Crown-Generation part 02

Waleed Alzamil

May 29, 2025

Contents

Chapter 1

Crown Generation

1.1 Introduction

Dental crown generation is the core objective of this project, aiming to automate the creation of anatomically accurate dental restorations from 3D scans. While traditional approaches depend on manual modeling and CAD tools, this work proposes a deep learning pipeline to synthesize crown shapes directly from input point clouds. This automation is expected to reduce clinical effort, improve turnaround time, and enhance reproducibility. However, data scarcity remains a significant obstacle to training and validating generative models.

1.2 Related Work

Crown generation from 3D scans overlaps with the broader field of point cloud completion. FoldingNet [8] introduced grid deformation for shape reconstruction, offering a base for surface generation. Transformer-based models like PoinTr [9] have advanced this with self-attention mechanisms, enabling better global context capture and flexible shape completion. However, their direct application to dental crowns is limited due to anatomical complexity and a lack of labeled crown datasets.

1.3 Approach

The proposed pipeline for crown generation is a modular architecture designed to generate anatomically accurate crowns from segmented intraoral 3D point clouds. As illustrated in Figure 1.1, the system consists of:

- An **EdgeConv**-based feature extractor for capturing local geometric features.
- A **geometry-aware transformer encoder** to aggregate contextual and global information.
- A **FoldingNet-based decoder** to reconstruct the final crown surface.

Due to the lack of high-quality labeled crown datasets, the full model was not trained so there will be no **EXPERIMENTS SECTION FOR THIS CHAPTER**. However, all components have been implemented and integrated in preparation for future training.

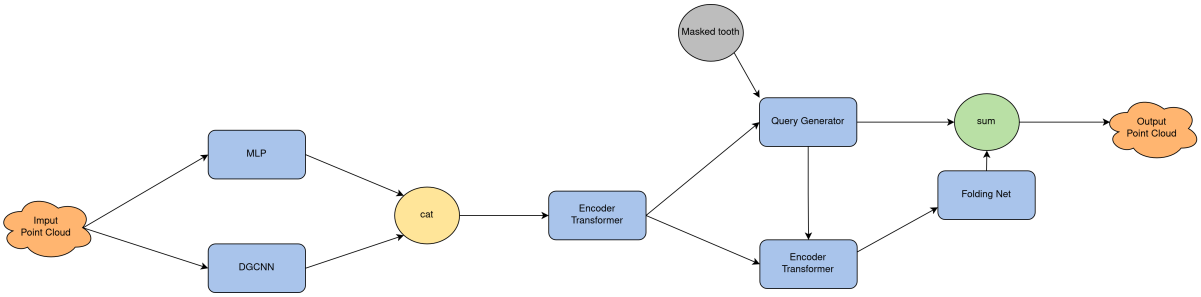

Figure 1.1: Crown generation pipeline architecture.

1.4 Discussion

The design of the crown generation pipeline was a major focus. Despite not being trained end-to-end, the architecture was carefully modularized to facilitate future experiments. The combination of EdgeConv [7], transformer-based encoding [9], and FoldingNet [8] decoding offers a promising structure for high-fidelity crown synthesis. The groundwork has been laid for applying this model once annotated training data becomes available.

Conclusion

Despite limitations in data availability particularly for crown supervision, the project made key advancements across segmentation, alignment, and modeling infrastructure, forming a strong base for future work.

The generative pipeline is now ready to be trained and evaluated once crown datasets are collected. The use of geometry-aware architectures gives it the potential to model fine anatomical structures, and the modularity allows for future flexibility in components and training strategy.

1.5 Future Work

The crown generation pipeline is currently untrained due to the lack of paired input-output data. Future work should prioritize:

- Establishing clinical collaborations to collect labeled 3D crown datasets from dental labs or clinics.
- Developing synthetic crown generation tools to bootstrap early-stage training data.
- Training and benchmarking the full pipeline using metrics such as Chamfer Distance and Earth Mover's Distance (EMD).
- Introducing shape priors or constraints to encourage anatomical realism.
- Evaluating clinical usability of generated crowns with expert feedback.

Chapter 2

Segmentation

2.1 Introduction

Segmentation serves as the foundational step in the crown generation pipeline. It involves isolating the prepared tooth and its neighboring structures from intraoral 3D scans. Accurate segmentation is critical, as any error propagates to downstream stages such as alignment and crown modeling. This thesis builds upon existing models like DGCNN and PCT, refining them through preprocessing enhancements and architectural tuning to improve their robustness and generalization across samples.

2.2 Related Work

Segmentation of dental structures in 3D point clouds benefits from point-based networks. PointNet [5] pioneered deep learning for unordered point sets but lacked local context modeling. PointNet++ [6] introduced hierarchical learning to address this, while DGCNN [7] used dynamic graphs to capture fine-grained local geometry, proving highly effective for anatomical segmentation. These methods form the backbone of most dental segmentation pipelines.

2.3 Approach

In the first semester, models such as DGCNN [7] and PCT [4] were used for segmenting the tooth and its context from intraoral scans. During the second semester, the segmentation accuracy was further improved through better preprocessing (sampling, cleaning, alignment) and architectural tuning. From the first semester, PCT [4] gives us the best results in both cases:

- Trained on a Fixed canonical space.
- Trained to be Invariant to rigid transformations.

Now we improved the model that is trained to be Invariant to rigid transformations and thus is illustrated in the Experiments section. Based on these results [1] we thought

[1] that it seems to be enough for our case in the first semester but now we will not use it for the same purpose as there will be no model for crown generation

that we can depend on the model that's trained on a Fixed canonical space instead of training the model to be Invariant to rigid transformations, but the issue comes when the input is not in the canonical space. For this reason, we decided to train a model that puts any input in the suitable canonical space that the segmentation model is trained based on it, and the idea is illustrated well in the Canonical Space Alignment chpater.

2.4 Experiments

While segmentation was the focus of the first semester, model refinement continued. Architectural tuning and preprocessing adjustments led to modest but measurable improvements in segmentation metrics (overall accuracy, per-class accuracy, and mIoU). And here is a comparsion between the best Invariant model that we get in the first semester and the new one.

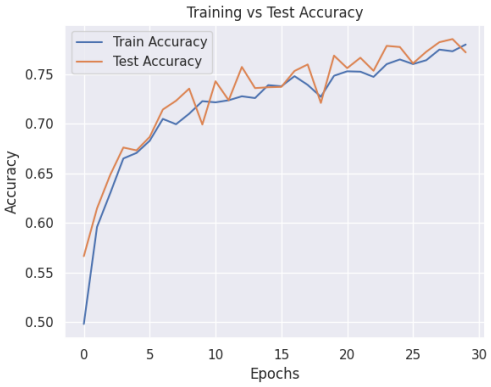

Figure 2.1: Accuracy from first semester.

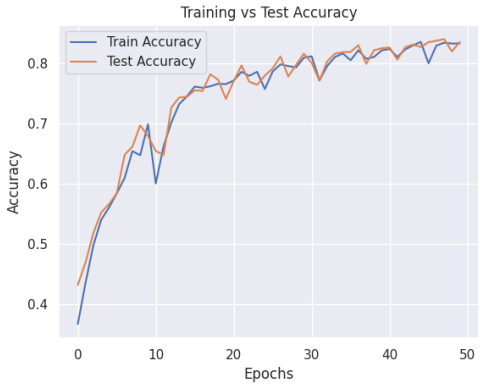

Figure 2.2: Accuracy from second semester.

Figure 2.3: Accuracy metrics

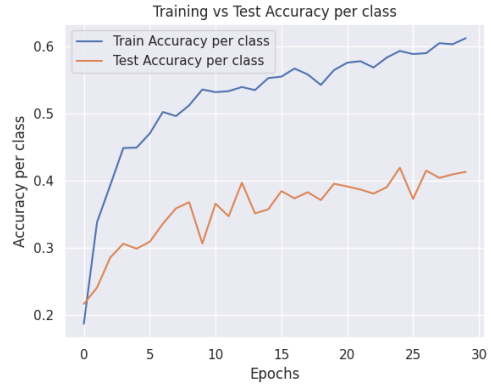

Figure 2.4: Average Accuracy per class from first semester.

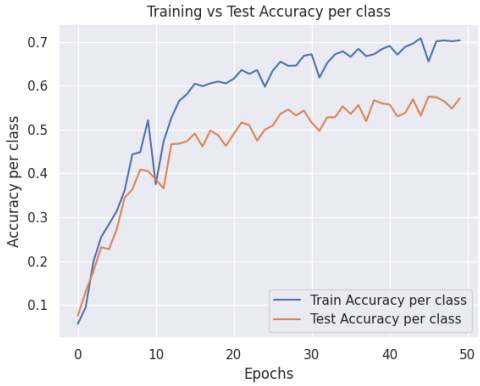

Figure 2.5: Average Accuracy per class from second semester.

Figure 2.6: Average Accuracy per class metrics

So even after refining the model, we couldn't reach the results that the model that is trained on a fixed canonical space.

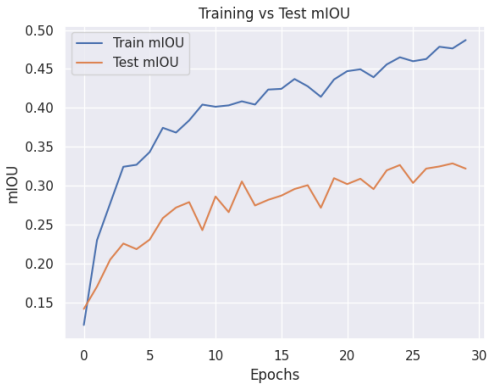

Figure 2.7: mIOU from first semester.

Figure 2.8: mIOU from second semester.

Figure 2.9: mIOU metrics

2.5 Discussion

The refinement of segmentation model PCT led to improved performance in extracting the prepared tooth and its anatomical context. But the results is still far from training the same model on a Fixed canonical space that we achieved (95% accuracy) in the first semester, and here was the motivation for moving forward training a T-Net to do this job for us.

Conclusion

Segmentation results showed state-of-the-art performance within the scope of intraoral scan data. The accuracy and robustness achieved are sufficient to support downstream tasks like margin-line detection and crown generation.

2.6 Future Work

Segmentation showed high performance but still leaves room for future refinements:

- Exploring semi-supervised or active learning to reduce annotation effort.
- Leveraging larger-scale multi-institution datasets for better generalization.
- Integrating segmentation directly into the generation pipeline for joint optimization.
- Improving domain robustness to handle variations in scanning resolution and occlusions.

Chapter 3

Canonical Space Alignment

3.1 Introduction

Canonical alignment ensures that input samples, regardless of orientation or position, are consistently represented in a fixed reference space. This uniformity is vital for reliable segmentation and generation. Since labeled alignment pairs are scarce, this thesis leverages self-supervised learning to predict corrective transformations, restoring each point cloud to a learned canonical pose using a trained spatial transformer network with post-processing via SVD to guarantee rigidity.

3.2 Related Work

Canonical alignment relies heavily on spatial transformer networks (STNs). While T-Net from PointNet [5] was an early attempt, it struggles with complex rotations. Recent advances incorporate more expressive networks with EdgeConv layers to capture local geometry. Self-supervised learning methods such as SimCLR [2]. VICReg [1], Barlow Twins [10], and BYOL [3] have been adapted for learning transformation-invariant features without labels, making them well-suited for unsupervised alignment.

3.3 Approach

To preserve a fixed canonical space across samples in the absence of explicit labels, we employ self-supervised learning techniques focused on geometric consistency. These methods rely on the application of random rigid transformations and the prediction of an appropriate inverse transformation to restore the original canonical pose.

Methodology: Each point cloud is initially centered to eliminate translation components. A random rigid transformation (rotation and optionally reflection) is applied to perturb the data. A neural network then predicts a 3×3 transformation matrix intended to map the transformed point cloud back to its canonical configuration.

Training: Figure 3.1 illustrates the sequence of mathematical operations in the training loop and is well illustrated in Algorithm 2.

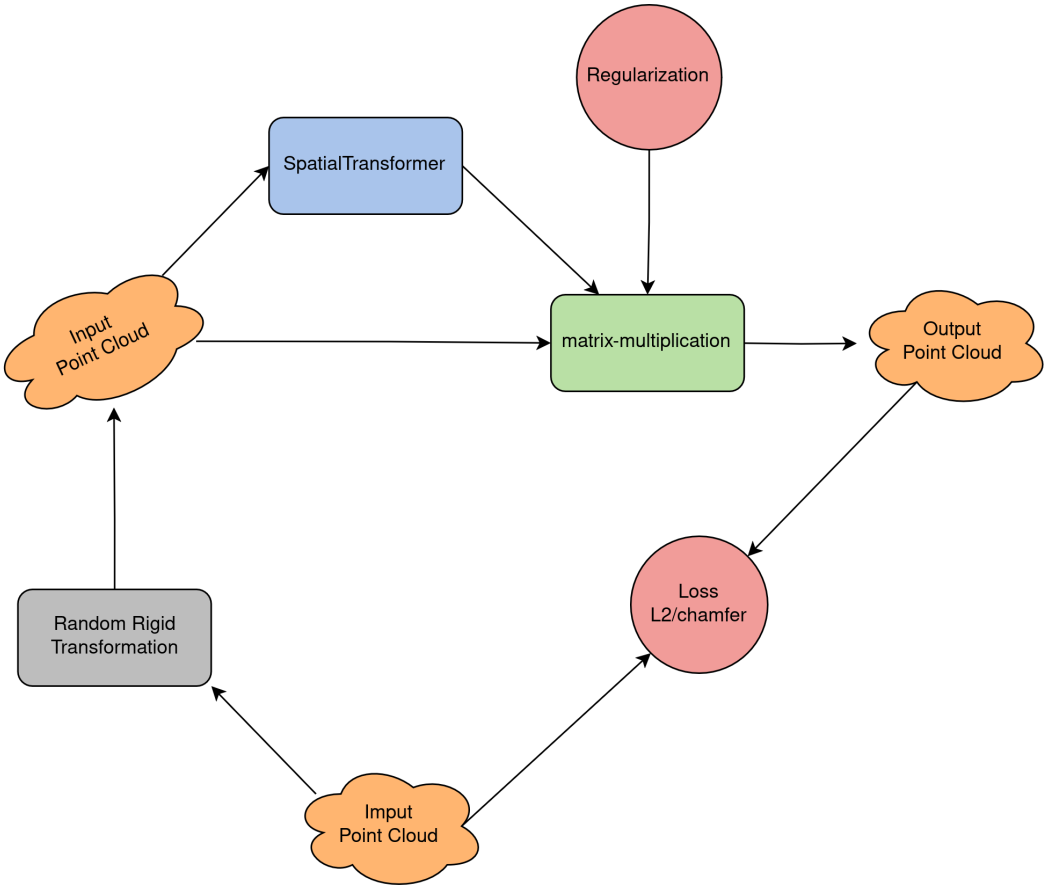

Figure 3.1: Training Diagram.

When predicting transformations in 3D (like aligning point clouds), it's easy for a neural network to predict matrices that are not strictly rotations (e.g., they may include shearing or scaling) that's why we put a regularization term on the output matrix as done in [5], the issue is not totally solved but its effect reduced.

Inference: To ensure the predicted transformation matrix is a valid rotation matrix and our point cloud still rigid we can't use the output matrix directly as in Figure 3.2, we need to apply a post-processing step using Singular Value Decomposition (SVD). By projecting the predicted matrix using Singular Value Decomposition (SVD) Figure 3.3, we guarantee that the result is a pure rotation, which preserves the rigid geometry of the point cloud, the algorithm is well illustrated in Algorithm 1.

This guarantees a valid rotation matrix and corrects minor deviations from orthogonality:

$$M = U\Sigma V^\top \quad \Rightarrow \quad R = UV^\top$$

This step effectively eliminates distortions such as shearing and non-rigid effects that can arise during learning.

Algorithm 1 Inference Algorithm for Canonical Alignment

Require: Input point cloud $X \in \mathbb{R}^{N\times 3}$, trained network f_θ
Ensure: Canonically aligned point cloud $\bar{X}$
Center the point cloud: $\hat{X} \leftarrow X - \frac{1}{N}\sum_{i=1}^{N} X^{(i)}$
Predict alignment matrix: $\hat{R} \leftarrow f_\theta(\hat{X})$
  $U, \Sigma, V^\top \leftarrow \text{SVD}(\hat{R})$
  $R \leftarrow UV^\top$
Apply alignment: $\bar{X} \leftarrow R \cdot \hat{X}$
return $\bar{X}$

Algorithm 2 Training Algorithm for Self-Supervised Rigid Alignment

Require: Point cloud dataset $\mathcal{D} = \{X_i \in \mathbb{R}^{N\times 3}\}_{i=1}^{M}$, I is 3×3 Identity matrix and λ is weight for a regularization term usally equal 0.001
Ensure: Trained alignment network f_θ
for each epoch **do**
  for each mini-batch $\{X_i\}_{i=1}^{B}$ **do**
    Center each point cloud: $\hat{X}_i \leftarrow X_i - \frac{1}{N}\sum_{j=1}^{N} X_i^{(j)}$
    Sample random rigid transformation T_i
    Apply transformation: $\tilde{X}_i \leftarrow T_i \cdot \hat{X}_i$
    Predict inverse transformation: $\hat{R}_i \leftarrow f_\theta(\tilde{X}_i)$
    Apply predicted inverse: $\bar{X}_i \leftarrow \hat{R}_i \tilde{X}_i$
    Compute loss: $\mathcal{L} \leftarrow \sum_{i=1}^{B} \|\bar{X}_i - \hat{X}_i\|_F^2 + \lambda \|I - \hat{R}_i^T \hat{R}_i\|_F^2$
    Backpropagate and update θ
  end for
end for

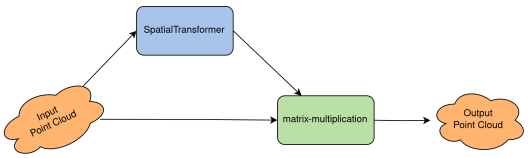

Figure 3.2: Inference pipeline for predicting transformation matrix.

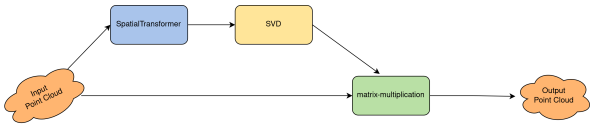

Figure 3.3: Final alignment result using SVD projection to enforce orthogonality.

Figure 3.4: Inference with learned transformation and post-SVD projection.

Architectures: Unlike the original T-Net introduced in PointNet [5], which suffers from poor alignment due to its reliance on pointwise MLPs and symmetric max pooling, we implemented a more expressive network that incorporates local geometric structures. Specifically, we integrated EdgeConv layers to capture neighborhood interactions, significantly improving alignment in complex geometries.

3.4 Experiments

We trained a self-supervised alignment network to recover canonical poses from randomly transformed point clouds. During training, random rigid transformations were applied, and the model learned to predict the inverse transformation. A regularization term was used to encourage near-orthogonality of the predicted 3×3 transformation matrix.

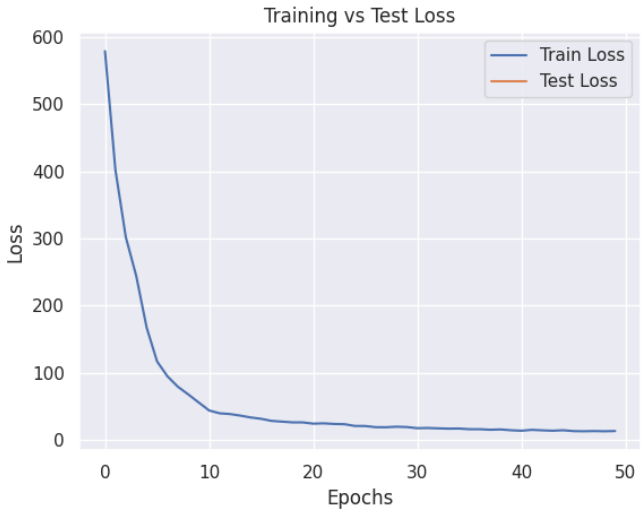

Figure 3.5: Training loss for canonical alignment network.

To qualitatively evaluate the model's performance, we visualize the transformed outputs before and after applying Singular Value Decomposition (SVD) to enforce orthonormality. The transformation before SVD shows moderate alignment, with a Chamfer loss of 3.567, L2 loss of 6.106, and a regularization loss of 0.458. After applying SVD, the matrix becomes orthonormal (determinant ≈ 0.999999), yielding visibly improved alignment with the ground truth.

Color legend for points in the images:

Green: Ground Truth PC **Red**: Model Without SVD **Blue**: Model With SVD

The alignment model showed significant improvement compared to PointNet's original T-Net which we couldn't present their results in the first semester, which suffered from poor convergence and limited spatial reasoning and couldn't learn a good inversion matrix. Incorporating EdgeConv layers resulted in better feature representation of local geometry, and SVD projection ensured that there is no shearing and the vectors are orthonormal.

3.5 Discussion

A major advancement was achieved in canonical alignment. Unlike the original T-Net (which underperformed due to its reliance on MLPs and symmetric functions), the proposed network leveraged EdgeConv to capture neighborhood features and restored canonical structure more effectively. Post-processing via SVD ensured the transformation was a valid rotation, preserving rigidity and eliminating distortions such as shearing.

Figure 3.6: Original input point cloud.

Figure 3.7: Raw transformed input point cloud.

Figure 3.8: Input sample

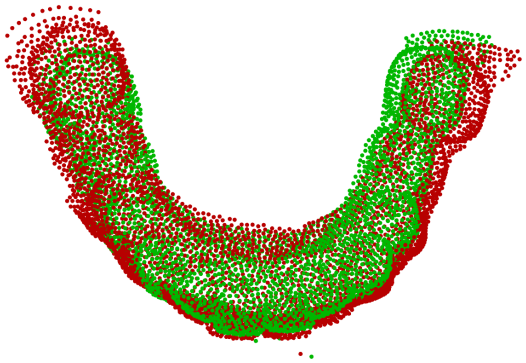

Figure 3.9: Ground truth and Predicted transformation (before SVD). Slight shearing.

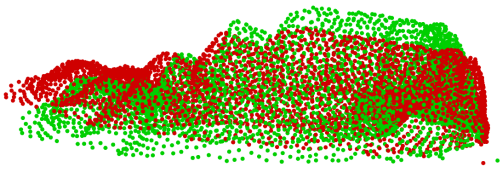

Figure 3.10: Ground truth and Predicted transformation (before SVD). Slight shearing. another view

Figure 3.11: Inference 01

Conclusion

Canonical space restoration enables consistent input formatting, improving the stability and performance of segmentation and generation modules. This network is now fully operational and can serve as a preprocessing standard in any future dental 3D modeling system.

3.6 Future Work

The canonical alignment module performed well, but can be extended further:

- Incorporating student-teacher architecture to learn more invariant representations.
- Testing generalizability across different datasets and scanner settings.

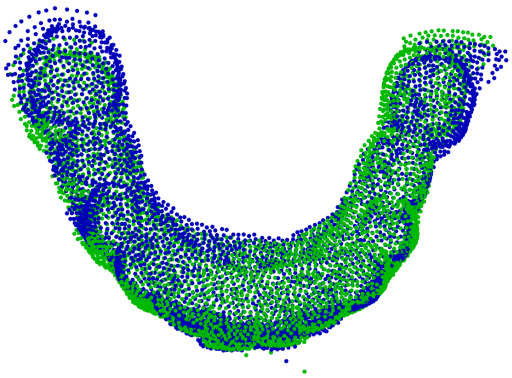

Figure 3.12: Ground truth and Final alignment result after SVD projection.

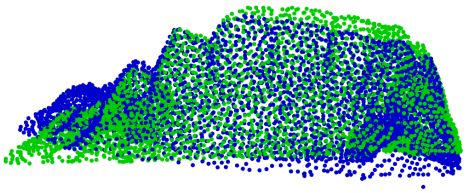

Figure 3.13: Ground truth and Final alignment result after SVD projection. another view

Figure 3.14: Inference 02

- Investigating joint optimization of alignment and segmentation tasks for tighter integration.

Chapter 4

Margin Line Estimation

4.1 Introduction

The margin line marks the boundary between the prepared tooth and surrounding tissue essential constraint for crown fitting and design. Detecting this line precisely is clinically critical, yet challenging due to subtle anatomical variations and limited labeled data. This work introduces a shape-completion-based approach using DGCNN to estimate margin lines from sparse and partial inputs, exploring feasibility and accuracy in a low-data setting.

4.2 Related Work

Margin-line estimation relates to boundary prediction and shape refinement. While not as extensively studied as full object segmentation, models like DGCNN [7] have shown promise in predicting fine boundary details through edge-based context reasoning. Margin estimation also draws from completion-based techniques like FoldingNet [8], allowing the model to infer missing geometry along margin contours, especially in low-data regimes.

4.3 Approach

Margin-line Estimation is a critical step in crown modeling, as it defines the boundary between the prepared tooth and the surrounding area. We adapted **DGCNN** [7] to perform shape completion conditioned on the tooth number.

Due to the limited dataset (approximately 178 cases), but the results of the trained model was just great. Nonetheless, qualitative results showed the model's potential to generalize margin-line structure given more data.

4.4 Experiments

DGCNN-based model was adapted to generate margin-line based on partial input scans, initial experiments showed promising qualitative outputs. With only 178 labeled samples, performance was great and really promising.

Color legend for points in the images:
Red: Case **Green**: Ground Truth **Blue**: Results

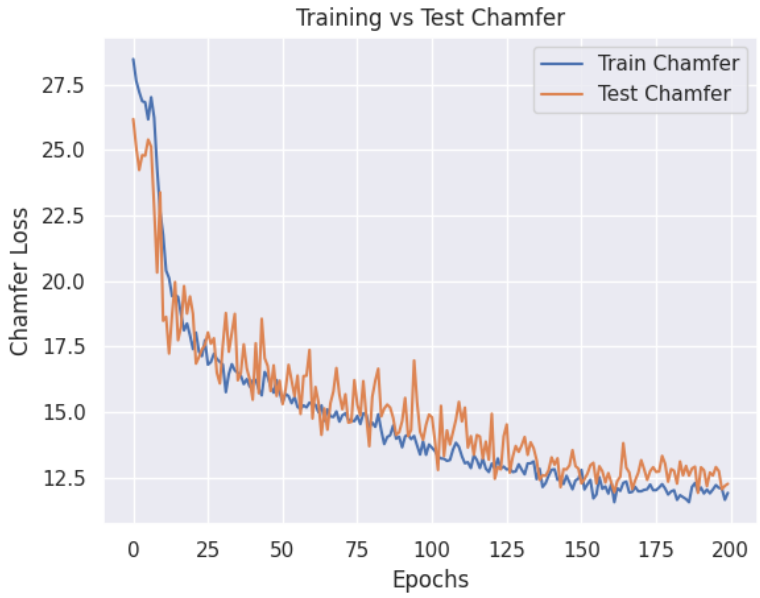

Figure 4.1: Loss.

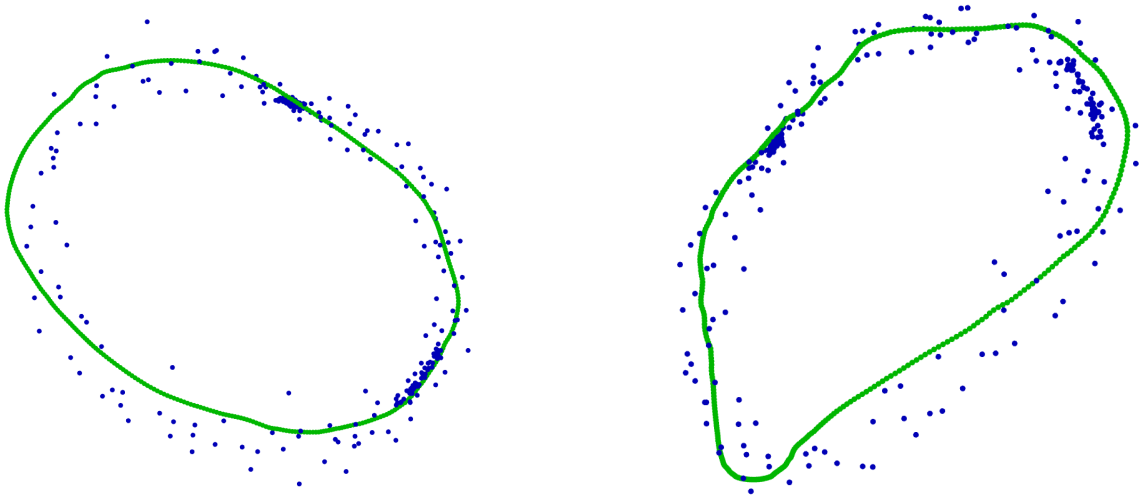

Figure 4.2: Margin-Line: Estimation and ground truth.

Although quantitative metrics for crown generation are unavailable due to the lack of ground truth crowns, the alignment module was evaluated visually and showed robust canonical restoration.

4.5 Discussion

Initial experiments with DGCNN-base model for margin-line Estimation showed the method's potential but were limited by dataset size. The results suggest that expanding the dataset and introducing guided refinement (e.g., using curvature or anatomical priors) could improve precision.

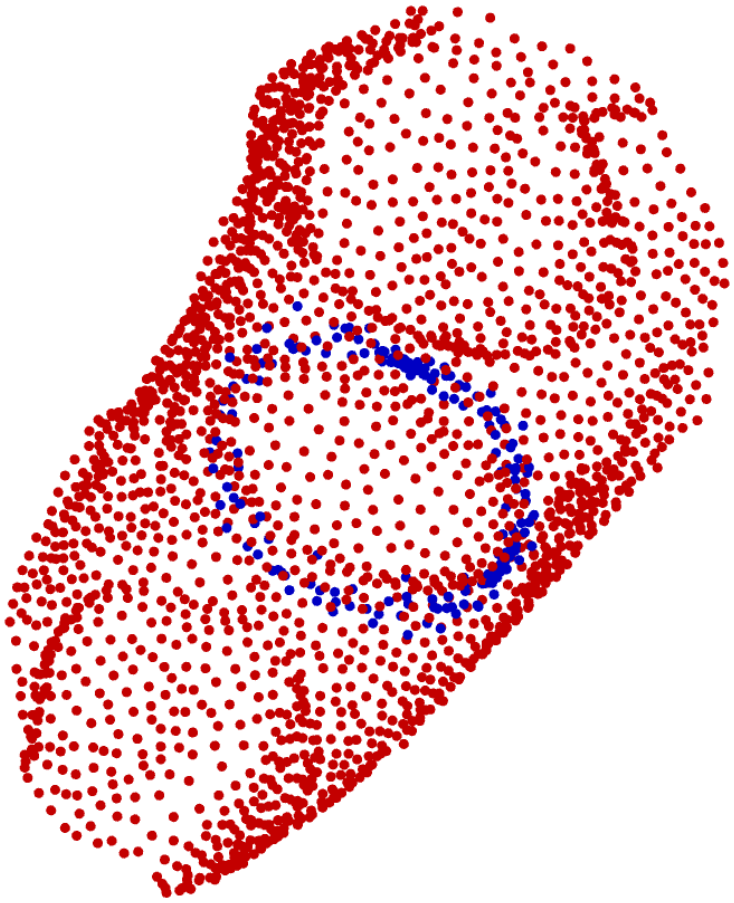

Figure 4.3: Sample output.

Conclusion

The margin-line model is a practical first step toward clinically reliable boundary estimation. While not quantitatively benchmarked, the architecture and visual performance validate the approach for expansion in future work.

4.6 Future Work

While promising, the margin-line estimation module was trained on limited samples. Future improvements include:

- Expanding the dataset and including more diverse tooth anatomies.
- Integrating anatomical priors, curvature features, or surface normals.
- Experimenting with graph-based or implicit surface modeling for smooth boundary estimation.
- Evaluating quantitative fit metrics to guide architectural changes.

Bibliography

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations.

[3] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning.

[4] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199.

[5] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation.

[6] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space.

[7] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds.

[8] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation.

[9] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse point cloud completion with geometry-aware transformers.

[10] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction.