Precision Tooth Segmentation Using CBCT: From Coarse Structures to Fine Details

Chenglong Ma[1] and Xukai Liu[1]

Chohotech Ltd., China, Hangzhou
mchenglong@chohotech.com

Abstract. Teeth segmentation in cone beam computed tomography (CBCT) images presents significant challenges due to the large number of voxels in their volumetric representations, which demand considerable computational resources. Furthermore, the high level of accuracy required in medical imaging exacerbates these challenges. In this paper, we introduce a novel 3-stage method designed to overcome these computational and accuracy challenges. Our approach effectively overcomes the unique complexities of CBCT data, producing highly accurate segmentation results. Our method achieved an average Instance Dice score of 0.922 and an average NSD of 0.966 for the teeth segmentation on the testing set using a GeForce RTX 3090 (24G). The average running time was 61 seconds for one image.

Keywords: CBCT · Instance Segmentation · Medical Imaging.

1 Introduction

Teeth segmentation in general, is a crucial step in various dental and orthodontic procedures, including diagnosis, treatment planning, and surgical interventions. Accurate segmentation involves precise identification of individual teeth and their respective boundaries, it serves as an initial input to numerous downstream tasks, and aiding professionals in crucial medical procedures, such as implant placement, root canal treatment, and orthodontic analysis.

However, the challenge in this task remains paramount, Cone Beam Computed Tomography (CBCT) has become a popular imaging modality in dental and maxillofacial applications due to its ability to provide high-resolution, 3D volumetric data with lower radiation doses compared to conventional CT. This unfortunately means it comes with an enormous amount of data, as the data composes of millions of voxels, which lead to dramatically larger image size compared to a regular image such as a panoramic X-Ray. Performing a 2D task via naively extending the model input dimension is implausible therefore, alternatively, simply down-sampling the input would complete dispose of its unique advantage in high resolution. A second challenge comes with the complicated anatomical structure of the human head itself. Any minute structures in the head which is unrelated to a specific task at hand serve as natural complication.

The teeth themselves are also quite complicated, they vary in sizes, orientations, and shapes, often overlapping with surrounding structures such as bone or soft tissues. On the opposed end of the spectrum, some of them present very little discernible differences structure-wise, presenting challenges in differentiation.

Historically, the segmentation of teeth from medical images has been performed manually by experts, which is time-consuming, labor-intensive, and prone to variability between operators. This led to the development of semi-automated and fully automated methods aimed at reducing the burden on clinicians.

3D CNNs have been applied to leverage the volumetric nature of CBCT images. These models process 3D patches of data to better capture spatial relationships in all three dimensions. However, the increased complexity and computational demand of 3D CNNs pose practical challenges, especially when processing large CBCT volumes.

In this solution, we devise a novel 3-stage pipeline.

In the first stage, we utilize a simple 2d detector to detect the region containing all the tooth. For this purpose, we generate Maximum Intensity Projections (MIPs), then use them as inputs to the detector. This allows us to discard much of the unrelated information in the scan, with a slight labelling overhead. However, in the later stages, this proves to allow us much more concentrated computing power and analysis focused on only the most relevant data.

In the next stage, we perform object detection on the cropped-out 3D volumetric data, using a model adapted from ultralytics yolov8[3]. We extensively modified the model to give it 3D functionality, this involves replacing all operators to a customized 3D variant that would function logically as before, and designing a special anchor strategy to replace the original 2D strategy. However, even with our previous optimizations, we are still required to sacrifice some resolution, with several other factors, the tooth boundaries on this model can still not achieve satisfactory results. Thus we introduce a third stage, with a simple U-Net that optimizes the tooth boundaries. Finally we will improve generalization of all our models using a pseudo-label generating strategy.

2 Method

We regard CBCT tooth segmentation as a classic instance segmentation task, with extensive research established in the field of computer vision, exemplified by methods such as Mask-RCNN and SOLO. Our objective is to develop an efficient solution for 3D CBCT tooth instance segmentation. Through comparative analysis and practical experience, we have chosen to implement the relevant network within the Ultralytics framework.

In contrast to 2D imaging data, 3D object detection imposes significantly greater computational demands on GPU resources, which poses a considerable challenge for effective network training. Initially, we observed that certain CBCT images exhibit substantial background areas. To mitigate the computational overhead associated with these backgrounds, we trained a tooth region detection network to effectively crop the areas of interest. Subsequently, we developed

a 3D tooth instance segmentation network based on YOLOv8. However, due to structural design constraints, the segmentation branch of YOLOv8 was unable to fully exploit low-level features inherent in medical imaging. To rectify this limitation, we ultimately implemented a boundary optimization network for individual teeth utilizing the U-Net architecture.

Consequently, our proposed methodology comprises three independent networks: the teeth region detection network, the teeth instance segmentation network, and the tooth boundary optimization network, as illustrated in the accompanying figure.

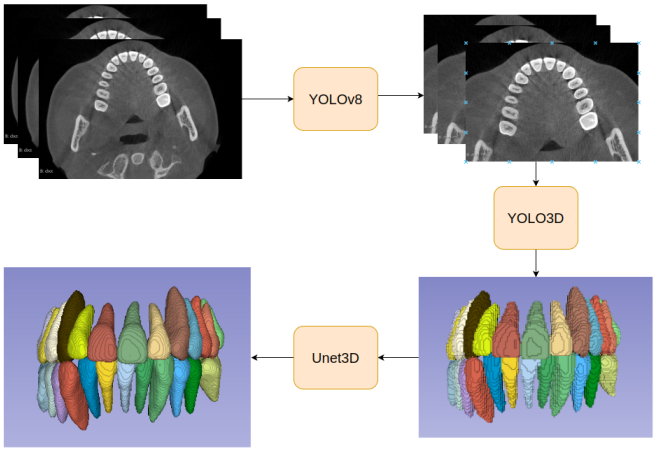

Fig. 1. Enter Caption

2.1 Teeth Region Detection Network

In this sub-network, we first pre-process the CBCT scans by projecting them into three orthogonal directions, namely axial, coronal, and sagittal. With this process, we generate the respective Maximum Intensity Projections. This essentially compresses the data into 3 images that contain all the information we require to locate the region where teeth are. In order to improve the differentiation between teeth and surrounding tissues, we then apply a equalized_adapthist supplied via *Scikit-Image*[4].

The accompanying figures will show these projections in each orthogonal direction. Once the projections are generated for all samples in the dataset, we use the *Labelme* annotation tool to manually label the tooth regions. For consistency and simplicity, we annotate these regions using bounding boxes.

Once we have pre-processed the data, the model itself would become a rather trivial network, here we utilize YOLOv8 from *ultralytics* to implement a simple

tooth region detection network, the output of this model is then used as the input to our next stage.

2.2 Teeth Instance Segmentation Network

Based on the results of the detection network from the first step, we scale the bounding boxes of the localized teeth by a certain factor and then crop the corresponding images. To avoid the interference of outliers in the dataset,We normalize HU from -1000 to 3000 to the range of -1 to 2 and clip the out-of-bounds values.

We modified Ultralytics to support 3D instance segmentation, specifically including the replacement of operators and the adaptation of anchor assigner strategies.

2.3 Tooth Refinement Network

We use the trained instance segmentation model to perform inference on all labeled data and compare the inference results with the ground truth. Teeth with an IOU greater than 0.6 are selected. The corresponding teeth images are cropped, and this is used as the final result for optimizing the tooth contours. The data pre-processing method is consistent with that in the "Teeth Instance Segmentation Network." Here we focus exclusively on the boundary of the tooth and disregard the rest. We pre-process the input to this model, and a tooth segmentation from the previous stage will become exclusively a representation of its boundaries. During inference, this result will be used to refine our raw coarse segmentation. The loss function used is a combination of binary cross-entropy loss and dice loss.

Essentially, this model and its operations comprises our post-processing operations.

3 Experiments

3.1 Dataset

The training set comprises 330 CBCT images, of which 30 are labeled and 300 are unlabeled[6][2][1]. The labeled cases contain detailed annotations for individual teeth, including corresponding Fédération Dentaire Internationale (FDI) tooth numbers, provided in the nii.gz format. The unlabeled cases, on the other hand, consist only of the images without any annotations. The validation set includes 20 CBCT images, which are used during the validation phase to evaluate algorithm performance metrics.

3.2 Evaluation metrics

In the STS 2024 challenge, the evaluation criteria have been expanded beyond segmentation accuracy to also account for algorithm efficiency and resource consumption, providing a more comprehensive assessment of performance. The primary segmentation accuracy metrics include the Dice Similarity Coefficient (DSC), Normalized Surface Distance (NSD), mean Intersection-over-Union (mIoU), and Identification Accuracy (IA), all of which are evaluated at both the instance and image levels. These metrics ensure a detailed analysis of how well the algorithms segment and identify objects across different granularities. Additionally, the challenge introduces efficiency metrics, such as inference running time and GPU memory consumption, encouraging participants to design algorithms that balance accuracy with computational efficiency. This holistic approach aims to optimize both the precision and practicality of segmentation models, with the final ranking being determined by a combination of accuracy and efficiency-based metrics.

3.3 Implementation details

Environment settings The development environment and system specifications are summarized in Table **??**. The system runs Ubuntu 18.04.5 LTS with an Intel(R) Xeon(R) Gold 6240R CPU at 2.40GHz. It is equipped with 128GB of RAM and four NVIDIA A100 80G GPUs. CUDA version 11.7 is installed to support GPU computations. Python 3.8.0 is used as the programming language, alongside the deep learning framework PyTorch (torch 2.0.1). These specifications outline the hardware and software environment for this study.

Table 1. Development environments and requirements.

System	Ubuntu 22.04.1 LTS
CPU	Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz
RAM	128GB
GPU (number and type)	Four NVIDIA A100 80G
CUDA version	11.7
Programming language	Python 3.8.0
Deep learning framework	torch 2.0.1

4 Results and discussion

4.1 Results

Our proposed method was evaluated on the testing set of the STS Challenge, and the results demonstrated outstanding performance across several key met-

Table 2. Training protocols.

Batch size	4
Patch size	Divisible Pad
Total epochs	400
Optimizer	Adam
Initial learning rate (lr)	0.001
Lr decay schedule	Cosine annealing
Loss function	3D yolov8 Loss
Number of model parameters	4.6M

Table 3. Training protocols for the boundary refinement model

Network initialization	
Batch size	8
Patch size	Divisible Pad
Total epochs	100
Optimizer	Adam
Initial learning rate (lr)	0.001
Lr decay schedule	Cosine annealing
Loss function	3D yolov8 Loss
Number of model parameters	4.8M

rics. As presented in Table 4 The final results show that our approach demonstrates advantages across several metrics. In instance-related metrics, such as tooth identification accuracy, the instance-based segmentation method proves its superiority. Semantic segmentation methods, which typically make pixel-level predictions, often require complex post-processing to remove false positives. This issue is particularly prominent in small-scale datasets, where undertrained models like Unet can struggle to accurately distinguish between different teeth. In contrast, our method first separates teeth at the instance level and then further refines their contours. By leveraging instance-level information, our approach not only generates more accurate results but also eliminates the need for complex post-processing.

Table 4. Quantitative evaluation results.

Metrics\Teams	ChohoTech	houwentai	madongdong	jichangkai	junqiangmler
Dice instance	0.922	0.887	0.779	0.733	0.658
Dice image	0.935	0.899	0.836	0.765	0.774
NSD instance	0.966	0.914	0.736	0.788	0.667
NSD image	0.974	0.922	0.809	0.853	0.774
mIoU instance	0.863	0.849	0.65	0.681	0.584
mIoU image	0.879	0.862	0.719	0.695	0.685
Identification Accuracy	0.984	0.922	0.882	0.724	0.656
Time	61	210	53	215	114
GPU Consumption	233660	829283	48267	377331	1004508

5 Conclusion

In this paper, we introduced a novel 3-stage method for teeth segmentation in CBCT images that effectively addresses the challenges posed by the large volume of data and the need for precise accuracy in medical imaging.

The key strengths of our method lie in its ability to balance computational speed with high segmentation accuracy. By carefully combining detection, instance segmentation, and post-processing steps, we successfully minimized errors in tooth boundaries.

The results of this study highlight the potential for this method to be applied in real-world clinical settings, where quick and precise segmentation is essential for treatment planning and decision-making.

Winning first place in the STS Challenge underscores the strength of our method, validating its effectiveness and relevance to the field of medical imaging.

Acknowledgements. The authors of this paper declare that the segmentation method they implemented for participation in the STS 2024 challenge has not used any additional datasets other than those provided by the organizers. The proposed solution is fully automatic without any manual intervention. We thank all the data owners for making the X-ray images and CT scans publicly available and Codebench [5] for hosting the challenge platform.

References

1. Cui, W., Wang, Y., Li, Y., Song, D., Zuo, X., Wang, J., Zhang, Y., Zhou, H., Chong, B.s., Zeng, L., et al.: Ctooth+: A large-scale dental cone beam computed tomography dataset and benchmark for tooth volume segmentation. In: MICCAI Workshop on Data Augmentation, Labelling, and Imperfections. pp. 64–73. Springer (2022) 4

2. Cui, W., Wang, Y., Zhang, Q., Zhou, H., Song, D., Zuo, X., Jia, G., Zeng, L.: Ctooth: a fully annotated 3d dataset and benchmark for tooth volume segmentation on cone beam computed tomography images. In: International Conference on Intelligent Robotics and Applications. pp. 191–200. Springer (2022) 4
3. Jocher, G., Qiu, J., Chaurasia, A.: Ultralytics YOLO (Jan 2023), https://github.com/ultralytics/ultralytics 2
4. Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: scikit-image: image processing in python. PeerJ **2**, e453 (2014) 3
5. Xu, Z., Escalera, S., Pavao, A., Richard, M., Tu, W.W., Yao, Q., Zhao, H., Guyon, I.: Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. Patterns **3**(7) (2022) 7
6. Zhang, Y., Ye, F., Chen, L., Xu, F., Chen, X., Wu, H., Cao, M., Li, Y., Wang, Y., Huang, X.: Children's dental panoramic radiographs dataset for caries segmentation and dental disease detection. Scientific Data **10**(1), 380 (2023) 4