ORIGINAL RESEARCH
published: 27 June 2022
doi: 10.3389/frcmn.2022.907388

Privacy-Preserved Federated Learning for 3D Tooth Segmentation in Intra-Oral Mesh Scans

Songshang Liu[1], Howard H. Yang[1], Yiqi Tao[1], Yang Feng[1,2], Jin Hao[3] and Zuozhu Liu[1]*

[1]ZJU-UIUC Institute, ZJU-Angelalign R&D Center for Intelligent Healthcare, Zhejiang University, Hangzhou, China, [2]Angelalign Inc., Shanghai, China, [3]Harvard School of Dental Medicine, Harvard University, Boston, MA, United States

Semantic segmentation over three-dimensional (3D) intra-oral mesh scans (IOS) is an essential step in modern digital dentistry. Many existing methods usually rely on a limited number of labeled samples as annotating IOS scans is time consuming, while a large-scale dataset of IOS is not yet publicly available due to privacy and regulatory concerns. Moreover, the local data heterogeneity would cause serious performance degradation if we follow the conventional learning paradigms to train local models in individual institutions. In this study, we propose the FedTSeg framework, a federated 3D tooth segmentation framework with a deep graph convolutional neural network, to resolve the 3D tooth segmentation task while alleviating data privacy issues. Moreover, we adopt a general privacy-preserving mechanism with homomorphic encryption to prevent information leakage during parameter exchange between the central server and local clients. Extensive experiments demonstrate that both the local and global models trained with the FedTSeg framework can significantly outperform models trained with the conventional paradigm in terms of the mean intersection over union, dice coefficient, and accuracy metrics. The FedTSeg framework can achieve better performance under imbalanced data distributions with different numbers of clients, and its overall performance is on par with the central model trained with the full dataset aggregated from all distributed clients. The data privacy during parameter exchange of FedTSeg is further enhanced with a homomorphic encryption process. Our work presents the first attempts of federated learning for 3D tooth segmentation, demonstrating its strong potential in challenging federated 3D medical image analysis in multi-centric settings.

Keywords: federated learning, medical image analysis, semantic segmentation, homomorphic encryption, tooth segmentation

OPEN ACCESS

Edited by:
Chen Feng,
University of British Columbia, Canada

Reviewed by:
Jianyu Niu,
Southern University of Science and Technology, China
Jianhang Tang,
Yanshan University, China

****Correspondence:***
Zuozhu Liu
zuozhuliu@intl.zju.edu.cn

Specialty section:
This article was submitted to Wireless Communications, a section of the journal Frontiers in Communications and Networks

Received: *29 March 2022*
Accepted: *16 May 2022*
Published: *27 June 2022*

Citation:
Liu S, Yang HH, Tao Y, Feng Y, Hao J and Liu Z (2022) Privacy-Preserved Federated Learning for 3D Tooth Segmentation in Intra-Oral Mesh Scans.
Front. Comms. Net 3:907388.
doi: 10.3389/frcmn.2022.907388

1 INTRODUCTION

The research on deep learning has brought remarkable progress in many interdisciplinary fields, such as medical image analysis Litjens et al. (2017) Lundervold and Lundervold, (2019) Zhou et al. (2021). Lots of novel designated neural network architectures and corresponding end-to-end solutions have been proposed to assist diagnosis and treatment planning in real-world clinical applications, with the goal of providing better and more intelligent healthcare services by improving the diagnosis efficiency of doctors and reducing the treatment cost of patients. These techniques become prevalent, especially in two-dimensional (2D) or three-dimensional (3D) volumetric medical image analysis. For example, for lung disease diagnosis, Gordienko et al. (2018) in their study performed lung segmentation in X-ray scans with a UNet-based Ronneberger et al. (2015) structure. A multi-scale convolutional neural network (MCNN) was proposed by Shen et al. (2015) to detect lung nodules in computed tomography (CT) slices with 3D volumetric images. More applications are also investigated, such as in liver segmentation by Christ et al. (2017) and Long et al. (2015) and breast cancer diagnosis, brain magnetic resonance imaging (MRI) analysis, etc. by Bejnordi et al. (2017). Besides regular 2D or 3D grid images, geometrical medical data, such as 3D meshes, point clouds, or molecules, also become even more popular in many medical applications, while satisfactory intelligent data analysis algorithms are yet under construction.

In dental research, intra-oral scanners (IOSs) have been popularized and extensively used in digital orthodontics. They can generate a 3D mesh of teeth anatomy, which is more accurate than a plaster model. In clinical diagnosis, an important step is to segment the individual tooth and gingiva precisely in the IOS meshes acquired from the scanners, i.e., 3D tooth segmentation. Specifically, given an IOS mesh consisting of more than 150,000 triangulated faces, 3D tooth segmentation would predict and classify each face to corresponding teeth or gingiva following the FDI Herrmann, (1967) notation. The segmented outputs are an indispensable prerequisite for subsequent steps, such as diagnosis and treatment planning in orthodontics and implanting. Several pioneering researchers carried out by Xu et al. (2019), Tian et al. (2019), Zanjani et al. (2019), Cui et al. (2021), Lian et al. (2020), and Hao et al. (2021) have worked on the 3D tooth segmentation problem with geometry-based or deep-learning-based methods, such as models based on conventional 2D or 3D convolutional neural networks (CNN) or designated tooth segmentation networks. These methods achieve good performance on their individual test set, but there still exist some limitations as elaborated below.

Accurate and automatic 3D tooth segmentation remains a challenging task on the following grounds. On one hand, the IOS samples vary significantly among patients, such as different tooth shapes (tooth with cavities or defective restoration or attrition), tooth numbers (missing teeth, hypodontia, or hyperdontia), tooth sizes (microdontia and macrodontia), positional varieties (tipped, rotated, or shifted tooth and crowding teeth), leading to a large data heterogeneity among patients samples Hao et al. (2021). Such a large data heterogeneity imposes serious challenges to developing robust and accurate 3D tooth segmentation solutions. On another hand, most of these methods are only evaluated with their own test set due to the lack of publicly available large-scale and multi-centric IOS datasets. For example, the method given in a study by Cui et al. (2021) is only evaluated with less than 50 samples collected from patients without third molars from the same center. Previous research also demonstrated that their performance would degrade a lot if they are evaluated on a large-scale dataset with more complicated cases Hao et al. (2021). Following the aforementioned two constraints, it is worthwhile if we could aggregate data samples from multiple hospitals and clinics, i.e., collecting much more data samples with high heterogeneity as they are collected from different sites and patients. However, data sharing and exchange among hospitals and clinics might be awkward due to privacy and regulatory concerns. Overcoming such data island issues is of high necessity to achieve clinically applicable solutions for 3D tooth segmentation in multi-centric scenarios.

Recently, the federated learning (FL) framework is proposed for collaborative and distributed learning across multiple participants (such as hospitals) without explicit data sharing McMahan et al. (2017). Participants, which are also termed as clients, no matter whether in large hospitals or small clinics, can utilize their computation resources to perform training based on premium local datasets, and share their model parameters to a central server. Within this collaborative mode, clients could contribute to the same global model and significantly boost the model performance without exchanging their local data. FL has been applied to many fields, including mobile-edge computing Lu et al. (2020) and the Internet of Things (IoT) Ren et al. (2019). There is also some pioneering work about federated learning in medical image analysis Kaissis et al. (2020); Rieke et al. (2020); Fan et al. (2021); Warnat-Herresthal et al. (2021). For example, Liu Q. et al. (2020) in their study improved prostate segmentation by learning the shared knowledge from heterogeneous datasets in multiple sites. However, to the best of our knowledge, there is no previous work exploring the feasibility of FL for 3D tooth segmentation due to challenges in 3D geometrical medical image analysis, though the demand becomes burning with the drastically increasing number of dental patients.

In this study, we propose the framework FedTSeg based on the general FL framework for distributed 3D tooth semantic segmentation with a privacy-preserved module under various settings. We first formulated the 3D tooth segmentation as a point cloud segmentation task and designed the corresponding segmentation architecture based on the EdgeConv blocks Wang et al. (2019). Under the general FedAvg setting McMahan et al. (2017) that can easily scale to a large-scale dataset with competing performance, we investigated the tooth segmentation performance of the FedTSeg with balanced or imbalanced distributions of data samples among clients. We also study the effect with different numbers of clients with heterogeneous IOS samples. Furthermore, to resist the potential parameters leakage during the federated process, we adopted a homomorphic privacy-preserving module in FedTSeg to strictly protect the communication between clients and the server. Comprehensive experiments with 500 IOS samples demonstrate that FedTSeg can achieve a mean intersection of union (mIoU), dice coefficient (DSC), and accuracy (ACC) of 81.49, 86.42, and 92.53%, respectively, significantly outperforming the conventional counterparts trained with a local paradigm. Moreover, the overall performance with FedTSeg is on par with the central model trained with the aggregated data from all clients with privacy-preserved distributed learning. Our work presents the first attempt in federated learning for 3D tooth segmentation over geometric medical data, demonstrating the strong potential of federated learning for challenging 3D medical image analysis tasks in the distributed multi-centric setting.

Our main contributions can be summarized as follows:

- We established the federated tooth segmentation (FedTSeg) framework based on the deep graph convolutional neural networks for privacy-preserved distributed 3D tooth segmentation and investigated the IOS tooth segmentation performance under various settings, such as a varying number of clients or different distributions of heterogeneous IOS mesh scans.

- We achieved privacy-preserving federated 3D tooth segmentation with a homomorphic encryption mechanism to prevent potential parameter leakage during communication.
- We demonstrated the effectiveness of the proposed FedTSeg framework with comprehensive experiments, which exhibit that FedTSeg could attain a better global model than conventional local training. Meanwhile, FedTSeg strictly protects the patient's privacy and secures the communication between clients and the central server.

The rest of the study is organized as follows. The related work is reviewed in **Section 2**, a system model used in this study is introduced in **Section 3**, methods are elaborated in **Section 4**, experiments performed are described in **Section 5**, the discussion and analysis of this work are given in **Section 6**,and conclusions of this study are given in **Section 7**.

2 RELATED WORK

2.1 3D Tooth Segmentation

We formulate the 3D tooth segmentation task as a 3D point cloud segmentation task, which is a specific branch of 3D shape segmentation. There has been substantial work for 3D shape segmentation. PointNet Qi et al. (2017a) and PointNet++Qi et al. (2017b) are widely used when it comes to dealing with point clouds. They support semantic segmentation of objects and scenes, but cannot capture the geometrical relationships between points, leading to inferior performance in complex scenes. In their study, Wang et al. (2019) came up with the dynamic graph convolutional neural work (DGCNN) based on the EdgeConv block, which can obtain both local and global representations. However, the performance of these methods for 3D tooth segmentation is not good as expected, as the IOS tooth mesh, with higher-resolution and complicated anatomical structures, is significantly different from nature objects. Recently, there have been several works toward improved performance of 3D tooth segmentation based on point cloud segmentation. Some methods first extract predefined geometrical features from the original mesh and then resolve the 3D tooth segmentation task via 2D/3D convolutional neural network (CNN) Xu et al. (2019) and Tian et al. (2019). Some specifically designed neural networks are also proposed to improve the performance of tooth segmentation, such as DC-Net Hao et al. (2021), TSegNet Cui et al. (2021), and MeshSegNet Lian et al. (2020). They achieve precise segmentation on the regular upper jaw or lower jaw scans but perform poorly if restricted to limited training data or tested with heterogeneous IOS samples, while the large-scale annotated dataset is not publicly available. In our work, we employ the FL framework with a segmentation backbone composed of EdgeConv blocks and CNNs, which can take advantage of distributed IOS samples from multiple hospitals and clinics for local training and global aggregation, to obtain a well-trained global model while getting around the privacy concerns.

2.2 Federated Learning

Federated learning was first raised to conduct distributed training from decentralized data across various client devices. In contrast to conventional centralized learning paradigms, FL does not require explicit data sharing among clients or institutions. A typical algorithm is federated averaging (FedAvg) McMahan et al. (2017), which gives a general platform consisting of a central server and a number of distributed clients. More advanced FL frameworks are also proposed to tackle different issues in FL, such as robustness, privacy, and heterogeneity [FedBN Li et al. (2021), FedProx Li et al. (2020), MFL Liu W. et al. (2020)]. But there still exist some challenges preventing federated learning from being applicable, such as performance decline in non-iid settings and heterogeneous data distribution, low communication efficiency under huge traffic pressure, and potential privacy leakage. In clinical situations, there have been some recent works adopting the federated learning framework for specific medical tasks, such as brain-tumor segmentation Sheller et al. (2020), COVID-19 screening Feki et al. (2021), and prostate cancer classification Yan et al. (2021). Besides, Kaissis et al. (2020) in their study presented an overview of cutting-edge secure methods in federated learning with medical imaging. Fan et al. (2021) in their study provided an FL framework for 3D brain MRI images. In their study, Warnat-Herresthal et al. (2021) developed a decentralized edge-computing framework for medical imaging with a permitted blockchain. In this work, we focus on the challenging 3D tooth segmentation task that is not investigated under the FL setting. The previous work by Yeom et al. (2018), Melis et al. (2019), and Song et al. (2020) has shown that an unreliable server could deduce the feature of training data via reverse engineering during model updating. These attackers could conduct inferences about label information and features of local datasets through gradient information uploaded by clients. Thus, secure protection for model parameters is needed during the communication between clients and servers. From the perspective of protecting patients' privacy, we further include the homomorphic encryption mechanism to strengthen privacy during parameter exchange between clients and the server.

2.3 Federated Learning for Medical Image Segmentation

There has been previous work focusing on improving the performance of segmentation with a federated learning system. In their work, Li et al. (2019) proposed an FL framework with differential privacy for brain-tumor segmentation to preserve patient data privacy. In their work, Bercea et al. (2021) proposed a framework for federated unsupervised brain anomaly segmentation. In their work, Lo et al. (2021) showed that a federated learning model could achieve similar results as models trained on fully centralized data for microvasculature segmentation. While these methods mainly focus on regular 2D

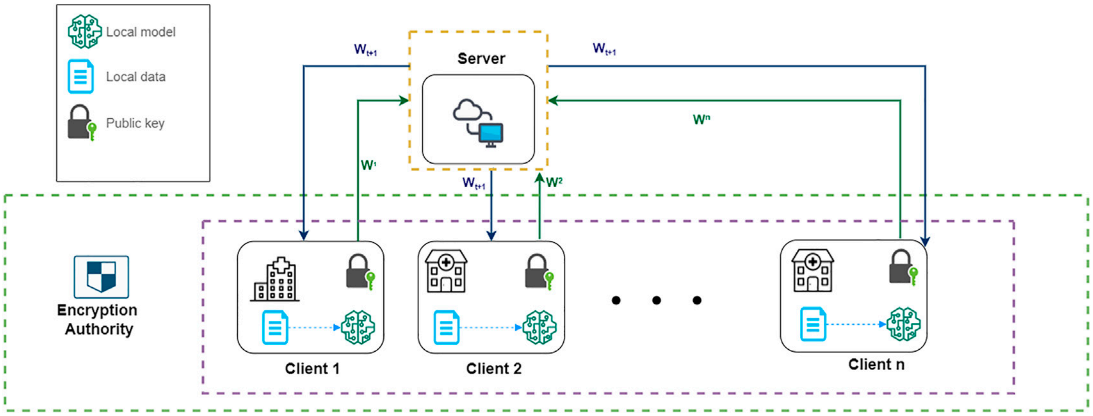

FIGURE 1 | A system model of our FL architecture. ω^1, ω^2, . . .ω^n represent the encrypted local model parameters of each client. ω_{t+1} represents the updated global model's parameter after aggregation.

segmentation tasks for grid images, this work, in contrast, investigates the more challenging 3D tooth segmentation task over complicated and heterogeneous geometrical medical data. To the best of our knowledge, there exists no prior work on federated learning for 3D medical image segmentation tasks, while our work presents the initial step on FL for tooth segmentation on large-scale heterogeneous 3D IOS meshes.

3 PROBLEM FORMULATION AND SYSTEM MODEL

We aim to solve the 3D tooth segmentation task by simulating cooperation among a large group of medical institutions with a federated learning framework. Let us first define the 3D tooth segmentation task on IOS meshes concretely. Let $\mathbf{m} = (\mathbf{V}, \mathbf{F})$ denote an IOS mesh, where $\mathbf{V}$ is the vertices and $\mathbf{F}$ is the triangular faces of the mesh. The goal of 3D IOS tooth segmentation is to assign a corresponding label y_t for each triangular faces f_t, where $y_t \in \{0, 11 - 18, 21 - 28, 31 - 38, 41 - 48\}$ denotes the gingiva and FDI Herrmann, (1967) notations for 32 permanent teeth, respectively.

Our system model is illustrated in **Figure 1**. The system contains three parts: n distributed clients, a global server, and an independent encryption authority component. The clients represent medical institutions participating in federated learning, such as clinics or hospitals, and the IOS scans from patients are securely stored in their local database. For large institutions, such as public hospitals, the amount and diversity of the dataset would be larger than small institutions, such as clinics, which is simulated in our experiments as well. Within this framework, the clients can make use of their data and computation capability to perform local training and participate in a federated learning process by sending their local model parameters w_i to a global server. The global server will aggregate the parameters from the distributed clients and send the updated parameters w to the local clients.

The federated learning process is further secured by an additional homomorphic encryption authority component. The encryption authority is a third party independent from the server. Before sending local parameters, clients will request public and private keys from an encryption authority, and use the public key to encrypt their local model's parameters. The encrypted model parameters from each client are aggregated by the server and distributed back to each client in one communication round. By doing so we can prevent information leakage to preserve privacy during federated communications, such as avoiding leakage of gradients that can be used to reconstruct the original training data. The detailed algorithm of federated learning and homomorphic mechanism will be introduced in the following sections.

4 METHODS

In this section, we systematically introduce the proposed FedTSeg framework. We first introduce the federated learning framework, which consists of the clients and server modules. Next, we present the details of our segmentation backbone for 3D IOS tooth segmentation, which is built with the EdgeConv blocks and inspired by the DGCNN. Finally, we mathematically define the homomorphic encryption process that can help secure the federated learning process.

4.1 Federated Learning Framework

The FedTSeg framework is mainly based on the general FedAvg architecture, which can easily scale to large-scale datasets. Below is a detailed description of the federated learning framework with multiple distributed clients, e.g., hospitals and clinics that can perform model training locally, and a global server that aggregates models from the clients.

4.1.1 Client Module

Assume there are n clients participating in the federated tooth segmentation, i.e., $\mathcal{C} = \{c_1, c_2, . . . c_n\}$, where $\mathcal{C}$ denotes all the clients, and c_k denotes the k-th client. Each client c_k is associated with a local dataset $\mathbb{D}_k = \{d_i^k\}_{i=1}^{N_k}$, where $N_k = |\mathbb{D}_k|$ denotes the size of the local dataset. Each data sample is defined as $d_i^k = (\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{x}_i$ represents the IOS scan from the i-th patient, and $\mathbf{y}_i$ denotes the corresponding annotated labels for each mesh face.

We define the feature extraction and learning process in each client. As mentioned above, we transform the original segmentation task over the 3D IOS surfaces to a segmentation task over 3D point clouds following. Suppose we have an IOS mesh $\mathbf{m} = (\mathbf{V}, \mathbf{F})$, where $\mathbf{V}$ are the vertices and $\mathbf{F}$ are the faces of the mesh. We first randomly sample N face centers from all triangular faces and extract a 15-dimensional feature for each point, leading to a point cloud $P = p_i|_{i=1}^{N}$. In particular, for each triangular face f_t, we have three vertices $\mathbf{v} = (x_i, y_i, z_i)|_{i=1}^{3}$, where x, y, and z are the 3D Cartesian coordinates of the vertex. The center of the triangle is denoted as $\mathbf{v_c} = \frac{1}{3}\sum_1^3 (x_i, y_i, z_i)$. We further extract more geometrical features from the original mesh for each point. Specifically, we compute the normal vector $\mathbf{h}_n$ of the triangular face and a structure descriptor $\mathbf{h}_s \in \mathbb{R}^9$. With three vertices $\mathbf{v}$ and triangle center $\mathbf{v_c}$, we have $\mathbf{h}_s = concat([\mathbf{v}_i - \mathbf{v_c}]_{i=1}^{3})$, where $concat$ () means concatenation of the vectors. Hence, each point p_i is associated with a feature vector $\mathbf{h} = \{\mathbf{v_c}, \mathbf{h}_s, \mathbf{h}_n\} \in \mathbb{R}^{15}$. Besides, each point p_i is also associated with a corresponding label y_i that assigns each face into different tooth codes, where $y_i \in \{0, 11-18, 21-28, 31-38, 41-48\}$ denotes the gingiva and FDI (Federation Dentaire Internationale) notations for 32 permanent teeth. In our framework, each client follows the same approach to perform feature extraction.

Before communicating with the central server, clients will conduct local training for E epochs based on the segmentation neural network, which is described in **Section 4.2**. The goal of each client is to find the local model's parameters, ω, that minimize the loss function ℓ_i (ω; $\mathbf{x}_i$, $\mathbf{y}_i$), which can quantify the distance between the predicted labels $\hat{y}_i$ (determined by ω) and true labels $\mathbf{y}_i$, as defined in **Section 4.2**. Consider all samples in the local dataset, the objective function for client k is

$$L_k(\omega) = \frac{1}{N_k}\sum_{i\in\mathbb{D}_k} \ell_i(\omega; \mathbf{x}_i, \mathbf{y}_i). \tag{1}$$

During local training, each client updates their local parameters using the stochastic gradient descent (SGD) method: $\omega \leftarrow \omega - \eta\nabla L_k(\omega)$, where η is the learning rate, and $\nabla L_k(\omega)$ represents the average gradient on local training. An advanced SGD optimizer, such as Adam by Kingma and Ba (2015), might be used for better convergence and performance during training. When the t-th round of local training is finished, the client will request a public key and a private key from the encryption authority, and encrypt the local model parameter ω_t^k with the public key. Then, the client will upload their ciphertext (local-encrypted parameters) to the server.

Upon receiving the updated encrypted message from the server, the client will decipher the message with the private key and get the updated model parameters ω_{t+1}^k. Afterward, it will begin the next round of local training. Due to the encryption–decryption mechanism, the model parameters are under strict protection during the communication period.

4.1.2 Server Module

The global server is responsible for aggregating model parameters from each client and distributing the updated global model back to clients. In the vanilla FedAvg framework, the server will collect the local model parameters from each client, perform weighted average (**Eq. 2**) operations, and then send the updated parameters back to each client. Here, we define the model aggregation process as follows:

$$\omega_{t+1}^k \leftarrow \sum_{k=1}^{n} \frac{N_k}{\mathbb{N}}\omega_t^k, \quad \text{where} \quad \mathbb{N} = \sum_{k=1}^{n} N_k. \tag{2}$$

In our FedTSeg framework, we modified the **Eq. 2** to incorporate the encryption mechanism to ensure secure parameter exchange. Let ζ^k denote the encrypted local model parameters. The server will update new model parameters ζ_{t+1} with **Eq. 3** following equation:

$$\boldsymbol{\zeta}_{t+1}^k \leftarrow \sum_{k=1}^{n} N_k \boldsymbol{\zeta}_t^k, \tag{3}$$

which excludes the division operation. The reason is that the additive homomorphism does not support the multiplication between an encrypted message and a float number. It supports the addition between two ciphertexts, easily inferring that it also supports the multiplication between ciphertext and a non-negative integer. Hence, in our FedTSeg setting, the server will send the number $\mathbb{N}$ back to each client, and ask each client to do the division operation locally.

4.2 3D Tooth Segmentation Network Architecture

In this work, we design the 3D tooth segmentation neural network based on the EdgeConv blocks as inspired by the DGCNN Wang et al. (2019), which is capable of processing point clouds and can be trained and evaluated in an end-to-end manner. The network architecture is illustrated in **Figure 2**.

4.2.1 Transform Net

We first align each input point cloud to canonical space with a transform Net Qi et al. (2017a), which will be fed into subsequent EdgeConv blocks for further representation learning and segmentation. Specifically, we sample n points on the input IOS mesh to generate the feature vector $\mathbf{x}$ with a shape of $n \times f$, where $f = 15$ denotes the dimension of features as defined in the feature extraction process above. In our implementation, we set $n = 10,000$. The transform net is composed of consecutive 2D convolutional layers (Conv2D), a max-pooling layer, and fully connected layers (FC). In particular, we use a transform net with four Conv2D layers with 64/128/128/1024 filters, respectively. Each convolutional layer uses a 1 × 1 kernel and a stride of 1, followed by batch normalization and ReLU activation. The output of the convolution layers is processed with a max-

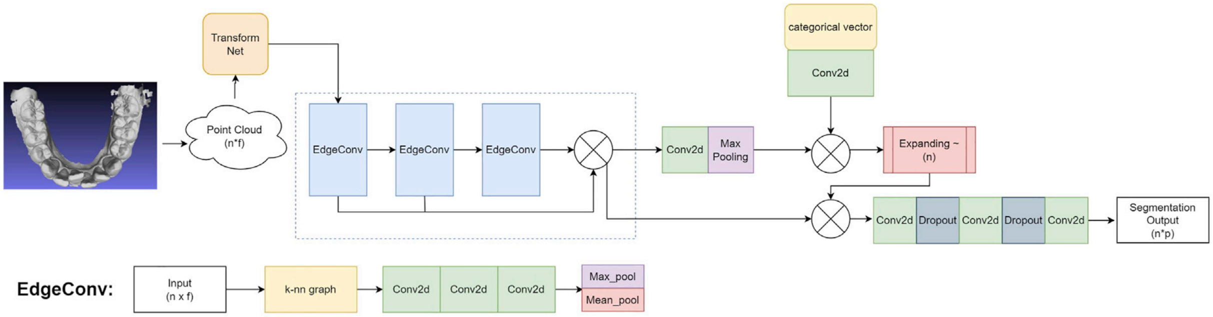

FIGURE 2 | The architecture of our 3D tooth segmentation network: Input mesh → Point cloud $x \in \mathbb{R}^{n \times f}$ → Transform net → EdgeConv → EdgeConv → EdgeConv → Conv2D[1024] → Maxpool → Conv2D[256] → Dropout → Conv2D[256] → Dropout → Conv2D[128] → Output. n denotes the number of sample points, and f denotes the number of features. In our implementation we set n = 10, 000 and f = 15. The segmentation output y $\in \mathbb{R}^{n \times p}$ gives pointwise prediction for p semantic labels. ⊗ represents concatenation. Conv2D[64] denotes a 2D convolutional layer with 64 filters.

pooling layer, followed by two consecutive FC layers with 256/512 hidden units. Finally, the input point cloud is transformed into a canonical space by a transformation matrix that is estimated by the transform net.

4.2.2 Network Architecture

The segmentation backbone is built on top of the EdgeConv block, which is illustrated in the bottom of **Figure 2**. The EdgeConv can capture local geometric features while preserving permutation invariance Wang et al. (2019). Stacking multiple EdgeConv layer can further capture high-level semantic features. Let $\mathcal{X} = \{x_1, x_2, \ldots x_n\} \in \mathbb{R}^{n \times f}$ represent a f-dimensional point cloud with n sample points. The point cloud can be transformed to a directed graph $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$, where $\mathcal{E}$ and $\mathcal{V}$ denote edges and vertices, respectively. The EdgeConv uses a k-NN graph to capture the edge feature e_{ij} between each point x_i and its k nearest neighbors $\left\{x_j \colon (i, j) \in \mathcal{E}\right\}$. Mathematically, in our network, e_{ij} is computed as

$$e_{ij} = ReLU\Big(\theta_m \cdot \left(x_j - x_i\right) + \varphi_m \cdot x_i\Big), \tag{4}$$

where $\{\theta_1, ..\theta_m, \phi_1, ..\phi_m\}$ are learnable parameters, $ReLU$ is the ReLU activation function, and features $\left\{x_i, x_j\right\} \in \mathbb{R}^{15}$. Compared with the vanilla EdgeConv block, our EdgeConv further applies channel-wise symmetric aggregation with both max-pooling and mean-pooling to better fuse neighbor information. This modification could help learn more expressive local and global representations. The architecture of our EdgeConv can be described as: a k-nn graph for edge feature extraction, three consecutive Conv2D layers with 64 filters, and a max/mean-pooling layer. In particular, we set k = 30 for the segmentation network, and we employ three stacked EdgeConv blocks to extract the hierarchical representations and concatenate each layer by another Conv2D layer and max-pooling. Furthermore, to include the category information about mandible and maxillary, we feed a one-hot categorical vector into a Conv2D layer, and the output is concatenated with the corresponding point representations learned from stacked EdgeConv blocks. The detailed settings of each layer is illustrated in **Figure 2**.

The overall schema and working mechanism of our tooth segmentation network can be summarized as follows. The input mesh is transformed into a 3D point cloud with a shape of $\mathbb{R}^{10000 \times 15}$, which will be sent to the transform net to aligned into a canonical space. Afterward, three EdgeConv are stacked, consecutively, to extract hierarchical point representations, i.e., the bottom EdgeConv block can learn local geometrical features, while the top EdgeConv block can learn high-level semantic features. Such hierarchical representation learning is achieved because we compute the k nearest neighbors based on the outputs from proceeding layers of EdgeConv blocks. Hence, the proximity is dynamically updated from layer to layer, leading to extracted local geometrical features in bottom EdgeConv blocks and high-level semantic features in top EdgeConv blocks. These hierarchical features are gathered by a Conv2D and max-pooling layer to obtain pointwise features fused with both local and global representations. After encoding the one-hot category information, the concatenated outputs will be fed into a series of convolution layers and dropout layers to learn global features and generate the final segmentation results $\hat{\mathbf{y}}_i \in \mathbb{R}^{10000 \times 33}$, where 33 is the number of classes in our segmentation task.

4.2.3 Loss function

We use the cross-entropy loss to train the network. To avoid overfitting, we further add an L2 regularization term in addition to the cross-entropy loss. The loss function is defined as follows:

$$L = \underbrace{-\frac{1}{N}\sum_{i=1}^{N} y_i \ln\left(\text{Softmax}\left(\hat{y}_i\right)\right)}_{Cross-Entropy-Loss} + \underbrace{\lambda \cdot \sum_{\upsilon \in \mho} \upsilon^2}_{L2-Loss}, \tag{5}$$

where N denotes the number of points in the point cloud, y_i and $\hat{y}_i$ represent the true label and predicted label of i-th point, respectively. υ denotes the weight parameters in the trained model $\mho$, and λ is a hyperparameter to adjust the significance of the L2 regularization term.

4.3 Encryption Authority

The encryption authority is a third-party agency that provides public and private keys for clients participating in federated learning. It is necessary that the encryption authority is independent of the server, i.e., the server should not obtain the private key for decryption. Here, we introduce homomorphic encryption (HE) in our federated learning framework. HE is an encryption method that allows the mathematical operation to be performed on the ciphertext, and the result of the operation after decryption is consistent with that of the direct operation on the plain text. Rivest et al. (1978) in their study raise the concept of homomorphic encryption. From the perspective of mathematical operation, it mainly consists of two branches: fully homomorphic encryption (FHE) and partial homomorphic encryption (PHE). The PHE is further divided into two parts: addition homomorphism and multiplication homomorphism. The FHE algorithm supports both multiplication and addition. Consequently, FHE would request a lot more computation resources than PHE. Normally an HE algorithm contains three parts: key generation, encryption, and decryption, denoted by *Keygen*, *Enc*(), and *Dec*(), respectively. *Keygen* will generate the public key K_p for encryption and private key K_v for decryption. Let m_1, m_2 denote the plain text, HE has the following property:

$$Enc\left(m_1, K_p\right) \otimes Enc\left(m_2, K_p\right) = Enc\left(m_1 \oplus m_2\right), \tag{6}$$

where $\otimes$ and $\oplus$ represent mathematical operators. If the encryption method satisfies **Eq. 6**, it conforms the homomorphism on $\oplus$ operation, such as addition or multiplication. Recent work by Kaissis et al. (2020) and Shah et al. (2021) has shown popularity and feasibility to include addition homomorphism in distributed learning. In our encryption authority, we employ the Paillier algorithm, which is an addition homomorphism invented by Paillier, (1999): $Enc(m_1, K_p) \cdot Enc(m_2, K_p) = Enc(m_1 + m_2)$.

During the communication, each client encrypts their local model parameters with public key requesting from the encryption authority, i.e., $\zeta_t^k \leftarrow Enc\left(\omega_t^k, K_p\right)$. Upon gathering all clients' uploaded messages, the server will perform aggregation on the messages as shown in **Eq. 3**, and then distribute the updated message ζ_{t+1}^k and total size of training data $\mathbb{N}$ to each client. The updated model parameters are calculated as: $\omega_{t+1}^k \leftarrow \frac{Dec(\zeta_{t+1}^k, K_v)}{\mathbb{N}}$.

4.3.1 Paillier Cryptosystem

Paillier cryptosystem is a probabilistic public key encryption system raised by Paillier in 1999. It supports addition homomorphism, i.e., the addition of two ciphertexts. It also supports ciphertext multiplied by a non-negative integer plaintext. The algorithm of key generation, encryption, and decryption is presented in **Algorithm 1**

Based on FedAvg McMahan et al. (2017), we established our privacy-preserving framework with a homomorphic encryption mechanism. Assume there are n clients collaboratively contributing to a global model, the objective is to minimize the global loss function:

$$\arg\min_{\omega} \sum_{k=1}^{n} \alpha_k L_k\left(\omega^k\right), \tag{7}$$

TABLE 1 | Symbol notations.

Parameter	Description
$\mathbb{D}_k$	The dataset of k^{th} client
d_i^k	The i^{th} patient sample in k^{th} client dataset
N_k	The size of the dataset in k^{th} client
ℓ_i	Loss function
ω^k	Local model parameters in k^{th} client
E	Number of epochs during a local training
ζ^k	Encrypted model parameters in k^{th} client
K_p, K_v	Public key and private key

Algorithm 1. Paillier cryptosystem.

Key generation
Pick two large prime numbers p and q, which satisfy
(1) $\gcd(p \cdot q, (p-1) \cdot (q-1)) = 1$ (2) p and q are of equal length.
Calculate n: $n = p \cdot q$
Calculate λ: $\lambda = \mathrm{lcm}(p-1, q-1)$
Define function $L(x)$: $L(x) = \frac{x-1}{n}$
Randomly select an integer $g \in Z_{n^2}^*$, define μ as below (If such μ doesn't exist, repeat from start):
$\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$
The public key for encryption is (n, g), and the private key for decryption is (λ, μ).

Encryption
Let m be the message (plaintext) where $0 < m \leq n$, and select random r where $r \in Z_n^*$. With the public key (n, g), the encrypted ciphertext c is calculated as:
$c = g^m \cdot r^n \bmod n^2$

Decryption
With the private key (λ, μ), the message could be decrypted from the following formula:
$m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$

Additive Homomorphism
1) $Enc(m_1) \cdot Enc(m_2) = g^{m1} \cdot r_1^n \cdot g^{m2} \cdot r_2^n \mod n^2 = g^{m1+m2} \cdot (r_1 r_2)^n \mod n^2 = Enc(m_1 + m_2)$
2) $Enc(m_1))^k = (g^{m_1} r^n)^k \mod n^2 = g^{m_1 \times k}(r^k)^n \mod n^2 = Enc(m_1 \times k)$

Algorithm 2. FedTSeg.

```
Initialize $\omega_0^k$ for $\forall k \in n$
for each round $t$=0,1,2,... do
    for each client $k \in n$ do
        for local epochs $e \in E$ do
            for batch $b \in \mathbb{D}_k$ do
                $\omega_t^k \leftarrow \omega_t^k - \eta \bigtriangledown \ell(\omega_t^k; b)$
            end for
        end for
        $\zeta_t^k \leftarrow Enc(\omega_t^k)$
        Send $\zeta_t^k$ and $N_k$ to the server
    end for
    $\zeta_{t+1} \leftarrow \sum_{k=1}^n N_k \zeta_t^k$
    Send $\zeta_{t+1}$ and $\mathbb{N}$ to each client                $\triangleright \mathbb{N} = \sum_{k=1}^n N_k$
    for each client $k \in n$ do
        $\omega_{t+1}^k \leftarrow \frac{Dec(\zeta_{t+1})}{\mathbb{N}}$
    end for
end for
```

where $\alpha_k = \frac{N_k}{\mathbb{N}}$ and $\sum_{k=1}^n \alpha_k = 1$. L_k () is the local loss function for client k defined in **Eq. 1**. In our framework, we let all participants update their local model for a certain training period E, where they use the stochastic gradient descent (SGD) method to calculate weight updates in each round. Finally, the overall algorithm for our FedTSeg is presented in **Algorithm 2**. Key notations are present in **Table 1**

5 EXPERIMENT

5.1 Experimental Setup

5.1.1 Dataset and Preprocessing

We collect 3,000 IOS meshes with labels annotated by experts. Each scan exhibits a 3D mesh for a patient's mandible or maxillary with corresponding labels for each face denoting the tooth and gingiva. We randomly split the dataset into 80% for the

TABLE 2 | Performance of five clients' local models when epoch = 100. "Size" means the proportion of the dataset for each client; w/FL means clients are within the FedTSeg framework; w/o FL means clients perform purely local training; *global* denotes the global model after aggregation; and *avg* calculates the average testing accuracy of five local models without the FL framework.

Distribution	Client (size)	w/FL			w/o FL		
		mIoU	DSC	Acc	mIoU	DSC	Acc
Balanced	c_1(20%)	70.68	77.49	87.4	70.12	78.12	86.76
	c_2(20%)	78.24	84.09	90.61	72.62	79.66	86.82
	c_3(20%)	78.54	84.60	90.19	73.69	80.83	87.67
	c_4(20%)	76.22	82.79	87.88	79.63	85.10	90.68
	c_5 (20%)	78.85	84.70	90.33	56.88	65.26	75.30
	global/avg	80.63	86.03	91.39	70.58	77.79	85.46
Imbalanced	c_1′(8%)	77.90	84.06	89.62	40.45	46.46	64.64
	c_2′(8%)	72.39	79.08	88.05	70.49	78.33	84.17
	c_3′(24%)	76.85	82.69	90.69	44.87	48.99	68.11
	c_4′(26%)	75.02	81.18	89.66	74.18	81.22	87.96
	c_5′(34%)	78.41	83.95	91.69	64.18	72.92	80.12
	global/avg	81.49	86.42	92.53	58.83	65.58	77
CL (as reference)	c_1 (100%)	82.56	86.97	93.26			

training set and 20% for the testing set. The training set will be distributed to clients following different experimental settings, while the testing set is the same for each client. The training samples are randomly transformed to get multiple augmentations to achieve better generalization ability. In particular, the transformation is defined as $\mathcal{T} = [\mathcal{R}|\mathcal{S}|\tau]$, where $\mathcal{R}, \mathcal{S}, \tau$ denote the rotation, scaling, and translation, respectively. For rotation $\mathcal{R}$, the mesh will rotate around an arbitrary axis with a random degree chosen from [0°, 5°, 10°]. The scaling $\mathcal{S}$ will scale the mesh with a factor randomly chosen from (83%, 115%). We will apply a global translation to the mesh under the 3D Cartesian coordinates. Each data sample would be augmented four times, and the corresponding features are extracted following the procedure given in **Section 4.1.1**.

5.1.2 Implementation Details

As shown in **Figure 2**, the network architecture is summarized as: Input mesh → Point cloud $x \in \mathbb{R}^{10000\times15}$ → Transform Net → EdgeConv → EdgeConv → EdgeConv → Conv2D [1024] → Maxpool → Conv2D[256] → Dropout → Conv2D[256] → Dropout → Conv2D[128] → Output. The number in the bracket denotes the number of filters in the 2D convolutional layer. Unless otherwise indicated, all the Conv2D layers use a kernel size of 1 × 1 and a stride of 1 with batch normalization and ReLU activation. We set $k = 30$ for k-nn graph in EdgeConv block. The dropout ratio is 0.6. The output layer is a Conv2d[33] without batch normalization and ReLU activation. The network is trained with a batch size of 4. During local training, each client uses an Adam optimizer for gradient descent, with a basic learning rate $\eta = 0.002$, an exponential decay rate of 0.7, and decay steps of 16,0000. To keep the fairness of performance evaluation, we measure all models with the same testing set. All experiments are conducted using a GPU of NVIDIA GeForce RTX 3090 (24G) and Intel(R) Xeon(R) Gold 6139M CPU @ 2.30GHz, including but not limited to training, testing, encryption/decryption process, and inference.

5.1.3 Federated Setting

In our experiments, we simulate a real-world situation where different medical institutions participate collaboratively in 3D tooth segmentation. We considered two distributions: "balanced" and "imbalanced." "Balanced" distribution means the size of the training set for each client is the same, and "imbalanced" distribution means the sizes are not necessarily the same. Let E denote local training epochs between each communication round. Unless specified, we set $E = 20$, i.e., for every 20 epochs, each client should send their local model's parameter to the server.

5.1.4 Metrics

In performance analysis, we include three metrics to evaluate the segmentation performance measuring the performance of models: mean intersection over union (mIoU), dice coefficient (DSC), and pointwise classification accuracy (Acc). Let X denote the segmentation results and Y denote ground truths, then $\mathrm{DSC} = \frac{2\cdot|X \cap Y|}{|X|+|Y|}$, $\mathrm{mIoU} = \frac{|X \cap Y|}{|X \cup Y|}$.

5.2 Results

5.2.1 FL *vs*. w/o FL

We first compare the performance between global/local models with FedTSeg (FL) and average/local models without FedTSeg (w/o FL). We simulate five independent clients under balanced distribution (c_1, c_2, ..c_5). Each client uses 20% of the original training set to carry on local training. The result is shown in **Table 2**. As for the global model with FedTSeg and the average performance of five local models without FL, the global model of FedTSeg reaches 80.63% mIoU, 86.03% DSC, and 91.39% Acc, outperforming the average model without FL by 10.05% mIoU, 8.24% DSC, and 5.93% Acc. Such a large performance margin demonstrates the effectiveness and necessity of employing federated learning for 3D tooth segmentation.

As for the local models, we can notice that, without FedTSeg, c_3's local model gets the highest performance with Acc 90.68%,

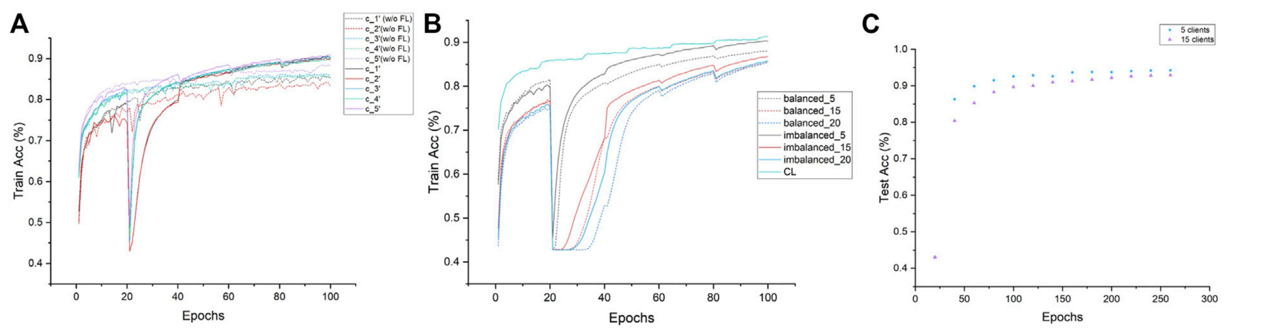

FIGURE 3 | Convergence analysis of each experiment. **(A)** Demonstration of the convergence of five clients under imbalanced distribution within FedTSeg (solid line) and without FedTSeg (dash) for 100 epochs. **(B)** Demonstration of the convergence of different clients and distributions with FedTSeg for 100 epochs. **(C)** Extension of the five-client experiment and 15-client experiment in **Table 3**, to 260 epochs.

DSC 85.10%, and mIoU 79.63%, while c_5's local model gets a terrible performance with Acc 75.30%, DSC 65.26%, and mIoU 56.88%. Though the sizes of training sets are the same, there is still a large variance among their final results, which indicates that the data heterogeneity significantly influences the local model performance. The global model of FedTSeg, trained with the aggregated dataset, significantly outperforms all local models without FedTSeg. Besides, there is a substantial decrease in the variance among five clients with FedTSeg. In particular, the standard deviation of Acc decreases from 5.27% (without FL) to 1.35% (with FL), indicating that the influence of data heterogeneity is materially alleviated by FedTSeg.

We further conduct experiments with five clients under the imbalanced distribution, i.e., the clients are denoted as (c_1′, c_2′, . . ., c_5′). As shown in the bottom part of **Table 2**, c_1′ and c_2′ hold the smallest training set with a size of 8%, and c_5′ holds the largest training set with a size of 34%, simulating the scenarios of small clinics and large hospitals in the real world. Without FedTSeg, the avg model under imbalanced distribution is worse than that under balanced distribution by 11.75% mIoU, 12.21% DSC, and 8.46% Acc. Since the size of the training set is too small, c_1′'s model only gets a result of 64.64% Acc, 46.46% DSC, and 40.45% mIoU, while c_2′'s model gets a better result of 84.17% Acc, 78.33% DSC, and 70.49% mIoU. Though using the same size of the training set, there still exists a large gap in performance between c_1′ and c_2′, indicating a significant influence of data heterogeneity. The variance of performance grows larger than that of balanced distribution with a standard deviation increase from 5.27 to 9.08% due to data heterogeneity and varying amounts of local training data under imbalanced distribution.

But with FedTSeg, the performance of local and global five-client models is significantly boosted, e.g., c_1′'s Acc increases from 64.64 to 89.62%. We can notice that all five clients' local models improve their performance while maintaining a smaller variance, as compared to the local model trained without FedTSeg. Moreover, the global model outperforms all local models, including those with FedTSeg. Thus, clients could obtain a better global model with FedTSeg, especially for those who only possess a small amount of training data.

We also study the training process and investigate the convergence properties of network training with or without FedTSeg under imbalanced distribution. The training process of the five-client setting is illustrated in **Figure 3A**. We can see that the convergence of clients c_1′ and c_2′ during training is relatively slower than client c_5′, which possesses a larger size than the local dataset. The convergence performance is positively correlated with local data size, while our FedTSeg could break the limit and boost the convergence of training, i.e., all five clients' convergence speed within FedTSeg is faster than clients with pure local training. Moreover, the clients exhibit a relatively smaller fluctuation of training accuracy during learning, as demonstrated by the smoother lines in **Figure 3A**.

Lastly, we trained a model under the centralized learning (CL) paradigm where we assumed that all data samples could be aggregated and available during learning. The performance of this CL model could be regarded as the upper bound of our 3D tooth segmentation task. The results are reported in **Table 2**. We can notice that the performance of the FedTSeg framework is on par with CL, i.e., CL only slightly outperforms the global model with FedTSeg by 1.07% mIoU, 0.55% DSC, and 0.73% Acc. Compared to the large margin between models with or without FedTSeg, such an improvement is relatively smaller, demonstrating the effectiveness of our FedTSeg model when data sharing is prohibited due to privacy concerns.

5.2.2 More Clients in Imbalanced and Balanced Distribution

In real-world scenarios, we suppose there are more clients participating in the FedTSeg framework. Thus, we simulate FedTSeg under balanced and imbalanced distribution with more clients, i.e., 15 and 20 clients. The detailed distribution of the local training dataset and the testing accuracy of each client's local model with FedTSeg under imbalanced distribution are displayed in **Figure 4**. The performance of the global model with different numbers of clients is reported in **Table 3**.

We can notice that the global model under imbalanced distribution gets better performance than that of balanced distribution. To better understand such an effect, we record the convergence performance of experiments related to

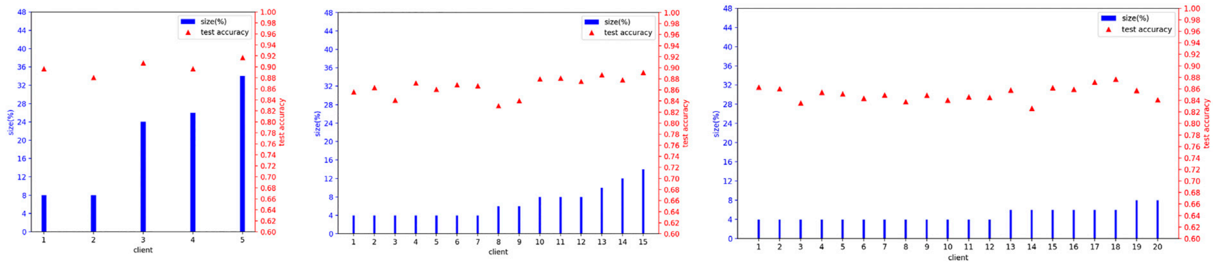

FIGURE 4 | Details of imbalanced distribution for local training set size and local model's test accuracy at Epoch = 100 before aggregation.

TABLE 3 | Performance of imbalanced and balanced distribution on different clients' number at epoch = 100.

Clients	Imbalanced			Balanced		
	mIoU	DSC	Acc	mIoU	DSC	Acc
5	81.49 (+0.86)	86.42 (+0.39)	92.53 (+1.14)	80.63	86.03	91.39
15	78.32 (+3.14)	84.59 (+2.38)	89.76 (+1.64)	75.18	82.21	88.12
20	76.13 (+1.99)	83.34 (+1.99)	88.12 (+0.36)	74.14	81.35	87.76

TABLE 4 | Performance of the FL global model for 5 clients and 15 clients at epoch = 100 and epoch = 260.

Clients	Epoch = 100			Epoch = 260		
	mIoU	DSC	Acc	mIoU	DSC	Acc
5	81.49	86.42	92.53	85.40	89.28	94.14
15	78.32	84.59	89.76	82.41	86.97	92.76

Table 3 in **Figure 3B**. For reference, we also add the convergence curve of the centralized learning paradigm. We can see that the convergence under imbalanced distribution is slightly faster than that under balanced distribution. Hence, we conjectured that the clients with large-scale high-quality data predominate during parameter updating, which could greatly contribute to an excellent global model, especially in the early stage of training. Such a conjecture is empirically demonstrated in **Figure 3C**;

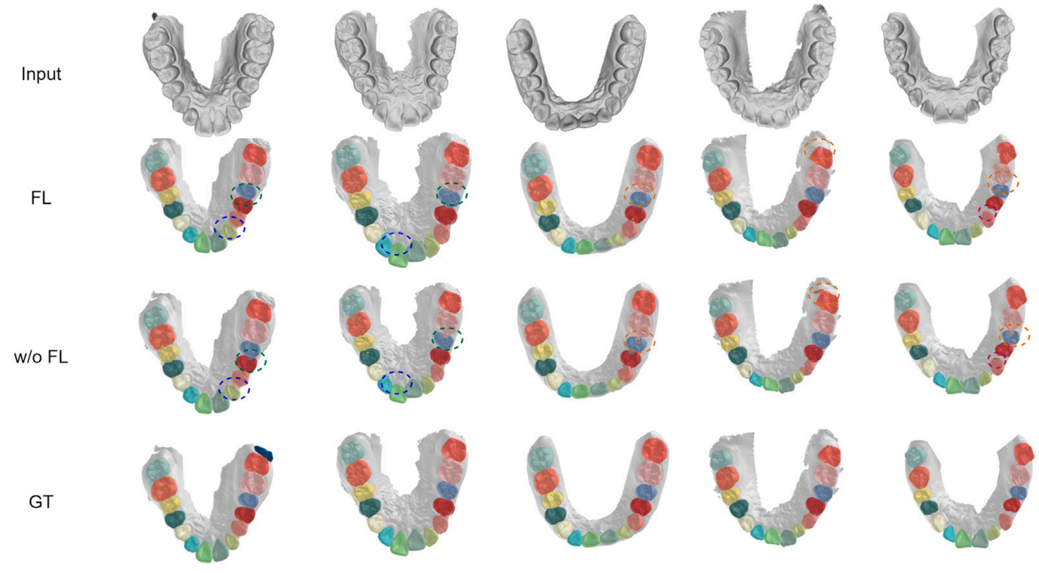

FIGURE 5 | Visualization of segmentation results under balanced distribution with FedTSeg framework and without FedTSeg framework. "FL" denotes the global model with FedTSeg and "w/o FL" denotes the local model under pure local training. The differences are annotated with dotted circles.

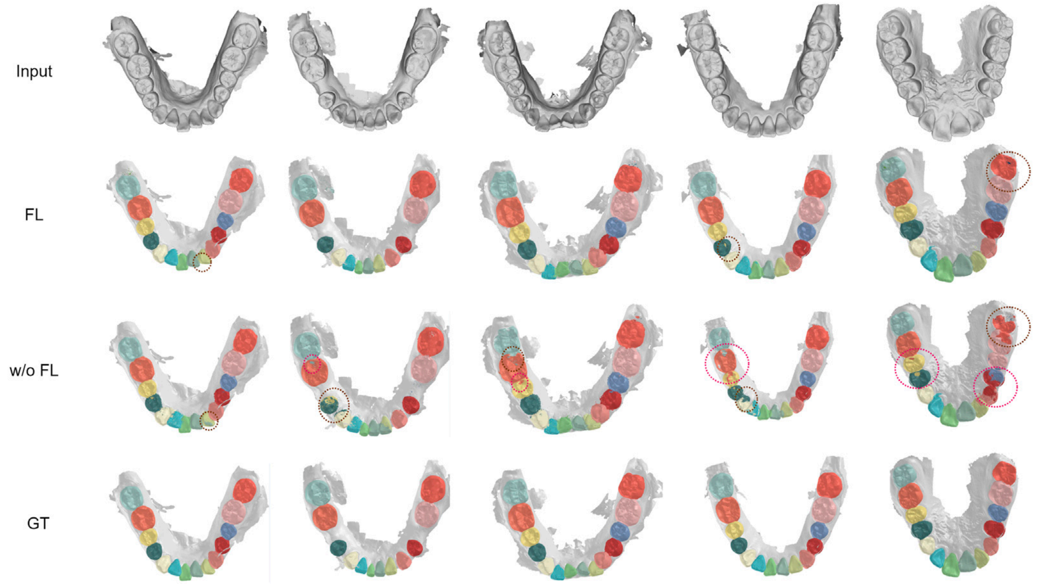

FIGURE 6 | Visualization of segmentation results under imbalanced distribution with FedTSeg and without FedTSeg. Some distinct errors are annotated with dotted circles.

Table 4, i.e., the gap of Acc between the five-client and 15-client global model shrinks from 2.77% (trained for 100 epochs) to 1.38% (trained for 260 epochs). In particular, though the performance of the global model with five clients is saturated when trained 100 epochs, the performance of the global model with 15 clients still can be slowly improved if the training epoch is increased. This is reasonable because the global model needs to be progressively improved, especially in settings with lots of clients each with a smaller number of local training samples.

We can also notice that both global models of 15 clients and 20 clients outperform the local models under balanced and imbalanced distribution, which is consistent with the above results of the five-client setting, demonstrating that FedTSeg can gather the advantages of multiple clients and boost the performance of local models. Under imbalanced distribution, the performance of the global model consisting of five clients is better than that of 15 clients by 3.17% mIoU, 1.83% DSC, and 2.77% Acc, and the 15-client global model is better than the 20-client model by 2.19% mIoU, 1.25% DSC, and 1.64% Acc. Under the balanced distribution, the five-client global model is better than 15-client global model by 5.45% mIoU, 3.82% DSC, and 3.27% Acc, and 15-client's better than 20-client's by 1.04% mIoU, 0.86% DSC, and 0.36% Acc. Since these three experiments (5, 15, and 20 clients) use the same original training data, the average size of clients would decrease when the number of clients increases, leading to performance degradation in experiments.

5.3 Visualizations

We demonstrate the effectiveness of our method with detailed case visualizations. Five visualization cases of segmentation results under balanced distribution are displayed in **Figure 5**, where each column represents a specific case. By comparing the segmentation results of models with or without FedTSeg, we can see that the global model of FedTSeg could predict a more precise segmentation result than pure local training under balanced distribution. Concretely, we can notice that the "w/o FL" model would mistakenly recognize part of the lateral incisor as cuspid, as shown in case 1. Without FedTSeg, there are errors of omission at the boundary between the central incisor and gingiva (tooth-gingiva boundary) in cases 2 to 5. More mistakes are also committed for the tooth–tooth boundary, i.e., there are boundary segmentation errors between the second bicuspid and first molar as in cases 2 and 5. As for the segmentation of molars, we can see that the predicted shape of the second molar by "w/o FL" model is visibly different from the ground truth, as shown in case 4. In contrast, the model trained with FedTSeg, though not perfect, rarely commits such mistakes, leading to much better segmentation results that would be more appealing to real-world clinical applications.

We visualized another five cases of segmentation results under imbalanced distribution, as shown in **Figure 6**. In cases 2, 4, and 5, there are obvious mistakes predicting tooth parts of the cuspid and first bicuspid without FedTSeg. In cases 2, 3, 4, and 5, the boundaries among the second molar, first molar, and second bicuspid are not correctly segmented. While with FedTSeg, these errors are significantly solved, which further demonstrates the effectiveness of the FedTSeg framework under imbalanced distribution.

5.4 Communication Analysis

In FedTSeg, each client will encrypt their model parameters and decrypt the ciphertext for each communication round using Paillier algorithm. To quantify the magnitude of external communication cost, we record the training time, encryption

TABLE 5 | Time analysis of the homomorphic encryption process. For training time, we record the average time for training one epoch with 20% data.

Key length	Training time (min)	Encryption time (min)	Decryption time (min)
256	6 (20% training set)	36	15
512		216	78
1024		1333	441

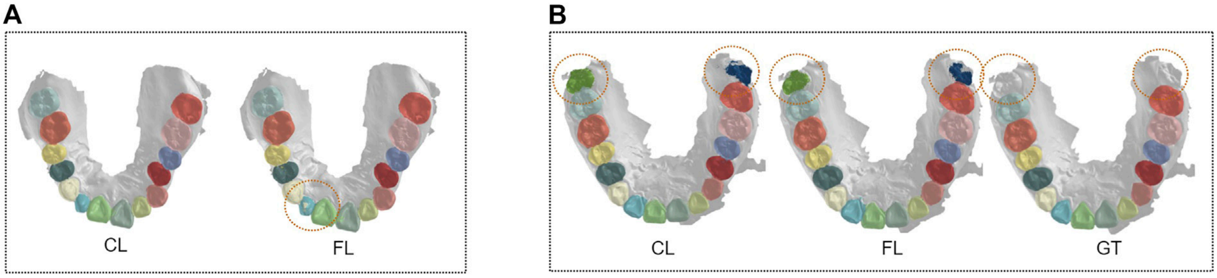

FIGURE 7 | Visualization of segmentation results under a centralized paradigm and federated learning framework. **(A)** Comparison between FL and CL. **(B)** Comparison between FL and GT.

time, and decryption time. Specifically, we used Tensorflow to train our network, where the model is stored in "ckpt" format. We used the open-source framework "Python-Paillier" to perform element-wise encryption on the model, i.e., each floating number inside the tensors is encrypted by the public key. For each model, there are 295 tensors with 545,6841 floating points to be encrypted.

The result is reported in **Table 5**. It takes about 6 min to train one epoch in the five-client setting with the balanced distribution. It takes 36 and 15 min to encrypt a local model and decrypt the upcoming global model using the 256-bit public key and private key. With the increase in key length, the ciphertext is more secure. However, the time cost for encryption and decryption also increases, respectively. What calls for special attention is that the computation ability of GPU/CPU can influence the training and homomorphic time cost, thus the value in **Table 5** only gives a referenced value in our experimental setup as described in **Section 5.1**.

6 DISCUSSION

There are nevertheless some limitations of our work. First, there still exists a gap between FL and the centralized paradigm. As shown in **Figure 7A**, there are some mistakes in annotating small tooth parts in the lateral incisor with FedTSeg, while the segmentation from CL is more precise. Meanwhile, though our FedTSeg can detect the missing unlabeled teeth and avoid manual mistakes, it is still inferior to the CL in extreme cases. Ideally, an applicable federated learning framework should be able to reach the same performance as a centralized learning paradigm or even surpass it. More domain-specific design might be required to better resolve the federated 3D tooth segmentation tasks, e.g., frameworks considering the anatomical and morphological features of different oral diseases.

Moreover, the federated learning framework should be able to generalize to settings with a large number of clients efficiently. In our preliminary results, more training time is required to better convergence of FedTSeg with more clients, though we simulate it with fewer samples in each client. It is reasonable that enlarging the local training set might help faster convergence of the FL framework with many clients. But novel designs are also appealing to deal with the scenarios where there are lots of distributed clients with a relatively smaller number of training samples. This is also the case when we have lots of small clinics rather than large hospitals for federated medical image analysis in practice.

Last but not least, further exploration is of high necessity to improve the communication and model aggregation efficiency, though some recent research already shed light on this direction. There exists computational heterogeneity among clients. For those hospitals with stronger computation ability, they would finish training and encryption faster than others, then they would have to wait for others to finish their communication. The cannikin law indicates that our framework would be limited by the slowest client. Asynchronous model aggregation might be adopted for faster convergence. Moreover, the weight parameters of the neural networks could be quantized or pruned to decrease the communication overload for better communication efficiency. Combining federate learning with cutting-edge deep-neural-network-compression techniques is a promising solution to learning powerful models efficiently with lots of mobile-edge devices.

7 CONCLUSION

In conclusion, we design and develop a federated learning framework FedTSeg for federated 3D IOS tooth segmentation, and achieve privacy-preserving via homomorphic encryption. Comprehensive experimental results reveal that FedTSeg could obtain much better global and local models than conventional local training paradigms, and could achieve segmentation performance on par with the centralized paradigms where we assure all data could be aggregated. Moreover, the data is strictly protected with homomorphic encryption, preventing attacks during parameter exchange. Our future work will focus on handling the performance degradation issue in scenarios with a large number of clients, each associated with a limited number of local training data, which is closely related to the long-tailed learning problem. Further techniques to resolve communication efficiency issues are also highly demanded to help deploy our framework to real-world applications.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

ZL initialized the project. SL, HY, and ZL designed the framework. JH and YF prepared the dataset. SL, YT, HY, and ZL implemented the framework and performed statistical analysis. SL and ZL wrote the manuscript. All authors significantly revised and approved the manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (62106222).

ACKNOWLEDGMENTS

The authors thank Xiang Li, Mingzhou Wu, and Chenxi Liu for their helpful discussion and initial work on this study.

REFERENCES

Bercea, C. I., Wiestler, B., Rueckert, D., and Albarqouni, S. (2021). Feddis: Disentangled Federated Learning for Unsupervised Brain Pathology Segmentation. *arXiv preprint arXiv:2103.03705*.

Christ, P. F., Ettlinger, F., Grün, F., Elshaera, M. E. A., Lipkova, J., Schlecht, S., et al. (2017). *Automatic Liver and Tumor Segmentation of Ct and Mri Volumes Using Cascaded Fully Convolutional Neural Networks* arXiv preprint arXiv: 1702.05970.

Cui, Z., Li, C., Chen, N., Wei, G., Chen, R., Zhou, Y., et al. (2021). Tsegnet: an Efficient and Accurate Tooth Segmentation Network on 3d Dental Model. *Med. Image Anal.* 69, 101949. doi:10.1016/j.media.2020.101949

Bejnordi, B. E., Veta, M., Johannes van Diest, P., van Ginneken, B., Karssemeijer, N., Litjens, G., et al. (2017). Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA* 318, 2199–2210. doi:10.1001/jama.2017.14585

Fan, Z., Su, J., Gao, K., Hu, D., and Zeng, L.-L. (2021). "A Federated Deep Learning Framework for 3d Brain Mri Images," in 2021 International Joint Conference on Neural Networks (IJCNN), 1–6. doi:10.1109/IJCNN52387.2021.9534376

Feki, I., Ammar, S., Kessentini, Y., and Muhammad, K. (2021). Federated Learning for Covid-19 Screening from Chest X-Ray Images. *Appl. Soft Comput.* 106, 107330. doi:10.1016/j.asoc.2021.107330

Gordienko, Y., Gang, P., Hui, J., Zeng, W., Kochura, Y., Alienin, O., et al. (2018). "Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for Chest X-Ray Analysis of Lung Cancer," in International conference on computer science, engineering and education applications (Springer), 638–647. doi:10.1007/978-3-319-91008-6_63

Hao, J., Liao, W., Zhang, Y., Peng, J., Zhao, Z., Chen, Z., et al. (2021). Toward Clinically Applicable 3-dimensional Tooth Segmentation via Deep Learning. *J. Dent. Res.* 101, 00220345211040459. doi:10.1177/00220345211040459

Herrmann, W. (1967). On the Completion of Fédération Dentaire Internationale Specifications. *Zahnarztl Mitt* 57, 1147–1149.

Kaissis, G. A., Makowski, M. R., Rückert, D., and Braren, R. F. (2020). Secure, Privacy-Preserving and Federated Machine Learning in Medical Imaging. *Nat. Mach. Intell.* 2, 305–311. doi:10.1038/s42256-020-0186-1

Kingma, D. P., and Ba, J. (2015). Adam: A Method for Stochastic Optimization. *ICLR. arXiv preprint arXiv:1412.6980*.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated Optimization in Heterogeneous Networks. *Proc. Mach. Learn. Syst.* 2, 429–450.

Li, W., Milletarì, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., et al. (2019). "Privacy-Preserving Federated Brain Tumour Segmentation," in International workshop on machine learning in medical imaging (Springer), 133–141. doi:10.1007/978-3-030-32692-0_16

Li, X., Jiang, M., Zhang, X., Kamp, M., and Dou, Q. (2021). Fedbn: Federated Learning on Non-iid Features *via* Local Batch Normalization. *ICLR arXiv preprint arXiv:2102.07623*.

Lian, C., Wang, L., Wu, T.-H., Wang, F., Yap, P.-T., Ko, C.-C., et al. (2020). Deep Multi-Scale Mesh Feature Learning for Automated Labeling of Raw Dental Surfaces from 3d Intraoral Scanners. *IEEE Trans. Med. Imaging* 39, 2440–2450. doi:10.1109/TMI.2020.2971730

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., et al. (2017). A Survey on Deep Learning in Medical Image Analysis. *Med. Image Anal.* 42, 60–88. doi:10.1016/j.media.2017.07.005

Liu, Q., Dou, Q., Yu, L., and Heng, P. A. (2020). Ms-net: Multi-Site Network for Improving Prostate Segmentation with Heterogeneous Mri Data. *IEEE Trans. Med. Imaging* 39, 2713–2724. doi:10.1109/tmi.2020.2974574

Liu, W., Chen, L., Chen, Y., and Zhang, W. (2020). Accelerating Federated Learning via Momentum Gradient Descent. *IEEE Trans. Parallel Distrib. Syst.* 31, 1754–1766. doi:10.1109/tpds.2020.2975189

Lo, J., Yu, T. T., Ma, D., Zang, P., Owen, J. P., Zhang, Q., et al. (2021). Federated Learning for Microvasculature Segmentation and Diabetic Retinopathy Classification of Oct Data. *Ophthalmol. Sci.* 1, 100069. doi:10.1016/j.xops.2021.100069

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully Convolutional Networks for Semantic Segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 3431–3440. doi:10.1109/cvpr.2015.7298965

Lu, Y., Huang, X., Dai, Y., Maharjan, S., and Zhang, Y. (2020). Differentially Private Asynchronous Federated Learning for Mobile Edge Computing in Urban Informatics. *IEEE Trans. Ind. Inf.* 16, 2134–2143. doi:10.1109/TII.2019.2942179

Lundervold, A. S., and Lundervold, A. (2019). An Overview of Deep Learning in Medical Imaging Focusing on Mri. *Z. für Med. Phys.* 29, 102–127. doi:10.1016/j.zemedi.2018.11.002

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). "Communication-efficient Learning of Deep Networks from Decentralized Data," in Artificial intelligence and statistics (PMLR), 1273–1282.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. (2019). "Exploiting Unintended Feature Leakage in Collaborative Learning," in 2019 IEEE Symposium on Security and Privacy (SP) (IEEE), 691. doi:10.1109/sp.2019.00029

Paillier, P. (1999). "Low-Cost Double-Size Modular Exponentiation or How to Stretch Your Cryptoprocessor," in International conference on the theory and applications of cryptographic techniques (Springer), 223–234. doi:10.1007/3-540-49162-7_18

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). "Pointnet: Deep Learning on Point Sets for 3d Classification and Segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 652–660.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Adv. neural Inf. Process. Syst.* 30, 5105–5114.

Ren, J., Wang, H., Hou, T., Zheng, S., and Tang, C. (2019). Federated Learning-Based Computation Offloading Optimization in Edge Computing-Supported Internet of Things. *IEEE Access* 7, 69194–69201. doi:10.1109/ACCESS.2019.2919736

Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., et al. (2020). The Future of Digital Health with Federated Learning. *NPJ Digit. Med.* 3, 119–127. doi:10.1038/s41746-020-00323-1

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* 21, 120–126. doi:10.1145/359340.359342

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional Networks for Biomedical Image Segmentation," in International Conference on Medical image computing and computer-assisted intervention (Springer), 234–241. doi:10.1007/978-3-319-24574-4_28

Shah, U., Dave, I., Malde, J., Mehta, J., and Kodeboyina, S. (2021). "Maintaining Privacy in Medical Imaging with Federated Learning, Deep Learning, Differential Privacy, and Encrypted Computation," in 2021 6th International Conference for Convergence in Technology (I2CT), 1. doi:10.1109/I2CT51068.2021.9417997

Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., et al. (2020). Federated Learning in Medicine: Facilitating Multi-Institutional Collaborations without Sharing Patient Data. *Sci. Rep.* 10, 12598. doi:10.1038/s41598-020-69250-1

Shen, W., Zhou, M., Yang, F., Yang, C., and Tian, J. (2015). "Multi-scale Convolutional Neural Networks for Lung Nodule Classification," in *Information Processing in Medical Imaging*. Editors S. Ourselin, D. C. Alexander, C.-F. Westin, and M. J. Cardoso (Cham: Springer International Publishing), 588–599. doi:10.1007/978-3-319-19992-4_46

Song, M., Wang, Z., Zhang, Z., Song, Y., Wang, Q., Ren, J., et al. (2020). Analyzing User-Level Privacy Attack against Federated Learning. *IEEE J. Sel. Areas Commun.* 38, 2430–2444. doi:10.1109/JSAC.2020.3000372

Tian, S., Dai, N., Zhang, B., Yuan, F., Yu, Q., and Cheng, X. (2019). Automatic Classification and Segmentation of Teeth on 3d Dental Model Using Hierarchical Deep Learning Networks. *IEEE Access* 7, 84817–84828. doi:10.1109/ACCESS.2019.2924262

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). Dynamic Graph Cnn for Learning on Point Clouds. *ACM Trans. Graph.* 38, 1–12. doi:10.1145/3326362

Warnat-Herresthal, S., Schultze, H., Shastry, K. L., Manamohan, S., Mukherjee, S., Garg, V., et al. (2021). Swarm Learning for Decentralized and Confidential Clinical Machine Learning. *Nature* 594, 265–270. doi:10.1038/s41586-021-03583-3

Xu, X., Liu, C., and Zheng, Y. (2019). 3d Tooth Segmentation and Labeling Using Deep Convolutional Neural Networks. *IEEE Trans. Vis. Comput. Graph.* 25, 2336–2348. doi:10.1109/TVCG.2018.2839685

Yan, Z., Wicaksana, J., Wang, Z., Yang, X., and Cheng, K.-T. (2021). Variation-aware Federated Learning with Multi-Source Decentralized Medical Image Data. *IEEE J. Biomed. Health Inf.* 25, 2615–2628. doi:10.1109/JBHI.2020.3040015

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. (2018). "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting," in 2018 IEEE 31st computer security foundations symposium (CSF) (IEEE), 268–282. doi:10.1109/csf.2018.00027

Zanjani, F. G., Moin, D. A., Claessen, F., Cherici, T., Parinussa, S., Pourtaherian, A., et al. (2019). "Mask-mcnet: Instance Segmentation in 3d Point Cloud of Intra-oral Scans," in International Conference on Medical Image Computing and Computer-Assisted Intervention (Springer), 128–136. doi:10.1007/978-3-030-32254-0_15

Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., et al. (2021). A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies with Progress Highlights, and Future Promises. *Proc. IEEE* 109, 820–838. doi:10.1109/JPROC.2021.3054390

Conflict of Interest: YF was employed by the company Angelalign Inc.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.