IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 31 January 2023, accepted 5 March 2023, date of publication 8 March 2023, date of current version 13 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3254206

RESEARCH ARTICLE

MPCNet: Improved MeshSegNet Based on Position Encoding and Channel Attention

HANQING HU, ZHENGXUN LI, AND WEICHAO GAO
School of Economics and Management, Beijing Information Science & Technology University, Beijing 100192, China
Beijing Key Laboratory of Big Data Decision-Making for Green Development, Beijing 100192, China
Corresponding author: Zhengxun Li (18801071003@163.com)

ABSTRACT In the process of orthodontic treatment, it is a very important step to accurately segment each tooth and jaw model with computer assistance. The use of deep learning technology methods for tooth segmentation can not only save a lot of manual interaction and time cost but also improve the treatment effect. 3D tooth segmentation is a hot topic of interest for international related scholars, and some end-to-end tooth segmentation methods based on dental mesh scanning models have been emerging in recent years. Due to the limited variety of existing models, they are not well suited for different 3D segmentation scenarios, and the feature extraction capability and segmentation effect of these models still need to be improved. In this paper, we propose a novel end-to-end tooth segmentation method, MPCNet, which adds multi-scale mesh density information to the input layer, uses position encoding and channel attention mechanism to improve MeshSegNet, and uses graph-cut post-processing to perform 3D tooth segmentation in real scenes. The effectiveness of MPCNet is evaluated on a real 3D scanned tooth segmentation dataset, which significantly outperforms the current mainstream segmentation methods.

INDEX TERMS Tooth segmentation, 3D deep learning, virtual orthodontics, 3D semantic segmentation, attention mechanism.

I. INTRODUCTION

With the continuous development of computer science, the importance of 3D digitization and artificial intelligence technologies in various fields is becoming more and more prominent. In the field of dentistry, the combination of deep learning and orthodontic treatment is also the focus of close attention by researchers in related fields [1], [2]. Through computer-assisted orthodontic treatment, i.e. virtual orthodontics, the orthodontist can understand the patient's oral condition more intuitively and give treatment plans to improve the efficiency and effectiveness of treatment [3], [4]. Virtual orthodontics begins by acquiring information about the patient's tooth and gum surfaces, including 3D shape and texture features, through an oral scanner [5]. This 3D dental scan is safer and more efficient than traditional dental mold image acquisition [6]. The segmentation and identification of the teeth is a crucial step in the entire virtual orthodontic treatment process, and is also the basis for tooth alignment and subsequent treatment plans [7]. However, achieving accurate end-to-end 3D dental model segmentation is a significant and challenging task because 3D oral scan data are unordered grid data and the variability of features among different teeth is not obvious, making it impossible for researchers to embed deep learning related methods into them as they do for 2D image data [8].

Although the traditional dental image segmentation method reduces some manual interaction operations and saves treatment time, its efficiency and accuracy still need to be improved [9], [10], [11]. With the rise of deep learning in the field of computer vision, deep learning in the field of medical image segmentation is also gradually taking a mainstream position [12], [13], [14]. Compared to traditional computer-assisted virtual orthodontics, the deep learning approach is not only more efficient and accurate, but also much less complex and less dependent on the dentist's expertise for the orthodontic treatment [15].

The associate editor coordinating the review of this manuscript and approving it for publication was Vishal Srivastava.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Dental scan mesh data is a more complex form of geometric data structure, which is difficult to combine with traditional deep learning image segmentation techniques due to its disordered arrangement. Therefore, researchers usually convert point clouds to 3D voxels or 2D image collections to make them orderly [16], [17]. However, this transformation of the data structure is not only relatively complicated to process, but also loses some of the 3D spatial information [18]. On the other hand, due to the continuous development of deep learning, end-to-end recognition and segmentation of 3D data is bound to be one of the major development trends in the future [19], [20]. With the emergence of PointNet [21], researchers began to focus on directly processing 3D data to realize end-to-end 3D recognition and 3D segmentation in a real sense. Subsequently, a series of improved works based on PointNet were born, such as PointNet++ [22], RSNet [23], PointConv [24], etc. These networks show excellent performance on the 3D segmentation task, however, due to the relatively similar shape features of different teeth, these networks cannot effectively extract the local feature information of the teeth, resulting in poor segmentation of the tooth edge regions. In this regard, Lian et al [25] proposed MeshSegNet, which also adopts the network architecture of PointNet, integrates a series of graph-constrained learning modules, and extracts local features in a hierarchical and multi-scale manner, which can effectively extract dental edge information, but its accuracy in practical applications still needs to be improved.

In this paper, we propose a new end-to-end tooth segmentation method MPCNet (Improved MeshSegNet Based on Position Encoding and Channel Attention) based on MeshSegNet network architecture. Compared with MeshSegNet, MPCNet can better extract the features of the dental scan grid data, significantly improve the accuracy of dental segmentation with little increase in training and inference time. The structure of MPCNet is shown in Figure 2. MPCNet is an extended version of MeshSegNet. First, the input layer of the model adds three density features of different scales (see 2.2 for details) to represent the local density in addition to the original 15 input features. In the actual tooth scanning model, the density of the mesh at different locations varies greatly, for example, the mesh located at the edge of the tooth tends to be more densely, and adding local density information at different scales to the input features is more beneficial to extract the local features of the tooth mesh. Second, we replace symmetric average pooling with locally symmetric positional encoding. The symmetric mean pooling in MeshSegNet [25] extracts only global symmetric location information when extracting location features, while the position encoding designed in this paper can extract local symmetric location information for better modeling of global features. Third, a channel attention mechanism is employed to handle the multi-level feature aggregation part of the network. In MPCNet, features from different stages of the network are densely linked before the output layer, and the channel attention mechanism can well assign weights to these features at different levels, thus improving the sensitivity of the network to features at different levels. Finally, the comparative experiments verify that MPCNet can significantly improve the effect of 3D tooth segmentation without increasing the inference time.

II. RELATED WORK

A. 3D SEGMENTATION

Traditional 3D object segmentation methods turn disordered point clouds or mesh data into ordered voxels, or convert these 3D data into a set of 2D images, and then segment them by 2D image object detection methods. For example, Milletari et al. [13] proposed a voxel-based, fully convolutional neural network for 3D image segmentation, which is trained and optimized by Dice coefficients to deal with different classes of data imbalance. Budzik et al. [26] scanned the patient's head by cone beam computed tomography (CBCT), obtained a geometric reconstruction model of the teeth using isotropic voxels, and performed tooth segmentation using the regional growth method. Tian et al. [27] proposed a 3D tooth segmentation method based on sparse voxel octrees and 3D convolutional neural networks (CNNs) using a three-level hierarchical method based on deep convolution for segmentation. Although these methods have achieved certain results, the data processing flow is complex, and the segmentation effect may be unstable for irregular tooth models.

PointNet [21] is the first model that truly implements end-to-end 3D point cloud data recognition and segmentation, can efficiently extract global features of point cloud data, and has achieved relatively promising results in both classification and semantic segmentation tasks of point clouds. After the emergence of PointNet, a series of studies based on PointNet for improvement were derived. Charles et al. [22] used a simplified PointNet to extract local features layer by layer and continuously expand from local features to global features like CNN to better achieve feature extraction of 3D objects. Wu et al. [24] regard the convolution of 3D data as a nonlinear function composed of weight function and density function, and improve the network performance by learning weight function and kernel density through multi-layer perceptron and density filter respectively. Aoki et al. [28] treated PointNet as a learnable imaging function and fused it with the LK algorithm into a trainable recursive deep neural network. Paigwar et al. [29] extended the theory of visual attention mechanisms to 3D point clouds and introduced a new recurrent 3D localization network module that significantly reduces the number of points to be processed and the inference time. However, tooth segmentation is different from the general point cloud segmentation or mesh segmentation, because of its irregular segmentation shape and the high similarity between teeth, it is very challenging to get better results in practical applications.

B. 3D DENTAL SEGMENTATION

3D tooth segmentation modeling focuses more on the extraction of local features to better segment the edge areas of the teeth. Cui et al. [30] used a tooth center-of-mass distance-aware voting scheme to detect all teeth and designed a cascade segmentation module with confidence perception to segment each individual tooth. Lian et al. [25] proposed MeshSegNet, an end-to-end deep learning method for the tooth segmentation task. Firstly, the downsampled tooth surface coordinate data and its derived features, totaling 15, will be used as input, and a series of graph-constrained learning modules combined with adjacency matrices will extract multi-level mesh features and location features. Then dense links will used to aggregate local features and global features, the prediction results will be post-processed by graph-cut and upsampled, eventually achieving better practical application results. Wu et al. [31] used edge convolution to improve MeshSegNet, and on the basis of tooth segmentation, used PointNet regression to automatically label tooth key points. However, in actual tooth segmentation scenarios, higher precision is often required. The MPCNet proposed in this paper, improves MeshSegNet by adding local density information, position coding and channel attention, which greatly improves the accuracy of tooth segmentation with almost no increase in computational load.

III. MATERIALS AND METHOD

A. DATA AND PRE-PROCESSING

The dataset used in this study is from 100 real case treatment plans annotated by professional dentist, and the lower dental model is used for training and testing. The original data model is first calibrated in the occlusal and midline planes, the extension direction is determined according to the occlusal and midline planes, and then the gingival base is generated. The processed lower jaw scan model contains approximately 40,000 mesh units.

Each lower jaw model contains one gums and 14 to 16 teeth (4 incisors, 4 canines, 4 premolars, 4-6 molars) [32]. Since the whole lower tooth model is symmetrical in the classification process, the category of the teeth is set as C = 9 classes (8 teeth and one gums) in this paper, as shown in Figure 1.

MPCNet uses 18-dimensional feature vectors to describe each mesh cell. In addition to the 15 features including mesh coordinates, mesh center coordinates, and normal vectors used in MeshSegNet and other mainstream methods, three different scales of density information are added to the input, and the specific input features are shown in Table 1.

MPCNet's input not only contains global location information, but the added density information also enriches the local features of the mesh cells. Preprocessing is required before the features are fed into the network. Firstly, the centroid of the model was translated to the origin, and then the unitized normal vector was calculated, the adjacency matrix [25] and multi-scale density information were obtained by calculating the distance matrix of the grid center coordinates. Finally, the position features such as grid coordinates, grid center coordinates, and unitized normal vectors were standardized by Z-score. The adjacency matrix and density features are normalized.

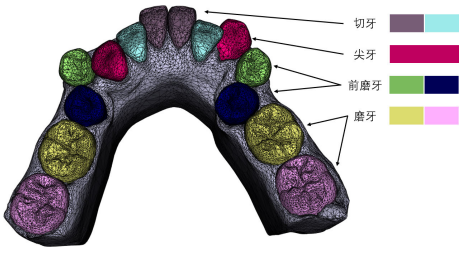

FIGURE 1. Example of tooth category labeling.

TABLE 1. Table of input features.

	Number of features	MeshSeg Net	MPC Net
Mesh Coordinate	(x,y,z) x 3 = 9	√	√
Normal	(x,y,z) = 3	√	√
Mesh Center	(x,y,z) = 3	√	√
Density information	(m_1,m_2,m_3) = 3	X	√

x, y, z are 3D coordinates, m is multi-scale density information.

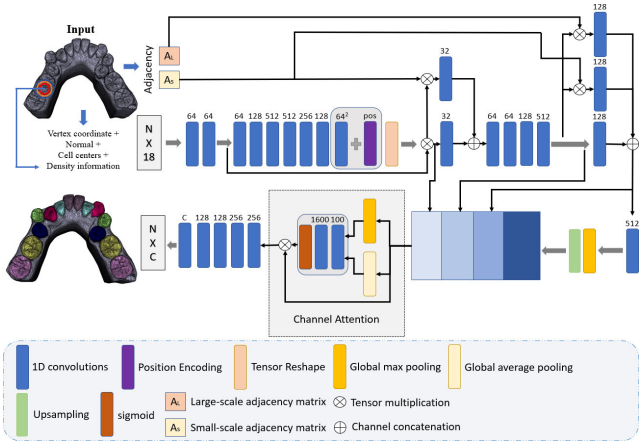

FIGURE 2. MPCNet network structure diagram.

B. MPCNET

The overall structure of MPCNet is shown in Figure 2. The input of the network is a matrix with the number of Meshes N multiplied by the features dimension 18, and the final output is a matrix with the number of Meshes N multiplied by the classes C, which represents the probability of each mesh belonging to the classes. The main idea of the network is to continuously extract the high-dimensional features of the whole Mesh scanning surface through 1D convolution (MLP) with kernel of 1.

MPCNet is similar to MeshSegNet in its overall structure. Compared to MeshSegNet, MPCNet has adopted the following main improvements:

1) Multi-scale density features are added to the feature input layer. MeshSegNet requires a lot of computing resources to perform data augmentation and calculate adjacency matrices (AS, AL) during training. In this paper, we believe that an important reason why MeshSegNet can effectively extract the neighboring grid features is that it utilizes the location adjacency matrix, so in order to effectively extract the local features, MPCNet adds the density information of neighboring locations to the input. This has another advantage that the distance between each point is already calculated when calculating the adjacency matrix (AS, AL), so there is almost no additional computation. The specific operation is to calculate how many mesh cells are adjacent to each mesh within a fixed radius of each mesh. MPCNet calculated the density information at three scales ($R_1 = 0.05$, $R_2 = 0.1$, $R_3 = 0.2$) during the formal training process and normalized to eliminate the magnitudes. The calculation process is as follows:

First, two N × N zero matrices $[S1, S2]$ are created and the adjacency matrices AS, AL are derived by calculating the normalized distance matrix $D^{N\times N}$ between the coordinates of the mesh centroids:

$$S1\,[D < 0.05] = 1, \quad S2\,[D < 0.05] = 1 \tag{1}$$

$$AS = \frac{S1}{sum\,(S1, axis = 1)} \tag{2}$$

$$AL = \frac{S2}{sum\,(S2, axis = 1)} \tag{3}$$

Second, three N × N zero matrices $[M1, M2, M3]$ are established as follows:

$$M1\,[D < 0.05] = 1 \tag{4}$$

$$M2\,[D < 0.1] = 1 \tag{5}$$

$$M3\,[D < 0.2] = 1 \tag{6}$$

The density information is calculated as $m1, m2, m3$:

$$m1 = sum\,(M1) \tag{7}$$

$$m2 = sum\,(M2) \tag{8}$$

$$m3 = sum\,(M3) \tag{9}$$

Normalized elimination of magnitudes:

$$m = [m1, m2, m3] \tag{10}$$

$$m = \frac{m - \min\,(m)}{\max\,(m) - \min\,(m)} \tag{11}$$

The radius selection strategy used for the multi-scale density information is described in detail during the ablation experiments in Section IV.

2) Positional encoding is used instead of symmetric average pooling. The position encoding adopted in this paper is shown in Figure 3. MeshSegNet adopts symmetric average pooling, which only records the global symmetric position information. Compared with the symmetric average pooling module, the advantage of position encoding is that it not only records the global symmetric position information, but also records the local position symmetry information, which is more in line with the local symmetric shape characteristics of the teeth. So that the feature extraction ability of the network can better adapt to the symmetric geometric structure of teeth [33].

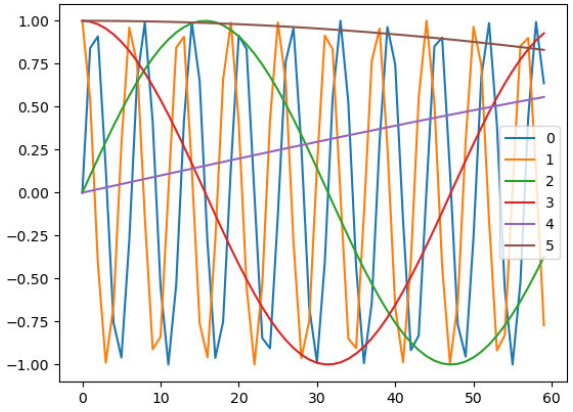

FIGURE 3. Position encoding.

3) The channel attention module is added to deal with the relationship between different levels of features. One reason for the success of MeshSegNet segmentation is feature convergence, where low-dimensional features are concatenated with high-dimensional features before the final classification layer, but for practical tasks low-dimensional features and high-dimensional features should not be equally important, and it is critical to handle features from different dimensions. CBAM is a very widely used attention method in image-based deep learning [34], [35], which integrates spatial attention and channel attention [36], [37]. However, spatial attention is not suitable for Mesh segmentation tasks, so our study modifies channel attention and applies it to 3D segmentation tasks.

The input to the channel attention module taken by MPCNet is a matrix X of N (number of mesh cells) × F (number of features). First, calculate the global maximum pooling F_{max} and the global average pooling F_{avg} for each feature dimension:

$$F_{max} = Maxpooling(X) \tag{12}$$

$$F_{avg} = Averagepooling(X) \tag{13}$$

Then, F_{max} and F_{avg} are fed into the same feature learning function G_{mlp} consisting of 1D convolution:

$$G_{max} = G_{mlp}(F_{max}) \tag{14}$$

$$G_{avg} = G_{mlp}\left(F_{avg}\right) \tag{15}$$

Finally, the results of attention learning G_{max} and G_{avg} are fed into the Sigmoid activation function (so that the distribution is between −1 and 1) to calculate the weight, and

the output is the Hadamard product of X and M:

$$M = Sigmoid\ (Gmax + Gavg) \tag{16}$$

$$X = X \odot M \tag{17}$$

where, $\odot$ is the Hadamard product.

The use of channel attention enables better handling of feature aggregation from different dimensions, allowing the network to discern which features are more important to learn.

In summary, MPCNet adds local density information to the input layer of the network by making full use of the distance matrix to compute multiscale density features. Feature transformations are performed to enhance local spatial information using locally symmetric position encoding. And learns weights to handle feature convergence in different dimensions through the channel attention mechanism. Therefore, MPCNet has stronger local feature extraction ability and global feature processing ability compared to MeshSegNet.

C. TRAINING AND DATA AUGMENTATION

The forward propagation part of the network is shown in Fig. 2. The input of the network is a matrix of N × 18, which first undergoes a set of 64 channels of 1D convolutions to extract features, and then performs feature encoding through a feature encoding matrix consisting of 1D convolution and position encoding. Then use 1D convolution combined with adjacency matrix to extract local features, and finally converge the low-dimensional features and high-dimensional features together for attention weight calculation. Finally, a set of 1D convolutions are linked to the output layer, and output an N × C-dimensional probability matrix by softmax.

The training data and test data consist of 100 real cases in total, and the labeling effect is shown in Figure 1.

The ratio of the training set to the test set is 8:2, the optimizer used for model training is AdamW (amsgrad) [38], [39], and the loss function is Dice loss [40], which is calculated as follows:

$$Ik = Pk \cap Tk \tag{18}$$

$$L = \sum_{k=1}^{c} w_c(1 - (\frac{2 * Ik + s}{Pk + Tk + s})) \tag{19}$$

where c is the number of classes, $c = 9$ in this paper, L is the Dice loss, w_c is the weight of each class loss, Pk is the number of the Kth category in the predicted value, Tk is the number of the Kth category in the true value, IK is the intersection of the predicted and true values of the Kth category, and s is the smoothing factor (to avoid the divisor being 0), and $s = 1$ in this paper. In order to enhance the generalization ability of the model, data augmentation was performed on the 3D dental model when extracting data in the training phase, mainly using the following methods:

1) Random rotation, three rotation angles are taken between [−30,30], rotation around x-axis, rotation around y-axis, and rotation around z-axis, respectively.

2) Randomly translate by taking three translations between [−10,10], respectively, along the x-axis, along the y-axis, and along the z-axis.

3) Random scaling, taking three scaling ratios between [0.8,1.2] for the x-axis, random scaling for the y-axis, and random scaling for the z-axis, respectively [41], [42], [43].

In order to further increase the diversity of the data and speed up the training, 10,000 grid planes are randomly selected from the original scan of approximately 40,000 grid planes for training. The above random sampling with data augmentation is performed for each batch of data during training, which greatly increases the diversity of data and improves the generalization ability of the model. For the hardware configuration, an RTX 3090 [44] with 24G video memory was used for all training and testing on Pytorch [45] in this paper.

D. APPLICATION

In this paper, the trained MPCNet is tested in a real application scenario. After the original data is downsampled and input to the model, the segmentation result may contain some isolated classified mesh surfaces.

Graph-cut is a progressive mesh-cutting tool proposed by Fan et al [46] to achieve efficient local cut optimization. Post-processing reclassification of these isolated mesh faces with uneven segmentation edges by Graph-cut can greatly improve the segmentation effect. For example, Xu et al. [9] proposed a deep convolutional neural network for 3D tooth segmentation with labeling optimization to refine the segmentation boundary by Graph-cut. Guo et al. [47] performed classification labeling of 3D dental meshes by deep convolutional neural networks, and then used Graph-cut for optimization of label continuity. Similarly, in this paper, the results of MPCNet are post-processed using Graph-cut, and the post-processed results are upsampled using KNN [48] or SVM [49] to obtain the segmentation results of the whole original plane. The actual segmentation effect is shown in Fig. 4, which proves that MPCNet has good practical application value.

IV. EXPERIMENT

A. COMPETING METHODS

In order to verify the advantages of MPCNet, based on the experimental configuration in Section II.C, this section uses PointNet [21], PointNet++ [22], MeshSegNet [25] and MPCNet for comparison experiments, and the evaluation metrics used are DSC (the Dice similarity coefficient), SEN (the sensitivity), and PPV (the positive prediction value), which are calculated as follows:

$$Ik = Pk \cap Tk \tag{20}$$

$$DSC = \sum_{k=1}^{c} w_k \left(\frac{2 * Ik + s}{Pk + Tk + s}\right) \tag{21}$$

$$SEN = \sum_{k=1}^{c} w_k((Ik + s)/(Tk + s)) \tag{22}$$

TABLE 2. Comparison results between MPCNet and other mainstream methods.

Metric	PointNet	PointNet++	MeshSegNet	MPCNet
DSC	0.822104692	0.588525343	0.882287943	**0.914855433**
SEN	0.810582149	0.679478741	0.887253451	**0.916068184**
PPV	0.844646609	0.593214989	0.927102542	**0.942890418**

$$PPV = \sum_{k=1}^{c} w_k((Ik + s)/(Pk + s)) \tag{23}$$

where c is the number of categories, in this paper $c = 9$, w_k is the weight of each category, in this paper $w_k = 1/9$, Pk is the number of the Kth category in the predicted value, Tk is the number of the Kth category in the true value, IK is the intersection of the predicted and true values of the Kth category, s is the smoothing coefficient (to avoid the divisor being 0), in this paper $s = 1$.

Details of the comparative experimental configuration are as follows:

1) PointNet: The PointNet used in this paper is basically the same as that in the original literature [21], the input is an Nx15-dimensional grid feature matrix, the learning rate is set to 0.0001, the batch_size is 4, the training rounds are 200 epochs, and the other parameters are set as in Section II.C.

2) PointNet++: The PointNet++ used in this paper is basically the same as in the original literature [22], the input is a grid feature matrix of N × 15, the learning rate is set to 0.0001, the batch_size is 4, the training rounds are 200 epochs, the input is a grid feature matrix of N × 15, and other parameters are set as in Section II.C.

3) MeshSegNet: The MeshSegNet [25] used in this paper is basically the same as in the original literature, where the input is an Nx15-dimensional grid feature matrix, the learning rate is set to 0.0001, the batch_size is 4, the training rounds are 200 epochs, and other parameters are set as in Section II.C.

4) MPCNet: The input of MPCNet is an Nx18-dimensional grid feature matrix, the network structure is the same as described in this paper, the learning rate is set to 0.0001, the batch_size is 4, the training rounds are 200 epochs, and other parameters are set as in Section II.C.

B. EXPERIMENT RESULTS

The summary results of the comparison experiments are shown in Table 2. The experimental results show that, firstly, the effects of MPCNet and MeshSegNet on DSC, SEN and PPV are far better than those of general 3D point cloud segmentation models such as PointNet and PointNet ++, which illustrate the superiority of MeshSegNet and MPCNet frameworks in the task of tooth segmentation. It also illustrates the importance of local features in 3D small object segmentation. Second, MPCNet outperformed MeshSegNet by 3.2 points in the DSC metric, 2.9 points in the SEN metric,

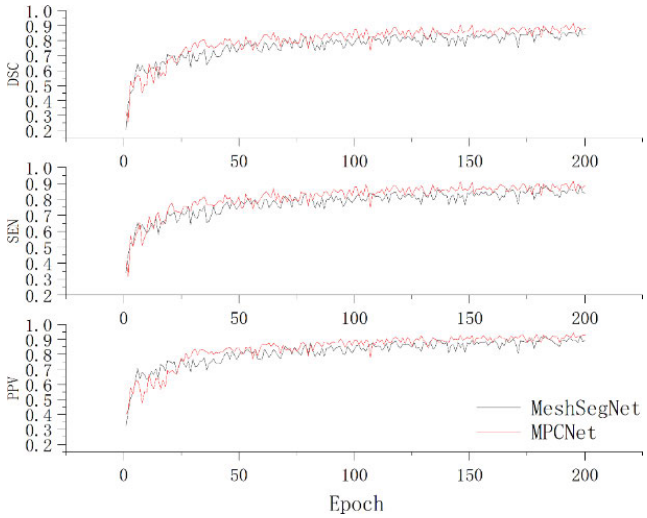

FIGURE 4. Comparison of MPCNet and MeshSegNet training process.

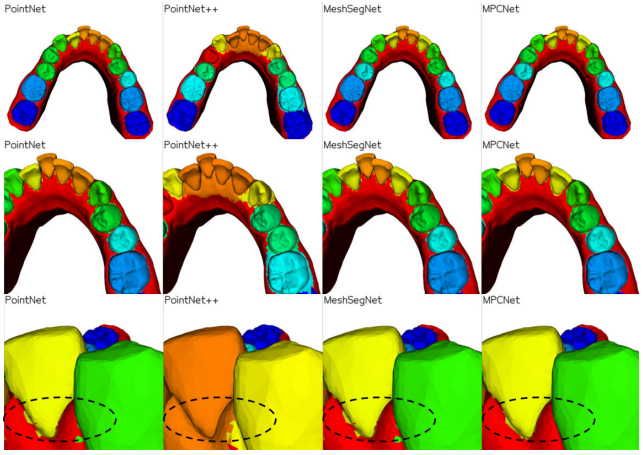

FIGURE 5. The actual segmentation effect.

and 1.6 points in the PPV metric, proving that MPCNet has surpassed MeshSegNet's performance on the tooth segmentation task. MPCNet, compared with MeshSegNet, firstly strengthens its local feature advantage in the input layer by adding multi-level density information, secondly adds local symmetric position coding to refine local feature extraction, and finally adds channel attention in the channel convergence layer to handle the relationship between high-dimensional features and low-dimensional features. These make MPCNet have better feature extraction ability and higher performance.

The comparison of the training process between MeshSegNet and MPCNet in Fig. 4 shows that MPCNet can significantly outperform MeshSegNet in terms of DSC, SEN and PPV by about 50 rounds of training, i.e., the convergence speed of MPCNet is also faster than that of MeshSegNet during training.

In the inference process, this paper also compares the inference speed of the two networks with a frame rate of

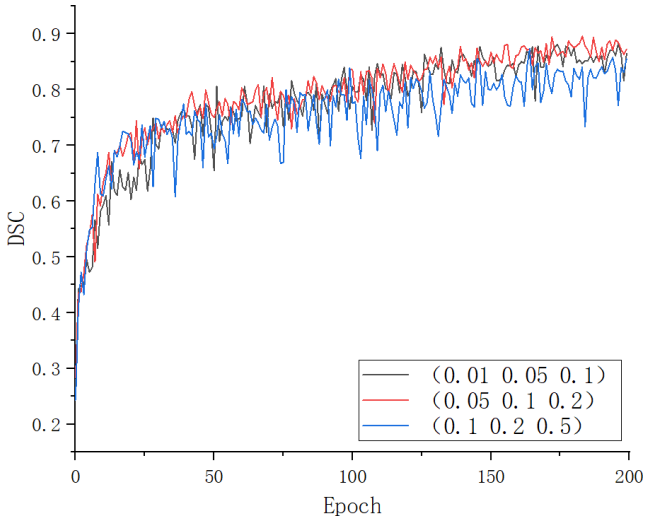

FIGURE 6. MeshSegNet training process under three different scales of density information.

0.585 fps for MeshSegNet and 0.582 fps for MPCNet [50], i.e., the increased inference time of MPCNet is almost negligible in practical applications.

The results after Graph-cut processing are shown in Figure 5. The comparison shows that during the practical application, PointNet and PointNet++ have the problems of inaccurate classification and insignificant edge detection, which are alleviated in MeshSegNet, indicating that MeshSegNet is useful for better local feature learning ability compared to PointNet and PointNet++, and MPCNet enhances this capability. Density information, position encoding and channel attention allow MPCNet to better learn the local features of the tooth scanning model and thus have better edge detection capability.

V. DISCUSSION

In order to verify the effectiveness and optimal parameter configuration of the density feature, position encoding, and channel attention approaches adopted in this paper, ablation experiments are also conducted on the same baseline model MeshSegNet.

Firstly, the density information at different scales is compared with the experimental configuration as in Section III-A, and the original MeshSegNet model is adopted, and the experimental results under different density radius settings are shown in Figure 6.

The legends in Figure 6 are shown as density information at different scales, such as (0.05, 0.1, 0.2) indicating the number of normalized meshes in the range of 0.05 radius, 0.1 radius, and 0.2 radius for each mesh. In this paper, we explored the effects of three different scales of density information on network effects, and obtained the highest DSC scores of 0.883, 0.896, and 0.873 for (0.01, 0.05, 0.1), (0.05, 0.1, 0.2), and (0.1, 0.2, 0.5) groups of radius range density information, respectively. From Table 1, it can be seen that the DSC score of MeshSegNet without adding density information is 0.882. Thus, it is obtained that adding (0.05, 0.1, 0.2) radius range density information is most beneficial for the network to learn local features. In the normalized scale, the radius range of 0.1 is approximately the single tooth coverage range, so it is more appropriate to select the local feature scale around the single tooth range in the tooth segmentation task.

TABLE 3. Comparison of the effect of density information, position encoding, and channel attention.

Metric	DSC	SEN	PPV
Base	0.8822 88	0.8872 53	0.9271 03
Base + Density Information	0.8956 89	0.8959 52	0.9445 17
Base + Position Encoding	0.8936 38	0.8958 21	0.9420 76
Base + Channel Attention	**0.9097 02**	**0.9140 57**	**0.9341 22**

In this paper, in addition to exploring the effects of density information at different scales, we also compare the effects of density information, location coding, and channel attention. Baseline uses the same MeshSegNet as set up in Section III-A, and adds density information, position encoding, and channel attention separately on top of that. The results of the ablation experiments are shown in Table 3.

The experimental results showed that density information and position encoding boosted the baseline by 1.3 and 1.1 DSC scores, respectively, while channel attention boosted the baseline by 2.7 DSC scores. It is demonstrated that in the process of learning features in the network, not only the learning of local features should be emphasized, but also the aggregation and processing of features, and different weights should be given to the features of different dimensions. The learning of these feature weights is very critical to the improvement of network performance.

VI. SUMMARIZE AND FUTURE WORK

For 3D tooth segmentation, this paper proposes a new end-to-end segmentation network MPCNet combining density information, position coding and channel attention. Experiments on real data show that MPCNet has better performance than the current mainstream methods in tooth segmentation.

The next step of the research can be carried out in two aspects. First, the training data of MPCNet contains only 100 lower jaw models, which may encounter some irregular tooth models in actual clinical applications, resulting in unsatisfactory segmentation results, and more training data need to be collected continuously to further improve the robustness of MPCNet. Secondly, the process of orthodontic treatment includes the whole process of tooth separation, filling and tooth arrangement. On the basis of MPCNet

tooth separation, it can be considered to carry out end-to-end modeling of other steps in orthodontic treatment to continue to optimize the diagnosis and treatment process.

REFERENCES

[1] S. B. Khanagar, A. Ehaideb, P. C. Maganur, S. Vishwanathaiah, S. Patil, H. A. Baeshen, S. C. Sarode, and S. Bhandi, "Developments, application, and performance of artificial intelligence in dentistry—A systematic review," *J. Dental Sci.*, vol. 16, no. 1, pp. 508–522, 2021.

[2] S. Murata, C. Lee, C. Tanikawa, and S. Date, "Towards a fully automated diagnostic system for orthodontic treatment in dentistry," in *Proc. IEEE 13th Int. Conf. e-Science (e-Science)*, New York, NY, USA, Oct. 2017, pp. 1–8.

[3] Y. Miki, C. Muramatsu, and T. Hayashi, "Classification of teeth in cone-beam CT using deep convolutional neural network," *Comput. Biol. Med.*, vol. 80, pp. 24–29, Jan. 2017.

[4] S. B. Khanagar, A. Al-Ehaideb, S. Vishwanathaiah, P. C. Maganur, S. Patil, S. Naik, H. A. Baeshen, and S. S. Sarode, "Scope and performance of artificial intelligence technology in orthodontic diagnosis, treatment planning, and clinical decision-making—A systematic review," *J. Dental Sci.*, vol. 16, no. 1, pp. 482–492, Jan. 2021.

[5] F. Mangano, A. Gandolfi, G. Luongo, and S. Logozzo, "Intraoral scanners in dentistry: A review of the current literature," *BMC Oral Health*, vol. 17, no. 1, p. 149, Dec. 2017.

[6] J. Winkler and N. Gkantidis, "Trueness and precision of intraoral scanners in the maxillary dental arch: An in vivo analysis," *Sci. Rep.*, vol. 10, no. 1, p. 1172, Jan. 2020.

[7] L. H. Son and T. M. Tuan, "A cooperative semi-supervised fuzzy clustering framework for dental X-ray image segmentation," *Expert Syst. Appl.*, vol. 46, pp. 380–393, Mar. 2016.

[8] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4548–4557.

[9] X. Xu, C. Liu, and Y. Zheng, "3D tooth segmentation and labeling using deep convolutional neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 7, pp. 2336–2348, Jul. 2018.

[10] H.-T. Yau, T.-J. Yang, and Y.-C. Chen, "Tooth model reconstruction based upon data fusion for orthodontic treatment simulation," *Comput. Biol. Med.*, vol. 48, pp. 8–16, May 2014.

[11] L. Wang, K. C. Chen, Y. Gao, F. Shi, S. Liao, G. Li, S. G. F. Shen, J. Yan, P. K. M. Lee, B. Chow, N. X. Liu, J. J. Xia, and D. Shen, "Automated bone segmentation from dental CBCT images using patch-based sparse representation and convex optimization," *Med. Phys.*, vol. 41, no. 4, 2014, Art. no. 043503.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241.

[13] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, New York, NY, USA, Oct. 2016, pp. 565–571.

[14] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sanchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[15] T. Takahashi, K. Nozaki, T. Gonda, T. Mameno, M. Wada, and K. Ikebe, "Identification of dental implants using deep learning—Pilot study," *Int. J. Implant Dentistry*, vol. 6, no. 1, p. 53, 2020.

[16] F. Schwendicke, T. Golla, M. Dreher, and J. Krois, "Convolutional neural networks for dental image diagnostics: A scoping review," *J. Dentistry*, vol. 91, Dec. 2019, Art. no. 103226.

[17] R. Jacobs, B. Salmon, M. Codari, B. Hassan, and M. M. Bornstein, "Cone beam computed tomography in implant dentistry: Recommendations for clinical use," *BMC Oral Health*, vol. 18, no. 1, p. 88, Dec. 2018.

[18] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2018, pp. 918–927.

[19] K. Kamnitsas, C. Ledig, V. F. J. Newcombe, J. P. Sirnpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med. Image Anal.*, vol. 36, pp. 61–78, Feb. 2017.

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Computer Vision–(ECCV)*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 746–760.

[21] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jul. 2017, pp. 77–85.

[22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Processing Syst. (NIPS)*. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. La Jolla, CA, USA, 2017, pp. 1–10.

[23] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2018, pp. 2626–2635.

[24] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2019, pp. 9613–9622.

[25] C. Lian, L. Wang, T. H. Wu, F. Wang, and P. T. Yap, "Deep multi-scale mesh feature learning for automated labeling of raw dental surfaces from 3D intraoral scanners," *IEEE Trans. Med. Imag.*, vol. 39, no. 7, pp. 2440–2450, Jul. 2020.

[26] G. Budzik, J. Burek, A. Bazan, and P. Turek, "Analysis of the accuracy of reconstructed two teeth models manufactured using the 3DP and FDM technologies," *Strojniški vestnik-J. Mech. Eng.*, vol. 62, no. 1, pp. 11–20, Jan. 2016.

[27] S. Tian, N. Dai, B. Zhang, F. Yuan, Q. Yu, and X. Cheng, "Automatic classification and segmentation of teeth on 3D dental model using hierarchical deep learning networks," *IEEE Access*, vol. 7, pp. 84817–84828, 2019.

[28] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "PointNetLK: Robust & efficient point cloud registration using PointNet," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2019, pp. 7156–7165.

[29] A. Paigwar, O. Erkent, C. Wolf, and C. Laugier, "Attentional PointNet for 3D-object detection in point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, New York, NY, USA, Jun. 2019, pp. 1297–1306.

[30] Z. Cui, C. Li, N. Chen, G. Wei, R. Chen, Y. Zhou, D. Shen, and W. Wang, "TSegNet: An efficient and accurate tooth segmentation network on 3D dental model," *Med. Image Anal.*, vol. 69, Apr. 2021, Art. no. 101949.

[31] T.-H. Wu, C. Lian, S. Lee, M. Pastewait, C. Piers, J. Liu, F. Wang, L. Wang, C.-Y. Chiu, W. Wang, C. Jackson, W.-L. Chao, D. Shen, and C.-C. Ko, "Two-stage mesh deep learning for automated tooth segmentation and landmark localization on 3D intraoral scans," *IEEE Trans. Med. Imag.*, vol. 41, no. 11, pp. 3158–3166, Nov. 2022.

[32] Y.-C. Huang, C.-A. Chen, T.-Y. Chen, H.-S. Chou, W.-C. Lin, T.-C. Li, J.-J. Yuan, S.-Y. Lin, C.-W. Li, S.-L. Chen, Y.-C. Mao, P. A. R. Abu, W.-Y. Chiang, and W.-S. Lo, "Tooth position determination by automatic cutting and marking of dental panoramic X-ray film in medical image processing," *Appl. Sci.*, vol. 11, no. 24, p. 11904, Dec. 2021.

[33] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2013, pp. 3476–3483.

[34] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Computer Vision–(ECCV)*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 3–19.

[35] Z. Cui, Q. Li, Z. Cao, and N. Liu, "Dense attention pyramid networks for multi-scale ship detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8983–8997, Oct. 2019.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2018, pp. 7132–7141.

[37] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jul. 2017, pp. 7398–7407.

[38] S. R. Dubey, S. Chakraborty, S. K. Roy, S. Mukherjee, S. K. Singh, and B. B. Chaudhuri, "DiffGrad: An optimization method for convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4500–4511, Nov. 2020.

[39] C. Chen, D. Han, and J. Wang, "Multimodal encoder–decoder attention networks for visual question answering," *IEEE Access*, vol. 8, pp. 35662–35671, 2020.
[40] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, M. J. Cardoso and T. Arbel, Eds. Cham, Switzerland: Springer, 2017, pp. 240–248.
[41] M. Musy, K. Flaherty, J. Raspopovic, A. Robert-Moreno, J. Richtsmeier, and J. Sharpe, "A quantitative method for staging mouse embryos based on limb morphometry," *Development*, vol. 145, no. 7, Jan. 2018, Art. no. dev154856.
[42] X. Diego, L. Marcon, P. Müller, and J. Sharpe, "Key features of Turing systems are determined purely by network topology," *Phys. Rev. X*, vol. 8, no. 2, Jun. 2018, Art. no. 021071.
[43] X. Lu, C. G. Farquharson, J.-M. Miehé, and G. Harrison, "3D electromagnetic modeling of graphitic faults in the Athabasca basin using a finite-volume time-domain approach with unstructured grids," *Geophysics*, vol. 86, no. 6, pp. B349–B367, Nov. 2021.
[44] M. Schütz, B. Kerbl, and M. Wimmer, "Rendering point clouds with compute shaders and vertex order optimization," *Comput. Graph. Forum*, vol. 40, no. 4, pp. 115–126, Jul. 2021.
[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, and G. Chanan, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, H. Wallach, H. Larochelle, A. Beygelzimer, F. D'Alche-Buc, E. Fox, and R. Garnett, Eds. La Jolla, CA, USA, 2019, pp. 1–12.
[46] L. Fan, L. Lic, and K. Liu, "Paint mesh cutting," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 603–612, Apr. 2011.
[47] K. Guo, D. Zou, and X. Chen, "3D mesh labeling via deep convolutional neural networks," *ACM Trans. Graph.*, vol. 35, no. 1, pp. 1–12, Dec. 2015.
[48] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 340–352, Jan. 2016.
[49] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, Jan. 2014.
[50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. La Jolla, CA, USA, 2015, pp. 1–9.

HANQING HU received the Ph.D. degree in management science and engineering from Beijing University of Technology, China.

He is currently a Master Supervisor with Beijing Information Science & Technology University, Beijing, China. He is also in-charge of national projects related to intelligent decision-making. His research interests include big data analysis and mining, machine learning algorithms, and business intelligence.

ZHENGXUN LI received the B.A. degree from Dalian Maritime University, Liaoning, China. He is currently pursuing the master's degree with Beijing Information Science & Technology University. He is mainly responsible for the projects related to artificial intelligence. His research interests include deep learning and intelligent fault diagnosis.

WEICHAO GAO received the B.E. degree from Shandong Technology and Business University, Shandong, China. She is currently pursuing the master's degree with Beijing Information Science & Technology University. She is mainly responsible for the projects related to intelligent manufacturing. Her research interests include big data and intelligent manufacturing.