ImplantFormer: Vision Transformer based Implant Position Regression Using Dental CBCT Data

Xinquan Yang, Xuguang Li, Xuechen Li, Peixi Wu, Linlin Shen, Xin Li and Yongqiang Deng

***Abstract*—Implant prosthesis is the most optimum treatment of dentition defect or dentition loss, which usually involves a surgical guide design process to decide the position of implant. However, such design heavily relies on the subjective experiences of dentist. To relieve this problem, in this paper, a transformer based Implant Position Regression Network, ImplantFormer, is proposed to automatically predict the implant position based on the oral CBCT data. The 3D CBCT data is firstly transformed into a series of 2D transverse plane slice views. ImplantFormer is then proposed to predict the position of implant based on the 2D slices of crown images. Convolutional stem and decoder are designed to coarsely extract image feature before the operation of patch embedding and integrate multi-levels feature map for robust prediction. The predictions of our network at tooth crown area are finally projected back to the positions at tooth root. As both long-range relationship and local features are involved, our approach can better represent global information and achieves better location performance than the state-of-the-art detectors. Experimental results on a dataset of 128 patients, collected from Shenzhen University General Hospital, show that our ImplantFormer achieves superior performance than benchmarks.**

***Index Terms*—Implant prosthesis, Dental implant, Vision transformer, Deep learning.**

I. Introduction

PERIODONTAL disease is the world's 11th most prevalent oral condition, which causes tooth loss in adults, especially the aged. Implant prosthesis is the most optimum treatment of dentition defect/dentition loss and implant position is the key to the long-term stability of the prosthesis. Using surgical guide in implant prosthesis leads to higher accuracy and efficiency. Cone-beam computed tomography (CBCT) and oral scanning are common data used for surgical guide design, among which the CBCT data is used to estimate the implant position, while the oral scanning data is employed to analyze the surface of the teeth. However, such process takes the dentist a long time to analyze the patient's jaw, teeth and soft tissue using surgical guide design software such as 3Shape, exoCAD, and guideMia. Artificial Intelligence (AI) based implant position estimation could significantly speed up such process.

Although deep learning has achieved great success in the field of dentistry [1] [2] [3], little research has involved dental implants [4] [5], especially the implant position estimation. As far as we are concerned, our work is the first one to apply deep learning techniques for implant position estimation.

Normally, dentists use the three-dimensional (3D) CBCT data and the surgical guide design software to simulate the implant position, which is usually subjective, tedious and inefficient. Fig.1 shows an example of virtual implant (marked with green) designed by a dentist in the professional software, in top, side and front view, respectively. The implant location in CBCT data can be considered as a 3D regression task. However, training of 3D deep neural network requires large amounts of training data, which requires lots of collection and labelling costs. Therefore, we transform the 3D CBCT data into a series of 2D transverse plane slice views and the task can now be modeled as the regression of the intersections of the center line of implant with each of the 2D slices, which avoids the usage of 3D neural network and alleviates the problem of insufficient training samples. However, the 2D slices captured from the tooth root usually look blurry and the shape of tooth root varies much more sharply across slices, which makes a big challenge to the training of the prediction network. In contrast, the texture in 2D slices corresponding to tooth crown areas are much more rich and stable.

As a result, we propose to train the prediction network using the 2D slices of crown images. Nevertheless, the implant is inserted into the alveolar bone, so that only the ground truth position at tooth root is available. To obtain the implant position label at tooth crown, we firstly fit the center line of implant using tooth root annotations and then extend the center line to the crown area. By this means, the intersections of implant center line with 2D slices of crown image, i.e. the implant position at tooth crown area, can be obtained,

This work was supported by the National Natural Science Foundation of China under Grant no. 91959108, 81901058 and the Research Fund for Overseas High-level Talents of Shenzhen (RC000336). (Xinquan Yang and Xuguang Li contribute equally to this work.) (Corresponding author: Linlin Shen and and Yongqiang Deng.)

X. Yang, Xuechen Li, and L. Shen are with Computer Vision Institute, College of Computer Science and Software Engineering, Shenzhen University, also with AI Research Center for Medical Image Analysis and Diagnosis, Shenzhen University, Shenzhen 518060, China, and the National Engineering Laboratory for Big Data System Computing Technology, Shenzhen 518060, China; Xuguang. Li, Xin. Li and Y. Deng are with the Department of Stomatology, Shenzhen University General Hospital, Shenzhen University, Shenzhen 518055, China, and also with the Institute of Stomatological Research, Shenzhen University, Shenzhen, 518060, China; P. Wu is with Health Science Center, the School of Dentistry, Shenzhen University, Shenzhen 518060, China (e-mail: yangxinquan2021@email.szu.edu.cn; lixuguang@szu.edu.cn; timlee@szu.edu.cn; WPX564@163.com; llshen@szu.edu.cn; lizn007@126.com; qiangyongdeng@sina.com).

which is then used as labels to train the prediction network. When the network is fully trained, the outputs of the network will be transformed back to the tooth root area, as the predicted positions of the implant. Inspired by the dentist who determines the implant position with reference to the neighboring teeth, we employ Vision Transformer (ViT) [6] as backbone for our Implant Position Regression Network (ImplantFormer). By dividing the input image into equal-sized patches and applying multi-head self-attention (MHSA) on image patches, all pixels can establish relationship with others, which is extremely important for implant position decision with reference to the texture of neighboring teeth. In addition, to further improve the performance of ImplantFormer, we design a convolutional stem to coarsely extract the image feature before patch embedding and then extract multi-level features from the backbone and introduce a convolutional decoder for further feature transformation and fusion.

The main contributions of this paper are summarized as follows.

- To the best of our knowledge, this is the first deep learning-based regression method for implant position prediction using CBCT data, which greatly accelerates the design of surgical guide.
- We creatively propose to predict the implant position using tooth crown image, which has better image quality than that of the tooth root.
- A transformer-based Implant Position Regression Network (ImplantFormer) is proposed to consider both local context and global information for more robust prediction.
- The experimental results show that the proposed ImplantFormer achieves superior performance.

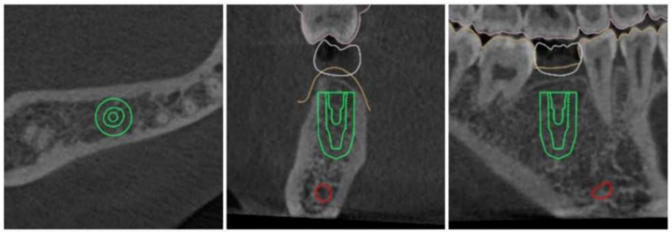


Fig. 1. The virtual implant (marked with green) in the surgical guide design software. From left to right are top view, side view and front view, respectively.

II. Related Work

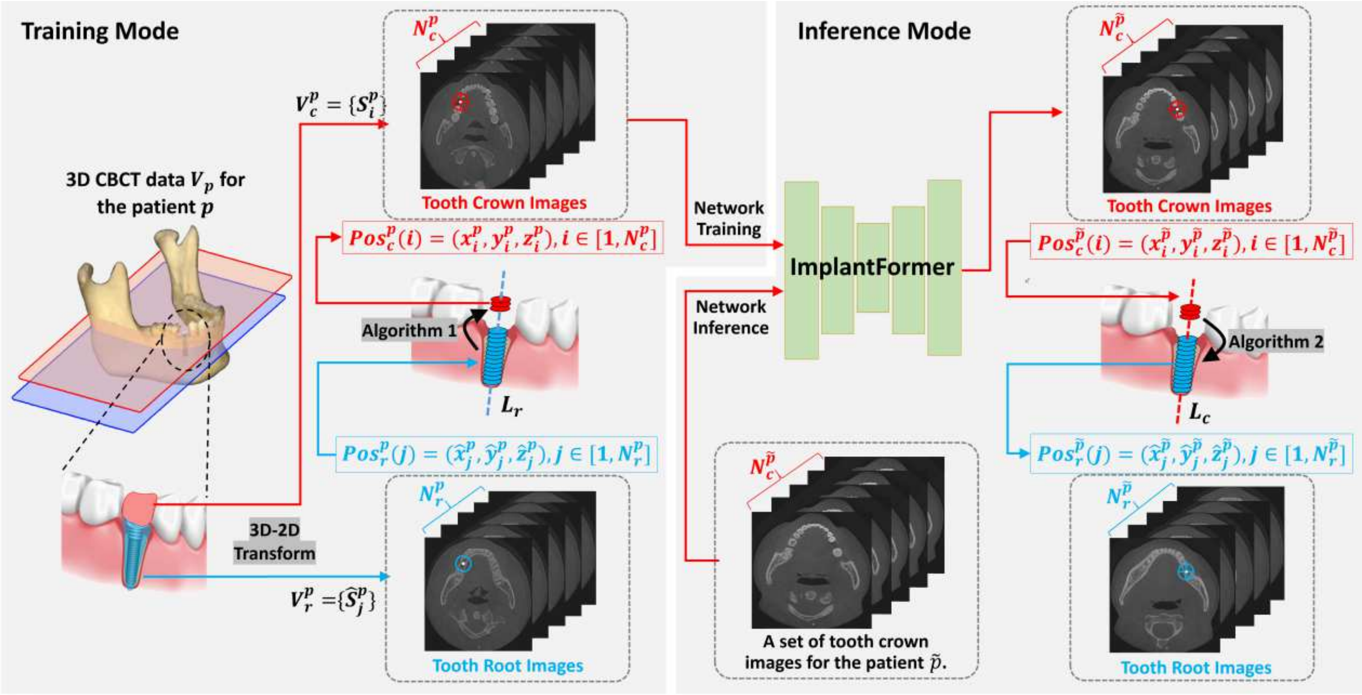
A. Deep Learning in Dentistry

Deep learning technology is widely used in many tasks of dentistry, such as dental caries detection, 2D and 3D tooth segmentation, and dental implants classification. For dental caries detection, Lee et al. [7], Schwendicke et al. [8] and Casalegno et al. [9] proposed deep convolutional neural networks (CNNs) for the detection and diagnosis of dental caries on periapical radiographs, Near-Infrared-Light Transillumination (NILT) and TI images, respectively. For tooth segmentation, Kondo et al. [1] proposed an automated method for tooth segmentation from 3D digitized image captured by a laser scanner, which avoids the complexity of directly processing 3-D mesh data. Xu et al. [2] presented a label-free mesh simplification method particularly tailored for preserving teeth boundary information, which is generic and robust in the complex appearance of human teeth. Lian et al. [3] integrated a series of graph-constrained learning modules to hierarchically extract multi-scale contextual features for automatically labeling on raw dental surface. Cui et al. [10] proposed a two-stage segmentation method for 3D dental point cloud data. The first stage use a distance-aware tooth centroid voting scheme to ensure the accurate localization of tooth and the second stage design a confidence-aware cascade segmentation module to segment each individual tooth and resolve the ambiguous cases. Qiu et al. [11] decomposed the 3D teeth segmentation task as the tooth centroid detection and tooth instance segmentation, which provides a novel dental arch estimation method and introduces an arch-aware point sampling (APS) module based on the estimated dental arch for tooth centroid detection. For the task of implant fixture system classification, Sukegawa et al. [4] evaluated five deep CNN models, i.e. a basic CNN with three convolutional layers, VGG16 and VGG19 transfer-learning models, and finely tuned VGG16 and VGG19, for implant classification. Kim et al. [5] proposed an optimal pre-trained network architecture for identifying four different types of implants, i.e. Brånemark Mk TiUnite Implant, Dentium Implantium Implant, Straumann Bone Level Implant and Straumann Tissue Level Implant on intraoral radiographs.

Different from these above mentioned studies, the proposed ImplantFormer focus on the implant position estimation. To the best of our knowledge, it is the first deep learning-based regression method for implant position prediction, which can greatly accelerate the design of surgical guide.

B. Deep Learning-based Object Detection

Current deep learning-based object detection approaches can be grouped into three categories, i.e. anchor-based, anchor-free and transformer-based. The anchor-based detection sets the pre-defined anchor box before training, which can be divided into two-stage and one-stage methods. Faster R-CNN [12] is a classical two-stage detector which consists of a region proposal network (RPN) and a prediction network (R-CNN [13]). Based on Faster R-CNN, large amounts of detection algorithms [14] [15] have been proposed to improve its performance. SSD [16] and YOLO [17] are classic one-stage detectors, which directly predict the bounding box and category of objects based on the feature maps. However, the anchor-based detector heavily relies on the pre-defined anchor box, which is not robust to the shape variation of the objects. The rise of anchor-free detector breaks this limitation. CornerNet [18] simplified the prediction of the object bounding box as the regression of the top-left corner and the bottom-right corner. CenterNet [19] further simplified CornerNet by regressing the center of object. FCOS [20] set all the locations inside the object bounding box as positives sample and a novel centerness score is used to detect objects. ATSS [21] further improved FCOS by redefining the foreground and background. VFNet [22] proposed an IOU-aware dense object detector, which merges the classification score with IoU to reformulate


Training Mode
3D CBCT data V_p for the patient p
$V_c^p = \{S_i^p\}$
N_c^p
Tooth Crown Images
$Pos_c^p(i) = (x_i^p, y_i^p, z_i^p), i \in [1, N_c^p]$
Algorithm 1
L_r
$Pos_r^p(j) = (\hat{x}_j^p, \hat{y}_j^p, \hat{z}_j^p), j \in [1, N_r^p]$
3D-2D Transform
$V_r^p = \{\hat{S}_j^p\}$
N_r^p
Tooth Root Images
Network Training
Network Inference
ImplantFormer
Inference Mode
$N_c^{\tilde{p}}$
Tooth Crown Images
$Pos_c^{\tilde{p}}(i) = (x_i^{\tilde{p}}, y_i^{\tilde{p}}, z_i^{\tilde{p}}), i \in [1, N_c^{\tilde{p}}]$
Algorithm 2
L_c
$Pos_r^{\tilde{p}}(j) = (\hat{x}_j^{\tilde{p}}, \hat{y}_j^{\tilde{p}}, \hat{z}_j^{\tilde{p}}), j \in [1, N_r^{\tilde{p}}]$
$N_c^{\tilde{p}}$
A set of tooth crown images for the patient $\tilde{p}$.
$N_r^{\tilde{p}}$
Tooth Root Images

(a) Transform the CBCT data into a series of 2D transverse plane slice views.

(b) Project the implant position annotation from tooth root to tooth crown.

(c) Train and Inference the ImplantFormer using the tooth crown images.

(d) Project the predicted implant position from tooth crown to tooth root.

Fig. 2. The whole procedure of the proposed implant position prediction method, , consists of two stages, i.e. training and inference mode. (a) shows the data transformation from 3D CBCT data $\boldsymbol{V_p}$ of patient $\boldsymbol{p}$ to 2D transverse plane slice views $\boldsymbol{S_j^p}$ and $\boldsymbol{\hat{S}_j^p}$ at tooth crown and root, respectively. In (b), the red circle marks the location of $\boldsymbol{Pos_c^p}$, which is the intersection of implant with 2D slices at tooth crown; the blue circle marks the location of $\boldsymbol{Pos_r^p}$, which is the ground truth position of implant at tooth root. Algorithm 1 is proposed to project the ground truth position from tooth root to tooth crown. In (c), the crown images with implant position annotations are used to train the ImplantFormer. In the inference stage, the crown images of patient $\boldsymbol{\tilde{p}}$ is input to the ImplantFormer for implant position prediction. In (d), the red circle denotes the prediction results $\boldsymbol{Pos_c^{\tilde{p}}}$. Algorithm 2 is proposed to transform back the prediction results from tooth crown to tooth root $\boldsymbol{Pos_r^{\tilde{p}}}$, which is marked by the blue circle.

a joint representation. Although anchor-free detection methods achieve great performance, it is limited by its CNN backbone, which is weak in representing the global information. Different from the above detector, transformer-based detector divides the input image into equal-sized patches and applies multi-head self-attention (MHSA) on image patches. The operation of patch embedding and MHSA enable transformer-based detector to model long-ranged dependencies, which can well represent global information. DETR [23] established a new paradigm for object detection, which employs ResNet as backbone and introduces a transformer-based encoder-decoder architecture for object detection task. Deformable DETR [24] extends DETR with a sparse deformable attention that reduces the training time significantly.

The location of implant requires the prediction network to consider both local context (implant position textures) and global information (neighboring teeth textures). Therefore, in this work, we employ ViT-Base model as backbone for the proposed ImplantFormer. The architecture of the detection network follows that of CenterNet, which is simple, efficient and does not require any non-maximum suppression.

III. Method

The whole procedure of the proposed implant position prediction method is shown in Fig. 2, which consists of four stages: 1) Firstly, we transform the 3D CBCT data V_p of patient p into a series of 2D transverse plane slice views, $V_c^p = \{S^p\}$ and $V_r^p = \{\hat{S}^p\}$. As the 3D implant regression problem is now modeled as a series of 2D regression of Pos^p, the prediction network can now be simplified from 3D to 2D and greatly alleviate the insufficiency of training data; 2) Use proposed Algorithm 1 to project the ground truth annotation Pos_r^p from tooth root to tooth crown Pos_c^p; 3) Train ImplantFormer using crown images S_i^p with projected labels and predict the implant position of the new patient $\tilde{p}$ at tooth crown; 4) in the end, Algorithm 2 is proposed to transform the predicted results $Pos_c^{\tilde{p}}$ back from tooth crown to tooth root $Pos_r^{\tilde{p}}$.

A. CBCT Data Pre-processing

As shown in Fig. 2(a), the 3D CBCT data of the patient p, V^p consists of two parts, i.e. tooth crown data V_c^p and tooth root data V_r^p, $V^p = V_c^p \bigcup V_r^p$. Here $V_c^p = \{S_i^p\}, i \in [1, N_c^p]$ and $V_r^p = \{\hat{S}_j^p\}, j \in [1, N_r^p]$, S_i^p and $\hat{S}_j^p$ are the 2D slices of tooth crown and tooth root, respectively. N_c^p and N_r^p are the total number of 2D slices of tooth crown and tooth root, respectively.

Clinically, the implant is inserted into the alveolar bone to act as the tooth root. However, the 2D slices captured from the tooth root usually look blurry and the shape of tooth root

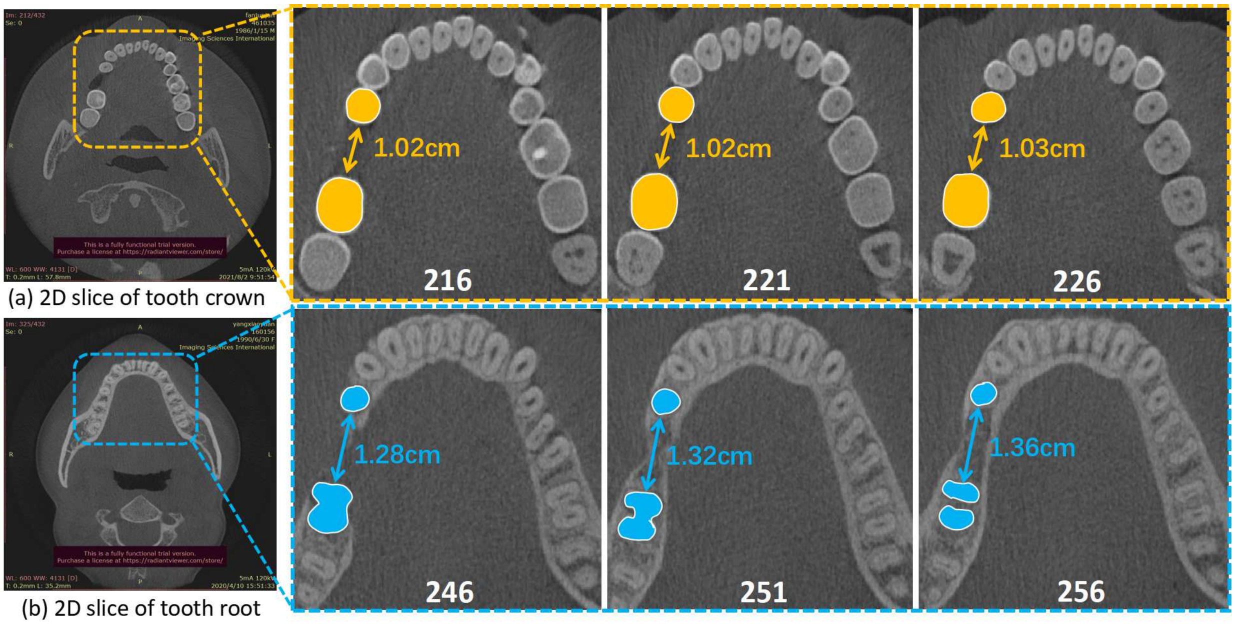


Fig. 3. Visual comparison of the 2D slice of tooth crown (first row) and tooth root (second row). The white number represents the serial number of slice. Areas covered in yellow and blue represent the total area of neighboring teeth at tooth crown and tooth root, respectively; and arrows show the distance between the neighboring teeth.

varies much more sharply across slices, which makes a big challenge to the training of the prediction network. In contrast, the texture in 2D slices corresponding to tooth crown areas are much richer and more stable. As shown in Fig. 3, with an interval of 10 slices, the distance between two neighboring teeth at tooth crown only increases by 0.01 centimeter while which is 0.08 at the tooth root. Meanwhile, the textures of tooth area (including the implant position and its neighboring teeth) in the crown images are richer and more stable than that of tooth root, which is beneficial for the network regression. More detailed comparison of tooth crown and root images are given in the experimental section.

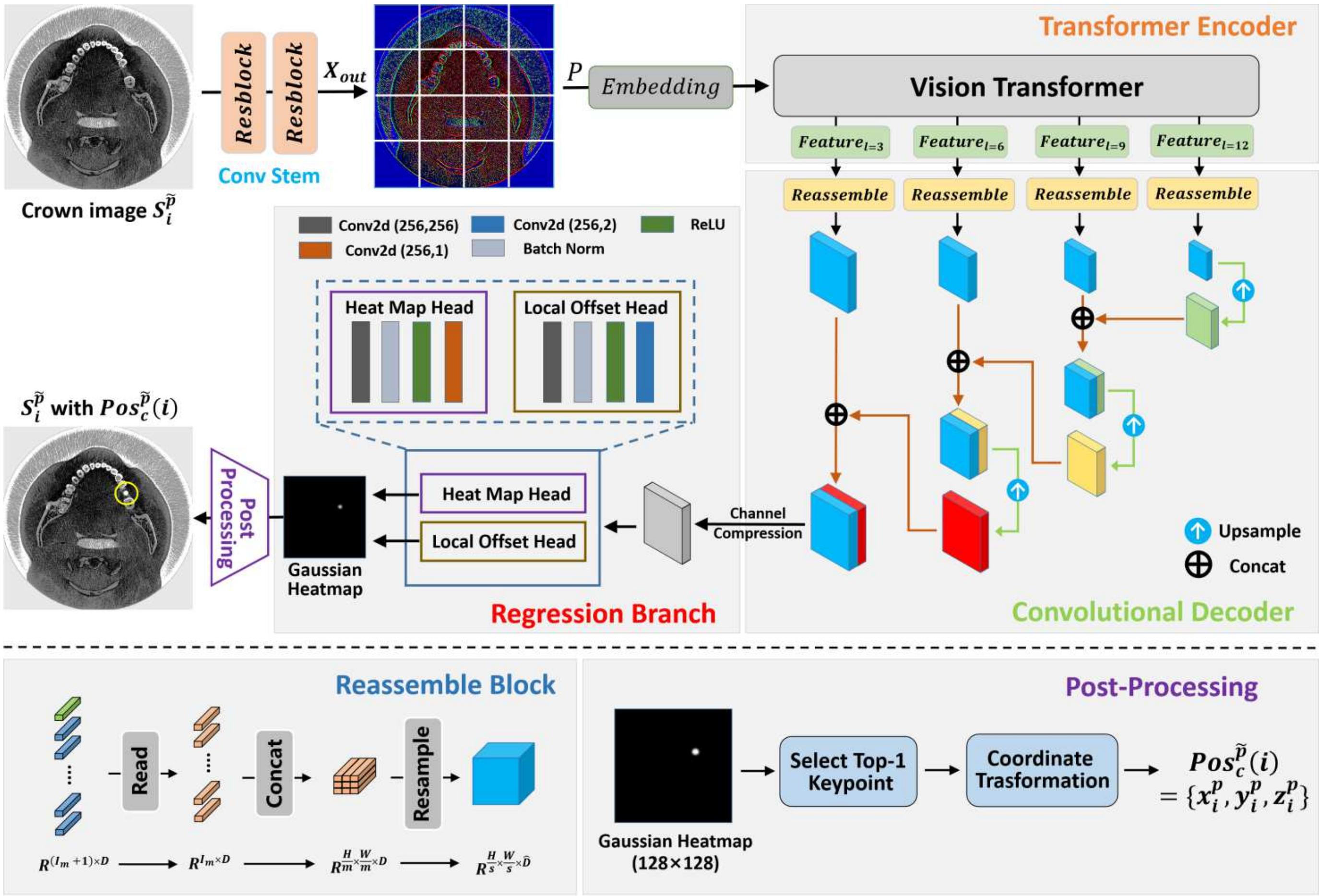
Hence, in this work, we propose to train the prediction network using crown images - V_c^p. However, the ground truth position of implant, defined as $Pos_r^p(j) = (\hat{x}_j^p, \hat{y}_j^p, \hat{z}_j^p), j \in [1, N_r^p]$, is annotated in V_r^p. Here $\hat{x}_j^p, \hat{y}_j^p$ are the coordinates of implant position in $\hat{S}_j^p$ and $\hat{z}_j^p$ is the slice index of $\hat{S}_j^p$ in V_r^p. To obtain the implant position annotation Pos_c^p in V_c^p, a space transform algorithm $T_{Pos_r^p \rightarrow Pos_c^p}$ is proposed. As shown in Fig. 2(b), firstly, we use Pos_r^p to fit the center line of implant, defined as L_r; then L_r is extended to the crown area and the intersections of L_r with V_c^p, Pos_c^p, can be obtained. The detailed algorithm is given in Algorithm 1. The input and output of the algorithm are Pos_r^p and Pos_c^p, respectively. Given the slice index of S_i^p as z_i^p, we define a space transform $T_{Pos_r^p \rightarrow Pos_c^p}$ to calculate the x_i^p and y_i^p. Here k_1, k_2, b_1 and b_2 can be calculated by minimizing the residual sum of square Q.

B. Transformer Based Implant Position Regression Network (ImplantFormer)

The input and output of the ImplantFormer is the 2D slice at tooth crown $S_i \in R^{H \times W \times 3}, (H = W = 512)$ and the corresponding implant position $Pos_c(i)$, respectively. Similar to the existing keypoint regression network [25] [26], ImplantFormer is based on the Gaussian heatmap and the center of the implant position (x_i, y_i) at 2D slice is set as the regression target. The network structure of ImplantFormer is given in Fig. 4, which mainly consists of four components: convolutional stem, transformer encoder, convolutional decoder and regression branch. Given a tooth crown image $S_i^{\tilde{p}} \in R^{H \times W \times 3}$ of the patient $\tilde{p}$ as the network input, we firstly introduce two Resblocks [27] as the convolutional stem for coarsely extracting the image feature, and then the general patch embedding is generated using the output of convolutional stem. Subsequently, we employ ViT-Base model as the transformer encoder and extract multi-level features. In convolutional decoder, we introduce reassemble blocks [28] to transform the multi-level features to multi-resolution feature maps and then integrate them by upsampling and concatenation. Finally, focal loss [29] and L1 loss are employed to supervise the heatmap head and the local offset head, respectively. The output of ImplantFormer is the implant positions at the tooth crown, which is extracted from the predicted heatmap by the post processing operation.

1) Convolutional Stem:

Adding a convolutional stem before transformer encoder is helpful in extracting finer feature representation. Inspired by the previous works [30], we introduce a convolutional stem before the transformer encoder, which consists of two Resblocks. The input-output channel of the two Resblocks are $\{3, 32\}$ and $\{32, 3\}$, respectively. Unlike previous work [31] that use the last stage feature of CNN backbone as the input of transformer encoder, we keep the size of the output tensor $X_{out} \in R^{H \times W \times 3}$ the same as the input image, so that the original ViT can be directly used as backbone.



(a) The structure of reassemble block

(b) The workflow of post processing

Fig. 4. The network structure of the proposed ImplantFormer. (a) the structure of the reassemble block and (b) the workflow of the post-processing. Given a crown image $S_i^{\tilde{p}} \in R^{H\times W\times 3}$,a convolutional stem firstly extracts coarse image features and then the general patch embedding operation is applied. ViT-Base model is used as backbone that takes the patch embedding as input and generates multi-level features. Convolutional decoder is proposed to transform the multi-level features to multi-resolution feature maps via the reassemble block and then the multi-resolution feature maps are integrated by upsampling and concatenation. The final output feature map is fed into the heatmap head and local offset head for generating the Gaussian heatmap. Finally, $S_i^{\tilde{p}}$ with implant position $Pos_c^{\tilde{p}}$ will be obtained after post-processing of the Gaussian heatmap.

2) Transformer Encoder: ViT strikes a remarkable success in computer vision, due to the property of modeling long-ranged dependencies. In our work, the regression of implant position heavily relies on the texture of the neighboring teeth, which is similar to the way dentist used to design the surgical guide. Consequently, the prediction network should possess the capacity of establishing relationship between long-ranged pixels, i.e. between the implant position and the neighboring teeth. ViT divides the input image into equal-sized patches $P \in R^{m\times m\times 3}$, and patch embedding is applied by flattening the patch into vectors using a linear projection. Then, multi-head self-attention (MHSA) is applied to establish relationship between different image patches. By this means, each pixel can perceive the adjacent pixels even with long distance. This characteristic is essential for our work that uses the texture of the neighboring teeth for implant position prediction.

Hence, in this work, we employ ViT-Base model as backbone for our ImplantFormer. The non-overlapping square patches are generated using the output of the convolutional stem. The patch size is set as 16 for all experiments. Meanwhile, to further improve the performance of ImplantFormer, we extract multi-level features from layers $l = \{3, 6, 9, 12\}$. The produced multi-level features with 256 dimensions are used to generate image-like feature representations for the convolutional decoder.

3) Convolutional Decoder: Unlike transformer-based encoder-decoder structures, we introduce the convolutional decoder for further feature transformation and fusion. As shown in Fig. 4, the reassemble block [28] firstly transform the multi-levels ViT features to multi-resolutions feature maps and then the image-like feature maps are integrated by upsampling and concatenation. The final concatenated feature map is fed into the 3×3 convolution for feature smooth and channel compression.

Specifically, we employ a simple three-stage reassemble block to recover image-like representations from the output tokens of arbitrary layers of the transformer encoder (shown as Fig. 4(a)), the reassemble operation is defined as follows:

$$Reassemble_s^{\hat{D}}(t) = (Resample_s \circ Concat \circ Read)(t) \quad (1)$$

where s denotes the output size ratio of the recovered representation with respect to the input image, $\hat{D}$ denotes the output feature dimension, and t refers to the readout token from transformer encoder.

$$Read = R^{I_m+1 \times D} \rightarrow R^{I_m \times D} \tag{2}$$

```
Algorithm 1 Space transform from $Pos_r^p$ to $Pos_c^p$.
Input: The groundtruth annotation of implant position in $V_r^p$:
    $Pos_r^p(j) = (\hat{x}_j^p, \hat{y}_j^p, \hat{z}_j^p), j \in [1, N_r^p]$.
Output: The annotation of implant position in $V_c^p$: $Pos_c^p(i) =$
    $(x_i^p, y_i^p, z_i^p), i \in [1, N_c^p]$.
  1) Define a space transform $T_{Pos_r^p \rightarrow Pos_c^p}$ based on $Pos_r^p$
        $T_{Pos_r^p \rightarrow Pos_c^p}(\hat{z}_j^p) = \begin{cases} \hat{x}_j^p = k_1 \hat{z}_j^p + b_1 \\ \hat{y}_j^p = k_2 \hat{z}_j^p + b_2 \end{cases}$
     where $k$ and $b$ are the coefficient and bias of $L_r$,
     respectively.
  2) Perform the residual sum of squares $Q$ on $T_{Pos_r^p \rightarrow Pos_c^p}$:
        $Q_1 = \sum_{j=1}^{N_r^p} (\hat{x}_j^p - k_1 \hat{z}_j^p - b_1)^2, Q_2 = \sum_{j=1}^{N_r^p} (\hat{y}_j^p - k_2 \hat{z}_j^p - b_2)^2$
  3) Employ the least square method to minimize $Q$, calculate the derivative of $Q$ with respect to $k$ and $b$,
     respectively, and set them to 0.
        $\frac{\partial Q_1}{\partial k_1} = \sum_{j=1}^{N_r^p} 2(\hat{x}_j^p - k_1 \hat{z}_j^p - b_1) \times (-\hat{z}_j^p) = 0$
        $\frac{\partial Q_1}{\partial b_1} = \sum_{j=1}^{N_r^p} 2(\hat{x}_j^p - k_1 \hat{z}_j^p - b_1) \times (-1) = 0$
        $\frac{\partial Q_1}{\partial k_2} = \sum_{j=1}^{N_r^p} 2(\hat{y}_j^p - k_2 \hat{z}_j^p - b_2) \times (-\hat{z}_j^p) = 0$
        $\frac{\partial Q_1}{\partial b_2} = \sum_{j=1}^{N_r^p} 2(\hat{y}_j^p - k_2 \hat{z}_j^p - b_2) \times (-1) = 0$
  4) Use $Pos_r^p$ to calculate $k$ and $b$.
        $k_1 = \frac{N_r^p \sum_{j=1}^{N_r^p} \hat{x}_j \hat{z}_j - \sum_{j=1}^{N_r^p} \hat{x}_j \times \sum_{j=1}^{N_r^p} \hat{z}_j}{N_r^p \sum_{j=1}^{N_r^p} z_j^2 - \sum_{j=1}^{N_r^p} z_j \times \sum_{j=1}^{N_r^p} z_j}$
        $b_1 = \frac{\sum_{j=1}^{N_r^p} \hat{x}_j \times k_1 \sum_{j=1}^{N_r^p} \hat{z}_j}{N_r^p}$
        $k_2 = \frac{N_r^p \sum_{j=1}^{N_r^p} \hat{y}_j \hat{z}_j - \sum_{j=1}^{N_r^p} \hat{y}_j \times \sum_{j=1}^{N_r^p} \hat{z}_j}{N_r^p \sum_{j=1}^{N_r^p} z_j^2 - \sum_{j=1}^{N_r^p} z_j \times \sum_{j=1}^{N_r^p} z_j}$
        $b_2 = \frac{\sum_{j=1}^{N_r^p} \hat{y}_j \times k_2 \sum_{i=1}^{N_r^p} \hat{z}_j}{N_r^p}$
  5) Substitute $k$, $b$ and $z_j^p$ into $T_{Pos_r^p \rightarrow Pos_c^p}$ to obtain $Pos_c^p$.
        $Pos_c^p(i) = (k_1 z_i^p + b_1, k_2 z_i^p + b_2, z_i^p), i \in [1, N_c^p]$
```

$$Concat = R^{I_m \times D} \rightarrow R^{\frac{H}{W} \times \frac{H}{W} \times D} \tag{3}$$

$$Resample_s = R^{\frac{H}{W} \times \frac{H}{W} \times D} \rightarrow R^{\frac{H}{s} \times \frac{H}{s} \times \hat{D}} \tag{4}$$

As shown in the above equations, reassemble block consists of three steps: Read, Concat and Resample. Read operation firstly map the $I_m + 1$ tokens to a set of I_m tokens, where $I_m = \frac{HW}{m^2}$ and D is the feature dimension of each token. After the Read block, the resulting I_m tokens are reshaped into an image-like representation by placing each token according to the position of the initial patch in the image. Then, a spatial concatenation operation is applied to produce a feature map of size $\frac{H}{m} \times \frac{W}{m}$ with channels. Finally, a spatial resampling layer is proposed to scale the representation to size $\frac{H}{s} \times \frac{W}{s}$ with $\hat{D}$ features per pixel.

After the reassemble operation, a simple feature fusion method is used to integrate the multi-resolution feature maps. We upsample the feature map by a factor of two and concatenate it with the neighboring feature map. For the final concatenated feature map, we use 3×3 convolution for feature smoothing and reducing the channel from 512 to 256.

4) Regression Branch: The outputs of ImplantFormer is an implant position heatmap $F \in [0,1]^{\frac{W}{g} \times \frac{H}{g}}$, where g is the down sampling factor of the prediction and set as 4. The heatmap F is expected to be equal to 1 at the center of implant position, and equal to 0 otherwise. Following the standard practice in CenterNet [19], the probability of pixel (x, y) being the target center point is modeled as a 2D Gaussian kernel:

$$L_k = \frac{-1}{N} \sum_{xy} \begin{cases} (1 - \hat{F}_{xy})^{\alpha} \log(\hat{F}_{xy}) & \text{if } F_{xy} = 1 \\ (1 - \hat{F}_{xy})^{\beta} \log(\hat{F}_{xy})^{\alpha} & \\ \log(1 - \hat{F}_{xy}) & \text{otherwise} \end{cases} \tag{5}$$

where α and β are hyper-parameters of the focal loss, $\hat{F}$ is the predicted heatmap and N is the number of keypoints in image. We use $\alpha = 2$ and $\beta = 4$ in all our experiments. To further refine the prediction location, the local offset head is used for each target center point, which is optimized by L1 loss. The loss of the local offset L_{off} is weighted by a constant $\lambda_{off} = 0.55$. The overall training loss is:

$$L_{total} = L_k + \lambda_{off} L_{off} \tag{6}$$

5) Post-processing: The output size of the heatmap is smaller in scale than that of the input image, due to the down sampling operation. We introduce a post-processing to extract the implant position from the heatmap and recover the scale of output. As shown in Fig. 4(b), the post-processing includes two steps: top-1 keypoint selection and coordinate transformation. We firstly select the prediction result with the highest confidence. Then, the Gaussian heatmap with the selected implant positions are directly transformed from the resolution of 128×128 to 512×512. The coordinate of implant position can be obtained by extracting the brightest point in the Gaussian heatmap.

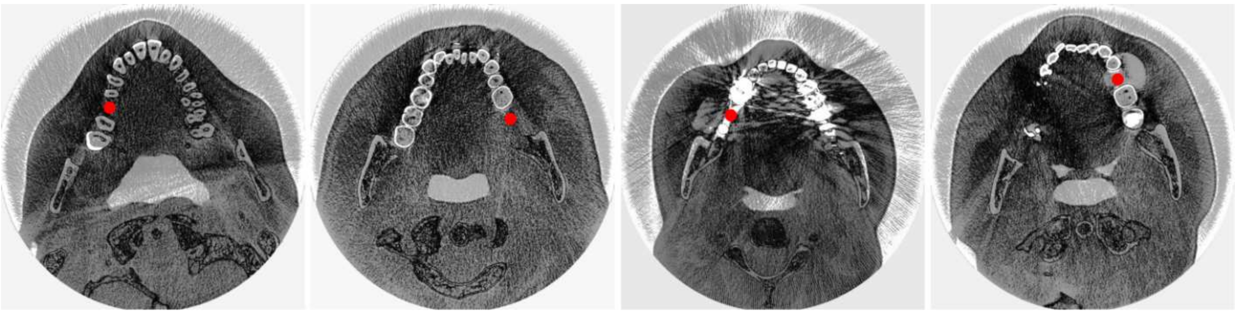


Fig. 5. Some sample images in our dataset. The red points denote the implant position annotation.

C. Project Implant Position from Tooth Crown to Tooth Root

After the post processing, a set of implant position at tooth crown $Pos_c^{\tilde{p}}$ are obtained. To obtain the real location of implant at tooth root $Pos_r^{\tilde{p}}$, we propose Algorithm 2 to transform the prediction results back from tooth crown to tooth root. Algorithm 2 has the same workflow as Algorithm 1, except that the input and output are reversed. As shown in Fig. 2(d), firstly, we use $Pos_c^{\tilde{p}}$ to fit the space line L_c; then L_c is extended to the root area and the intersections of L_c with $V_r^{\tilde{p}}$, $Pos_r^{\tilde{p}}$, can be obtained. The detailed algorithm is given in Algorithm 2.

```
Algorithm 2 Space transform from $Pos_c^{\tilde{p}}$ to $Pos_r^{\tilde{p}}$.
Input: The outputs of ImplantFormer for patient $\tilde{p}$:
    $Pos_c^{\tilde{p}}(i) = (x_i^{\tilde{p}}, y_i^{\tilde{p}}, z_i^{\tilde{p}}), i \in [1, N_c^{\tilde{p}}]$.
Output: The predicted implant position at tooth root:
    $Pos_r^{\tilde{p}}(j) = (\hat{x}_j^{\tilde{p}}, \hat{y}_j^{\tilde{p}}, \hat{z}_j^{\tilde{p}}), j \in [1, N_r^{\tilde{p}}]$.
  1) Define a space transform $T_{Pos_c^{\tilde{p}} \to Pos_r^{\tilde{p}}}$ based on $Pos_c^{\tilde{p}}$
       $T_{Pos_c^{\tilde{p}} \to Pos_r^{\tilde{p}}}(\hat{z}_i^{\tilde{p}}) = \begin{cases} x_i^{\tilde{p}} = \hat{k}_1 z_i^{\tilde{p}} + \hat{b}_1 \\ y_i^{\tilde{p}} = \hat{k}_2 z_i^{\tilde{p}} + \hat{b}_2 \end{cases}$
     where $\hat{k}$ and $\hat{b}$ are the coefficient and bias of $L_c$, respectively.
  2) Perform the residual sum of squares $\hat{Q}$ on $T_{Pos_c^{\tilde{p}} \to Pos_r^{\tilde{p}}}$:
       $\hat{Q}_1 = \sum_{i=1}^{N_r^{\tilde{p}}} (\hat{x}_i^{\tilde{p}} - k_1 \hat{z}_i^{\tilde{p}} - b_1)^2, \hat{Q}_2 = \sum_{i=1}^{N_r^{\tilde{p}}} (\hat{y}_i^{\tilde{p}} - k_2 \hat{z}_i^{\tilde{p}} - b_2)^2$
  3) Calculate the derivative with respect to $k$ and $b$ on $\hat{Q}$, and set them to 0.
  4) Use $Pos_c^{\tilde{p}}$ to calculate $\hat{k}$ and $\hat{b}$.
  5) Substitute $\hat{k}$, $\hat{b}$ and $\hat{z}_j^{\tilde{p}}$ into $T_{Pos_c^{\tilde{p}} \to Pos_r^{\tilde{p}}}$ to obtain $Pos_r^{\tilde{p}}$.
       $Pos_c^{\tilde{p}}(j) = (\hat{k}_1 \hat{z}_j^{\tilde{p}} + \hat{b}_1, \hat{k}_2 \hat{z}_j^{\tilde{p}} + \hat{b}_2, \hat{z}_j^{\tilde{p}}), j \in [1, N_c^{\tilde{p}}]$
```

IV. Experiments and Results

A. Dataset Details

We evaluate our method on a dental implant dataset collected from Shenzhen University General Hospital (SZUH). The dataset contains the CBCT data of 128 patients, which were captured using the KaVo 3D eXami machine, manufactured by Imagine Sciences International LLC. Dentists firstly designed the virtual implant based on the CBCT data using the surgical guide design software. Then the implant position can be determined as the center of the virtual implant.

In our dataset, the location of the virtual implant was determined by three experienced dentists. We select 115 patients for training and 13 patients for testing. As previously discussed, we transform the 3D CBCT data of each patient into a series of 2D slices and project the ground truth position from tooth root to tooth crown. Only tooth crown images are used for network training and testing. Finally, our dataset contains 1958 training images and 239 test images. The data distribution is depicted in Table I. Some sample images of the dataset are shown in Fig. 5.

TABLE I

THE DETAILS OF THE DENTAL IMPLANT DATASET.

	Train	Test	Total
Patients	115	13	128
2D slices	1958	239	2197

B. Implementation Details

For our experiments, we use a batch size of 6, Adam optimizer and a learning rate of 0.0005 for network training. The network is trained for 140 epochs and the learning rate is divided by 10 at 60th and 100th epochs, respectively. Three data augmentation methods, i.e. random crop, random scale and random flip are employed in the network training. The original images are with size 776 × 776 and cropped to 512 × 512 at the network training stage. In the inference stage, images are directly resized to 512 × 512. All the models are trained and tested on the platform of NVIDIA GeForce RTX TITAN.

C. Evaluation Criteria

The PASCAL VOC evaluation criteria, i.e. average precision (AP) is used to evaluate the predicting results. As the average radius of implants is around 20 pixels, a bounding-box with size 21 × 21 centered at the keypoint is generated for performance evaluation. The calculation of AP is defined as follows:

$$Precition = \frac{TP}{TP + FP} \tag{7}$$

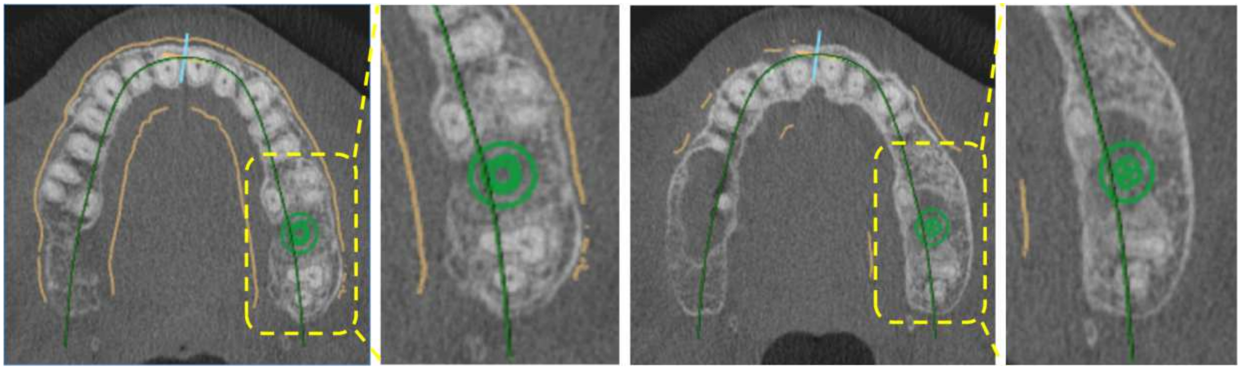


Fig. 6. The CBCT imaging of tooth root in the middle and the bottom slice of the implant (the implant is marked with green).

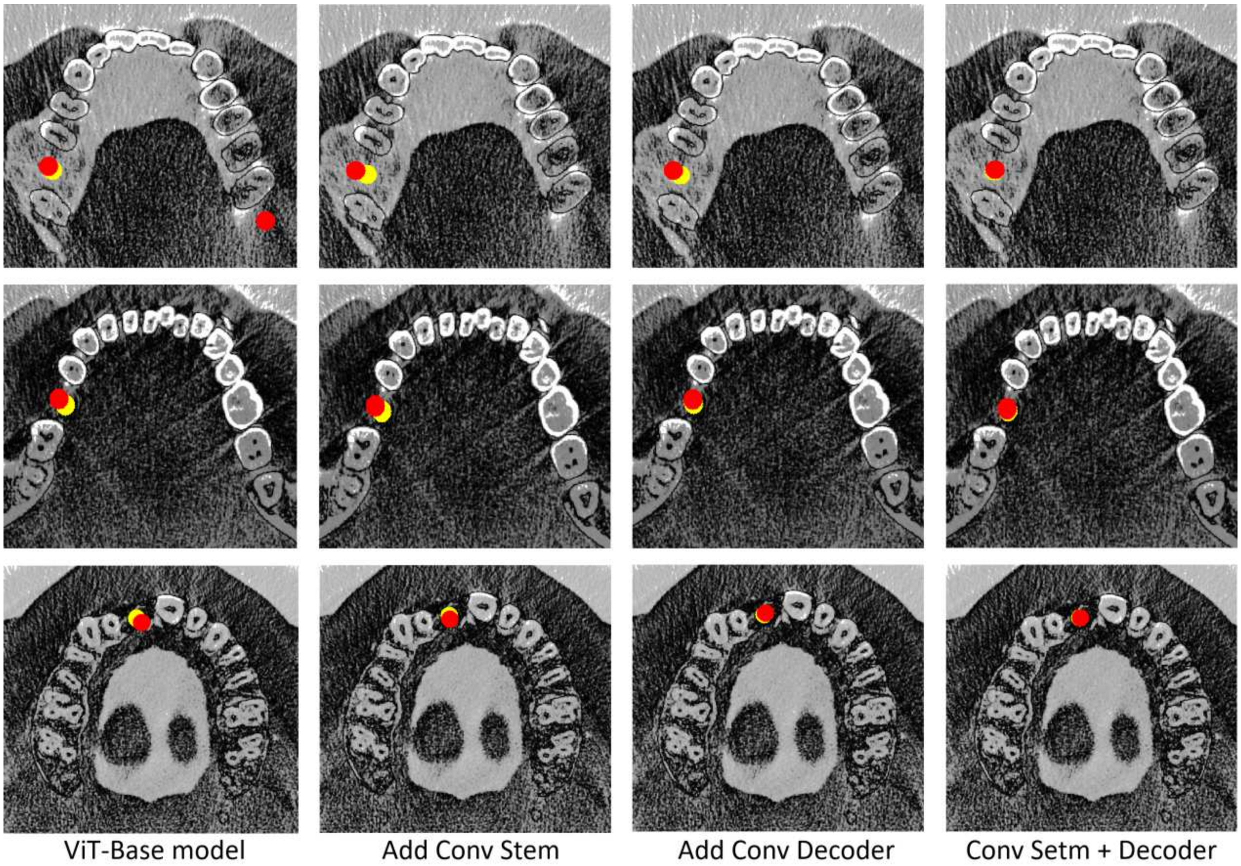


Fig. 7. Visual comparison of detection results of the proposed component. The red and yellow circles represent the predicted implant position and ground truth position, respectively.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$AP = \int_0^1 P(r)dr \tag{9}$$

Here TP, FP and FN are the number of correct, false and missed predictions, respectively. P(r) is the PR Curve where the recall and precision act as abscissa and ordinate, respectively. As implant position prediction requires high accuracy, AP_{50} can not meet the requirement in clinical practice. Therefore, we employed AP_{75} as the evaluation criteria in this work.

D. Performance Analysis

1) Tooth Crown vs. Tooth Root: The CT imaging of tooth root is blurry since it is surrounded by the gingiva. Fig. 6 shows the CT imaging of tooth root in the middle and the bottom slices of the implant, respectively. We can observe from the figure that the texture of the neighboring teeth is blurry in the middle slice of the implant (left image) and even vanish in the bottom slice of the implant (right image), which will influence the regression of implant position, since the prediction is based on the texture of the neighboring teeth. To resolve this problem, in this work, we propose to train the prediction network using the tooth crown image. To justify the effectiveness of tooth crown images for implant position prediction, we train the ImplantFormer using both tooth crown and tooth root images, and compare their results in Table II.

We can observe from the table that the prediction using tooth crown images performs much better than that using the tooth root images. When setting the IOU value as 0.5 and 0.75, the prediction of ImplantFormer using crown images achieves 24.8% and 13.5% AP, respectively, which is 14.3% and 13.1% higher than using tooth root images. The experimental results demonstrate the efficiency of tooth crown images for implant position prediction.

TABLE II
THE DETECTION PERFORMANCE OF IMPLANTFORMER TRAINED USING TOOTH CROWN AND TOOTH ROOT IMAGES.

CBCT slice	$AP_{50}\%$	$AP_{75}\%$
Tooth crown	**24.8**	**13.5**
Tooth root	10.5	0.4

2) Component Ablation: To demonstrate the effectiveness of the proposed components of the ImplantFormer, i.e. convolutional stem and decoder, we conduct ablation experiments on both components to investigate their effects for ImplantFormer. When these components are removed, the patch embedding is generated on the input image and the multi-resolution feature maps are directly upsampled and element-wise added with the neighboring feature map. As discussed previously, we use AP_{75} as the evaluation criteria in the following experiments.

The comparison results are shown in Table III. We can observe from the table that the proposed components are beneficial for ImplantFormer, among which convolutional stem and decoder improves the performance of ImplantFormer by 0.7% and 1.6%, respectively. When combining both of these components, the improvement of performance reaches 3.1%.

In Fig. 7 we visualize some examples of the detection results of different components for further comparison. We can observe from the figure that, for the ViT-base model, the predicted positions are relatively far away from the ground truth position. In the image of the first row, there is also a false positive detection at the right side. Both convolutional stem and decoder can reduce the distance between prediction and ground truth, thus improve the prediction accuracy. When combining convolutional stem and decoder, the model achieves more accurate predictions and reduces the number of missing or false positive detection.

TABLE III
THE DETECTION PERFORMANCE OF IMPLANTFORMER TRAINED USING TOOTH CROWN AND TOOTH ROOT IMAGES.

Backbone	Conv stem	Conv decoder	$AP_{75}\%$
ViT-Base			10.4
	✓		11.1(▲0.7)
		✓	12.0(▲1.6)
	✓	✓	**13.5**(▲3.1)

3) Comparison to The State-of-the-art Detectors: In Table IV, we compare the AP value of ImplantFormer with other state-of-the-art detectors. As no useful texture is available around the center of implant, where teeth are missing, the regression of the implant position is mainly based on the texture of neighboring teeth. Therefore, anchor-free methods (VFNet [22], ATSS [23], RepPoints [36], CenterNet [21]) and transformer-based method (Deformable DETR [26]) are more suitable for this regression problem. Nevertheless, we also employed two classical anchor-based detectors - Faster RCNN [13] and Cascade RCNN [15] for comparison. Resnet-50 is employed as the feature extraction backbone for these detectors. To further determine the validity of vision transformer, we introduce ResNet-50 as the backbone for ImplantFormer, in which we remove the convolutional stem and the patch embedding operation, but keep the feature fusion block.

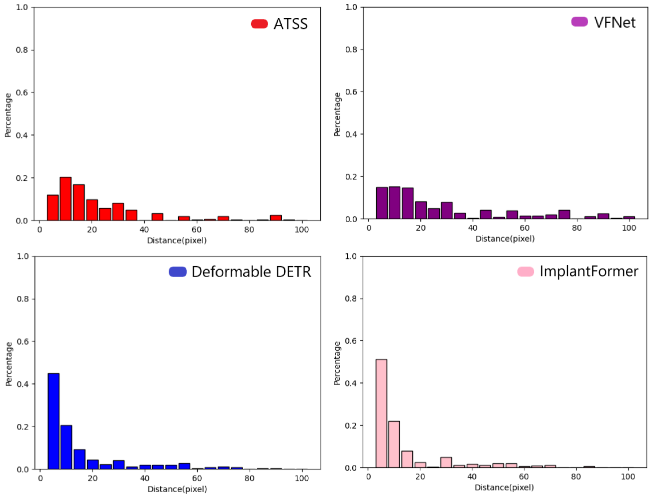


Fig. 8. The distribution of euclidean distances between the ground truth position and the predictions of different detectors.

From Table IV we can observe that the anchor-based methods fail to predict the implant position, which confirms our concern. The transformer-based methods perform better than the anchor-free networks (e.g. Deformable DETR achieved 12.3% AP, which is 0.7% higher than the best performed anchor-free network - ATSS). The ViT-Based ImplantFormer achieves 13.5% AP, which is 2.9% and 1.2% higher than ResNet-based ImplantFormer and Deformable DETR, respectively. The experimental results proved the effectiveness of our method, and the ViT-based ImplantFormer achieves the best performance among all benchmarks.

We choose two detectors from both anchor-free (e.g. ATSS and VFNet) and transformer-based (e.g. Deformable DETR and ImplantFormer) methods, respectively, to further demonstrate the superiority of ImplantFormer in the implant position prediction. Fig. 8 shows the euclidean distance between the ground truth position and the predictions of these detectors. The euclidean distances are summed in an interval of five. Smaller the distance, more accurate the implant position prediction. From the figure we can observe that, for anchor-free detection methods, the distance distributes equally from 0 to 30 pixels. Only 35% predictions distribute within 10 pixels from groundtruth position. In contrast, for transformer-based detection methods, the distance mainly distributes in the range of 0 to 20 pixels. More than 70% of the predictions

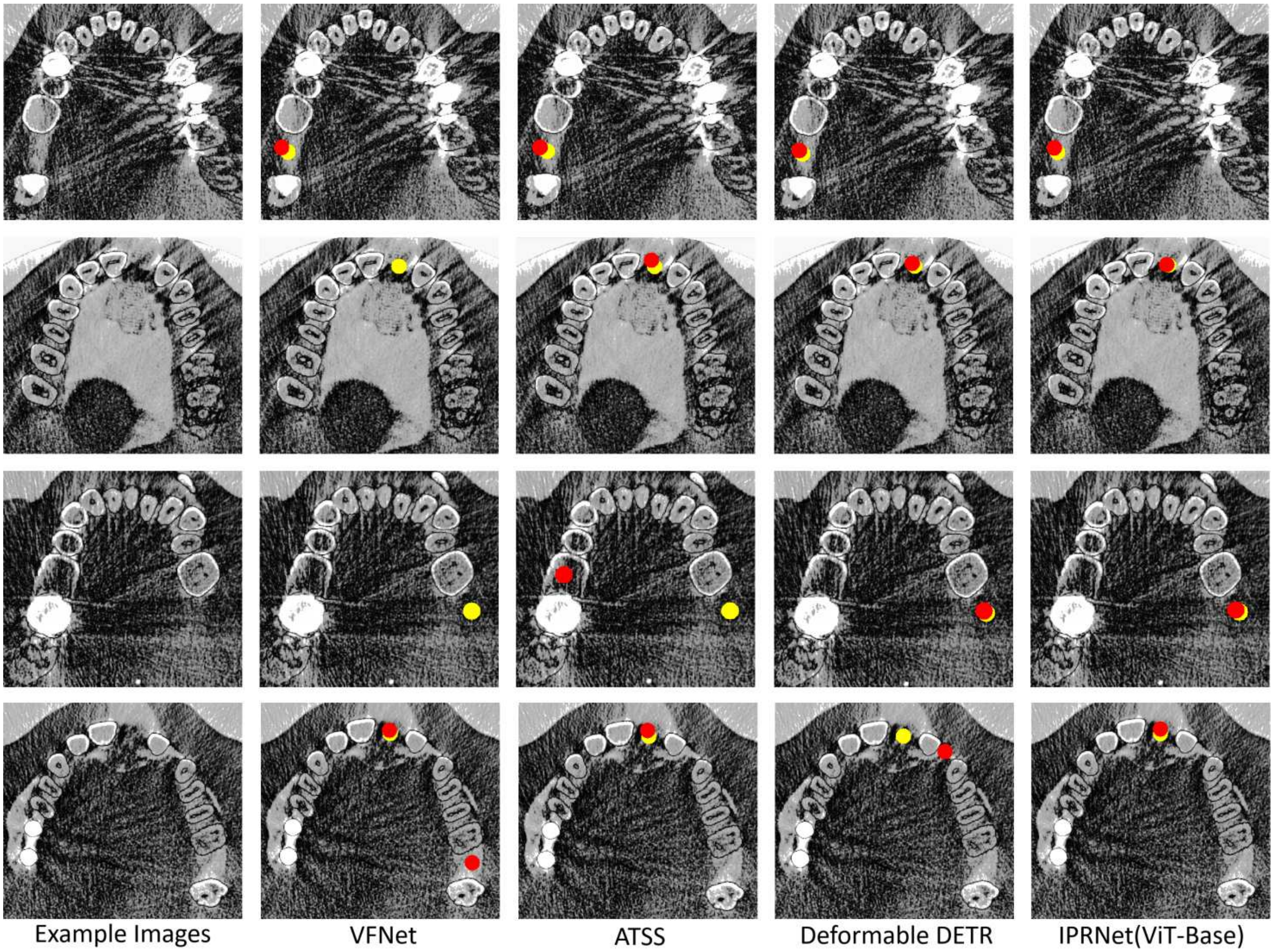


Fig. 9. Visual comparison of the detection results of different detectors. The red and yellow circles represent the predicted implant position and ground truth position, respectively.

located within 10 pixels from groundtruth. Considering that the diameter of implant is around 20 pixels, the predictions with distance more than 10 pixels are meaningless. Therefore, the transformer-based detectors are more suitable for implant prediction task. Compared to Deformable DETR, the ImplantFormer generates about 8% more predictions with distance less than 5 pixels, which indicates that the ImplantFormer achieve much better performance even its AP is only 1.2% higher.

In Fig. 9, we also visualize the detection results of these detectors for four example images. We can observe from the first row of the figure that the transformer-based methods perform better than the anchor-free methods, which is consistent with the distance distribution. As shown in the second and third row, the anchor-free methods generate false detections, while the transformer-based methods perform accurately. In the last row, an example of hard case is given, where the patients teeth are sparse in several places, which leads to error detection of VFNet and Deformable DETR, while the proposed ImplantFormer still performs accurately.

TABLE IV

VERIFICATION OF THE PROPOSED METHOD AND OTHER STATE-OF-THE-ART DETECTORS ON OUR DATASET. "-" REPRESENTS 0.

Methods	Network	Backbone	$AP_{75}\%$
Anchor-based	Faster RCNN	ResNet-50	-
	Cascade RCNN		-
Anchor-free	CenterNet		9.6
	ATSS		11.6
	VFNet		10.9
	RepPoints		9.7
	ImplantFormer		10.6
Transformer-based	Deformable DETR		12.3
	ImplantFormer	ViT-Base	**13.5**

4) Visualization of Attention Map: To verify whether the location mechanism of implant position of the ImplantFormer is in line with the dentist, we visualize the attention map of the ImplantFormer in the Fig. 10. The enlarged portion in Fig. 10(a) shows the location mechanism of implant position by the dentist, which is determined by the edge of neighboring teeth. Fig. 10(b) is the attention map of ImplantFormer, from which we can observe that the network attention is on the edges of neighboring teeth of implant position, which is in accordance with the location mechanism of implant by

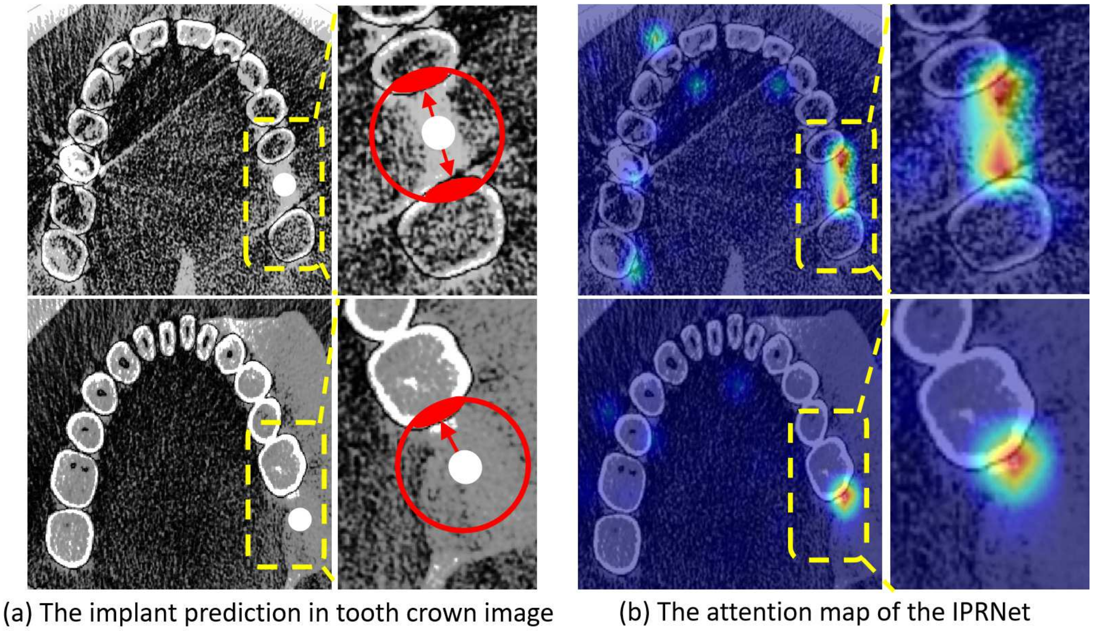

(a) The implant prediction in tooth crown image
(b) The attention map of the IPRNet

Fig. 10. Visual comparison of the location mechanism of implant position. (a) The groundtruth position in tooth crown image. The red areas in the enlarged rectangle represent the edge of neighboring teeth. (b) The attention map of the ImplantFormer, where the highlighted rectangle shows the attention region of the network.

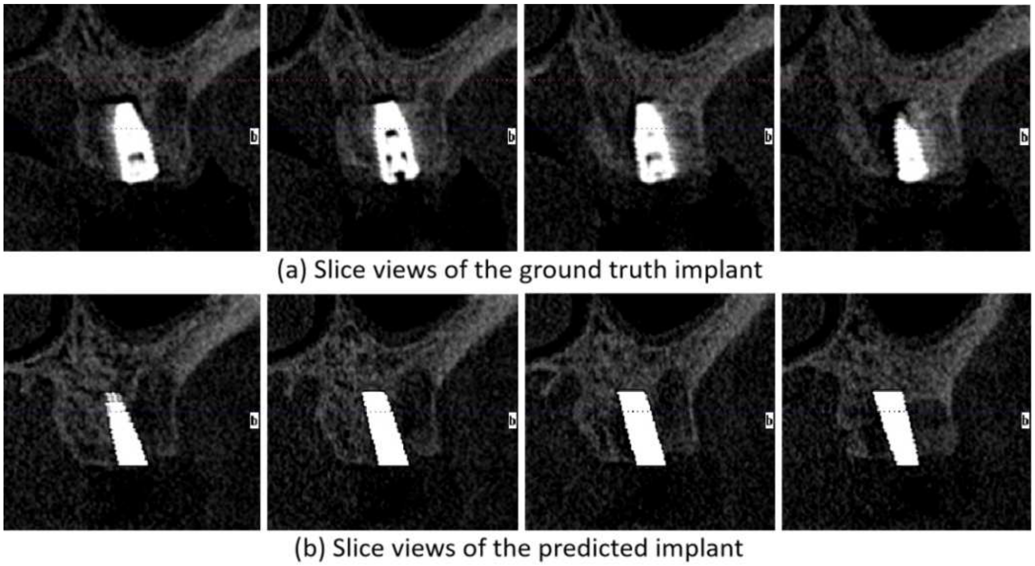

(a) Slice views of the ground truth implant
(b) Slice views of the predicted implant

Fig. 11. Visual comparison of the ground truth implant (first row) and the implant predicted by ImplantFormer (second row) in different slice views.

the dentist. Simultaneously, the network attention around the neighboring teeth also indicates that the vision transformer can establish the relationships between long-ranged pixels, i.e. from the implant position to the neighboring teeth.

5) Comparison of Slice Views of Implant: To validate the effectiveness of the proposed ImplantFormer, we compare four slice views of the actual implant with the predicted implant in Fig. 11. The slice view is the different longitudinal views of CBCT data of the implant, which can well demonstrate the direction and placement of the implant. For the predicted implant, a cylinder with a radius of 10 pixels centered at Pos_r, is generated, and the implant depth is manually set. The pixel value of CBCT data inside the cylinder area is set as 3100. For fair comparison, the longitudinal views from the CBCT data of both ground truth and predicted implant are selected at the same position.

From the figure, we can observe that the implant position and direction generated by ImplantFormer are consistent with the ground truth implant, which confirms the accuracy of the proposed ImplantFormer.

V. Conclusions

In this paper, we introduce an implant position regression network, ImplantFormer, for CBCT data based implant position prediction. By transforming the 3D CBCT data into a series of 2D slices, the 3D implant regression task is simplified as 2D key point regression. A transformer-based Implant Position Regression Network (ImplantFormer) is proposed to include both local context and global information for more robust prediction. We creatively propose to train ImplantFormer using tooth crown images by projecting the annotations from tooth root to tooth crown via the space transform algorithm. In the inference stage, the outputs of ImplantFormer will be projected back to the tooth root area as the predicted positions of the implant. Visualization of attention map and slice view of implant demonstrated the effectiveness of our method and qualitative experiments show that the proposed ImplantFormer achieves superior performance than the state-of-the-art object detectors.

References

[1] T. Kondo, S. Ong, and K. Foong, “Tooth segmentation of dental study models using range images,” IEEE Trans. Med. Imag., vol. 23, no. 3, pp. 350–362, Mar. 2004.

[2] X. Xu, C. Liu, and Y. Zheng, “3D tooth segmentation and labeling using deep convolutional neural networks,” IEEE Trans. Visual. Comput. Graph., vol. 25, no. 7, pp. 2336–2348, Jul. 2019.

[3] C. Lian et al., “Deep multi-scale mesh feature learning for automated labeling of raw dental surfaces from 3D intraoral scanners,” IEEE Trans. Med. Imag., vol. 39, no. 7, pp. 2440–2450, Jul. 2020.

[4] S. Sukegawa et al., “Deep neural networks for dental implant system classification,” Biomolecules, vol. 10, no. 7, p. 984, 2020.

[5] J. Kim et al., “Transfer learning via deep neural networks for implant fixture system classification using periapical radiographs,” J Clin Med, vol. 9, no. 4, p. 1117, 2020.

[6] D. Alexey et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, arXiv:2010.11929.

[7] J. Lee, D. Kim, S. Jeong and S, Choi, “Detection and diagnosis of dental caries using a deep learning-based convolutional neural network algorithm,” J Dent, vol. 77, pp. 106–111. 2018.

[8] F. Schwendicke et al., “Deep learning for caries lesion detection in near-infrared light transillumination images: a pilot study,” J Dent, vol. 92, p. 103260, 2020.

[9] F. Casalegno et al., “Caries detection with nearinfrared transillumination using deep learning,” J Dent Res, vol. 98, no. 11, pp. 1227–1233. 2019.

[10] Z. Cui et al., “TSegNet: an efficient and accurate tooth segmentation network on 3D dental model,” Med. Image Anal., vol. 69, p. 101949. 2021.

[11] L. Qiu et al., “DArch: Dental Arch Prior-Assisted 3D Tooth Instance Segmentation With Weak Annotations,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2022, pp. 20752–20761.

[12] S. Ren et al., “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS), 2015, pp. 91–99.

[13] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 580–587.

[14] Z. Cai and N. Vasconcelos, “Cascade r-cnn: Delving into high quality object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 6154–6162.

[15] K. He et al., “Mask r-cnn,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2961–2969.

[16] W. Liu et al., “Ssd: Single shot multibox detector,” in Proc. Eur. Conf.Comput. Vis. (ECCV), 2016, pp. 21–37.

[17] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788.

[18] H. Law and J. Deng, “Cornernet: Detecting objects as paired keypoints,” in Proc. Eur. Conf.Comput. Vis. (ECCV), 2018, pp. 734–750.

[19] X. Zhou, D. Wang, and P. Krähenbühl, Objects as points, 2019, arXiv:1904.07850.

[20] Z. Tian, C. Shen, H. Chen and T. He, “Fcos: Fully convolutional one-stage object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2019, pp. 9627–9636.

[21] S. Zhang et al., “Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 9759–9768.

[22] H. Zhang, Y. Wang, F. Dayoub and N. Sunderhauf, “Varifocalnet: An iou-aware dense object detector,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2021, pp. 8514–8523.

[23] N. Carion, “Cornernet: Detecting objects as paired keypoints,” in Proc. Eur. Conf.Comput. Vis. (ECCV), 2018, pp. 734–750.

[24] X. Zhu et al., Deformable detr: Deformable transformers for end-to-end object detection, 2020, arXiv:2010.04159.

[25] E. Nguyen et al., “Circle Representation for Medical Object Detection,” IEEE Trans. Med. Imag., vol. 41, no. 3, pp. 746–754, 2021.

[26] Z. Liu et al., “Training-time-friendly network for real-time object detection,” 2019, arXiv:1909.00700.

[27] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

[28] R. Ranftl, A. Bochkovskiy and V. Koltun, “Vision Transformers for Dense Prediction,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2021, pp. 12179–12188.

[29] T. Lin, P. Goyal, R. Girshick and K. He, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2980–2988.

[30] T. Xiao et al., “Early convolutions help transformers see better,” in Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS), 2021, pp. 30392–30400.

[31] D. Meng et al., “Conditional detr for fast training convergence,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 3651–3660.

[32] Z. Yang et al., “Reppoints: Point set representation for object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2019, pp. 9657–9666.