A Self-Training Pipeline for Semi-Supervised 2D Teeth Instance Segmentation

Kaiwen Fu[0009−0001−9008−5268], Chengyuan Chang[0009−0007−8983−7569], Jiahui Chen[0009−0009−5624−9124], and Qinjie Hu[0009−0004−7610−900X]

School of Artificial Intelligence, Xidian University, Xi'an, China
{23171214520}@stu.xidian.edu.cn

Abstract. The 2D panoramic X-ray image examination is an efficient way for dentists to determine invisible caries, impacted teeth, and supernumerary teeth among children. However, identifying and manually annotating teeth in panoramic X-ray images is time consuming and labor intensive, which limits the availability of labeled cases and hinders the development of deep learning algorithms for tooth segmentation and disease analysis. In this work, we propose a self-training pipeline for semi-supervised 2D teeth instance segmentation. Our pipeline employ a multi-model ensemble strategy and morphological operations to generate more accurate pseudo-labels for self-training. Furthermore, we expand the output channels of the segmentation model to better handle overlapping regions among the teeth. Our method achieved average scores on the validation set of 77.55% for instance-level DSC, 89.57% for image-level DSC, 82.05% for instance-level NSD, 93.12% for image-level NSD, 69.02% for instance-level mIoU, 81.19% for image-level mIoU, and 79.36% for identification accuracy. Our method achieved an average inference speed of 0.161 seconds per image on a NVIDIA GeForce RTX 4090. Our code is available at https://github.com/Liaaaar/2024-MICCAI-STS-2D.

Keywords: Self-Training · Instance Segmentation.

1 Introduction

Computer-aided diagnosis tools are gaining popularity in modern dental practice, particularly for treatment planning and comprehensive prognosis evaluation. Among these, 2D panoramic X-ray imaging is an efficient method for dentists to detect hidden caries, impacted teeth, and supernumerary teeth in children. However, identifying teeth from panoramic X-ray images and manually annotating them is time-consuming and labor-intensive. As a result, the availability of a large number of labeled cases is often limited, which hinders the development of deep learning algorithms for tooth segmentation and automated disease analysis. Semi-supervised learning offers a promising solution by leveraging useful information from unlabeled cases. Numerous advanced semi-supervised image segmentation methods have been proposed, yielding impressive results. However, developing a semi-supervised algorithm for teeth instance segmentation remains a highly challenging task. First, due to privacy concerns and

the high cost of annotation, the number of labeled cases is extremely limited, restricting the amount of supervised information available for effectively training the model. Second, the task involves the segmentation of 52 distinct tooth categories, some of which exhibit overlap, further complicating accurate segmentation and precise classification of each individual tooth.

Semi-supervised image segmentation has gained significant attention in recent years due to its ability to leverage a large amount of unlabeled data alongside a limited set of labeled samples, reducing the dependency on costly manual annotation. More recent advancements have focused on deep learning techniques. Methods like consistency regularization [7] and entropy minimization [4] have demonstrated promising results by enforcing model predictions to be consistent across perturbed versions of input data and reducing prediction uncertainty on unlabeled samples. Additionally, the use of pseudo-labeling [1] has become a popular approach, where the model generates labels for unlabeled data during training to iteratively refine the segmentation boundaries. This technique, known as self-training [2] , allows the model to gradually improve by leveraging the pseudo-labeled data. Despite their effectiveness, existing methods are primarily designed for semantic segmentation, making them difficult to apply in instance segmentation scenarios where instances may overlap.

To address these problems, we propose a self-training pipeline for semi-supervised 2D teeth instance segmentation. We take advantage of existing self-training framework and propose two improvements. First, we propose a pseudo-label generation strategy based on multi-model ensemble and morphological operations to produce more reliable pseudo-labels for self-training. Additionally, to address the issue of instance overlap, we expand the output channels of the segmentation model to more accurately segment each instance. Our algorithm maintains fast inference speed while achieving high segmentation accuracy. On the validation set, we achieved results of 77.55% for instance-level DSC, 89.57% for image-level DSC, 82.05% for instance-level NSD, 93.12% for image-level NSD, 69.02% for instance-level mIoU, 81.19% for image-level mIoU, and 79.36% for identification accuracy, with an average inference time of only 0.161 seconds per image on a NVIDIA GeForce RTX 4090.

We summarize our contributions as follows:

- We propose a simple yet effective semi-supervised self-training pipeline for 2D teeth instance segmentation, which leverages information from unlabeled images to enhance the segmentation of teeth.
- We propose a pseudo-label generation strategy based on multi-model ensemble and morphological operations for self-training.
- Our algorithm achieves competitive results on the validation set while maintaining a fast inference speed, making it practical for dental examinations.

2 Method

In this section, we will provide a detailed description of our method, including preprocessing, our proposed self-training pipeline, and post-processing.

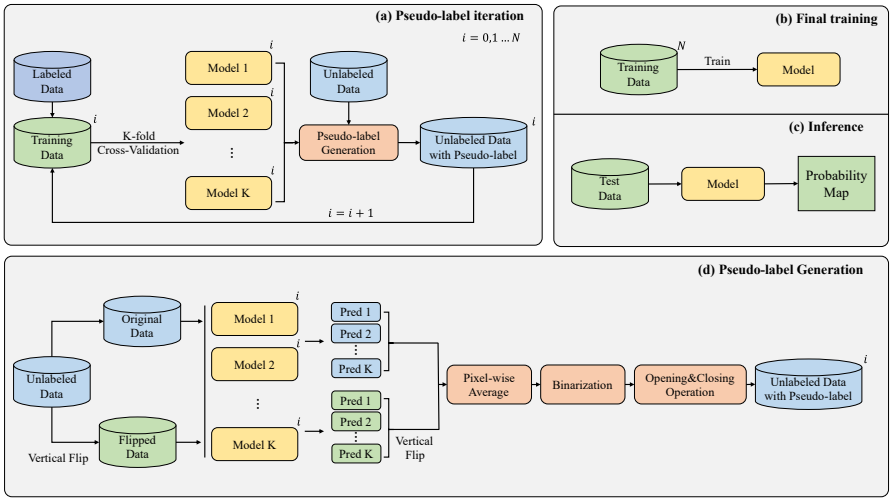

Fig. 1. The framework of the self-training pipeline we proposed. (a) Iteratively refining the pseudo-labels to enhance their accuracy. (b) Train the final model using the unlabeled data, the final pseudo-labels, and the labeled data. (c) Perform inference with the final model to generate the probability map for segmentation. (d) Our proposed pseudo-label generation strategy. We employ a multi-model ensemble strategy and morphological operations to generate more accurate pseudo-labels.

2.1 Preprocessing

The images in the Semi-supervised Teeth Segmentation MICCAI Challenge dataset are grayscale, with pixel values ranging between [0, 255], and come in two resolutions: 1127×1991 and 942×2000. First, to standardize the image size and reduce computational load, we resized all images to 320×640. Subsequently, we divided the images by 255 to normalize them to the range [0, 1].

There are 52 categories of teeth. To facilitate training, we organize the tooth numbering in ascending order and map them to category labels ranging from 0 to 51. For each labeled image, we create a mask with 52 channels initialized to zero, and then sequentially populate each channel based on the contour information of the teeth. In this way, we obtain the masks used for training.

2.2 Proposed Method

The self-training pipeline we proposed is illustrated in Fig. 1. Our training consists of two stages: pseudo-label iteration and final training. As illustrated in Fig. 1(a), in the pseudo-label iteration phase, we first train initial models using only the labeled data and employ these models to generate pseudo-labels. Second, the unlabeled data, along with their associated pseudo-labels, are integrated into the training set to retrain new models. These new models are then utilized to further refine the pseudo-labels. The second step is repeated N times

to continuously enhance the quality of the pseudo-labels. In this phase, we employ K-fold cross-validation, meaning that each iteration trains K new models, with K set to 5. After completing N iterations, we utilize the unlabeled data with pseudo-labels and the labeled data to train a single final model, as illustrated in Fig. 1(b), which we refer to as final training. At last, we utilize the final model to infer the test data.

The specific pseudo-label generation process is shown in Fig. 1(d). To improve accuracy, we use a multi-model ensemble strategy and morphological operations. Given an input image, the image is vertically flipped and subsequently processed through K models, resulting in a total of $2K$ probability maps. These $2K$ probability maps are averaged pixel-wise, followed by a thresholding operation using a threshold of 0.5 to produce an initial label. By utilizing model ensemble and test time augmentation, we can reduce the variance of the predictions, thereby enhancing the quality of the pseudo-labels. Finally, morphological opening and closing operations are performed on the initial label channel by channel, with a disk size of 5, to eliminate small area noise in the pseudo-labels.

In this pipeline, we utilize DeepLabv3+ [3] as our model. To address the overlapping regions between teeth, we replace the final softmax function of the network with a sigmoid function, resulting in an output of 52 channels. Each channel is dedicated to segmenting a specific category of teeth.

2.3 Inference and Post-processing

During the inference stage, we resize the input image to 320×640 and subsequently input it into the model, resulting in a probability map with a shape of $52 \times 320 \times 640$. We utilize test time augmentation by performing inference twice for each image—once on the original image and once on its vertically flipped version. The final probability map is computed as the mean of the two outputs.

After obtaining the probability map, we first resize it back to the original image dimensions and then apply a threshold of 0.5 for binarization. Next, we use the findContours function in OpenCV to identify connected regions channel by channel, thereby obtaining the segmentation contours for each tooth. To filter out noise in the results, we remove regions with contour points fewer than 45. Finally, we remap the tooth categories from 0 to 51 back to the original tooth numbering and write the results into a JSON file to obtain the final output.

3 Experiments

3.1 Dataset

The dataset [9] for the 2D semi-supervised teeth segmentation challenge contains panoramic images of both children and adult dentition, with a relatively fewer number of images for children patients. The training set includes 2,380 panoramic X-ray images, comprising 30 cases with pixel-level instance labels and 2,350 unlabeled cases, while the validation set contains 20 panoramic X-ray images. The images in the dataset are grayscale, with resolutions of 1127×1991 and 940×2000, encompassing a total of 52 dental categories.

3.2 Evaluation metrics

The segmentation inference results are evaluated using several metrics, including Dice Similarity Coefficient (DSC), Normalized Surface Dice (NSD), Intersection over Union (IoU), Identification Accuracy (IA), Running Time (RT), and Area under the GPU Memory-Time Curve (GPU-area). DSC and IoU are employed to measure regional errors, while NSD assesses boundary errors. The IA metric evaluates object-level localization performance for teeth. RT measures inference speed, and the GPU utilization-time curve assesses GPU consumption.

3.3 Implementation details

Environment settings We develop our algorithm on a Linux machine, with the specific development environments and requirements detailed in Table 1.

Table 1. Development environments and requirements.

System	Ubuntu 20.04.6 LTS
CPU	Intel(R) Xeon(R) Gold 6326 CPU@2.90GHz
RAM	8 × 32GB; 3200MT/s
GPU (number and type)	1 × NVIDIA RTX 4090 24G
CUDA version	11.8
Programming language	Python 3.8.11
Deep learning framework	Torch 1.13.1, Torchvision 0.14.1
Specific dependencies	Opencv-Python 4.6.0
Code	VSCode, XShell 7

Training protocols In all our experiments, we utilize the DeepLabV3+ segmentation network, employing ResNet50 [5] as the backbone. The specific training protocols and hyperparameter settings are illustrated in Table 2. We utilize a combination of 0.2 times Dice loss and 0.8 times Binary Cross-Entropy (BCE) loss as our loss function, which has been confirmed to be the most effective in the ablation study presented in Table 4. We default to training for 200 epochs; however, when using only labeled data for training, we adjust the epochs to 1000. In K-fold cross-validation, we employ an early stopping strategy, retaining only the model that achieves the highest Dice coefficient on the validation set. In other experiments, we retain only the model from the last epoch. We use exponential moving average (EMA) to smooth the model parameters, with a momentum coefficient set to 0.999. During training, we employ data augmentation techniques, including vertical flipping, as well as a combination of random gamma adjustment, brightness and contrast enhancement, blurring, and optical

distortion. Furthermore, we incorporate elastic transformation, grid distortion, motion blur, and hue saturation adjustments. These methods introduce variability in the training data, thereby enhancing the model's robustness.

Table 2. Training protocols and hyperparameter settings.

Network initialization	He initialization
Batch size	4
Total epochs	200
Optimizer	AdamW [6]
Betas	(0.9, 0.999)
Weight decay	0.01
Initial learning rate (lr)	0.001
Lr decay schedule	CosineAnnealingLR
Loss function	Dice and BCE
Number of model parameters	26.69M
Number of flops	29.11G

4 Results and discussion

4.1 Quantitative results on validation set

The quantitative results on the validation set are shown in Table 3. When training only with labeled data, the scores for image-level Dice, image-level IoU, image-level NSD, instance-level Dice, instance-level IoU, instance-level NSD, and IA are 87.86%, 78.49%, 91.36%, 69.09%, 64.71%, 77.98%, and 72.14%, respectively, and our overall score is 77.38%. In our self-training pipeline, we generate pseudo-labels for unlabeled data and use these to augment the training set. This is a simple yet effective strategy. Through continuous iterations, we can obtain more reliable pseudo-labels, leading to the training of models with enhanced performance. As shown in Table 3, the model's performance progressively improves with the increasing number of pseudo-labeling iterations N. When $N = 3$, the scores for image-level Dice, image-level IoU, image-level NSD, instance-level Dice, instance-level IoU, instance-level NSD, and IA are 89.57%, 81.19%, 93.12%, 77.54%, 69.02%, 82.05%, and 79.36%, respectively, and our overall score is 81.70%. When $N = 3$, the overall score of DeepLabV3+ improves by 4.32% compared to training with only labeled data, and as N increases, the model's performance may further improve.

Additionally, at the beginning of the training, we examined the impact of the loss function to determine subsequent training strategies, with results presented in Table 4. The combination of 0.2 times Dice loss and 0.8 times BCE loss yielded the best results, and we used this loss function in subsequent training.

Table 3. Quantitative results on validation sets. Supervised denotes training conducted solely with labeled data, while Ours refers to our self-training pipeline, where N represents the number of iterations for pseudo-labels.

Method	image-level			instance-level				Avg.(%)
	Dice(%)	IoU(%)	NSD(%)	Dice(%)	IoU(%)	NSD(%)	IA(%)	
Supervised	87.86	78.49	91.36	69.09	64.71	77.98	72.14	77.38
Ours (N=0)	88.65	79.77	92.05	69.42	67.72	80.68	77.47	79.40
Ours (N=1)	88.67	79.80	92.03	73.77	67.65	80.53	76.72	79.88
Ours (N=2)	89.41	80.93	92.95	**77.80**	68.58	81.65	78.80	81.45
Ours (N=3)	**89.57**	**81.19**	**93.12**	77.54	**69.02**	**82.05**	**79.36**	**81.70**

Table 4. The impact of the loss function when training solely with labeled data.

Loss function	image-level			instance-level				Avg.(%)
	Dice(%)	IoU(%)	NSD(%)	Dice(%)	IoU(%)	NSD(%)	IA(%)	
Dice	88.05	78.82	91.39	61.72	66.35	79.06	59.15	74.93
0.5Dice+0.5BCE	88.41	79.35	91.83	60.18	64.98	77.69	72.34	76.40
0.2Dice+0.8BCE	87.86	78.49	91.36	**69.09**	64.71	77.98	72.14	**77.38**
BCE	**89.22**	**80.69**	**92.76**	46.36	**67.69**	**80.07**	**73.80**	75.80

4.2 Qualitative results on validation set

Our visualization results are shown in Fig. 2. The first two rows are examples of easy samples, while the last two rows illustrate hard samples. In two simple samples, the teeth appear relatively aligned, and there are many similar images in the training set. Regardless of whether unlabeled data is used, good segmentation results can be achieved. In the two challenging samples, the teeth are misaligned to some extent, and these types of images are relatively scarce in the training set, making accurate segmentation and classification quite difficult. Through the visualization results, we can see that there is a significant reduction in false positives and the classification results become more accurate when trained using unlabelled data. This is mainly attributed to our pseudo-label generation and iterative strategy. The segmentation accuracy improves with an increase in the number of iterations. However, our pseudo-label generation strategy also introduces certain flaws. The use of opening and closing operations causes the model to lean towards obtaining smoothed predictions, leading to a decline in the segmentation quality for certain details, such as tooth roots.

4.3 Results on final testing set

We obtained test set scores of 77.55% for instance-level DSC, 89.57% for image-level DSC, 82.05% for instance-level NSD, 93.12% for image-level NSD, 69.02%

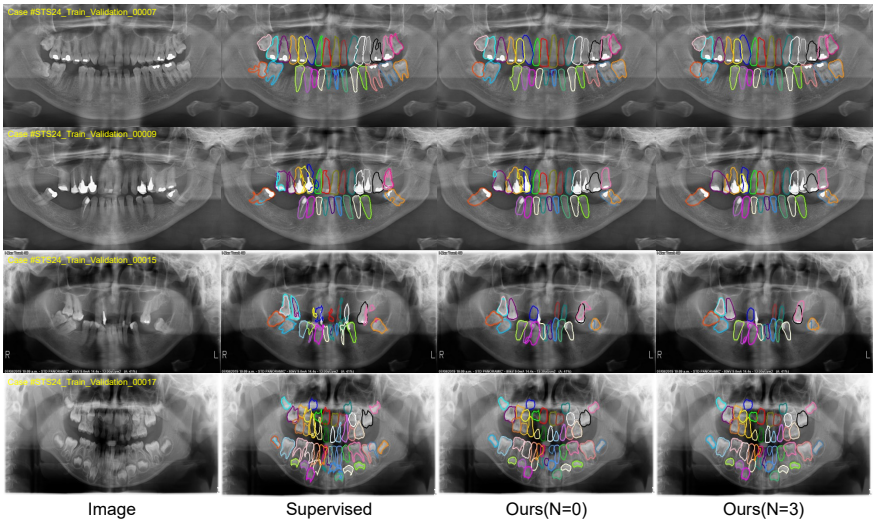

Fig. 2. Qualitative results on easy (case Validation 0007 and 0009) and hard (case Validation 0015 and 0017) examples. N represents the number of pseudo-label iteration.

for instance-level IoU, 81.19% for image-level IoU, and 79.36% for identification accuracy. The average time latency on the test set was 13.27 seconds, with an average area under the GPU memory-time curve of 14,250.98 MB·s.

4.4 Limitation and future work

Despite achieving competitive results, our method has certain limitations. First, the use of morphological opening and closing operations in our proposed pseudo-label generation strategy can lead to overly smooth pseudo-labels, which may weaken the model's ability to segment fine structures, such as dental roots. Additionally, the cost of generating iterative pseudo-labels is relatively high due to our use of 5-fold cross-validation. In future work, we plan to introduce uncertainty metrics to retain only high-confidence pseudo-labels, thereby better mitigating noise in the pseudo-labels.

5 Conclusion

In this work, we propose a self-training pipeline for semi-supervised 2D teeth instance segmentation. Through our proposed pseudo-label generation strategy based on multi-model ensemble and morphological operations, we can generate pseudo-labels for unlabeled data and gradually optimize the quality of the pseudo-labels through continuous iterations, thereby better utilizing the information in the unlabeled data. Our method achieved competitive results on the online validation and test sets, and possesses a high inference speed.

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 [8] for hosting the challenge platform.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: 2020 International joint conference on neural networks (IJCNN). pp. 1–8. IEEE (2020) 2
2. Cascante-Bonilla, P., Tan, F., Qi, Y., Ordonez, V.: Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 6912–6920 (2021) 2
3. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018) 4
4. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. Advances in neural information processing systems **17** (2004) 2
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 5
6. Loshchilov, I.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 6
7. Luo, X., Wang, G., Liao, W., Chen, J., Song, T., Chen, Y., Zhang, S., Metaxas, D.N., Zhang, S.: Semi-supervised medical image segmentation via uncertainty rectified pyramid consistency. Medical Image Analysis **80**, 102517 (2022) 2
8. 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) 9
9. 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

Table 5. Checklist Table. Please fill out this checklist table in the answer column.

Requirements	Answer
A meaningful title	Yes
The number of authors (≤6)	4
Author affiliations and ORCID	Yes
Corresponding author email is presented	Yes
Validation scores are presented in the abstract	Yes
Introduction includes at least three parts: background, related work, and motivation	Yes
A pipeline/network figure is provided	3
Pre-processing	3
Strategies to use the partial label	3,4
Strategies to use the unlabeled images.	3,4
Strategies to improve model inference	4
Post-processing	4
The dataset and evaluation metric section are presented	4,5
Environment setting table is provided	5
Training protocol table is provided	6
Ablation study	7
Efficiency evaluation results are provided	8
Visualized segmentation example is provided	8
Limitation and future work are presented	Yes
Reference format is consistent.	Yes