Mask-MCNet

Neurocomputing 453 (2021) 286–298

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Mask-MCNet: Tooth instance segmentation in 3D point clouds of intra-oral scans

Farhad Ghazvinian Zanjani [a,*], Arash Pourtaherian [a], Svitlana Zinger [a], David Anssari Moin [b], Frank Claessen [b], Teo Cherici [b], Sarah Parinussa [b], Peter H.N. de With [a]

[a] Eindhoven University of Technology, 5612 AJ Eindhoven, The Netherlands
[b] Promaton Co. Ltd., 1076 GR Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 8 January 2020
Revised 3 April 2020
Accepted 9 June 2020
Available online 27 January 2021

Keywords:
Deep learning
3D point cloud
Instance object segmentation
Intra-oral scan

ABSTRACT

Computational dentistry uses computerized methods and mathematical models for dental image analysis. One of the fundamental problems in computational dentistry is accurate tooth instance segmentation in high-resolution mesh data of intra-oral scans (IOS). This paper presents a new computational model based on deep neural networks, called *Mask-MCNet*, for end-to-end learning of tooth instance segmentation in 3D point cloud data of IOS. The proposed Mask-MCNet localizes each tooth instance by predicting its 3D bounding box and simultaneously segments the points that belong to each individual tooth instance. The proposed model processes the input raw 3D point cloud in its original spatial resolution without employing a voxelization or down-sampling technique. Such a characteristic preserves the finely detailed context in data like fine curvatures in the border between adjacent teeth and leads to a highly accurate segmentation as required for clinical practice (e.g. orthodontic planning). The experiments show that the Mask-MCNet outperforms state-of-the-art models by achieving 98% Intersection over Union (IoU) score on tooth instance segmentation which is very close to human expert performance.

1. Introduction

Dentistry has witnessed a rapid growth of technological innovations in advanced imaging methods. Such advances in imaging systems are playing an important role in efficient diagnoses, treatment, and surgeries. Computational dentistry involves computerized methods for automated analysis of digital dental images. It utilizes mathematical and/or data-driven models to facilitate data analysis, e.g. for accurate treatment planning and diagnostic purposes. Computational dentistry may incorporate multiple sources of imaging data obtained by both extra-oral (e.g. X-ray panoramic, cephalometric and cone-beam computed tomography) and intra-oral optical imaging (e.g. laser or structured light projection scanners).

Intra-oral scanners are advanced imaging devices for optical capture of the surface profiles of anatomical structures inside of the patient's mouth. Similar to other 3D scanners, intra-oral scanners project a light source (laser, or structured light) on the surface of objects to be scanned, in this case, the dental arches. Based on the employed technique, the time-of-flight of the laser or the deformation of the projected pattern on the subject's surface is measured by the imaging sensors and processed by the scanning software, which generates a highly accurate 3D point cloud. The obtained point cloud might be further processed and converted to a 3D surface model (mesh) by using *triangulation* techniques.

The obtained 3D point cloud represents the 3D geometrical profile of tooth crowns and gingiva (i.e. gum) in a very high spatial resolution, in the order of 30–80 points per mm^2 [1] with a spatial accuracy of less than 20 μm. Such a precise 3D model is widely used for implant treatment and orthodontic planning. A scan of one dental arch consists of a large set of points (e.g. hundreds of thousands) in an *arbitrary* 3D Cartesian coordinate system. Here, the term arbitrary refers to an unaligned coordinate system between different scans or even for two acquisition trials from the same subject. Each point in a point cloud is represented by its 3D coordinates and, depending on the type of scanner, other attributes such as color. Automated processing of such a large 3D point cloud by a computational model that preserves highly detailed information from anatomic structures of teeth crowns is highly beneficial for many clinical dental applications.

In this study, we introduce a computational model for instance tooth segmentation using such advanced imaging and point cloud information. The segmentation of an individual tooth requires that the imaging data of a point cloud is analyzed to the level of

* Corresponding author.

https://doi.org/10.1016/j.neucom.2020.06.145
0925-2312/

identifying an individual tooth. The purpose of our study is to propose a new methodology for the automation of the clinical workflow and to improve the quality thereof. Here, the instance tooth segmentation of an intra-oral scan (IOS) refers to the assignment of a unique label to all the points belonging to each instance (i.e. an individual tooth), using a computational model. For clinical practice, after segmentation of the tooth instances, a post-processing stage may follow for the standardization of the labels. This stage provides a look-up table for the conversion of the labels assigned to each detected tooth into one final label prescribed by the *Federation Dentaire Internationale* (FDI) dental notation for adult dentition.

In the following, we briefly introduce the related work on IOS segmentation, recent advances of deep learning in semantic instance segmentation on point cloud data, and our contributions to both. Afterwards, we explain our proposed method and the obtained results in detail. Lastly, we provide discussions and conclusions.

2. Related work

Related literature can be divided into two parts: conventional IOS segmentation methods and available deep learning solutions for instance segmentation in 3D point clouds.

2.1. Conventional IOS segmentation approaches

The existing literature on IOS segmentation covers mostly conventional computer vision/graphics techniques, which are limited to finding the best handcrafted features, manual tuning of several parameters and lack of generalization and robustness [2]. Among the proposed methods, one generic approach is first projecting the 3D IOS mesh on one or multiple 2D plane(s) and then applying standard computer vision algorithms. Afterwards, the processed data is projected back into the 3D space. For example, Kondo et al. [3] propose gradient orientation analysis and Wongwaen et al. [4] apply a boundary analysis on a 2D projected panoramic depth images for finding teeth boundaries. Most of the other studies are based on curvature analysis [5–9], *fast marching watersheds* [10], *morphological operations* [9], 2D [11] and 3D [12] active contour (snake) analysis and tooth-target harmonic fields [13] for segmenting the teeth and gingiva. Some other works follow a semi-automatic approach by manually setting a threshold [6], picking some representative points [8], or interactively involve a human operator for the analysis [7,9]. As already mentioned, such a method is always limited to finding the best handcrafted features and their inherent constraints in applying them within computer-aided design (CAD) systems.

2.2. Deep learning approaches

The teeth segmentation problem can be formulated in two different ways. First, the teeth segmentation task can be formulated as a multi-class semantic segmentation problem. Therefore, each tooth instance is considered as a semantic class. The points in the point cloud are assigned to one of those classes, i.e. a *tooth number* or the gingiva. Considering gingiva and a maximum of 16 teeth on each dental arch, each point has a probability of belonging to each of 17 semantic classes. This probability is expected to be predicted at the output of the model. The work of [2] is an example of employing a deep learning model for semantic segmentation of teeth in the IOS data. As a second way, the tooth segmentation problem is formulated in the context of a semantic instance segmentation problem. To do so, only one semantic class is defined that is the class of *tooth*. The points which do not belong to any tooth (e.g. to the gingiva), are considered as the undefined class. In the semantic instance segmentation, apart from assigning a semantic label to each point, indicating whether it belongs to a tooth, the model should assign a secondary unique *arbitrary* label to those points as an indication of individual tooth instance. It is worth mentioning that in contrast to the first approach, the assigned labels to the instances convey no semantic meaning (i.e. they convey no information regarding whether the tooth is an incisor or molar) and is used only to separate the teeth from each other.

Although the former approach is straightforward, its performance for teeth segmentation suffers from two main shortcomings. The first shortcoming originates from the fact that there is low inter-class variability between the crown shape of neighbouring teeth, especially among the molar and premolar teeth. Hence, an accurate prediction of the labels requires not only the local geometrical information (i.e. crown shapes) but also the global context of e.g. the relative position, teeth arrangement and possible absence of other teeth. The second shortcoming is that because of preserving the global context, employing patch-based analysis and processing those individually is not feasible. Hence, due to hardware limitation (e.g. GPU memory), training and inference on the whole point cloud require down-sampling of the point cloud. Such a down-sampling would be detrimental to a precise teeth segmentation where preserving high-frequency information such as curvature at the border of teeth is crucial. Employing *adversarial* training and *non-uniform* resampling of the point cloud are two proposed techniques for addressing these two shortcomings in the work of Zanjani et al. [2].

Alternatively, formulating the problem as semantic instance segmentation does not suffer from missing global context and dependencies between the labels, since it localizes the teeth by fitting a 3D bounding box and simultaneously assigns a unique label to all points belonging to each instance inside the detected bounding box. Accordingly, it locally processes the cropped 3D patches from the point cloud. Later, by aggregating all detections in the processed patches, the model performs the inference on the whole point cloud. As a consequence, the point cloud data is divided in local data processing actions in a natural way, so that the later processing of patches is possible in full quality, preserving the original spatial resolution of the data (without down-sampling). This property greatly facilitates the machine learning algorithms for performing automated instance segmentation.

2.3. Deep learning models for instance segmentation in 3D point cloud

Among the proposed deep learning models for point cloud analysis, only a few researchers have addressed the challenging issue of 3D instance segmentation. To better compare and position our proposed method, we briefly survey recent deep learning models, all related to instance segmentation in a 3D point cloud.

FrustumNet [14] proposes a hybrid framework involving two stages. The first stage detects the objects bounding boxes in a 2D image. The second stage processes the 3D point cloud in a 3D search space, partially bound by the initially set 2D bounding boxes. The *3D-SIS* model [15] also first processes the 2D images rendered from the point cloud through a 2D convolutional network (ConvNet). Afterwards, the learned features are back-projected on the voxelized point cloud data, where the extracted 2D features and the geometric information are combined to obtain the object proposal and per-voxel mask prediction. The dependency on the 2D image(s) of both preceding models limits the application of them for 3D point cloud analysis. In another approach, *GSPN* [16] is a deep learning framework that follows an analysis-by-synthesis strategy and instead of directly finding the object bound-

ing boxes in a point cloud, it utilizes a conditional variational auto-encoder (CVAE). However, GSPN training requires a separate two-stage training of the CVAE part and the region-based networks (which perform the detection, localization and mask generation on the object proposals).

In an alternative approach to detect object proposals, *SGPN* [17] and *MASC* [18] methods perform clustering on the processed points for segmenting the object instances. *SGPN* [17] uses a similarity matrix between the features of each pair of points in the embedded feature space, to indicate whether the given pair of points belong to the same object instance or not. Although computing the pair-wise distance for the small point clouds is practical, it is crucial for large point clouds and especially for IOS data, where down-sampling would significantly affect the detection/segmentation performance. *MASC* [18] voxelizes the point cloud for processing the volumetric data by a 3D U-Net model. Similar to SGPN, MASC uses a clustering mechanism to find similarities between each pair of points by comparing their extracted features in several hidden layers of a trained U-Net. Unfortunately, as mentioned before, voxelization of a large fine-detailed point cloud significantly limits the performance of such approaches.

In another approach, *VoxelNet* [19] first divides a point cloud into equally spaced 3D voxels and then transforms a group of points within each voxel into a feature space. Subsequently, it uses a RPN to generate 3D box detections. The preliminary division of the 3D input space into voxels facilitates the processing of sparse point clouds, such as those collected by LIDAR sensors, since the 3D convolutions can be applied on the constructed volumetric space. However, such voxelization techniques for dense point clouds (like those obtained from a dental scan) may lead to an inhomogenoues feature extraction for neighbouring points, which have been assigned to two adjacent voxels. This can happen for a large number of points in a dense point cloud. To mitigate this effect, decreasing the number of bins would reduce the relevant population of border points in each voxel, however at the expense of the RPN losing spatial accuracy. *VoteNet* [20], instead of converting a 3D point cloud to a regular grid, directly votes for a virtual center of objects from the point clouds. It generates a group of high-quality 3D object proposals by aggregating vote features and predicting offset vectors to the corresponding object centers for seed points, followed by a clustering module to generate object proposals. *PointRCNN* [21] directly segments a 3D point cloud to obtain the foreground points. Afterwards, a bin-based 3D bounding-box generation is performed only around the foreground points to produce high-quality 3D boxes. The authors showed that such an approach achieves state-of-the-art performance on the car detection task in the sparse LIDAR point cloud. Our proposed model is similar to PointRCNN by using a backbone network as a feature extractor and using bins as a search space for the RPN. However, in contrast, we do not segment the input points into two classes of foreground and background. This is because the objects, teeth, are located closely on the dental arch, so the foreground points are the vast majority, and the technique would not be effective to reduce the search space. Instead, we use larger bins to search the entire 3D space. To compensate for such a coarse quantization step, the backbone first encodes the dense point cloud to a high-dimensional feature space and then uses an adaptive pooling operator to transfer the information to the bins.

Summarizing the mentioned research work of the literature indicates that instance segmentation is more suitable for our segmentation problem, but the available studies do not reveal a suitable computational model for processing of the large dense 3D point cloud data, containing fine-detailed information. This results in the following contributions.

2.4. Novelty of the approach

In this paper, we propose an end-to-end deep learning model for instance segmentation in 3D point cloud data. Our contribution is threefold.

1. We present a new instance segmentation model, called Mask-MCNet. Our proposed model is applied directly to an irregular 3D point cloud on its original spatial resolution and predicts the 3D bounding boxes of object instances along with their masks, indicating the segmented points of each instance.
2. To the best of our knowledge, this is the first study which both detects and segments tooth instances in IOS data by a deep learning model.
3. We conduct an extensive experimental evaluation and show that the proposed model significantly outperforms state-of-the-art in IOS segmentation.

The remainder of this paper starts with a detailed description of the individual processing modules of our proposed model in Section 3. Afterwards, the performed experiments and results are presented in Section 4. Section 5 provides discussions and conclusions.

3. Proposed method

Deep learning models for 3D point cloud analysis computationally differ from the ConvNet-based models, since they are applied on non-grid data (unstructured samples). However, at a high level, our proposed Mask-MCNet model is similar to ConvNet-based Mask R-CNN [22] as it includes three main parts: the backbone network, Region Proposal Network (RPN), and three branches of predictor networks for detection, localization by fine-tuning, and mask generation (see Fig. 1). Each part is explained in detail below.

3.1. Backbone network

The *backbone* is a deep network based on the Multi-layer Perceptron (MLP) architecture, which is applied on the entire or cropped (depending on hardware limitations) point cloud and acts as a feature extractor. Every input patch includes n points (varying across patches), where each point is represented by its (x, y, z) 3D coordinates and might have other attributes such as color or a normal vector. A point cloud does not explicitly convey the information from neighbouring points. Therefore, in order to aggregate over local neighborhoods, most existing methods augment the input point cloud with the surface normal vectors or resort to a neighbor searching mechanism (e.g., KNN [23] or ball query [24]). In Section 4, we evaluate the impact of augmenting the input with the normal vectors. Hence, in case of using normal vectors, the input to the backbone model is an $n \times 6$ matrix. The output of the backbone model is a high-dimensional feature vector per given input point (e.g. a matrix of $n \times 256$) which contains rich geometrical information around each point. In this study, we choose to employ a PointCNN [24] for its fine-detail processing capacity and its small model size [24,2]. To demonstrate the generality of our approach, we instantiate the proposed Mask-MCNet with two different backbone architectures. In an ablation study, we also report using PointNet [25] as an alternative choice for a backbone network.

3.2. Region Proposal Network (RPN)

Since the points in the point cloud are distributed solely on the surface of objects, the computed features from the backbone network only contain local geometrical representations on a manifold

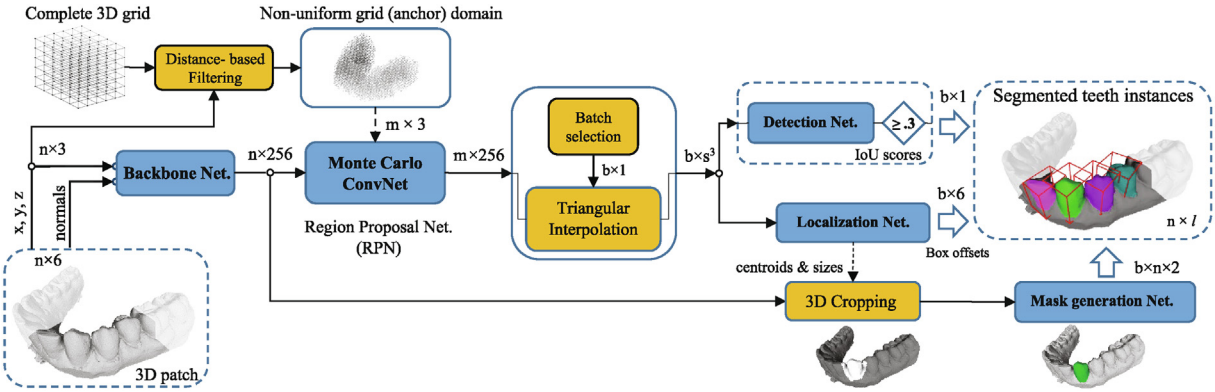


Fig. 1. Block diagram of the Mask-MCNet (training mode).

in 3D space. However, for accurate localization of a 3D bounding box, encompassing an object, the network is required to be aware of several parts (or sides) of each object. Such awareness leads to a reliable learning and consequently an accurate prediction of the center and size of each bounding box. Hence, voxelization of the data and employing a 3D ConvNet on the obtained volumetric data is a common approach. However, the shortcomings of such an approach have been mentioned already. Therefore, as an alternative method, for distributing and transferring the computed geometrical information from the surface of objects to the entire 3D space (e.g. into void space inside of the objects as well as the centroid of a 3D bounding box), we employ a *Monte Carlo ConvNet* (MCCNet) [26]. The MCCNet is a multi-layer network of which each layer consists of several MLP-based sub-networks. Each sub-network consists of a two hidden layers MLP whose functions resemble a set of convolution kernels. Each sub-network (called kernel) receives a set of features at the location of points in its spherical *receptive field* and computes an output feature vector at the center of its receptive field. Similar to standard convolution kernel, by positioning the kernel on any points in the point cloud the kernel maps the input feature set into that point. However, in contrast to a standard convolution, such a mapping is performed by applying the transfer function of an MLP-based kernel. In order elaborate in the motivation behind employing the MCCNet in the framework of our proposed Mask-MCNet, in the following, we will briefly explain the principle of Monte Carlo convolution in the work of Hermosilla et al. [26].

3.2.1. Monte Carlo Convolution Network (MCCNet)

Let's assume that the feature function f defined on the surface of an object which we have a set $\mathscr{S}$ of discrete samples $y_i \in \mathscr{S}$ (our data points). The f represents a mapping between the 3D position of each point and its representation in the embedded feature space ($f : \mathbb{R}^3 \mapsto \mathbb{R}^N$). In our particular case, $N = 256$ which is the dimension of the backbone features. By defining a convolution kernel g with a spherical receptive field (centered at 0 with radius equal to 1), the discreet convolution operator can be written as:

$$(f * g)(x) = \sum_{i \in \mathscr{N}(x)} f(y_i) \cdot g(x - y_i), \tag{1}$$

where $\mathscr{N}(x)$ is the set of neighborhood indices in the receptive field of g which is centered around point x. The convolution kernel g was suggested to be implemented by a 2-hidden layer MLP network [26]. Since the defined discrete convolution in Eq. (1) assumes that the sampled points are distributed uniformly on f, Hermosilla et al. [26] suggested to estimate and incorporate the probability density function (PDF) in the field of view of g to address a valid approximation of the convolution operator when the points are distributed non-uniformly on f. Therefore, Eq. (1) is modified to be written as:

$$(f * g)(x) \approx \frac{1}{|\mathscr{N}(x)|} \sum_{i \in \mathscr{N}(x)} \frac{f(y_i) \cdot g(\frac{x - y_i}{r})}{p(y_i|x)}, \tag{2}$$

where r is the distance of each point y_i in the receptive field of g from its center x. Here, $p(y_i|x)$ is the PDF at point y_i. Since the PDF is unknown, the author proposed using a kernel density estimation h for estimating the PDF at the location of y_i as below:

$$p(y_i|x) \approx \frac{1}{|\mathscr{N}(x)| \cdot \sigma^3} \sum_{k \in \mathscr{N}(x)} \left\{ \prod_{d=1}^{3} h\left(\frac{y_{i,d} - y_{k,d}}{\sigma}\right) \right\}, \tag{3}$$

where h is the *density estimation kernel*, a non-negative function of which the integral equals 1 (e.g. a Gaussian) and σ is its bandwidth which determining the smoothness of estimation (e.g. $\sigma = .25r$). It is worth mentioning that the derivatives of Eq. 2 with respect to the parameters of network g and employing the back-propagation technique for training the MCCNet are straightforward. For more information regarding the mechanism of MCCNet, we refer to the original paper [26].

As mentioned earlier, we aim to transfer (i.e. map) and then distribute the extracted backbone features at the location of samples in the set $\mathscr{S}$ into the entire 3D space at the location of nodes of a 3D grid (G). The domain G spans the whole 3D space and is bounded by the bounding box of the input scan. Fig. 2 illustrates such a mapping between $\mathscr{S}$ and G. For performing such a mapping, we proposed using a set of Monte Carlo convolution kernels because of two important properties: (1) its capability of computing the

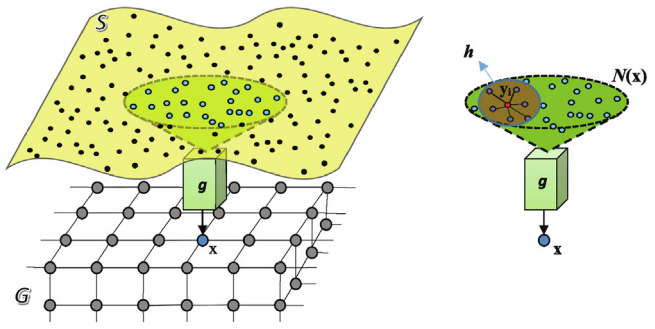


Fig. 2. (left) Mapping the backbone features from the location of set $\mathscr{S}$ (point cloud) into point x from a 3D grid domain G by a learning kernel g; (right) estimating the PDF at each point y_i by kernel density estimation h (see Eq. (3)). The spherical receptive field of g is shown in a green color. In an ablation test, the Monte Carlo convolution is replaced with a max-pooling operator (i.e. the mapped feature vector on point x is $f_x = max(y_i)$ where. $i \in N(x)$).

convolution on an arbitrary output point (x) within the kernel's receptive field, regardless of its presence within the set of input points ($\mathscr{S}$); (2) its capability of handling the non-uniform distribution of points when computing the mapping. The first property makes it possible to transfer the computed backbone features on an arbitrary new domain such as the node of the regular 3D grid G, while the second property facilitates processing of a non-uniform distribution of points on the surface of object.

By aggregating the features of surface points on a regular 3D grid, only the nodes close to the surface will have a valid feature vector. The nodes near the object center are likely to have no feature vector at all (empty space). As a result, aggregating the shape context in the vicinity of object centers creates difficulties. Simply increasing the receptive field does not solve the problem because as the network captures a larger context, it also causes more inclusion of nearby objects and clutter. Hence,after mapping the features into domain G through the set of convolutional kernels (g's) in the first layer of MCCNet, the geometrical information is passed on and further processed in the deeper layer of MCCNet and is distributed to the neighbour nodes in domain G based on the predefined field of view of kernels in the hidden layers of MCCNet. Applying the MCCNet on the output of the backbone (e.g. $n \times 256$ matrix), a high-dimensional feature vector (e.g. 256-dimensional vector) at the location of each node of the 3D grid domain is computed. By assuming m nodes for domain G, the network returns in total a feature matrix of size $m \times 256$.

3.2.2. Object proposal (anchor) and Triangular Interpolation

To generate object proposals (i.e. 3D cubes encompassing teeth), we follow the idea of using *anchors* which is adopted from *Faster-RCNN* [27], but modified to a 3D space. Here, each 3D anchor is indicated by a cube, which is represented with its central position $[x_c, y_c, z_c]$ and its size $[w, d, h]$. The orientation of 3D boxes is ignored in our modeling approach. This is because some of the tooth crowns (e.g. premolars teeth) have a symmetric and semi-cylindrical shape. Considering the orientation for fitting a 3D box imposes a high degree of uncertainty to the learning module and contributes little towards point segmentation, which is the main goal of processing an intra-oral scan. Fig. 3 visualizes examples of such anchors with different sizes and aspect ratios that are localized at different positions in 3D space. Making no assumptions regarding the possible positions of objects (which leads to a more generic approach), the centers of the anchors should be located on the nodes of a regular 3D grid which is domain G of which the MCCNet computes the geometrical information at its node. We will discuss the spatial resolution of grid G in a later section.

Since the 3D tooth size varies along the dental arch and between different subjects, we use multi-sized ($k > 1$) anchors which are centered around each node of G. Since each node on G indicates the center of one ($k = 1$) or multi-sized ($k > 1$) anchor (s), the total amount of anchors is $k \times m$. For the teeth segmentation problem, by considering various sizes of teeth (incisor and molar teeth), we position 4 different anchors (2 different sizes and 2 different aspect ratios).

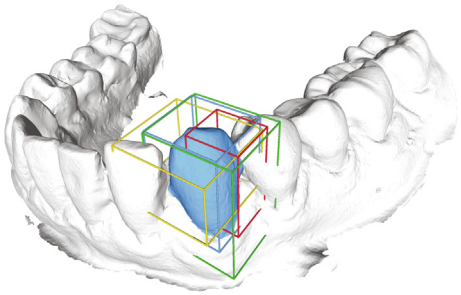


Fig. 3. Example of typical 3D anchors that are used for tooth localization. The ground truth (colored in blue) has 3D IoU scores equal to 0.35, 0.42 and 0.60 with red, yellow and green anchors, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Generally the idea of using anchors is to change the problem of object localization into the anchor classification. To do so, the anchor which has a high overlap with an object should be classified as positive class, otherwise the anchor should be classified as negative class. Anchor classification and, later on, fine-tuning the position and size of positive anchors to precisely fit the 3D bounding box of the object, requires a fix-length feature vector to be extracted from variable-sized anchors. Here, we use an idea similar to *ROI alignment*, which was introduced in Mask R-CNN [22] but in a 3D space of point cloud data. We re-sample a fixed number of points inside each anchor by applying *Triangular interpolation* by finding a set of three nearest neighbour nodes of the G and weighting their feature vectors based on their distance to the new node in 3D space. Fig. 4 shows an example of performing ROI alignment for an anchor and the triangular interpolation. In our experiment, we use a 3D grid of $s \times s \times s$ nodes (e.g. $s = 5$) for interpolating the features inside each anchor.

3.3. Predictor networks

The predictor networks consist of three parallel branches for *detection*, *localization* by fine-tuning, and *mask generation*. The detection and localization branches both consist of a fully-connected MLP network that receives the interpolated feature set inside each anchor and performs a regression task. The detection branch aims to predict the IoU score of each anchor (in the unity interval), which indicates their maximum overlap with any object. Later, by applying a threshold on the predicted IoU scores, the anchors are classified into positive and negative classes. The assigned class indicates if an anchor acceptably encompasses an object instance or not (in total, $k \times m$ anchors). Fig. 5 shows two examples of input 3D patches and their overlaid positive detected anchors.

The localization branch aims to predict the adjustment values of the location and size of each positive anchor to fit tightly to the true bounding box of the corresponding object. Since the 3D position of each anchor has been encoded by its centroid and size, the localization branch aims to predict 6 values for each positive anchor at its output. Instead of estimating the absolute values of such parameters directly, the network only predicts the difference (i.e. residual vectors) between the center and size of the anchor with the center and size of the corresponding object. Learning such

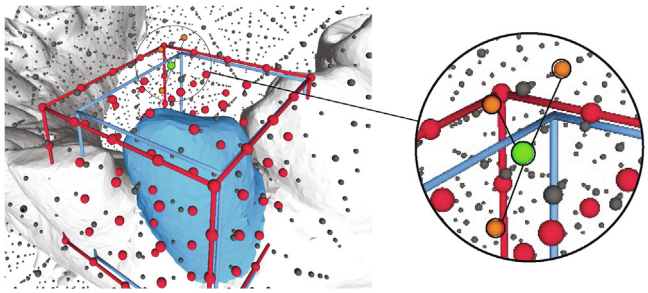


Fig. 4. Example of 3D ROI alignment. By applying the triangular interpolation in domain G (gray nodes), a set of features is computed at the locations of red nodes (with 5^3 nodes) on a grid inside the red anchor. Hence, a fix-length feature set is assigned to the anchor. Here, the blue box is the ground truth and the orange nodes are examples of using three nearest neighbor nodes of G to the green node for interpolating the feature vector on the green node. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

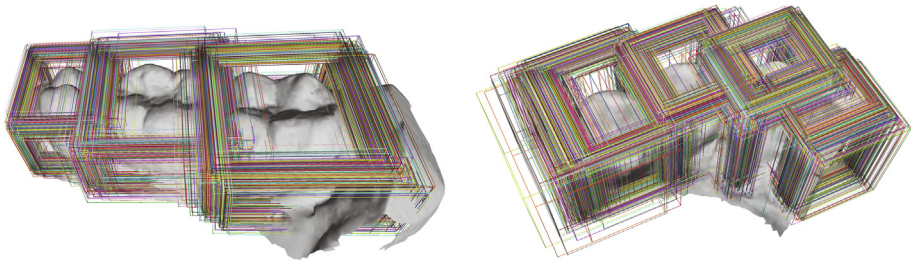


Fig. 5. Examples of positive detections for two 3D patches.

residual vectors in the form of difference values is easier and imposes less complexity to the learning model because its computation is performed locally in a canonical coordinate system. Since the detection and localization branches work in parallel, the feature matrix which is supplied to these two branches has a size of $k \times m \times s^3$ elements. As mentioned earlier, in the training phase, an anchor is labeled positive if it has a high IoU score with any single tooth instance above a threshold (e.g. 0.4) and it is labeled negative if the IoU to be lower than a certain threshold (e.g. 0.2). Since the number of positive and negative anchors are highly imced, about 50% of each training batch is selected from the positive and 50% from the negative anchors. The marginal anchors ($0.2 < IoU < 0.4$) are not utilized when training the model.

The mask-generation branch aims to segment the points that are located inside the 3D bounding box of each positive anchor. The outcome of the mask generator has the form of a binary mask that indicates whether each point inside the bounding box belongs to the corresponding tooth instance or not. To perform such a binary classification, the points inside a positive anchor along with their features from the backbone network are supplied to the mask-generation branch. Since the number of points in such a classification is highly imbalanced, we trained the mask generation branch on points sampled equally from both classes. In our architecture, the mask generator has similar computational layers to the backbone architecture, but it consists of only three layers. Fig. 6 shows the details of Mask-MCNet architecture.

3.4. Loss function

Mask-MCNet performs a multi-task learning, including the estimation of 3D bounding-box overlap (i.e. IoU score), center and size offset estimation, and mask generation. To perform these, the loss function of the model consists of an equal contribution of three terms. The first term ($\mathscr{L}_{det}$) is a mean-squared error for estimating the IoU scores of each anchor at the output of the detection branch. The second term ($\mathscr{L}_{loc}$) is also a mean-squared error at the linear output layer of the localization branch. Finally, the third term ($\mathscr{L}_{mask}$) is a binary cross-entropy loss for classification of all points in each positive anchor at the output softmax layer of the mask branch. The localization and mask losses are involved only if the examined anchor is labeled positive. Thus, the total loss function can be written as:

$$\mathscr{L}_{total} = \mathscr{L}_{det}^{\{p,n\}} + \mathscr{L}_{loc}^{\{p\}} + \mathscr{L}_{mask}^{\{p\}}, \tag{4}$$

where the superscript $^{\{p,n\}}$ indicates that the term is calculated for both positive and negative anchors in the training batch.

3.5. Implementation details

3.5.1. Training

of the entire Mask-MCNet model is done end-to-end by using gradient descent optimization and the Adam learning adaptation technique for 1,000 epochs with a batch size of 32 (equally balanced between positive and negative anchors). The preprocessing of the input intra-oral scan only consists of normalizing the whole point cloud to have zero mean and unit variance. The input to the Mask-MCNet is a randomly cropped patch of the point cloud, which usually contains 2–4 tooth instances. As explained, the centers of the anchors are positioned on a regular 3D grid with spatial resolution of 0.03 in a normalized coordinate system. As to impose sufficient overlap between anchors and both small and large objects (e.g. incisor and molar teeth, respectively), two types ($k = 2$) of anchors are employed (with a size of [0.3, 0.3, 0.2] and [0.15, 0.15, 0.2]). For computational efficiency, reducing the number of anchors is desirable. Such a reduction can be considered in two ways. Firstly, by choosing a minimal number of types of 3D anchors that differ by their aspect ratios. This minimal number of types depends on the variation of object (tooth) sizes. Hence, the box sizes are adapted to the tooth instance sizes. Secondly, by reducing the number of nodes which are the central position of the anchors, the total number of anchors required to be examined by the model for the presence of a tooth, are reduced.

For the sake of obtaining a high recall in tooth detection, the resolution of grid G cannot be reduced too much. Instead, we can remove the nodes that are very unlikely to be close to the center of a tooth. To do so, we remove nodes of the grid, based on their distance to the closest point in the point cloud. Hence, the nodes and consequently the anchors that have a distance higher than a certain threshold are suppressed. Such a suppression mainly removes the points in void space, close to the center of the dental arch and decreases the total computational time of the model.

3.5.2. Inference

When applying the model on all patches (about 5 patches per jaw) that are extracted from the input point cloud and aggregating the result, we can perform inference on the whole point cloud in its original resolution. Since for each tooth, several overlapping detections are obtained, we need to aggregate the detections to generate exactly one detection per tooth. Furthermore, handling the overlapping patches also requires using such a aggregation step. Since each detection has a predicted IoU score, we can use a non-maximum suppression algorithm, which is a common practice in object detector models. However, to compensate for the effect of false positive detection, instead of using a simple non-maximum suppression, we employ the DBSCAN clustering algorithm to group the detected 3D bounding boxes, based on their centroid position and their size. Such a clustering is preferred because it is able to detect outliers (i.e. false positive detections). Fig. 7 shows an example of a clustering result for a test intra-oral scan.

Recursive regression: As explained earlier, the regression branch aims to predict the offset values of centroid and size of each positive anchor. Since such an inference is based on given interpolated features inside each anchor, the predicted values for the anchors that are not highly overlapping with a tooth are more prone to errors. For compensating the source of such an error and improving

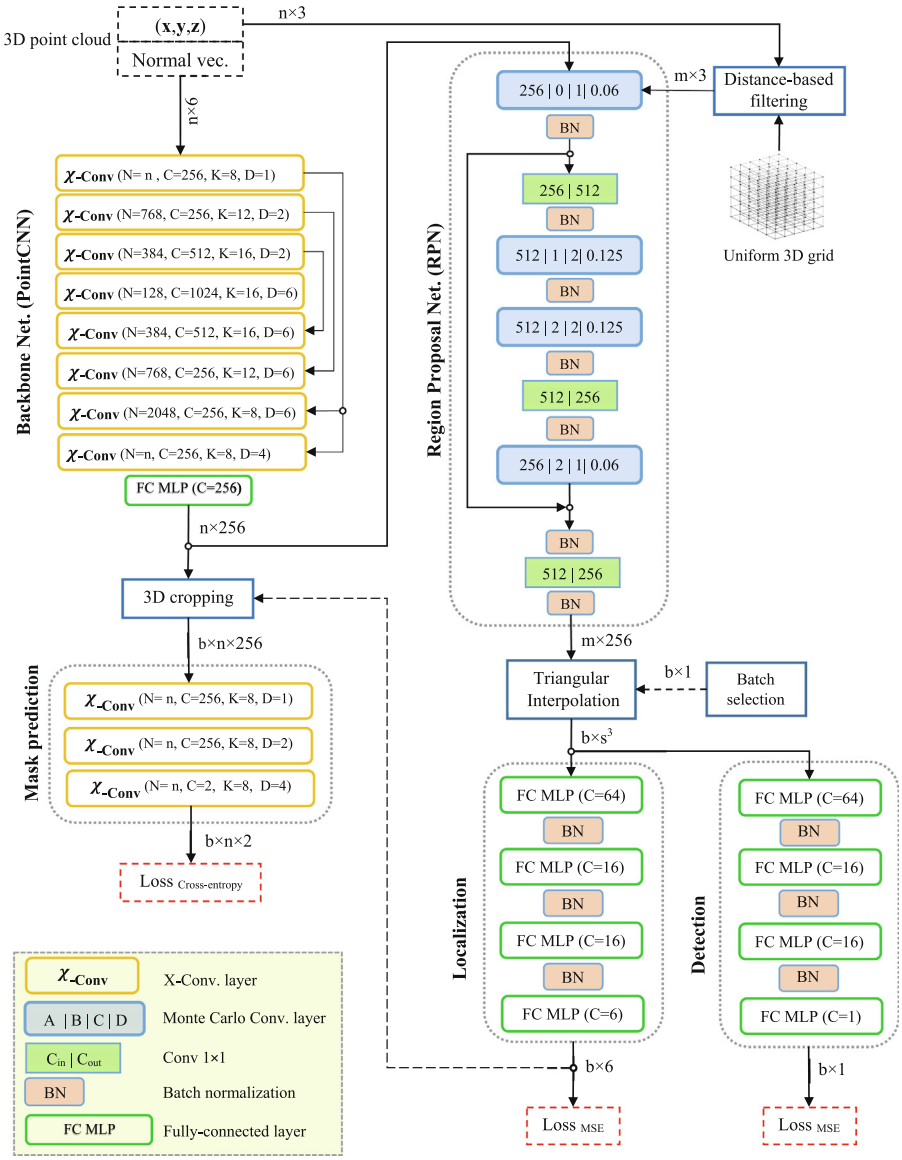


Fig. 6. Mask-MCNet architecture (PointCNN as backbone).

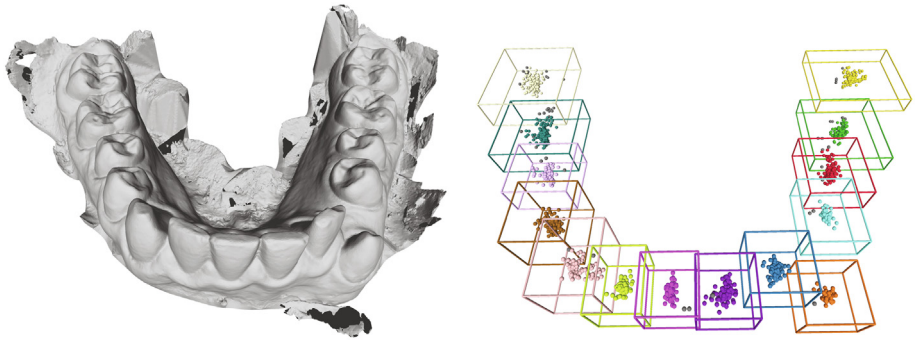


Fig. 7. Clustering of all positive anchors (detections) for assigning only one 3D bounding box to each tooth instance.

the predicted offset values, in inference mode, we employ a simple *recursive* scheme that applies multiple executions of the regressor on the relocated anchor, according to the last predicted values. Such a recursive scheme works as a *negative feedback* in the model. Applying this mechanism, the predicted offset values are used for relocating the bounding box of the anchor and then by re-applying the triangular interpolation and re-estimating the feature set at the location of the updated point set (with s^3 points), the model predicts the offset values again. Since in the following iteration, the updated bounding box is more likely closer to the

bounding box of a tooth, the predicted values will become more precise. Fig. 8 visualizes the predicted offset values for the positive anchors for two iterations. As can be observed, in the second iteration, the replacement vectors are more concentrated on the centroid of each tooth instance in the input patch. Fig. 9 shows the changes of average estimation error (Euclidean distance) of anchor centroids across several iterations at inference time. The values are measured for the scans, normalized with zero mean and unit standard deviation. In our experiments, the regression branch was executed for two iterations that slightly improved the 3D box detection.

4. Experiments and results

4.1. Data

In this study, we used two datasets which have been collected from two different types of scanners. The first dataset called *Dataset I*, is used for both training and testing the models by using the cross-validation technique. The second dataset called *Dataset II*, is only used for evaluating the robustness of Mask-MCNet across different scanner types.

4.1.1. Dataset I

This dataset consists of 120 optical scans of dentitions from 60 adult subjects, each containing one upper and one lower jaw scan. The optical scans were recorded from *dental impressions* by a 3Shape D500 optical desktop scanner (3Shape AS, Copenhagen, Denmark), which uses stereo-vision cameras and three free-axes motion system for 3D reconstruction. The scanner has high spatial accuracy with a tolerance smaller than 20 μm and obtains about 180 k points per scanned jaw on average (varying in a range interval of [100 k, 310 k]). The dataset includes both healthy dentitions and a variety of abnormalities in dentition among subjects.

4.1.2. Dataset II

This dataset consists of 48 optical scans of 24 adult subjects. The scans are captured by a 3Shape Trios Move intra-oral scanner that uses confocal laser scanning microscopy and structured light projection. As mentioned earlier, this dataset is only used for evaluating the trained model across different scanner types.

All optical scans were manually annotated by using Meshmixer 3.4 (Autodesk Inc, San Rafael CA, USA) and their respective points were categorized, according to the FDI standard into one of the 32 classes by a dental professional and reviewed and adjusted by one dental expert. Annotation of each optical scan took 45 min on average, which shows that it forms an intensive laborious task for a human.

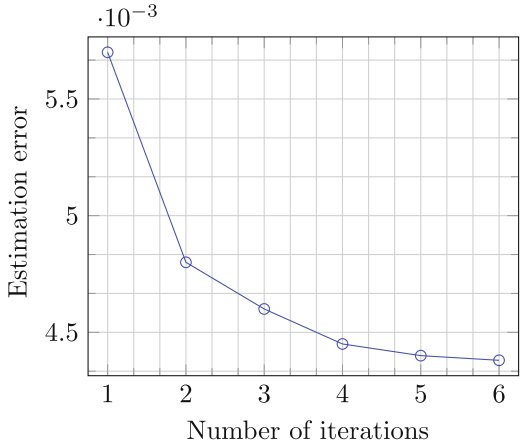


Fig. 9. Changes of average estimation error (Euclidean distance) of the anchor centroids using an iterative regression.

4.2. Experimental setup

The performance of the Mask-MCNet in comparison with state-of-the-art systems is evaluated by fivefold cross-validation. The average Jaccard Index (also known as mIoU) of all teeth instances is measured. On top of the measured mIoU, by treating each class individually as a binary (one-versus-all) segmentation problem and then by averaging on all measured precision and recall scores, we report the mean average precision (mAP) and mean average recall (mAR) for evaluating the multi-class teeth segmentation performance.

Although the instance segmentation performance is coupled with the result of 3D bounding box detection in the proposed Mask-MCNet model, for evaluating the intermediate steps, we also report the performance of 3D bounding box detection. To do so, as is common in object detection problems, we measure the average precision value for recall value of 0 and 1 when the predicted bounding box overlaps with the ground truth by applying a threshold of 0.25 (which was proposed in [14]) and 0.5 on 3D IoU.

4.3. Main results

We evaluated the proposed Mask-MCNet in comparison with the competitive models for 3D point cloud semantic segmentation on Dataset I. The performance of each method was tested by fivefold cross validation. The obtained results for instance tooth segmentation are shown in Table 1. The results show that the Mask-MCNet outperforms state-of-the-art models by achieving 0.98 mIoU on the instance tooth segmentation. Fig. 10 visualizes the

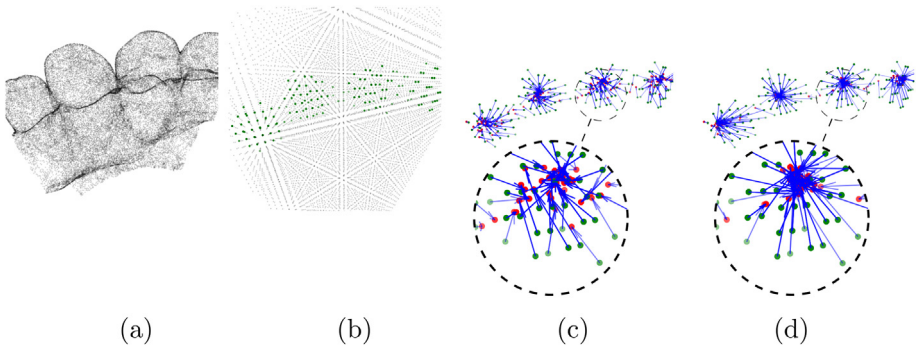


Fig. 8. Iterative regression of anchors centroids; (a) input 3D patch; (b) 3D grid domain (G) and positive detected centroids; (c) predicted displacement vector of positive anchors; (d) predicted displacement after 2nd iteration.

Table 1
Results of tooth segmentation on Dataset I by the proposed Mask-MCNet in comparison with state-of-the-art deep learning models. The mean IoU (mIoU), mean average precision (mAP), mean average recall (mAR), and the execution time are reported.

Method	Defined task		Metric			Exec.time (sec.)*
	Semantic Seg.	Instance Seg.	mIoU	mAP	mAR	
PointNet [25]	✓	–	0.76	0.73	0.65	**0.19**
PointGrid [28]	✓	–	0.80	0.75	0.70	0.88
MCCNet [26]	✓	–	0.89	0.88	0.84	1.01
PointCNN [24]	✓	–	0.88	0.87	0.83	0.66
PointCNN++ [2]	✓	–	0.94	0.93	0.90	6.86
MASC [18]	–	✓	0.93	0.92	0.89	1.31
VoxelNet [19]	–	✓	0.94	0.94	.91	0.54
PointRCNN [21]	–	✓	0.97	0.96	0.97	0.23
Mask-MCNet (ours)	-	✓	**0.98**	**0.98**	**0.97**	14.6

* NVIDIA Titan-X GPU

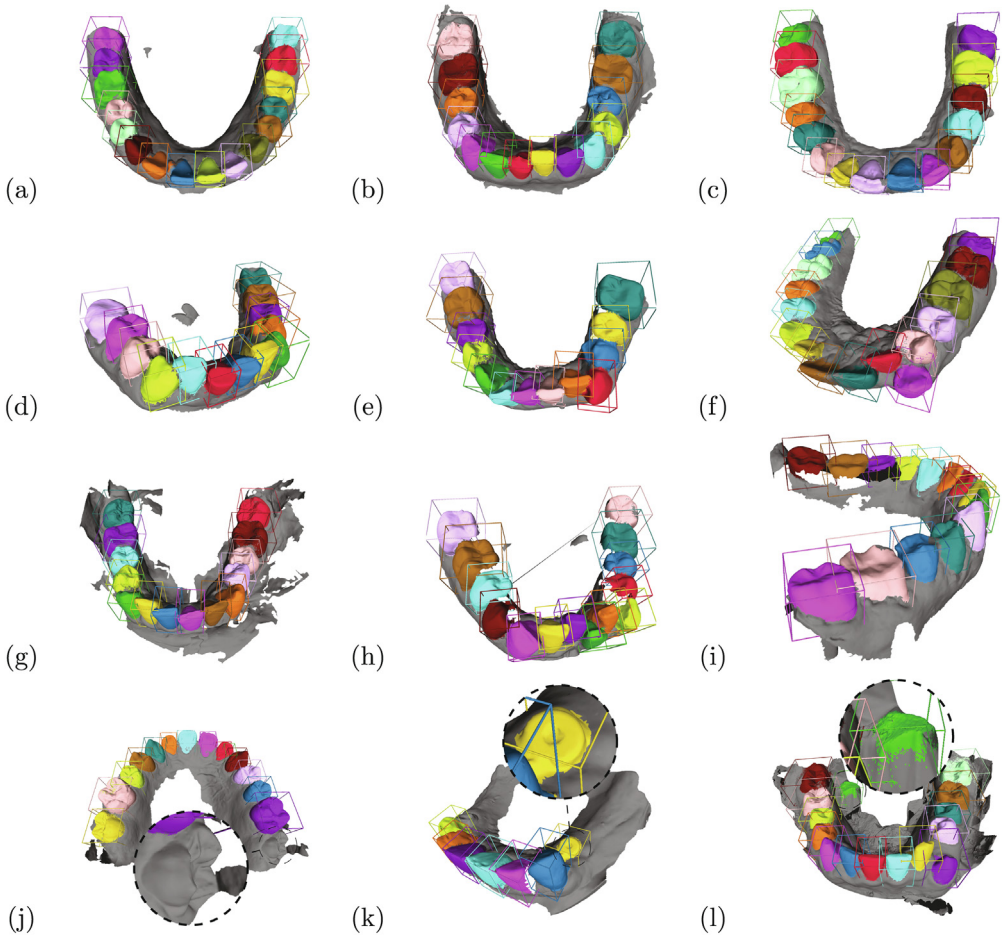


Fig. 10. Examples of tooth instance segmentation on test data by Mask-MCNet; (a–c) normal dentition (d–f) subjects with different abnormality in dentition; (g) artifacts in scanning (h–i) typical missing data; (j–l) failure cases.

segmentation result for a number of test scans with variable abnormality in dentition and in the scanning artifacts.

The execution time of Mask-MCNet is relatively high because of two reasons. Firstly, in contrast with compared semantic segmentation models, the Mask-MCNet is able to process the input scans at their original spatial resolution by using a patch processing technique. This results in a dense prediction without employing a down-sampling at the expense of a longer computational time. Secondly, in our implementation, executing the triangular interpolation on a multi-threading CPU slows the inference. However, the computation time of Mask-MCNet is still a small fraction of the time needed by a human expert for annotating of an intra-oral scan.

For evaluating the robustness of Mask-MCNet across different scanner types, the trained model on Dataset I is tested on Dataset II. The results in Table 2 show only a 1% drop in the mIoU score which indicates acceptable robustness of the proposed model across different scanner types.

4.4. Ablation experiments

In ablation experiments, we evaluate multiple basic instantiations, which allow us to demonstrate the robustness of the model and analyze the effects of core factors in the proposed Mask-MCNet. We have examined the performance of Mask-MCNet across different backbone architectures, different surface normal-vector

Table 2
Results of tooth instance detection and segmentation on the Dataset II, in comparison to state-of-the-art.

Model	Mask			Bounding Box		
	mIoU	mAP	mAR	mIoU	$mAP_{.25}$	$mAP_{.5}$
MASC [18]	0.90	0.89	0.88	0.48	0.92	0.78
VoxelNet [19]	0.92	0.92	0.88	0.50	0.90	0.80
PointRCNN [21]	0.95	0.95	0.92	0.52	0.97	0.83
Mask-MCNet	0.97	0.96	0.94	0.53	0.99	0.84

estimation, coupling mechanisms between backbone and Monte Carlo network, and the granularity (i.e. spatial resolution) of the grid domain, where the anchors are distributed.

4.4.1. Backbone architecture

As mentioned earlier, we have chosen PointCNN as the backbone network in the framework of the proposed Mask-MCNet because of its lower number of training parameters, compared with other deep networks for point cloud analysis. Here, by replacing the PointCNN with PointNet [25] as an alternative choice for the backbone, the performance of Mask-MCNet in both tooth instance segmentation and 3D bounding-box detection are measured. Table 3 shows the performance of Mask-MCNet by employing each of these two backbones. The results show that the extracted features from the PointCNN lead to a slightly higher accuracy in both segmentation and detection tasks. This observation is in agreement with what we expected because the PointCNN extracts a richer set of geometrical features by incorporating KNN points in its representation at $\chi - Conv$ layers [24].

4.4.2. Data augmentation with local surface geometry

A point cloud does not explicitly convey the information from neighbouring points. Therefore, in order to aggregate over local neighborhoods, most existing methods augment the input point cloud with the surface normal-vectors or resort to a neighbor searching mechanism [23] or ball query [24]. Since some scanner's software additionally exports the surface mesh data, computing the normal vectors per point (mesh vertices) is trivial. In this case, a normal vector per query point can be simply computed by averaging over all normal vectors of faces connecting to the point. However, this approach is prone to noise. An alternative approach, is to use an approximation method such as an analysis of the local covariance matrix, which is computed from the neighbors of the query point. To compute the local covariance matrix, we used all points within a fixed distance from the center of a sphere on the query point in the 3D space. This approach is less vulnerable to the number of points (resolution) than using KNN. The computed local covariance matrix of size 3-by-3 can be either vectorized to a size of 1-by-9 and directly used as an attribute per point [29] or can be analyzed by eigenvector analysis (e.g. PCA) [30]. Table 4 shows the impact of using the local surface geometry on teeth instance detection and segmentation. The results show that adding neighbour point information can slightly improve the results. Since the employed PointCNN as the backbone, internally computes the features on K nearest neighbour points, it may make such an augmentation less effective.

4.4.3. Coupling mechanism

The mechanism of *coupling* the backbone with the Monte Carlo network indicates the way that the extracted backbone features at the location of points in the input point cloud are transferred to the nodes of the grid domain G. Here, we assume two different coupling mechanisms. In the first approach, as explained earlier, each MLP-based kernel in the first layer of the MCCNet, maps the backbone feature vectors of all points within its receptive field into its center, where the node of G has been located. Thus, transferring the geometrical information between these two domains is performed by a set of learning convolutional kernels. In an alternative approach, simply a max-pooling operator can aggregate (i.e. pool) the backbone feature vectors in a predefined spherical receptive field and assign a single feature vector to a node of G, where it has been located at the center of its receptive field. For comparing the performance of these two mechanisms, we kept the radii of their receptive fields identical. Table 5 shows the performance of Mask-MCNet in tooth instance segmentation and detection with these two coupling mechanisms. The results show that employing the learning kernel has slightly higher performance than max-pooling for transferring the information between these two spatial domains.

4.4.4. Granularity of anchor domain

As mentioned earlier, the nodes of grid domain G indicate the central position of one ($k = 1$) or multiple anchors ($k > 1$). The spatial resolution of G affects the performance of tooth detection and

Table 3
Ablation test results on choice of backbone architecture. Mask-level and box-level AP for two models are reported.

Backbone Arch.	Mask			Bounding Box		
	mIoU	mAP	mAR	mIoU	$mAP_{.25}$	$mAP_{.5}$
PointCNN [24]	**0.98**	**0.98**	**0.97**	0.64	1.0	0.94
PointNet [25]	0.97	0.97	0.94	0.65	1.0	0.72

Table 4
Ablation test with/without data augmentation by using local surface geometry. The normal vectors are computed based on mesh data or PlanePCA method. Alternatively, the local covariance matrix can be vectorized and used as an attribute for each input point.

Data augmentation	Mask			Bounding Box		
by local surface geometry	mIoU	mAP	mAR	mIoU	$mAP_{.25}$	$mAP_{.5}$
–	0.97	0.97	0.96	0.62	1.0	0.93
Mesh surface normal	0.98	0.98	0.97	0.64	1.0	0.94
PlanePCA normal [30]	0.98	0.97	0.97	0.64	1.0	0.94
Local covariance [29]	**0.98**	**0.98**	**0.97**	**0.65**	1.0	**0.95**

Table 5
Ablation test results on type of coupling backbone and Monte Carlo networks. The backbone is PointCNN.

Coupling	Mask			Bounding Box		
	mIoU	mAP	mAR	mIoU	$mAP_{.25}$	$mAP_{.5}$
Max-pooling	0.97	0.96	0.96	0.64	0.99	0.94
MLP Conv. filter	**0.98**	**0.98**	**0.97**	0.71	1.0	0.95

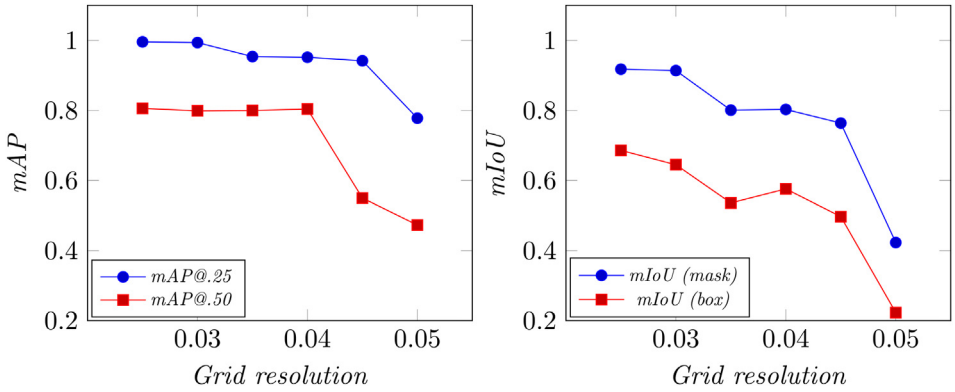


Fig. 11. Impact of granularity of grid domain G on the tooth instance detection and segmentation tasks. The Y-axis denotes the spatial resolution of G in the normalized 3D space (where the data has a mean of zero and a standard deviation equal to unity); (left) Performance of Mask-MCNet on 3D bounding box detection at 3D IoU thresholds equal to 0.25 (mAP@.25) and 0.5 (mAP@.25); (right) performance of the model for tooth instance segmentation (blue) and localization (red).

localization. On one hand, employing a dense high-resolution grid helps for posing a sufficient number of anchors around each tooth (especially around small teeth such as incisors). This increases the chance of positioning a set of highly overlapped anchors with each tooth and consequently detecting several positive anchors for each tooth instance. On the other hand, employing a low-resolution hence sparser grid is more likely to miss some of the teeth, because the number of detected positive anchors may not be sufficient enough to form a cluster at a later stage. However, employing a very high-resolution grid requires the processing of more anchors and hence a higher computation time. Fig. 11 shows the performance of the model, both in detection and segmentation of tooth instances against changing the granularity of the G domain. The left plot in Fig. 11 illustrates that the rate of tooth instance detection (vertical axis) drops when decreasing the spatial resolution of G (horizontal axis). Since the 3D coordinates of points in the input point cloud are normalized to have a standard deviation equal to unity, the unit of spatial resolution is reported here by its normalized value. Fig. 11 (right) plots the mIoU scores as a measure of tooth instance segmentation performance (i.e. mask generation). It also plots the mIoU scores between the 3D bounding boxes of the ground truth and the prediction. Because of a trade-off between the granularity of G and the computational cost of the model, in all our experiments, we adjust the grid resolution to be equal to 0.03 (normalized value).

5. Discussions and conclusions

In this study, we have presented a new end-to-end learning framework, called Mask-MCNet, for tooth instance segmentation in a 3D point cloud of intra-oral scan data. Accurate tooth instance segmentation is an important step towards automated computational dentistry with many clinical applications in implantology and orthodontics.

In contrast to alternative deep learning models, the proposed Mask-MCNet does not employ a voxelization or down-sampling step for processing the input 3D point cloud. Instead, by first localizing the 3D bounding box of teeth in extracted patches and then segmenting the points that belong to each tooth instance, the model is able to process a large point cloud in patches. Hence, a large point cloud is processed at its native high resolution, thereby preserving the finely detailed geometrical information, which is crucial for accurate teeth segmentation.

In the architecture of Mask-MCNet, the Monte Carlo ConvNet transfers the information from the point cloud where it is spread over the surface of objects into the entire 3D space (e.g. the void space inside of objects). This property facilitates the inference on the center and size of objects (i.e. teeth). Furthermore, the employed Monte Carlo ConvNet in the Mask-MCNet can handle processing of non-uniformly distributed samples. This feature leads to an efficient search of object proposals which is important for scalability of the method, such that it is applicable for processing intra-oral scan data with large point clouds (more than 180 k points).

Our experiments have shown that the proposed model achieves a 98% mIoU on the test data, thereby outperforming the state-of-the-art networks in tooth instance segmentation. This level of performance is close to the human level and obtained in only a few seconds of processing time, whereas for a human it would form a lengthy and labour intensive task.

CRediT authorship contribution statement

Farhad Ghazvinian Zanjani: Conceptualization, Methodology, Software, Validation, Writing - original draft, Visualization. **Arash Pourtaherian:** Software, Validation, Formal analysis, Investigation. **Svitlana Zinger:** Supervision, Project administration, Resources. **David Anssari Moin:** Funding acquisition, Project administration, Resources, Investigation. **Frank Claessen:** Visualization, Formal analysis, Software. **Teo Cherici:** Visualization, Formal analysis, Software. **Sarah Parinussa:** Software, Data curation, Resources. **Peter H.N. de With:** Supervision, Funding acquisition, Writing - review & editing, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

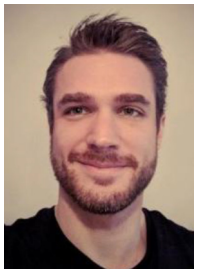
[1] P. Medina-Sotomayor, A. Pascual-Moscardó, I. Camps, Relationship between resolution and accuracy of four intraoral scanners in complete-arch impressions, J. Clin. Exp. Dentistry 10 (4) (2018) e361.
[2] F. Ghazvinian Zanjani, D. Anssari Moin, B. Verheij, F. Claessen, T. Cherici, T. Tan, P. de With, Deep learning approach to semantic segmentation in 3D point cloud intra-oral scans of teeth, in: Proc. 2nd Int. Conf. Medical Imaging with Deep Learning (MIDL), Vol. 102, PMLR, 2019, pp. 557–571..
[3] T. Kondo, S. Ong, K.W. Foong, Tooth segmentation of dental study models using range images, IEEE Trans. Med. Imaging 23 (3) (2004) 350–362.
[4] N. Wongwaen, C. Sinthanayothin, Computerized algorithm for 3D teeth segmentation, in: International Conference on Electronics and Information Engineering (ICEIE), IEEE, 2010, pp. 277–280.
[5] T. Yuan, W. Liao, N. Dai, X. Cheng, Q. Yu, Single-tooth modeling for 3D dental model, J. Biomed. Imaging 2010 (2010) 9.
[6] Y. Kumar, R. Janardan, B. Larson, J. Moon, Improved segmentation of teeth in dental models, Computer-Aided Des. Appl. 8 (2) (2011) 211–224.
[7] M. Yaqi, L. Zhongke, Computer aided orthodontics treatment by virtual segmentation and adjustment, in, in: International Conference on Image Analysis and Signal Processing (IASP), IEEE, 2010, pp. 336–339.
[8] S.M. Yamany, A.M. El-Bialy, Efficient free-form surface representation with application in orthodontics, in: Three-Dimensional Image Capture and Applications II, Vol. 3640, International Society for Optics and Photonics, 1999, pp. 115–125..
[9] M. Zhao, L. Ma, W. Tan, D. Nie, Interactive tooth segmentation of dental models, in, in: 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS), IEEE, 2006, pp. 654–657.
[10] Z. Li, X. Ning, Z. Wang, A fast segmentation method for stl teeth model, in: International Conference on Complex Medical Engineering (ICME), IEEE, 2007, pp. 163–166.
[11] M. Grzegorzek, M. Trierscheid, D. Papoutsis, D. Paulus, A multi-stage approach for 3D teeth segmentation from dentition surfaces, in: International Conference on Image and Signal Processing, Springer, 2010, pp. 521–530.
[12] T. Kronfeld, D. Brunner, G. Brunnett, Snake-based segmentation of teeth from virtual dental casts, Computer-Aided Design Appl. 7 (2) (2010) 221–233.
[13] B. Zou, S. Liu, S. Liao, X. Ding, Y. Liang, Interactive tooth partition of dental mesh base on tooth-target harmonic field, Comput. Biol. Med. 56 (2015) 132–144.
[14] C.R. Qi, W. Liu, C. Wu, H. Su, L.J. Guibas, Frustum pointnets for 3D object detection from RGB-D data, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 918–927..
[15] J. Hou, A. Dai, M. Nießner, 3D-SIS: 3D semantic instance segmentation of RGB-D scans, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4421–4430..
[16] L. Yi, W. Zhao, H. Wang, M. Sung, L. Guibas, GSPN: Generative shape proposal network for 3D instance segmentation in point cloud, arXiv preprint arXiv:1812.03320..
[17] W. Wang, R. Yu, Q. Huang, U. Neumann, SGPN: Similarity group proposal network for 3D point cloud instance segmentation, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2569–2578..
[18] C. Liu, Y. Furukawa, MASC: Multi-scale affinity with sparse convolution for 3D instance segmentation, arXiv preprint:1902.04478..
[19] Y. Zhou, O. Tuzel, Voxelnet: End-to-end learning for point cloud based 3d object detection, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4490–4499.
[20] C.R. Qi, O. Litany, K. He, L.J. Guibas, Deep hough voting for 3d object detection in point clouds, in, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9277–9286.
[21] S. Shi, X. Wang, H. Li, Pointrcnn: 3d object proposal generation and detection from point cloud, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 770–779.
[22] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proc. IEEE Int. Conf. Computer Vision, 2017, pp. 2961–2969..
[23] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Advances in neural information processing systems, 2017, pp. 5099–5108..
[24] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, PointCNN: Convolution on X-transformed points, in: Advances in Neural Information Processing Systems (NIPS), 2018, pp. 820–830.
[25] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: Proc. IEEE Computer Vision and Pattern Recognition (CVPR) 1 (2) (2017) 4..
[26] P. Hermosilla, P.-P. Ritschel, Tobias and Vázquez, À. Vinacua, T. Ropinski, Monte Carlo convolution for learning on non-uniformly sampled point clouds, in: SIGGRAPH Asia 2018 Technical Papers, ACM, 2018, p. 235..
[27] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems (NIPS), 2015, pp. 91–99..
[28] T. Le, Y. Duan, PointGrid: A deep network for 3D shape understanding, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 9204–9214..
[29] Y. Yang, C. Feng, Y. Shen, D. Tian, Foldingnet: Point cloud auto-encoder via deep grid deformation, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 206–215.
[30] K. Jordan, P. Mordohai, A quantitative evaluation of surface normal estimation in point clouds, in: International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 4220–4226.

Farhad Ghazvinian Zanjani received his MSc. in Biomedical Eng. from Amirkabir University of Technology in Tehran. He also received a Master degree (with honor) in Machine Learning from Radboud University of Nijmegen, The Netherlands. Currently, he is doing a PhD in Artificial Intelligence at Eindhoven University of Technology, working on deep learning models for medical image analysis.

Arash Pourtaherian received his PhD in 2018 on medical image analysis from the Electrical Engineering faculty at Eindhoven University of Technology. Currently, he is a postdoc fellow researcher working on deep learning models for 3D ultrasound data analysis.

Svitlana Zinger received her PhD degree in 2004 on interpolation and re-sampling of 3D data from the Ecole Nat. Superieure Telecommunications, Paris, France. In 2006-2008, she was associated researcher at the AI Dept. of the University of Groningen, the Netherlands. She is currently assistant professor at the Video Coding & Architectures Research group at Eindhoven University of Technology, the Netherlands. Research interests of dr. Zinger concern image analysis for computer-aided diagnosis and prognosis.

David Anssari Moin obtained a PhD in 3D printing of patient specific implants from the Vrije Universiteit Amsterdam, The Netherlands. He is also a dentist and a frequent speaker on international conferences in dentistry. He has a large international collaboration and partner network in the dentistry. He is the founder of Dental Centre Altman and Promaton Ltd., involved in both research and clinical healthcare in dentistry.



Frank Claessen obtained his master's degree in Design for Interaction from the Delft University in 2010. Since march of 2017 he has been the data scientist at Promaton Ltd., specifically focusing on the application of state-of-the-art machine learning technique in the field of dental software.

Teo Cherici obtained his master degree in Mechanical Engineering at the Delft University of Technology, working on robotics and deep learning solutions with Deep Reinforcement Learning. Since 2017, he is with Promaton Ltd., working on innovative AI dental software solutions.

Sarah Parinussa obtained her BSc. and MSc. in Computer Science from University of Amsterdam. Since 2017, she works as a machine learning engineer for developing computer-aided diagnosis in computational dentistry at Promaton Ltd. in Amsterdam.

Peter H.N. de With received his PhD degree from Univ. of Technol. Delft, The Netherlands. In 1984-1997, he worked for Philips Research Eindhoven on video compression and programmable TV architectures. In 1997-2000, he was full professor at the Univ. of Mannheim, Germany, Computer Engineering, and chair of Digital Circuitry and Simulation. He was also part-time professor at the Eindhoven Univ. of Technology (TU/e), heading the chair on Video Coding and Architectures. Since 2011, he is full professor at TU/e and appointed scientific director of the Center for Care and Cure board member of the SA Health program. He has set up an image analysis program with multiple regional and national. He is also international expert in video surveillance for safety and security and has been involved in multiple EU projects on video/object analysis with the industry. He is Fellow of the IEEE, has (co-)authored over 350 papers on video coding, video analysis, architectures, 3D processing and their realization. He is (co-) recipient of multiple papers awards of the IEEE CES, VCIP and Transactions papers and a Eurasip Signal Processing award.