Instance Segmentation of Teeth in Panoramic X-Ray Images Using YOLO v9

Caiyi Chen

College of Computer Science and Network Security Fujian Normal University,Fuzhou
Fujian 350117,P.R. China

Abstract. We stand on the shoulders of giants, utilizing the latest YOLOv9e-seg [1] model for panoramic instance segmentation. By appropriately selecting the segmentation model and fine-tuning predictions, we effectively address the small sample segmentation problem. Our method achieved an average Dice score of 0.8491%, an average mIoU of 0.7463, NSD_image of 0.882, and Identification Accuracy of 0.5403 on the validation set for the tooth segmentation task, using an NVIDIA GeForce RTX 3060. The average inference time per image is 0.8 seconds. In the classification of types, our method achieved a top ranking in the weighted score.

Keywords: YOLO · panoramic images · medical image segmentation

1 Introduction

1.1 Background and difficulty of this challenge

2D panoramic X-ray images and 3D dental cone-beam computed tomography (CBCT) examinations are effective methods for dentists to determine occult caries, impacted teeth, and supernumerary teeth in children. However, identifying teeth from panoramic X-ray images or CBCT scans and manually annotating them is both time-consuming and labor-intensive. Consequently, we often cannot obtain a large number of labeled cases, which limits the development of deep learning algorithms for tooth segmentation and automatic disease analysis. As a potential alternative, semi-supervised learning can explore useful information from unlabeled cases. The difficulty of this semi-supervised 2D panoramic image instance segmentation lies in:

(1)The classification types are numerous. There are 52 different types of teeth, and the differences among the same type of teeth in different individuals are small. For example, the shape of different types of adult teeth is highly similar, making it difficult to capture feature differences and posing a significant challenge for type identification.

(2)There is a scarcity of labeled data. The provided dataset only contains thirty panoramic images with complete labels, making it difficult to extract features for segmentation targets, and the correct segmentation of teeth from the background and the correct type labeling of different teeth are challenging.

(3)The evaluation of results considers multiple factors. When submitting for validation, it is necessary to consider both image-level and instance-level Dice similarity coefficients and IoU, as well as the model efficiency in terms of time and GPU memory consumption. This means that in addition to correctly segmenting teeth from the background, it is also necessary to consider the accurate typing of individual teeth and the prediction efficiency and GPU occupancy of the model. This makes the task not only a competition of accuracy but more like the proposal of a practical engineering task solution.

1.2 Related work

Medical image segmentation is an important part of computer vision tasks, especially in assisting doctors in diagnosis and helping patients determine their conditions. Early medical image segmentation mainly relied on traditional image processing techniques, such as threshold segmentation, region growing, edge detection, and other methods. With the popularization of deep learning methods, Convolutional Neural Networks (CNNs) have become a common approach in medical image segmentation tasks in recent years. Long J et al. [2] introduced the Fully Convolutional Network (FCN)into the encoder-decoder structure for medical image segmentation, using deconvolution and skip connections in the decoder to perform spatial upsampling and obtain more refined segmentation results. However, deep network structures can lead to varying degrees of loss of image details and structural information. To alleviate this issue, Ronneberger et al. [3] proposed the U-Net model for medical image segmentation based on FCN. This model is based on a symmetric encoder-decoder structure composed of convolutions, spatial downsampling, and spatial upsampling, and uses skip connections to directly combine feature maps from the encoder's lateral outputs with high-level semantic information at the decoder end. This allows the decoding module to recover the details lost due to downsampling during the encoding phase from top to bottom, ultimately generating the predicted segmentation map at the output end.

However, despite the significant progress of Convolutional Neural Networks in the field of medical image segmentation, the limitation of convolution kernel size makes it difficult to capture long-range contextual information, requiring the continuous stacking of network depth, which also leads to the loss of detail information. The success of Transformers [4] in the field of natural language processing has led many researchers to apply it to image processing. The Vision Transformer (ViT) [5]has proven that a pure Transformer-based architecture can achieve performance comparable to CNNs in image recognition tasks. The Swin Transformer [6] builds on the hierarchical feature map approach of CNNs and introduces a sliding window-based self-attention mechanism with linear complexity. UNETR [7], designed directly for three-dimensional volume data, replaces the entire U-Net encoder with a Transformer architecture, capable of capturing long-range dependencies.

However, when facing instance segmentation that requires precise distinction of object contours, consideration of object spatial distribution, and identification of individual object identities, it seems insufficient. Instance segmentation technology is actually a combination of object detection (determining object categories) and pixel-level classification (precisely distinguishing objects from the background). YOLO (You Only Look Once) [8] is a popular real-time object detection system that solves the object detection task as a regression problem. The core idea of the YOLO algorithm is to divide the input image into grids and process each grid to predict bounding boxes and class probabilities. A significant feature of YOLO is that it treats the entire object detection task as a single regression problem, rather than the traditional two-stage detection methods (such as the R-CNN series). Therefore, YOLO, with its end-to-end training and precise identification of types, has a significant advantage in handling the multi-type instance segmentation task in this competition.

1.3 My motivation and solution

Due to the large number of segmentation categories and the relatively small labeled dataset of only 30 cases, preserving as much feature information as possible is key to solving this problem. On the one hand, the mainstream application of YOLO is in object detection tasks, so it has an inherent advantage in the precise identification of target types, which is of great application value for the 52 different types of teeth segmentation in this task. On the other hand, YOLO v9 can effectively address the issue of information loss and has high parameter utilization and computational efficiency. This is of great significance for the time-weighted score. Therefore, the YOLOv9e-seg model selected in this paper can be well-suited to the complex task at hand.

2 Method

2.1 Preprocessing

Converting label data format: The YOLO annotation format for instance segmentation tasks is highly suitable for this competition. In the YOLO instance segmentation format, annotations are stored in separate .txt files corresponding to each image, with the image and annotation files sharing the same base filename. The data format provided by the competition is in JSON files, which will be processed into .txt files to match the YOLO input format. In the label, the format is as follows: <class-index> <x1> <y1> <x2> <y2> ... <xn> <yn>, representing the normalized bounding coordinates around the object segmentation area, ensuring that annotations can scale across images of different sizes, which is fully consistent with YOLO. The labels and images will be organized into two folders, with corresponding sets of images and labels using the same filenames. Subsequently, we will extract the category_id and the polygons of the current object instances for normalization in a loop.

We will configure the required dataset into a separate YAML file to fit the training pipeline of any YOLO model. This file contains paths to the training, validation, and testing image directories, as well as the class names and the number of classes in the dataset. The training set path, validation set path, class count, and class name list will be set accordingly. Finally, we will load the YOLOv9e model from Ultralytics to initiate training.

2.2 Main Method

This competition is an application task, so the approach is to select the appropriate model to accurately extract features for prediction.In the field of object detection, YOLOv9 is undoubtedly a significant advancement, deeply optimized based on Ultralytics YOLOv5. With its outstanding efficiency, accuracy, and adaptability, it has become a leader in the industry. Moreover, YOLOv9 embodies the spirit of community collaboration and open-source development, expanding research and practical applications in the field of image segmentation with remarkable results. The success of YOLO in various domains is attributed to its combination of two innovative technologies: Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN). Its excellent performance in this competition fully demonstrates its powerful capabilities.

The YOLOv9e-seg model is particularly well-suited for this task, cleverly utilizing the information bottleneck principle and reversible functions to effectively address information loss in deep learning, ensuring that important data is retained across layers. This innovative strategy not only enhances the structural efficiency of the model but also ensures precise detection capabilities, allowing for keen detail capture even in lightweight models. Thus, unlabeled images were not utilized.

Programmable Gradient Information (PGI): The core idea of PGI is to provide complete input information when computing the objective function for the target task, thereby obtaining reliable gradient information for updating network weights. It consists of a main branch, an auxiliary reversible branch, and multi-level auxiliary information. During inference, only the main branch is used, while the auxiliary reversible branch addresses detail loss due to downsampling. This design reduces information loss during the training of deep networks and mitigates the gradient bottleneck that may arise from increased network depth, preventing the loss function from failing to generate effective gradient updates for the weights.

The advantages are as follows: 1. In some main branches, important information may be lost due to bugs in the information bottleneck; however, the proposed method can receive driving parameters for learning from the auxiliary reversible branch, helping to extract the correct important information. 2. It does not force the main branch to retain complete original information but generates useful gradients to update the original information through an auxiliary supervisory mechanism. The proposed method can also be applied to shallower networks. 3. The auxiliary reversible branch can be removed during the inference phase, preserving the inference capability of the original network. 4. Any reversible architecture can be selected in PGI to act as the auxiliary reversible branch.

Multi-level auxiliary information addresses the problem of error accumulation in deep supervision. By precisely controlling the flow and utilization of gradients, it ensures that useful information for the final task is retained at every layer of the network, inserting an integrated network between the auxiliary supervised feature pyramid layers and the main branch, and utilizing it to combine the gradients returned by different prediction heads. Multi-level auxiliary information aggregates the gradient information of all target objects, which is then passed to the main branch for parameter updates. Thus, even in deep networks, the model can effectively learn while maintaining high predictive accuracy.

The advantages are: 1. The method can alleviate the information fragmentation problem in deep supervision. 2. Any integrated network can utilize multi-level auxiliary information.

Generalized Efficient Layer Aggregation Network (GELAN): GELAN is a new lightweight network architecture based on gradient path planning design, aimed at optimizing information flow and parameter utilization within the network. Through a meticulously designed network structure, GELAN can effectively aggregate and transmit information between different computational blocks, reducing information loss during transmission while maintaining the lightweight and efficiency of the network. The design of GELAN allows it to operate efficiently on various inference devices while providing excellent object detection performance.

2.3 Post-processing

In this study, the YOLO v9 (You Only Look Once) model is employed for panoramic dental instance segmentation tasks. The model efficiently performs feature extraction and object detection on input images of size 1024x1024 pixels. By setting the confidence threshold to 0.01, the model is capable of recognizing and predicting various dental features, outputting corresponding masks and bounding box information.

The prediction results are stored in a structured JSON format, containing labels for each detected object along with their polygon coordinates. This format not only facilitates subsequent data processing and analysis but also provides a solid basis for model evaluation. The generated JSON files are named according to the submission format, allowing for organized access and ensuring the reproducibility and scalability of the experimental results.

3 Experiments

3.1 Dataset

The event organizer provided a training set of 2,380 panoramic X-ray images, including 30 labeled dental panoramic images and 2,350 unlabeled panoramic

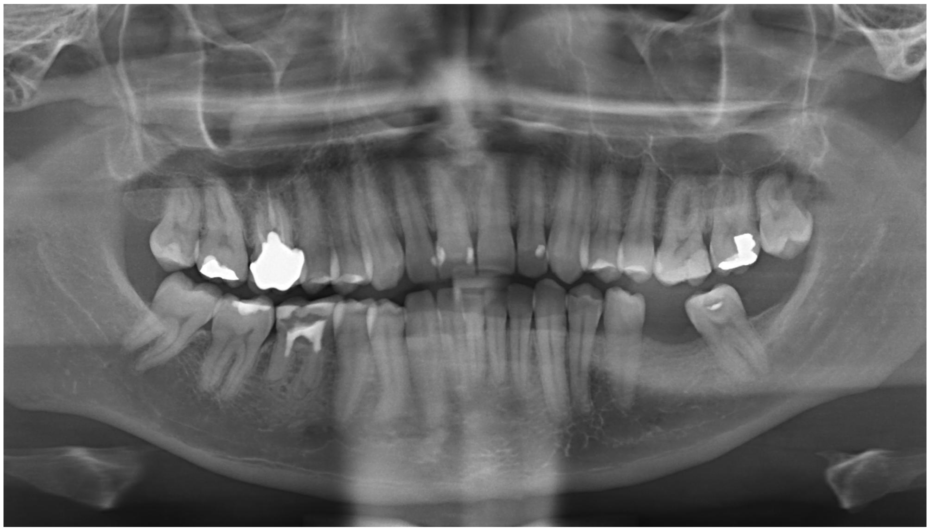

Fig. 1. STS2024 Competition Tooth Panoramic Images Example

images. The size of the labeled images is either (942,2000) or (1127,1991), and all have a channel count of 3. The validation set consists of 20 panoramic X-ray images. Labels are stored in JSON file format, and predictions should also be output in JSON file format. The predicted results of the 20 validation images will be submitted, and the ranking on the validation set will be updated in real-time. The final ranking will be determined by the final submission in the form of a Docker, using an inaccessible test set.There are other similar tooth datasets that can be referenced. [9] [10] [11]

3.2 Evaluation Metrics

Segmentation Accuracy Metrics

- **Dice Similarity Coefficient (DSC)**:

$$\mathrm{DSC} = \frac{2|A \cap B|}{|A| + |B|}$$

  where A and B are the sets of predicted and ground truth pixels. This metric can be evaluated at both instance-level and image-level.
- **Normalized Surface Distance (NSD)**:

$$\mathrm{NSD} = \frac{1}{N} \sum_{i=1}^{N} |D_i - G_i|$$

  where D_i and G_i are the distances between predicted and ground truth surfaces, applicable at both instance-level and image-level.

- **Mean Intersection-over-Union (mIoU)**:

$$\mathrm{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{|A_c \cap B_c|}{|A_c \cup B_c|}$$

  This metric is computed at both instance-level and image-level.
- **Identification Accuracy (IA)**:

$$\mathrm{IA} = \frac{TP + TN}{TP + TN + FP + FN}$$

  Applicable at both instance-level and image-level.

Segmentation Efficiency Metrics

- **Running Time**: A tolerance of 45s (including Docker starting time) is set, with inference time per case within 60s; otherwise, the case is regarded as failed.
- **GPU Memory Consumption**: Area under the GPU memory-time curve:

$$\mathrm{AUC} = \int_{t_1}^{t_2} M(t)\, dt$$

3.3 Implementation detail

The development environment and requirements are detailed in Table 1. The system operates on Windows 10 as the operating system. The CPU used is the 12th Gen Intel(R) Core(TM) i5-12490F, with a clock speed of 3.00 GHz. The system has a total of 32.0 GB of RAM. It is equipped with a single NVIDIA 3060 12G GPU. The installed CUDA version is 11.1. The programming language used for development is Python 3.20. The deep learning frameworks employed include torch 1.9.1, torchvision 0.10.0, and ultralytics 8.2.103.

Table 1. Development environments and requirements.

System	Windows 10
CPU	12th Gen Intel(R) Core(TM) i5-12490F @ 3.00 GHz
RAM	32.0 GB
GPU	NVIDIA 3060 12G
CUDA version	11.1
Programming language	Python 3.20
Deep learning frameworks	torch 1.9.1, torchvision 0.10.0, ultralytics 8.2.103

4 Results and discussion

4.1 Quantitative results on validation set

The final results are shown in Table2.The final submission ranked sixth overall, with a significant advantage in Identification Accuracy (IA) compared to subsequent competitors. This indicates that the YOLOv9-seg model can effectively recognize the presence of certain objects during segmentation; however, it struggles with accurately predicting the precise shapes and positions of these objects, and the handling of edges is not optimal.

Table 2. Performance Metrics

Metric	**Value**
Dice Instance	0.524
Dice Image	0.8491
NSD Instance	0.542
NSD Image	0.882
mIoU Instance	0.4761
mIoU Image	0.7463
Identification Accuracy (IA)	0.5403

4.2 Results on final testing set

The predictions on the test set are overall similar to the best validation set submission metrics, with a slight increase in instance metrics and a decrease in image metrics. The Identification Accuracy has increased significantly, and there is a considerable advantage in terms of time. The GPU memory consumption is relatively low, which also reflects the effective parameter utilization and lightweight advantages of the YOLO model.

Table 3. Evaluation Metrics

Metric	Value
Dice Instance	0.5301807649619877
Dice Image	0.8200493049621582
NSD Instance	0.565597369249702
NSD Image	0.8546617119122323
mIoU Instance	0.4919185700523667
mIoU Image	0.702889706492424
Identification Accuracy (IA)	0.5740806372380626
Time (seconds)	19.531428571428574
GPU Consumption (MB)	26666.571428571428

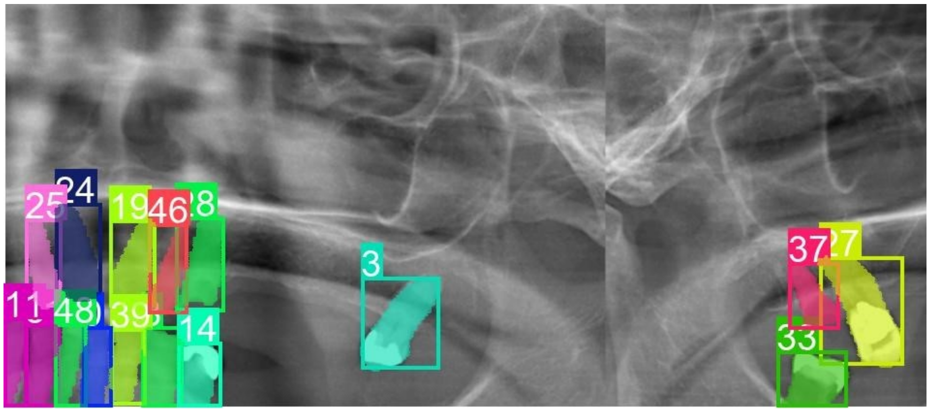

Fig. 2. STS2024 Competition Tooth Panoramic Images Example

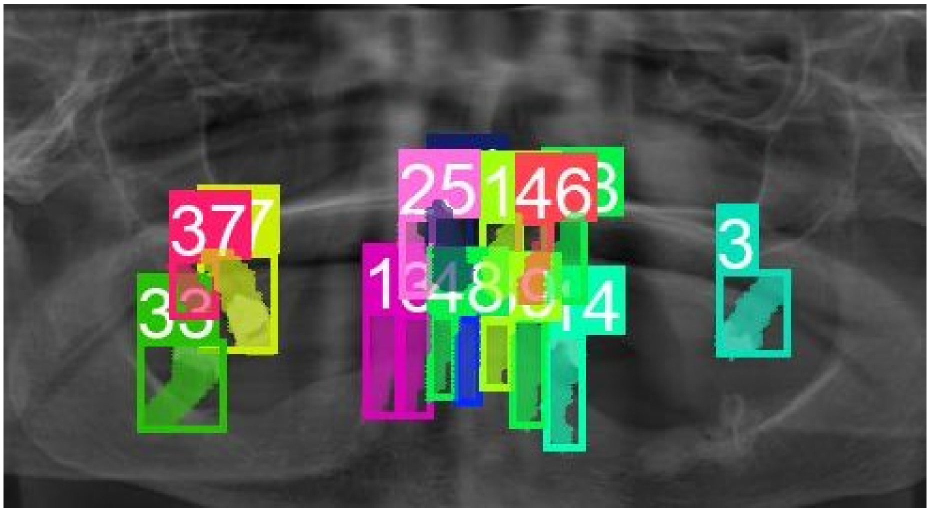

Fig. 3. STS2024 Competition Tooth Panoramic Images Example

4.3 Results Presentation

All 30 cases were treated as the training and validation sets. The prediction results indicate that the model performs well in the training set, showing correct predictions and good convergence, allowing it to learn the features of the current samples effectively. As shown in Fig2 and Fig3, the visualization of predictions in the training set illustrates the model's performance.

4.4 Limitation and future work

The model demonstrates strong overall comprehension of dental structures but has poorer sensitivity in capturing details and edges. This reflects an inherent issue with YOLO, which is primarily designed for object detection rather than segmentation. While it excels at recognizing various types of segmented targets, it falls short on detail. Future work could consider incorporating attention mechanisms or adopting a two-stage segmentation approach, where the model first locates the teeth and their types before performing detailed segmentation.

5 Conclusion

In this competition, YOLOv9 was used for the segmentation of 2D panoramic dental images, achieving sixth place on the STS validation leaderboard. The model demonstrates a significant advantage in class recognition, but improvements are needed in detail handling and segmentation precision. This paper explores the potential capabilities of YOLOv9 in medical imaging and dental segmentation, providing a convenient and lightweight segmentation network approach. It also validates the immense potential of YOLO in instance segmentation. Future targeted explorations of YOLO's segmentation capabilities represent a promising area of research.

References

1. Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024.
2. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
3. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
4. A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
5. Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
6. Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
7. Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584, 2022.

8. J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
9. Yifan Zhang, Fan Ye, Lingxiao Chen, Feng Xu, Xiaodiao Chen, Hongkun Wu, Mingguo Cao, Yunxiang Li, Yaqi Wang, and Xingru Huang. Children's dental panoramic radiographs dataset for caries segmentation and dental disease detection. *Scientific Data*, 10(1):380, 2023.
10. Weiwei Cui, Yaqi Wang, Qianni Zhang, Huiyu Zhou, Dan Song, Xingyong Zuo, Gangyong Jia, and Liaoyuan Zeng. 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*, pages 191–200. Springer, 2022.
11. Weiwei Cui, Yaqi Wang, Yilong Li, Dan Song, Xingyong Zuo, Jiaojiao Wang, Yifan Zhang, Huiyu Zhou, Bung san Chong, Liaoyuan Zeng, 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*, pages 64–73. Springer, 2022.