pre-training models, nor any data sets other than those provided by the organizers. The proposed solution is fully automated and requires no human intervention.

# **4. References**

1. Isensee, Fabian, et al. "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation." Nature methods 18.2 (2021): 203-211.
2. Ho, Jonathan, et al. "Axial attention in multidimensional transformers." arXiv preprint arXiv:1912.12180 (2019).
3. Luu, Huan Minh, and Sung-Hong Park. "Extending nn-UNet for brain tumor segmentation." International MICCAI Brainlesion Workshop. Cham: Springer International Publishing, 2021.
4. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
5. Woo, Sanghyun, et al. "Cbam: Convolutional block attention module." Proceedings of the European conference on computer vision (ECCV). 2018.
6. Akobo, Seleipiri, et al. "The odontoid process: a comprehensive review of its anatomy, embryology, and variations." Child's Nervous System 31 (2015): 2025-2034.
7. Lee, Dong-Hyun. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks." Workshop on challenges in representation learning, ICML. Vol. 3. No. 2. 2013.
8. Huang, Ziyan, et al. "Revisiting nnU-net for iterative pseudo labeling and efficient sliding window inference." MICCAI Challenge on Fast and Low-Resource Semi-supervised Abdominal Organ Segmentation. Cham: Springer Nature Switzerland, 2022. 178-189.