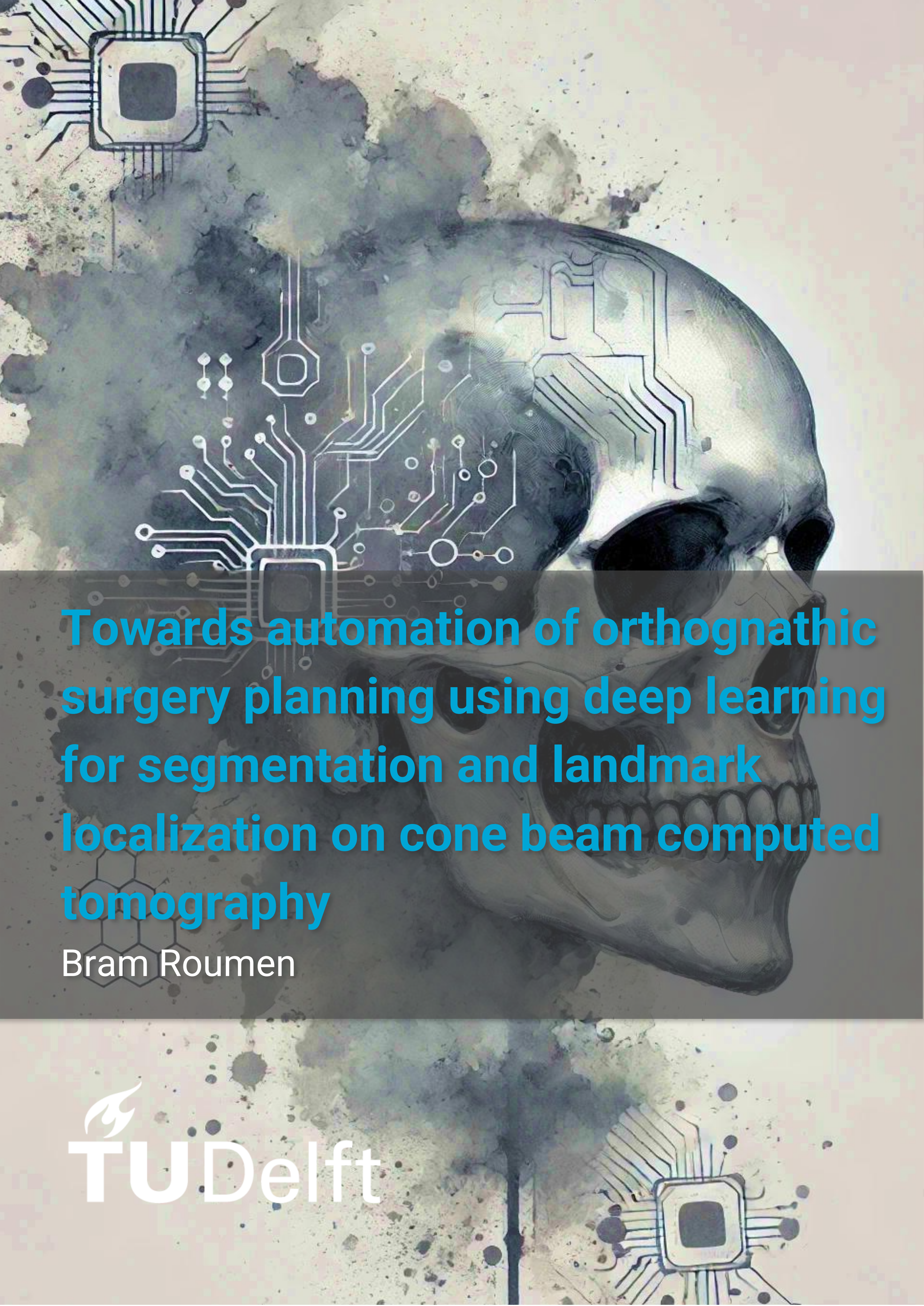


Towards automation of orthognathic surgery planning using deep learning for segmentation and landmark localization on cone beam computed tomography

Bram Roumen

Towards automation of orthognathic surgery planning using deep learning for segmentation and landmark localization on cone beam computed tomography

By

Bram Roumen
Student number : 4610474
9-7-2024

Thesis in partial fulfilment of the requirements for the joint degree of Master of Science in Technical Medicine

Leiden University ; Delft University of Technology ; Erasmus University Rotterdam,

Master thesis project (TM30004 ; 35 ECTS)

Dept. of Biomechanical Engineering, TUDELFT

Project duration:	January 15, 2023 – July 24, 2024	
Thesis committee:	Prof. Dr. E.B. Wolvius	Erasmus MC (chair)
	Dr. B.M.W. Cornelissen	Erasmus MC
	Dr. R.P.J. Van Den Ende	LUMC

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Cover image generated using DALL-E by OpenAI.

Preface

This thesis represents the final product of seven years of dedicated work towards becoming a Technical Physician. It marks the end of my journey as a student, a period during which I have grown both professionally and personally far beyond what I could have anticipated when I left my family home to study in Delft. I am immensely grateful for this experience, which has allowed me to learn, grow, and create lasting relationships.

The successful completion of this project would not have been possible without the guidance of my supervisors. I want to thank Dr. Cornelissen for his constant support throughout this project. Your in-depth knowledge and expertise in 3D craniomaxillofacial surgery planning have been important in shaping my research. Your critical and thorough feedback have significantly enhanced my research skills. Prof. Dr. Wolvius, I am profoundly grateful for your supervision. Your insights into the clinical context of this research, especially by allowing me to assist during the orthognathic surgery procedures, gave me a front-row experience and an unforgettable learning opportunity. Special thanks to Dr. Su for always being available and helping me delve into the field of computer vision. Additionally, I extend my gratitude to all the staff members and residents of the craniomaxillofacial department at Erasmus MC for their time and effort in enriching my medical and dental knowledge..

Lastly, I wish to express my dearest appreciation to my friends, family, and Noor. Your support and encouragement have pushed me to exceed my limits and never lose faith in myself. Without your belief in me, neither this project nor my growth as an individual would have reached their current heights.

With sincere gratitude,

Bram Roumen
Rotterdam, July 2024

1. Abstract

Introduction: 3D imaging has become the standard for orthognathic surgery planning. Manual segmentation and landmark localization are the foundation for assessing deformities and planning of the corrective surgery. However, in 3D imaging, these manual procedures are complex and time consuming. Deep learning has emerged as a promising solution to automate this workflow.

Objective: To develop an automated segmentation and landmark localization workflow to reduce time and efforted required for orthognathic surgery planning using CBCT.

Methods: The dataset consisted of 57 presurgical Cone Beam CT (CBCT) scans. Manual segmentations were created for the mandible, maxilla, skin, mandibular canals (MC) left and right. Additionally, 43 landmarks (13 mandibular, 10 maxillary, 10 dental, and 10 surgical) were manually annotated. Automated landmark localization was approached as a segmentation task, with spherical segmentation of adjacent tissue with radii of 4mm, 5mm, or 6mm around landmarks. The dataset was split into 64% train, 16% validation 20% testset. Seven nnU-Net models were trained: one for segmentation and six (two models for each radius configuration) for landmark localization. Landmark positions were determined by calculating the center of mass of predicted segmentations. Performance was evaluated by comparing results to manual ground truth segmentations and landmark locations.

Results: The nnU-Net was successfully trained to identify five segmentations and 43 landmarks. The overall median [Q1 – Q3] volumetric dice coefficient (vDSC) was 0.91 [0.79 - 0.96] for the segmentation model. The overall median radial error (MRE) for the landmarking models using 4mm, 5mm and 6mm spherical segmentations (each based on two models) were 0.98 [0.58 - 1.60] mm, 1.03 [0.63 - 1.67] mm, and 1.08 [0.66 - 1.82] mm, respectively. The successful detection rate below the clinical acceptability threshold of < 2mm ranged from 78.6% to 81.6%. The inference workflow required 17.7 minutes per patient on average.

Conclusion: The nnU-Net achieved accurate craniomaxillofacial (CMF) anatomy segmentations and precise localization of landmarks, maintaining a clinical acceptability level with an error margin of less than 2mm. This fully automated workflow has the potential to enhance the efficiency of CMF planning.

2. Keywords

Orthognathic surgery, Landmark localization, Segmentation, Cone Beam (CBCT), Deep learning

Table of contents

3. Introduction

In craniomaxillofacial (CMF) surgery, a wide range of congenital and acquired conditions affecting the head, face, jaws, and associated structures are treated. Accurate surgical planning is important for CMF surgery to ensure the desired functional and aesthetic outcomes (1). Traditional planning methods use 2-dimensional (2D) lateral cephalograms for assessing distance and angular measurements within the sagittal plane, forming the basis for cephalometric analysis and subsequent surgical planning (2, 3). However, the 2D representations are limited in their ability to analyze complex three-dimensional (3D) morphology in the craniofacial region (4). The prevalence of facial asymmetry in patients with dentofacial deformities highlights the importance of this limitation, with Severt and Proffit et al. reporting facial laterality occurred in 5%, 36%, and 74% of deformities in the upper, middle, and lower facial regions, respectively (5).

The increased use of cone-beam computed tomography (CBCT) and computed tomography (CT) for cranio-maxillofacial surgery planning provides CMF planners and surgeons with 3D data that allows for a more comprehensive assessment of dentofacial deformities (6). These imaging modalities facilitate bilateral landmarking and enable the quantification of these 3D deformities, thus allowing for 3D cephalometric analysis and accurate 3D surgical planning (7).

Landmark localization and segmentation are crucial steps in CMF imaging analysis that form the foundation for 3D quantification and surgical planning. Landmark localization identifies anatomical reference points necessary for quantitative analysis, such as angle and distance measurements, while segmentation involves delineating anatomical structures for creating virtual models. Using these models, virtual surgery can be performed by planning the osteotomy planes and corrections and assessing the corrected anatomy. This virtual planning allows for the design and creation of 3D printable patient-specific surgical guides and implants on the virtual models. These guides help the surgeon to precisely follow the planned osteotomy planes during the actual surgery. Additionally the implants help to fixate the corrections at a millimeter level, ensuring accuracy and improving surgical outcomes. In orthognathic surgery, this is beneficial for complex yet standard procedures such as the bilateral sagittal split osteotomy (BSSO) and Le Fort I reconstructions, where 3D planning has led to a significant increase in the accuracy of the surgical outcome (8-10).

Manual landmark localization and semi-automatic segmentation are still gold-standard approaches for 3D CMF imaging analysis. The transition from 2D to 3D imaging has made the manual planning process increasingly complex and time-consuming, requiring training and expertise (11). For instance, manual placement of landmarks has been reported to take approximately 15 minutes on CBCT (12, 13). Similarly, semi-automated segmentation of CMF anatomy takes an average of 48.4 minutes, making both methods less feasible for routine clinical use (14). Furthermore, the entire surgical planning process can take up to approximately 60 minutes for a single jaw surgery and 70 minutes for a double jaw surgery when conducted by an experienced surgeon (15).

Semi-automatic segmentation and delineation of craniofacial structures present significant challenges due to several factors. The lack of contrast in the temporomandibular joint and low contrast-to-noise ratio near the field of view (FoV) boundaries. Furthermore, the frequent occurrence of artifacts from orthodontic materials, fixation implants, dental fillings, or crowns makes the process trivial and time-consuming (16). These challenges emphasize the need for a robust and efficient automated workflow to manage these challenges.

Recent advancements in machine learning and deep learning have significantly improved the accuracy and efficiency of automatic segmentation and landmark localization. Deep learning-based methodologies, including customized architectures based on deep reinforcement learning and long-short term memory networks, have achieved clinical acceptability standards, frequently attaining landmark localization accuracy with errors under 2 mm (17-31).

Deep learning models have demonstrated impressive performance in CMF segmentation tasks as well, with several studies reporting volumetric dice similarity coefficients (vDSC) above 0.90 for various craniofacial structures (11, 31, 32). These advancements are important for applying automated workflows in clinical settings, as they ensure reliable and precise anatomical delineation necessary for effective surgical planning.

Among the various deep learning architectures, nnU-Net has emerged as an up-and-coming solution (33). nnU-Net is a well-maintained, open-source, robust, out-of-the-box segmentation framework that automatically adapts to a wide range of biomedical segmentation problems without requiring extensive manual tuning. Its flexible design and overall strong performance make it promising for addressing the complex challenges of CMF segmentation and landmark localization. The nnU-Net remains one of the best-performing algorithms available, offering a streamlined approach to achieving high-quality 3D imaging analysis (34).

4. Objective

The aim of this master thesis is to reduce the time and effort required for CMF surgical planning by developing an efficient and accurate automated segmentation and landmark localization workflow using the existing nnU-Net framework.

5. Methods

5.1 Dataset

The dataset for this research consists of 57 presurgical CBCT images. The CBCT data originates from patients who have been referred to the Erasmus Medical Centre for orthognathic surgery between 2017 and 2022. Ethical approval had been obtained from the ethics board of the EMC prior to this study. The patient demographics and the diagnosis were not available, since the data had been anonymized. Median image shape was 681 x 681 x 481 voxels and ranges from 561 x 561 x 101 to 685 x 693 x 495. The voxel sizes were either 0.25 x 0.25 x 0.25 mm3 or 0.25 x 0.25 x 1.0 mm3.

5.1.1 Segmentations

To provide the deep learning network with consistent input data a segmentation protocol was created. This protocol included guidelines for segmenting the mandible, maxilla and cranial bones, the left and right mandibular canals (MCL & MCR), and the skin. The complete protocol can be found in **Appendix A**. Segmentations were made using Mimics software version 26.0 (Materialise, Leuven, BE).

Segmentations were created of the mandible, maxilla and cranial bones, the MCL & MCR and the skin. Dental segmentations were included in the mandible or maxilla and cranial bones. Thresholding was used for the initial segmentation after which details were manually corrected such as noise induced by brackets, low contrast in the temporomandibular joint region and filling holes in thin bone structures. The mandibular canal was segmented using a curve with standard diameter. The diameter was set to 2.5 mm along the complete length of the MC (35). The start and endpoint were defined as the entrance and outlet for the inferior alveolar nerve in the mandible.

The ground truth segmentation dataset was validated by an expert, who had more than three years of experience with CMF segmentation and surgery planning. In total six cases (10%) were randomly drawn for the dataset for review and were thoroughly checked. Systematic errors were manually corrected for all patients if needed.

5.1.2 Landmarks

To ensure consistency in landmark placement, a detailed protocol was established and documented. **Table 1** provides an overview of the landmarks used in this study, with the protocol for their placement. The landmarks are organized alphabetically by category: mandibular, maxillary, dental, and surgical. This table guided the placement of 14 mandibular, 12 maxillary, 10 dental, and 10 surgical landmarks on the CMF anatomy. The bony and dental landmarks were chosen based on their common usage in scientific literature and within Erasmus MC. The surgical landmarks were selected because these were used as reference points for calculating surgical cutting planes in the Erasmus MC for the BSSO and Le Fort I osteotomy.

Landmarks were manually placed in Slicer 5.6.2, chosen for its ability to save landmarks in accessible file format (JSON), facilitating future research with this dataset (36). Additionally,

Table 1, *Overview of landmarks with their general descriptions and locations, including annotation details in sagittal, coronal, and axial views. The landmarks are listed alphabetically within each anatomical region in the following order: Mandible, Maxilla, Dental, and Surgical. The abbreviations used are: AP = Anterior Point, MP = Midpoint, LP = Lowest Point, PP = Posterior Point, UP = Upper Point, M2 = second molar.*

Landmark name	Anatomical description	Sagittal	Coronal	Axial
B Point (B)	Most PP on anterior surface mandibular symphysis	Most PP	MP determined antero-posteriorly by the two	AP+MP
Condyle left & right (Con-L/R)	UP condylar head	UP + most PP	Most UP+MP	Most UP+MP
Coronoid left & Right (Cor-L/R)	UP head coronoid process	UP + most MP	Most UP+MP	Most UP+MP
Foramen Mentale left & right (MF-L/R)	Cavity in mandibular cortex	MP in line with the cortex	MP in line with the cortex	MP in line with the cortex
Gnation (Gna)	MP between Pogonion and Menton	Most convex curvature of anterior lower cortex	MP	MP
Gonion left & right (Go-L/R)	Most PP edge branch. Bisection tangents posterior edge branch and lower body	Most PP	Most PP+MP	Most PP determined. supero-inferiorly by the 2
Menton (Me)	LP mandibular symphysis	LP	LP +MP	LP+MP
Pogonion (Pog)	Most AP mandibular symphysis	Most AP	MP	AP+MP
Sigmoid Notch left & right (SN-L/R)	LP of sigmoid notch curvature	LP	UP+MP	MP at LP of sigmoid notch
Anterior Nasal Spine (ANS)	Most AP maxillary process nasal floor region	Most AP	AP+MP	AP+MP
Basion (Ba)	Most AP foramen Magnum	Most PP+LP	MP foramen, determined antero-posteriorly by the 2	Most AP+MP anterior contour
Infraorbitale left & right (Inf O- L/R)	Most LP infraorbital margin	Most UP + MP	UP+MP	MP at LP of infraorbital margin
Nasion (Na)	Most AP frontonasal suture	Most AP	MP	Most AP+MP anterior contour
A Point (A)	Most PP maxillar curvature, between anterior nasal spine and supradental point	Most PP	MP determined antero-posteriorly by the 2	AP+MP
Porion left & right (Por L/R)	UP and MP external left roof auditory canal	UP+MP	UP, where skin lies directly on the bone	MP determined supero-inferiorly by the 2
Posterior Nasal Spine (PNS)	Most PP maxillary process nasal floor region	Most PP	PP+MP	PP+MP
Sella (S)	MP Hypophyseal fossa	MP	MP	MP
Supraorbitale left & right (Sup O-L/R)	Most ASP supraorbital margin orbital	Most LP + MP	LP+MP	MP at UP of infraorbital margin
Element 13 (13)	Most LP of element 13	LP	Most LP+MP	Most LP
Element 16 (16)	PP point mesio buchal cusp of element 16	UP	LP +MP	MP
Element 23 (23)	Most LP of element 23	LP	Most LP+MP	Most LP
Element 26 (26)	PP point mesio buchal cusp of element 26	UP	LP +MP	MP
Element 33 (33)	Most UP of element 33	UP	Most UP+MP	Most UP
Element 36 (36)	UP point Mesiobuchal cusp of element 36	LP	UP +MP	MP
Element 43 (43)	Most UP of element 43	UP	Most UP+MP	Most UP
Element 46 (46)	UP point Mesiobuchal cusp of element 46	LP	UP +MP	MP
Lower Incisal midpoint (IsL1)	MP between incisal edge of left and right lower incisers	Most UP + MP	MP between incisal edge	MP between incisal edge
Upper incisal midpoint (IsU1)	MP between incisal edge of left and right upper incisers	Most LP + MP	MP between incisal edge	MP between incisal edge
Inferior border left & right (Inf B-L/R)	Inferior border of mandible right below the M2-line landmark	MP inferior cortex	LP	Most PP
Lingula left & right (Lin-L/R)	Ligula point	Most PP	Most medial point	Most medial + PP
Nasal notch left & right (NN-L/R)	Most concave point of Nasal notch	AP	AUP	Most AP+MP anterior contour
M2-line left & right (M2-L/R)	Oblique line in line with distal roots of M2	MP	Lateral point	lateral point in line with posterior root of the M2
Zygomatic Process left & right (Zyg L/R)	Most concave point of zygomatic process of the maxilla	Most LP	Most concave point	Most Lateral point

this software provides simultaneous axial, sagittal, coronal, and 3D views of the anatomy, aiding in precise landmark placement. The landmarks were first roughly annotated on the 3D models of the segmented mandible and maxilla and cranial bones in the 3D viewer. Then corrections were made on the axial, sagittal and coronal view. The CBCT images were dominant for the placement, since small errors could occur in the segmentation of the anatomy. Therefore the ground truth landmark location was not necessarily on the surface of the segment structures. A complete list of (missing) landmarks is available in **Appendix B.** The same expert checked the validity of landmark localization with 6 randomly selected cases (10%).

The field of view (FoV) of the CBCT scans ranged from the menton to the orbita on average. Consequently, the supraorbitale (left and right) landmarks were excluded from the model training dataset since only one scan was eligible for annotating these landmarks. Additionally, the gnathion was excluded as it does not provide relevant information beyond what is offered by the pogonion and menton.

5.2 Study design

The landmarking task was approached as a segmentation problem, requiring two deep-learning models to avoid excessive overlap between adjacent landmark segmentations, which could reduce localization precision. Initially, landmark locations were used to create spherical segmentations around each landmark, as can be seen in **Figure 1**. The optimal radius for the landmark localization sphere was unknown and tested during this study. Preliminary results indicated that a radius of 3 mm was too small and 10 mm was too large, whereas a radius of approximately 5 mm provided the most accurate results. Therefore, radii of 4 mm, 5 mm, and 6 mm were tested.

This study involved training seven distinct deep learning models: one for the segmentation task and six for the landmark localization task (two models for each of the three radius configurations). Each model could operate independently.

5.2.1 data preprocessing

Anatomy segmentations

The open-source nnU-Net requires Neuroimaging Informatics Technology Initiative (NIfTI) files as input. Masks created in Mimics 26.0 were first converted to parts as STL files, subsequently converted to masks, and saved as NIfTI files utilizing 3D Slicer. The labeling of the segmentations was checked for consistency and adjusted if needed.

Landmark segmentation

The images for the landmark localization task were resampled to a voxel size of 0.5 x 0.5 x 0.5 mm using cubic interpolation and an image size of 340 x 340 x 240 voxels using the SciPy v1.14.1 library. Empty NIfTI masks were created to match the patient's scan in shape, voxel size, and orientation. The manually annotated landmark coordinates were translated to the image coordinate system and used to segment a single voxel, followed by spherical segmentation using a radius of 4mm, 5mm, or 6mm.

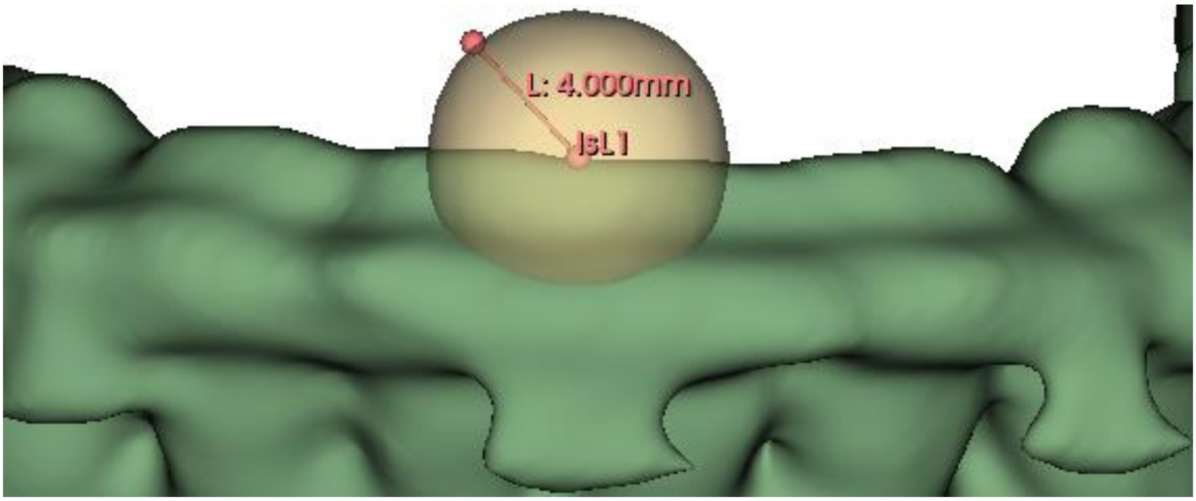


***Figure 1**, an example of landmark IsL1 as segmentation (yellow), with a 4 mm radius.*

For each patient, a mask is created for each landmark, ensuring individual labeling. Once all landmarks for a patient were processed, the segmented landmarks were assembled based on their labels.

To prevent overlap between segmentations of adject landmarks, landmarks were divided into two groups: A and B, see **Appendix C.** Consequently, two models were trained for each radius, resulting in a total of six landmarking models: two models for each configuration (4 mm, 5 mm, and 6 mm). The models were referred to as model_[radius]mm, for example, the models trained for group A and B using a 4 mm radius were referred to as model_4mm.

5.2.2 nnU-Net

The nnU-Net framework, presented by Isensee et al., provides an out-of-the-box segmentation solution by automatically configuring itself to any dataset through a dataset fingerprint (34). This fingerprint includes extracting parameters like image shape, voxel size, and intensity distribution from the train dataset. These parameters inform the resampling strategy, intensity normalization, and patch size, ensuring optimal preprocessing, network architecture, training, and post-processing configurations.

In this study, the 3D-full resolution configuration was used, with a five-fold cross-validation strategy over 1000 epochs. CBCT images were split into training, validation, and test sets in a 64/16/20 ratio. For each deep learning network trained during this study, the same data split was used. The data augmentation in the training pipeline was modified by disabling x-axis mirroring to preserve the distinction between left and right mandibular canal segmentations and bilateral landmarks. Training was performed using 8x NVIDIA A40, 48 RAM available at the GPU cluster in the Erasmus MC. Training times for each model were monitored.

5.2.3 Data post-processing

For both the segmentation and landmark localization models, nnU-Net automatically created a post-processing pipeline. This pipeline utilized connected component analysis to identify the largest segmented voxel group in each mask, removing all smaller, irrelevant segmentations. The pipeline then tested if the performance increased after removing these small objects for each label. Based on the results, the post-processing was enabled or disabled for each label accordingly.

After postprocessing the segmentations produced by the segmentation model were ready for subsequent analysis. For the landmark localization, an additional script was developed to determine the coordinates of the center of each segmented sphere by calculating the center of mass. The coordinates of the center of mass were chosen to reflect the center of the sphere and thus the landmarks location.

5.2.4 External dataset validation

MICCIA 2024 challenge was used to acquire external datasets to validate the robustness of the trained segmentation model for the mandible, maxilla and cranial bones and mandibular canal segmentations (37, 38). Two datasets were available for this purpose: The ToothFairy2F dataset (TF2F), and the ToothFairy2P dataset (TF2P). The datasets differ in their FoV. Dataset TF2F had a large FoV, ranging from the menton to the zygomatomaxillary suture, whereas dataset TF2P covers a smaller FoV, extending from the menton to the upper dentition, thereby missing most or all of the maxilla.

The segmentation labels were adjusted for comparability. In the MICCAI datasets, each individual dental element, as well as implants, crowns, and bridges, were labeled. Furthermore, the maxillary sinus and the pharynx were labeled as well and the skin was not. Segmentation labels for dental elements were assigned to either the mandible or the maxilla, or for the maxillary sinus and pharynx, removed, to match the labeling in the training dataset using Python. Dental implants, crowns, and bridges were manually assigned to the mandible or maxilla. This was necessary because no distinction was made between mandibular dental implants, crowns, and bridges, and their maxillary counterparts in the original labeling.

External dataset preprocessing

The MICCAI CBCT images and masks, which were in the MetaImage (.mha) format, were converted to NIfTI format, similar to the training data. This conversion was performed using the Insight Segmentation and Registration Toolkit (SimpleITK v2.3.2) in Python.

In the preprocessing stage, both the CBCT images and the corresponding masks resampled to match the shape and size of the images in the initial train dataset and ensure uniformity across all datasets. The original voxel size of 0.3 x 0.3 x 0.3 mm was resampled to 0.25 x 0.25 x 0.25 mm. For the images, cubic interpolation was used during the resampling process to maintain the continuity and smoothness of the image data. On the other hand, the binary masks were resampled using nearest neighbor interpolation to preserve the discrete values of the mask labels.

Following resampling, the images and masks were padded to a uniform size of 681x681x481 voxels to ensure consistency with the training dataset, to which the model architecture was configured. Padding involved adding extra voxels to the edges of an image or mask to achieve the specified dimensions. In this study, padding was performed by adding the minimum voxel value, corresponding to air or background. This approach ensured compatibility with the trained nnU-Net model.

5.3 Analysis

For the analysis of the predictions for the anatomy segmentation, the volumetric Dice similarity coefficient (vDSC) was used to measure the overall overlap between the predicted and actual volumes. The vDSC can be less sensitive to large structures due to their already large overlapping volumes and can be overly sensitive to small structures where the boundary delineation is more challenging. To address these limitations, the surface Dice similarity coefficient (sDSC) with a 1 mm tolerance was also utilized. The sDSC evaluates the ratio of successful boundary delineation within a margin of 1 mm, providing a more nuanced assessment of surface accuracy (39). Additionally, the average symmetric surface distance (ASSD) was calculated to assess the average distance between the surfaces of the ground truth and the predicted segmentations, offering a detailed evaluation of the segmentation accuracy (40). Together, these metrics provide a comprehensive and nuanced analysis of the segmentation performance. For comparability with other scientific literature, the intersection over union (IoU) and the robust Hausdorff distance 95% (HSD95) were calculated. All metrics were calculated using scripts provided by the Medical Segmentation Decathlon, which used the SciPy v1.11.4 library (41).

To evaluate the model's performance on the TF2F and TF2P datasets, the data were tested for normality using the Shapiro-Wilk test, and based on the results, a one-way ANOVA or Kruskal-Wallis test was conducted to compare the vDSC, sDSC, and ASSD for the mandible, maxilla, and MC left and right. The skin was not included since it was not labeled in either MICCAI dataset. This statistical test was used to determine if there were significant differences between the model's performance on external datasets. A p-value of less than 0.05 was considered as the level of statistical significance.

Landmark localization accuracy was assessed by calculating the Euclidian distance between ground truth and prediction and is reported as median radial error (MRE) and the number of landmarks detected within the clinical acceptability error of < 2 mm is noted as the successful detection rate (SDR). Data were represented as median [Q1 - Q3] if they were not normally distributed.

Statistical analyses and visualizations were conducted using R version 3.6.0 and Excel.

5.4 Inference Workflow

The inference workflow is the combined workflow of all models, including pre-processing and post-processing steps, to achieve a fully automated workflow. The inference was run on the GPU cluster with 24GB RAM available. The workflow comprises several steps:

1. **Preprocessing**: The input image should be in NIfTI format. For the segmentation model, the image is resampled to a spacing of 0.25 x 0.25 x 0.25 mm. The image is then padded with the minimum voxel value to achieve a shape of 681x681x481 voxels. For the landmark localization models, the image is resampled to 0.5x0.5x0.5 mm and padded to achieve a shape of 340x340x240 voxels.
2. **Segmentation Model Inference**: Inference is performed on the preprocessed image using the segmentation model. Post-processing is automatically applied following the prediction stage. The segmentation file is exported.
3. **Landmark Model Inference**:

   - **Landmark Model 1**: Inference is conducted for the first group of landmarks.
   - **Landmark Model 2**: Inference is conducted for the second group of landmarks.
4. **Post-Processing for Landmark Localization**: The analysis script identifies the center of the spherical segmentations and translates these into world coordinates. The coordinates are stored in a JSON file, including the name and coordinates of each landmark, using the Slicer template for landmarks.
5. **Patient data**: The exported landmarks files and segmentation file and cephalometric analysis files are saved in a specified folder for accessibility.

The inference workflow can be found in **Appendix D**. This workflow contains three main parts. Two Python files, one for managing the pre-processing and the other for post-processing. The third script is a bash script, which handles model installation, executes the aforementioned Python scripts, and runs the prediction of the models in the GPU cluster environment.

6. Results

6.1 Dataset

In this study, an extended dataset has been created involving segmentation of the mandible, maxilla and cranial base, MCL, MCR, and the skin. Furthermore, 43 CMF landmarks were annotated which could be used for 3D cephalometric analysis and surgical planning. **Figure 2** illustrates the results of the manual segmentation and landmark localization process for a randomly selected patient from the dataset. It shows all landmarks and segmented structures, except for the skin structure (for visibility purposes) and the supraorbital landmarks (left and right) which fall outside the FoV. Among the 57 annotated patients, one patient was missing the right mandibular canal due to a fibula reconstruction of the right body of the mandible. Among the 57 annotated patients, one patient missed the right mandibular canal due to a fibula reconstruction of the right body of the mandible. On average, 1.9 ± 2.0 landmarks were missing per patient, primarily due to limitations in the CBCT FoV or missing dentition.

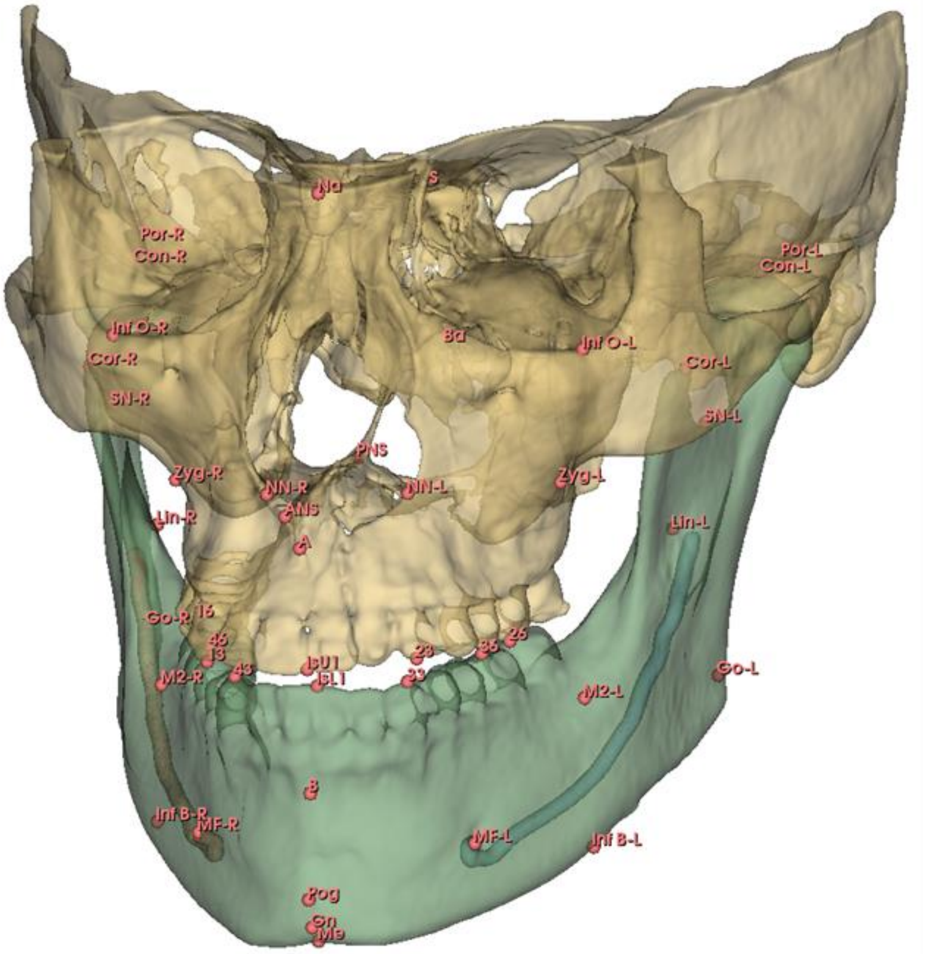


__Figure 2__, Example of one patient, showing all landmarks and segmented structures except for the skin structure to enhance visibility and supraorbital (left and right) landmarks.

6.2 Segmentation model performance on test dataset

The performance of the segmentation model was thoroughly evaluated across the three datasets, with the test dataset results displayed in **Figure 3**. The Shapiro-Wilk test indicated that the results were not normally distributed, so the Kruskal-Wallis test was used for statistical analysis.

The most common error in the predicted segmentations was partially missing or discontinuous mandibular canals. No systematic errors were attributable to dental implants, crowns, or similar factors, although in the TF2P dataset, some maxillary dentition was incorrectly labeled as mandibular dentition. For two examples from the test set, **Appendix F** shows the ground truth and predicted results for both the worst-performing patient and a randomly selected patient representing an average test case.

The vDSC, sDSC, and ASSD are presented for the test dataset, TF2F, and TF2P datasets in **Table 2**. The mandible was segmented with the highest vDSC and sDSC and lowest ASSD across all three datasets. Furthermore, the median ASSD was below 0.5 mm for almost all segmented structures. The highest in ASSD was observed for the maxilla segmentation in the TF2P dataset 7.45 mm [0.81 – 9.81]. The results of the IoU and HSD95 results can be found in **Appendix E**.

The Kruskal-Wallis test showed statistically significant differences among the datasets. Significant differences were found between all datasets for all metrics, except for the MCL's vDSC and ASSD.

***Table 2**, Average symmetric surface distance (ASSD) in mm, Volumetric dice scores (vDSC), surface dice scores (sDSC) at 1 mm tolerance and Kruskal - Wallis statistical test results for the test dataset, TF2F and TF2P datasets.*

		Test dataset	**TF2F**	**TF2P**	**Kruskal-**
	Metric	**(n=11)**	**(n=30)**	**(n=30)**	**Wallis test**
		Median [Q1 - Q3]	**Median [Q1 - Q3]**	**Median [Q1 - Q3]**	**(*p* -value)**
Mandible	ASSD	**0.06** [0.06 - 0.07]	**0.12** [0.11 - 0.13]	**0.15** [0.13 - 0.18]	< 0.005
	sDSC	**1.00** [0.99 - 1.00]	**0.99** [0.98 - 0.99]	**0.98** [0.97 - 0.99]	< 0.005
	vDSC	**0.98** [0.98 - 0.98]	**0.97** [0.97 - 0.97]	**0.97** [0.96 - 0.97]	< 0.005
Maxilla	ASSD	**0.25** [0.20 - 0.38]	**0.33** [0.26 - 0.42]	**7.45** [0.81 - 9.81]*	< 0.005
	sDSC	**0.95** [0.93 - 0.96]	**0.94** [0.92 - 0.96]	**0.70** [0.59 - 0.85]*	< 0.005
	vDSC	**0.91** [0.90 - 0.93]	**0.92** [0.91 - 0.95]	**0.69** [0.57 - 0.84]*	< 0.005
Mandibular canal left	ASSD	**0.24** [0.20 - 0.30]	**0.19** [0.16 - 0.29]	**0.15** [0.14 - 0.27]	**0.073**
	sDSC	**0.98** [0.95 - 0.99]	**0.98** [0.95 - 0.99]	**0.99** [0.96 - 1.00]	0.032
	vDSC	**0.77** [0.73 - 0.79]	**0.80** [0.76 - 0.82]	**0.82** [0.77 - 0.84]	**0.822**
Mandibular canal right	ASSD	**0.26** [0.21 - 0.40]	**0.18** [0.15 - 0.24]	**0.16** [0.14 - 0.21]	0.010
	sDSC	**0.98** [0.90 - 0.98]	**1.00** [0.99 - 1.00]	**0.99** [0.96 - 1.00]	< 0.005
	vDSC	**0.75** [0.65 - 0.79]	**0.82** [0.76 - 0.84]	**0.82** [0.79 - 0.84]	0.046
Skin	ASSD	**0.55** [0.47 - 0.61]	-	-	-
	sDSC	**0.87** [0.85 - 0.88]	-	-	-
	vDSC	**0.95** [0.94 - 0.96]	-	-	-
Overall	**ASSD**	**0.25 [0.18 - 0.44]**	**0.19 [0.14 - 0.32]**	**0.18 [0.14 - 0.61]**	**-**
	sDSC	**0.96 [0.89 - 0.99]**	**0.98 [0.94 - 0.99]**	**0.98 [0.89 - 0.99]**	**-**
	vDSC	**0.91 [0.79 - 0.96]**	**0.86 [0.81 - 0.95]**	**0.83 [0.75 - 0.96]**	**-**

** n=24 due to absence of ground truth segmentations*

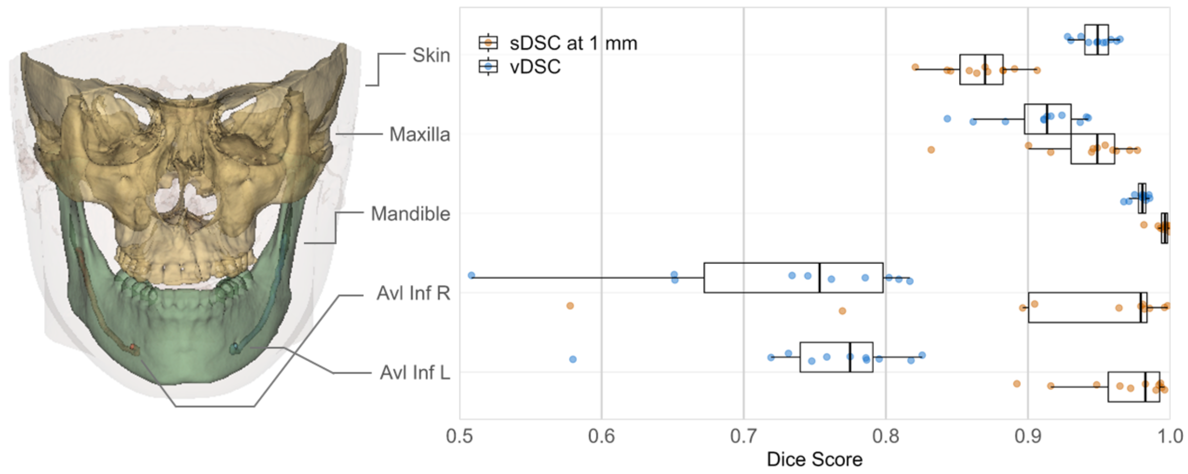


***Figure 3**, boxplots of the vDSC (blue) and sDSC (orange) for the performance of the segmentation model on the test dataset. The colored dots represent individual data points.*

6.3 Landmark localization model performance on the test dataset

A total of 453 landmarks were automatically localized in the 11 test patients. Due to missing dental elements or locations outside the FoV, 20 landmarks were missing in the test dataset. The predicted landmarks were not always segmented as perfect spheres and had slightly lower volumes than the segmented landmarks, see **Appendix F**.
Among the models, the two models trained with landmarks annotated using spheres of 4 mm radius, configuration model_4mm, achieved the lowest overall MRE of 0.98 [0.58 – 1.60] mm. Additionally, model_4mm exhibited the highest SDR < 2 mm, with an SDR of 81.8%. Each of the three configurations demonstrated the lowest MRE for dental landmarks and the highest MRE for surgical landmarks. **Table 3** presents the results of the landmarking models for each region. The most accurate landmark was element 23, predicted by model_4mm with a MRE of 0.45 [0.39 - 0.81] mm; the largest MRE was observed for the Inferior border L predicted by model_6mm with a MRE of 2.88 [1.44 - 4.89] mm. The results of each individual landmark prediction can be found in **Appendix G.** The distribution of MRE for all landmarks in the test dataset for each configuration is shown in **Figure 4**.

6.4 Training and Inference

Training one fold of the segmentation model took approximately 50h and training of one fold for the landmark localization models took approximately 45h.
The inference workflow used the model_4mm for landmark localization. Without manual intervention, the total inference time from preprocessing to the export of patient data was approximately 17.7 minutes per patient. Segmentation accounted for the longest duration, taking 13.2 minutes to complete. Each landmarking model took 1.8 minutes. The preprocessing steps required on average 35 seconds and 20 seconds, respectively.

***Table 3**, landmark localization performance on test dataset for mandibular, maxillary, dental, and surgical landmarks for each of the three configurations. FP = false positive, FN = fasle negative, FPR = false positive rate, and FNR = false negative rate.*

	Landmarks	MRE median [Q1 -Q3]	SDR < 2mm	FP	FN	FPR	FNR
Model_4mm	mandible	0.94 [0.54 - 1.53]	83.1%	0	1	NA*	0.7%
	maxilla	0.88 [0.55 - 1.89]	76.3%	2	5	14.3%	4.9%
	dental	0.75 [0.47 - 1.07]	96.9%	1	6	11.1%	5.6%
	surgical	1.37 [0.81 - 2.09]	71.3%	0	2	NA*	1.8%
Model_5mm	mandible	1.00 [0.64 - 1.61]	82.4%	0	1	NA*	0.7%
	maxilla	0.99 [0.59 - 2.13]	73.4%	2	4	14.3%	3.9%
	dental	0.77 [0.54 - 1.21]	68.2%	1	5	11.1%	4.7%
	surgical	1.40 [0.97 - 2.27]	96.9%	0	0	NA*	0.0%
Model_6mm	mandible	1.05 [0.63 - 1.77]	81.1%	0	0	NA*	0.0%
	maxilla	1.14 [0.65 - 2.15]	69.4%	2	0	14.3%	0.0%
	dental	0.84 [0.57 - 1.23]	94.9%	3	4	27.3%	3.8%
	surgical	1.46 [0.97 - 2.27]	69.1%	0	0	NA*	0.0%
Model_4mm	**overall**	**0.98 [0.58 - 1.60]**	**81.8%**	**3**	**14**	**13.0%**	**3.0%**
Model_5mm	**overall**	**1.03 [0.63 - 1.67]**	**80.1%**	**3**	**10**	**13.0%**	**2.2%**
Model_6mm	**overall**	**1.08 [0.66 - 1.82]**	**78,6%**	**7**	**4**	**25.9%**	**0.9%**

* No missing landmarks, so FPR cannot be computed.

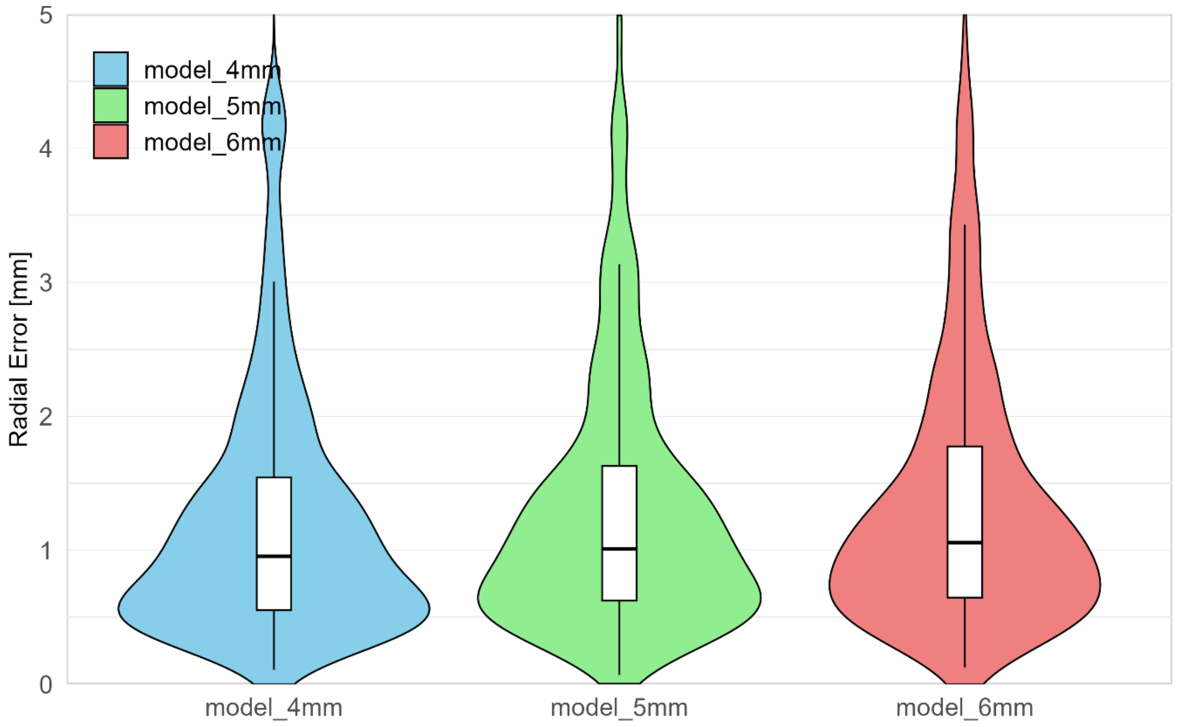


***Figure 4,** voilin plot of MRE of all landmarks (n = 543) in the test dataset for the three configurations*

7. Discussion

The accuracy of surgical planning in orthognathic surgery has significantly increased with the adoption of CBCT scans (8-10). However, the process of manual segmentation and landmark localization in 3D is complex and time-consuming, creating the need for an automated, efficient, and accurate workflow (11). The proposed solution in this study integrates a segmentation network and two landmark localization networks into a fully automated workflow using the open-source nnU-Net. This approach achieved accurate segmentation and landmark localization results, requiring 17.7 minutes per patient without the need for manual intervention.

The volumetric overlap of the segmentation results in the test set was large according to the overall vDSC of 0.91 [0.79 - 0.96]. The mandible, maxilla, and skin all achieved vDSC greater than 0.9. The mandible segmentation, in particular, was highly accurate, with a mean vDSC of 0.98 [0.98-0.98] and a near-perfect mean sDSC of 1.0 [0.99 – 1.0] and ASSD 0.06 [0.06 -0.07] mm. These overall segmentation results are consistent with those reported by Dot et al. and Zhang et al. (11, 31). Despite the overall high performance for the vDSC, the mandibular canal segmentations achieved lower vDSCs of 0.75 [0.65 – 0.79] and 0.77 [0.73 – 0.79] mm compared to the other structures. This discrepancy is likely due to the relatively small volume of the canals, which poses a challenge for volumetric segmentation metrics (40). However, the median ASSDs of 0.25 [0.20 – 0.35] mm demonstrated precise boundary delineation for the MCs, and the sDSCs of 0.95 [0.9 – 1.00] indicated the model's capability to accurately delineate boundaries within a 1 mm margin. The model's performance is similar to dedicated methods proposed by Jaskari et al. and Abdolali et al., which reported a mean (± SD) ASSDs of 0.45 ± 0.12 mm and 0.81 ± 0.2 mm, respectively (42, 43). While direct comparison between mean and median values can be challenging, the low median ASSD indicates that the majority of the test cases had very precise boundary delineations.

The low ASSD values for the mandible (< 0.2 mm) and maxilla (< 0.4 mm) across datasets indicate that the segmentation model's output is suitable for the subsequent planning of patient-specific surgical guides and implants, which improve the accuracy of surgical outcomes (44, 45). However, the ASSD for the maxilla in the TF2P dataset showed a higher value of 7.45 [0.81–9.81] mm. This discrepancy is likely due to inaccuracies in the ground truth labeling in this dataset and was expected, as some patients lacked partial or complete upper dental segmentation despite its presence in the CBCT. Furthermore some upper dentition was incorrectly labeled by the segmentation model as mandibular dentition. The absence of the maxillary bone might have been the reason for this. Another segmentation error was observed for the MCs. The MC was partially missing or discontinuous in some cases, this might be caused by disabling the data augmentation for mirroring along the x-axis. Giving both the MCL and MCR the same label and training the model using mirroring in the x-axis, and separating the labels in post-processing might solve this problem, however, the MC is difficult to segment due to low contrast between the canal and surrounding tissue in the mandible and compromised the integrity of the tubular structure (46).

The worst-performing case in the test set was attributed to overall low scan quality and a unique post-maxillary advancement scan in the dataset. This scan resulted in artifacts not present in the training set, specifically due to the presence of four fixation plates at the maxillary surface.

Furthermore, the robustness of the segmentation model has been validated using the MICCAI 2024 ToothFairy2F and ToothFairy2P datasets (37, 38). Although the model's performance was statistically different for most labels in the three datasets, the metrics still indicated accurate results across different patient populations and imaging conditions. The validation is particularly significant given the wide range of artifacts present in the data, such as implants, crowns, and braces, which can complicate the segmentation process. The segmentation model shows the ability to maintain high performance despite these challenges. The model could be further improved by training on a wider variety of patient deformities, CBCT scanners, and imaging conditions.

In order for automatic landmark localization on CBCT to be implemented in daily clinical use, consistent and reproducible results are needed. The models trained in this study achieved clinically acceptable standards with errors below 2 mm (25). The three configurations (model_4mm, 5mm, and 6mm) achieved an SDR < 2mm, of around 80% for all landmarks, indicating the feasibility of automated landmark localization using a segmentation approach. The 4 mm model achieved the highest accuracy, with an SDR < 2 mm of 81.8% and an MRE of 0.98 [0.83-1.23].

Dental landmarks were located with the highest overall accuracy with a MRE of 0.75 [0.47 - 1.07] and a SDR of 96.9% and surgical landmarks were located with the least accurate results with MRE 1.37 [0.81 - 2.09] and SDR of 71.3% using model_4mm. The location of the landmark could be the reason for this as the dental landmarks were located at distinct locations, such as crown tips. However most surgical landmarks such as the Zyg-L & R and the Inf B-L & R are located along broad curvatures and indistinct boundaries and therefore more erroneous for identification (47). This was also seen for the Pog and Inf O-L & R landmarks on the mandible and maxilla, in contrast to the accurate localization of the ANS and PNS.

Furthermore the Nasion was occasionally missed during prediction due to its low occurrence in the dataset (n = 12). Additionally similar to the MC in the segmentation model, differences in MRE were observed between bilateral landmarks, such as Go-L & R and Condyle L & R. This could also be caused by the disabled mirroring in the x-axis for data augmentation.

Although some predicted landmark segmentations were not perfect spheres, as illustrated in the worst-case example in **Appendix F**, the center of mass proved to be a robust method for managing these irregular shapes. Segmenting a sphere around landmarks introduced an additional step compared to direct object detection, but the results, after translating the volume back to the center coordinate using the center of mass, were comparable to object detection studies, achieving MREs below 2 mm (17-31). This suggests that segmentation could be a viable option for landmark localization, similar to the findings of Weingart et al., who used 2 mm radial segmentations of landmarks on CT scans (27).

It should be noted that high segmentation overlap and low landmark localization error levels do not necessarily translate to clinical usable segmentations and landmarks. Segmentations and landmarks must be carefully reviewed and adjusted if needed to ensure their clinical applicability.

Overall, the use of nnU-Net has proven to be very suitable for automating segmentation and landmark localization in surgical planning. The models were accurate and its extensive documentation, active maintenance, and frequent usage make it a reliable and easily reproducible tool.

7.1 Limitations

The limitations in this study are multifaceted and could impact the overall accuracy and efficiency of the outcomes.

Firstly, this was a single-center study with anonymized data and unknown patient characteristics. No evaluation of different CBCT scanner models and imaging settings on the segmentation performance was included. Variations in scanner hardware and acquisition protocols can introduce inconsistencies in image quality, potentially affecting the generalizability of the model. The worst-performing case in the test set highlights the importance of including a diverse range of imaging conditions in the training dataset to improve model robustness.

Secondly, the workflow required segmentations made in Materialise software to be exported as STL files, then imported into Slicer for conversion back to masks, and finally saved as NIfTI files. This multi-step conversion process introduced potential sources of error. Although no significant errors were observed, the process inherently risks introducing small inaccuracies that could affect the overall segmentation quality.

7.2 Recommendations

To enhance the robustness and generalizability of the models, it is recommended to incorporate a broader range of scans from diverse patient populations, various scan modalities, and different fields of view (FoV). This expansion will ensure that the models are exposed to a wider variety of anatomical variations and imaging conditions, thereby improving their performance across different clinical scenarios.

Furthermore, the planning pipeline should be extended to include a comprehensive 3D cephalometric analysis with a wide variety of measurements implemented. This extension would allow for a more detailed and accurate assessment of craniofacial structures. As suggested by Gateno et al., integrating such a detailed cephalometric analysis can significantly enhance the assessment of deformities (48). Furthermore, they proposed a detailed method for consistent 3D assessment of the CMF anatomy, including size, shape, position, and orientation of the different facial units, which might be suitable to incorporate in the proposed workflow in this study. Evaluating the influence of landmarking inaccuracies on clinical implications, such as diagnosis and treatment planning, is important for this integration. The fact that landmarks are located in free space rather than on the surface of created segmentation for the anatomy might also influence landmarking accuracy, either positively or negatively. Future studies should focus on quantifying these effects and developing strategies to mitigate any negative impacts, thereby ensuring the automated workflow remains reliable and clinically relevant.

To further validate the efficacy of the proposed nnU-Net-based workflow, it should be implemented and tested in a clinical setting. This will help determine if it indeed enhances efficiency and maintains acceptable accuracy in real-world craniomaxillofacial surgical planning.

8. Conclusion

In conclusion, this study evaluated the use of nnU-Net for developing an automated segmentation and landmark localization workflow in craniomaxillofacial surgery planning. The results demonstrated high accuracy in segmenting the main CMF anatomical structures and achieving precise landmark localization. This automated approach shows great potential to enhance the efficiency of CMF surgical planning, reducing the need for manual intervention and improving overall workflow efficiency.

9. References

1. Cheong Y-W, Lo L-J. Facial asymmetry: etiology, evaluation, and management. Chang Gung Med J. 2011;34(4):341-51.
2. Ong T, Banks R, Hildreth A. Surgical accuracy in Le Fort I maxillary osteotomies. British journal of oral and maxillofacial surgery. 2001;39(2):96-102.
3. Buchanan EP, Hyman CH, editors. LeFort I osteotomy. Seminars in plastic surgery; 2013: Thieme Medical Publishers.
4. Akhil G, Kumar KPS, Raja S, Janardhanan K. Three-dimensional assessment of facial asymmetry: a systematic review. Journal of Pharmacy and Bioallied Sciences. 2015;7(Suppl 2):S433-S7.
5. Severt T, Proffit W. The prevalence of facial asymmetry in the dentofacial deformities population at the University of North Carolina. The International journal of adult orthodontics and orthognathic surgery. 1997;12(3):171-6.
6. Troulis M, Everett P, Seldin E, Kikinis R, Kaban L. Development of a three-dimensional treatment planning system based on computed tomographic data. International journal of oral and maxillofacial surgery. 2002;31(4):349-57.
7. Caloss R, Atkins K, Stella JP. Three-dimensional imaging for virtual assessment and treatment simulation in orthognathic surgery. Oral and maxillofacial surgery clinics of North America. 2007;19(3):287-309.
8. Mazzoni S, Bianchi A, Schiariti G, Badiali G, Marchetti C. Computer-aided design and computer-aided manufacturing cutting guides and customized titanium plates are useful in upper maxilla waferless repositioning. Journal of Oral and Maxillofacial Surgery. 2015;73(4):701-7.
9. McDonnell J, McNeill R, West R. Advancement genioplasty: a retrospective cephalometric analysis of osseous and soft tissue changes. Journal of Oral Surgery (American Dental Association: 1965). 1977;35(8):640-7.
10. Monson LA, editor Bilateral sagittal split osteotomy. Seminars in plastic surgery; 2013: Thieme Medical Publishers.
11. Dot G, Schouman T, Dubois G, Rouch P, Gajny L. Fully automatic segmentation of craniomaxillofacial CT scans for computer-assisted orthognathic surgery planning using the nnU-Net framework. European radiology. 2022:1-10.
12. Dot G, Rafflenbeul F, Kerbrat A, Rouch P, Gajny L, Schouman T. Three-dimensional cephalometric landmarking and Frankfort horizontal plane construction: reproducibility of conventional and novel landmarks. Journal of Clinical Medicine. 2021;10(22):5303.
13. Hassan B, Nijkamp P, Verheij H, Tairie J, Vink C, van der Stelt P, et al. Precision of identifying cephalometric landmarks with cone beam computed tomography in vivo. The European Journal of Orthodontics. 2013;35(1):38-44.
14. Nogueira-Reis F, Morgan N, Suryani IR, Tabchoury CPM, Jacobs R. Full virtual patient generated by artificial intelligence-driven integrated segmentation of craniomaxillofacial structures from CBCT images. Journal of Dentistry. 2024;141:104829.
15. Steinhuber T, Brunold S, Gärtner C, Offermanns V, Ulmer H, Ploder O. Is virtual surgical planning in orthognathic surgery faster than conventional planning? A time and workflow analysis of an office-based workflow for single-and double-jaw surgery. Journal of Oral and Maxillofacial Surgery. 2018;76(2):397-407.
16. Torosdagli N, Liberton DK, Verma P, Sincan M, Lee J, Pattanaik S, et al., editors. Robust and fully automated segmentation of mandible from CT scans. 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017); 2017: IEEE.
17. Chen R, Ma Y, Chen N, Liu L, Cui Z, Lin Y, et al. Structure-Aware Long Short-Term Memory Network for 3D Cephalometric Landmark Detection. IEEE Trans Med Imaging. 2022;41(7):1791-801.

18. Chen X, Lian C, Deng HH, Kuang T, Lin HY, Xiao D, et al. Fast and Accurate Craniomaxillofacial Landmark Detection via 3D Faster R-CNN. IEEE Trans Med Imaging. 2021;40(12):3867-78.
19. Dot G, Schouman T, Chang S, Rafflenbeul F, Kerbrat A, Rouch P, et al. Automatic 3-Dimensional Cephalometric Landmarking via Deep Learning. J Dent Res. 2022;101(11):1380-7.
20. Gillot M, Miranda F, Baquero B, Ruellas A, Gurgel M, Al Turkestani N, et al. Automatic landmark identification in cone-beam computed tomography. Orthod Craniofac Res. 2023;26(4):560-7.
21. Hwang IK, Kang SR, Yang S, Kim JM, Kim JE, Huh KH, et al. SinusC-Net for automatic classification of surgical plans for maxillary sinus augmentation using a 3D distance-guided network. Sci rep. 2023;13(1):11653.
22. Kang SH, Jeon K, Kang SH, Lee SH. 3D cephalometric landmark detection by multiple stage deep reinforcement learning. Sci Rep. 2021;11(1):17509.
23. Lang Y, Lian C, Xiao D, Deng H, Thung KH, Yuan P, et al. Localization of Craniomaxillofacial Landmarks on CBCT Images Using 3D Mask R-CNN and Local Dependency Learning. IEEE Trans Med Imaging. 2022;41(10):2856-66.
24. Lu G, Shu H, Bao H, Kong Y, Zhang C, Yan B, et al. CMF-Net: craniomaxillofacial landmark localization on CBCT images using geometric constraint and transformer. Phys Med Biol. 2023;68(9).
25. Palazzo S, Bellitto G, Prezzavento L, Rundo F, Bagci U, Giordano D, et al. Deep Multi-stage Model for Automated Landmarking of Craniomaxillofacial CT Scans. 2021:9982-7.
26. Tao L, Li M, Zhang X, Cheng M, Yang Y, Fu Y, et al. Automatic craniomaxillofacial landmarks detection in CT images of individuals with dentomaxillofacial deformities by a two-stage deep learning model. BMC Oral Health. 2023;23(1):876.
27. Weingart JV, Schlager S, Metzger MC, Brandenburg LS, Hein A, Schmelzeisen R, et al. Automated detection of cephalometric landmarks using deep neural patchworks. Dentomaxillofac Radiol. 2023;52(6):20230059.
28. Xu M, Liu B, Luo Z, Ma H, Sun M, Wang Y, et al. Using a New Deep Learning Method for 3D Cephalometry in Patients With Cleft Lip and Palate. J Craniofac Surg. 2023;34(5):1485-8.
29. Xu M, Liu B, Luo Z, Sun M, Wang Y, Yin N, et al. Using a New Deep Learning Method for 3D Cephalometry in Patients With Hemifacial Microsomia. Ann Plast Surg. 2023;91(3):381-4.
30. Zhang J, Liu M, Wang L, Chen S, Yuan P, Li J, et al. Joint Craniomaxillofacial Bone Segmentation and Landmark Digitization by Context-Guided Fully Convolutional Networks. Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv. 2017;10434:720-8.
31. Zhang J, Liu M, Wang L, Chen S, Yuan P, Li J, et al. Context-guided fully convolutional networks for joint craniomaxillofacial bone segmentation and landmark digitization. Med Image Anal. 2020;60:101621.
32. Torosdagli N, Liberton DK, Verma P, Sincan M, Lee JS, Bagci U. Deep Geodesic Learning for Segmentation and Anatomical Landmarking. IEEE Trans Med Imaging. 2019;38(4):919-31.
33. Isensee F, Jaeger PF, Kohl SA, Petersen J, Maier-Hein KH. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods. 2021;18(2):203-11.
34. Isensee F, Wald T, Ulrich C, Baumgartner M, Roy S, Maier-Hein K, et al. nnu-net revisited: A call for rigorous validation in 3d medical image segmentation. arXiv preprint arXiv:240409556. 2024.

35. Trzeciak M, Michalczak M, Niziolek M, Lipski M, Musiał A, Skrzat J, et al. The surgical anatomy of the inferior alveolar nerve: a meta-analysis with clinical implications. Folia Morphologica. 2023.

36. Fedorov A, Beichel R, Kalpathy-Cramer J, Finet J, Fillion-Robin J-C, Pujol S, et al. 3D Slicer as an image computing platform for the Quantitative Imaging Network. Magnetic resonance imaging. 2012;30(9):1323-41.

37. Cipriano M, Allegretti S, Bolelli F, Di Bartolomeo M, Pollastri F, Pellacani A, et al. Deep segmentation of the mandibular canal: a new 3D annotated dataset of CBCT volumes. IEEE Access. 2022;10:11500-10.

38. Cipriano M, Allegretti S, Bolelli F, Pollastri F, Grana C, editors. Improving segmentation of the inferior alveolar nerve through deep label propagation. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2022.

39. Nikolov S, Blackwell S, Zverovitch A, Mendes R, Livne M, De Fauw J, et al. Clinically applicable segmentation of head and neck anatomy for radiotherapy: deep learning algorithm development and validation study. Journal of medical Internet research. 2021;23(7):e26151.

40. Reinke A, Tizabi MD, Sudre CH, Eisenmann M, Rädsch T, Baumgartner M, et al. Common limitations of image processing metrics: A picture story. arXiv preprint arXiv:210405642. 2021.

41. Antonelli M, Reinke A, Bakas S, Farahani K, Kopp-Schneider A, Landman BA, et al. The medical segmentation decathlon. Nature communications. 2022;13(1):4128.

42. Abdolali F, Zoroofi RA, Abdolali M, Yokota F, Otake Y, Sato Y. Automatic segmentation of mandibular canal in cone beam CT images using conditional statistical shape model and fast marching. International journal of computer assisted radiology and surgery. 2017;12:581-93.

43. Jaskari J, Sahlsten J, Järnstedt J, Mehtonen H, Karhu K, Sundqvist O, et al. Deep learning method for mandibular canal segmentation in dental cone beam computed tomography volumes. Scientific reports. 2020;10(1):5842.

44. Kuehle R, Scheurer M, Bouffleur F, Fuchs J, Engel M, Hoffmann J, et al. Accuracy of Patient-Specific Implants in Virtually Planned Segmental Le Fort I Osteotomies. Journal of Clinical Medicine. 2023;12(18):6038.

45. Rios O, Lerhe B, Chamorey E, Savoldelli C. Accuracy of segmented le fort I osteotomy with virtual planning in orthognathic surgery using patient-specific implants: A case series. Journal of Clinical Medicine. 2022;11(19):5495.

46. Zhao H, Chen J, Yun Z, Feng Q, Zhong L, Yang W. Whole mandibular canal segmentation using transformed dental CBCT volume in Frenet frame. Heliyon. 2023;9(7).

47. Sam A, Currie K, Oh H, Flores-Mir C, Lagravere-Vich M. Reliability of different three-dimensional cephalometric landmarks in cone-beam computed tomography: A systematic review. The Angle Orthodontist 2019;89:317;32

48. Gateno J, Xia JJ, Teichgraeber JF. New 3-dimensional cephalometric analysis for orthognathic surgery. Journal of oral and maxillofacial surgery. 2011;69(3):606-22.

10. Appendices

Appendix A – Segmentation protocol

Erasmus MC, Department of Oral and Maxillofacial Surgery

Semi-Automatic Craniomaxillofacial Segmentation of CBCT images using Mimics 26.0

Killian Zijlstra, Bram Roumen
e-mail: k.zijlstra@erasmusmc.nl, NetID: 4652460
e-mail: b.roumen@erasmusmc.nl, NetID: 4610474

July 9, 2024

Supervisors:

Prof. Dr. E.B. Wolvius
Dr. B.M.W. Cornelissen
Dr. ir. T. van Walsum
Dr. R. Su

Versions:
Document drafted on Nov. 25, 2022, by K. Zijlstra
Created the Segmentation Protocol for the mandible in Mimics 24.0
Document revised on Mar. 11, 2024, by B. Roumen
Extended the segmentation protocol for maxilla, mandibular canals and skin segmentations. Adjusted the protocol for applicability in Mimics 26.0

Abstract

Background: During the last decades, three-dimensional (3D) virtual planning in orthognathic surgery has become the golden standard. Cone beam computed tomography (CBCT) is most commonly used as imaging modality for 3D virtual planning. In order to create the 3D virtual planning, mandible, maxilla, mandibular canal and skin segmentations are required.
Objective: This article aimed to create a quick and easy protocol for segmenting the mandible, maxilla and cranial bones, mandibular canals and skin in CBCT scans using Mimics 26.0.
Method: Two experienced technical physicians provided segmentation training. After completion of the training six test CBCT images were segmented and reviewed by the technical physician. Next, the protocol was drawn-up and the rest of the segmentations were performed.
Results: A total of 57 patient were segmented using this protocol. Each segmentation takes approximately one and a half hour to complete.
Conclusion: A step by step protocol was constructed for the segmentation of CBCT images using Mimics 26.0. The resulting segmentations are watertight STL files that can be used for a variety of future studies.

Keywords

Craniomaxillofacial, Segmentation, Cone-Beam Computed Tomography (CBCT)

Introduction

Orthognathic surgery is a sub-specialty of oral and maxillofacial surgery that corrects disproportion of the facial skeleton. Surgery can be performed to create or restore function and aesthetics [1]. Traditionally, conventional two-dimensional (2D) radiographs and manual model surgery were the standard for orthognathic surgery. In the last decades, three-dimensional (3D) pre-operative planning is commonly used to provide accurate and reproducible treatment planning. Compared to conventional treatment planning, 3D planning reduces operative time and costs [2-4].

Cone-beam computed tomography (CBCT) can be used as imaging modality to create the virtual 3D planning [5,6]. CBCT has lower costs and reduces radiation exposure compared to regular computed tomography (CT) [7].

However, CBCT has a few disadvantages compared to regular CT such as higher noise levels and lower contrast. Also, the gray values of a CBCT scanner are not absolute as in regular CT with fixed Hounsfield units (HU). This means that tissue with a similar density can be assigned a different gray value based on their position in the field of view of the scanner [5].

Most (semi-)automatic segmentation programs are based on thresholding. Due to the non absolute gray values of CBCT, threshold based segmentation becomes more difficult since the gray level and contrast decrease towards the condyles [5]. Therefore, it is impossible to find one and the same threshold for all bone tissue to be segmented resulting in a labor intensive and time consuming manual segmentation process [8].

In order to create segmentations using CBCT images for virtual preoperative planning, manual or semi-automatic segmentation is currently used in clinical practice. Since no segmentation protocol for Mimics 26.0 is described in the literature, the goal of this article is to provide a quick and easy protocol for semi-automatic segmentation of the craniomaxillofacial anatomy on CBCT scans using Mimics 26.0.

Method

Under guidance of two experienced technical physicians segmentation training was provided using Mimics 26.0. The training consisted of instructions how to use Mimics 26.0 segmentation tools and showing their current workflow for bone surface segmentations used in clinical practice.

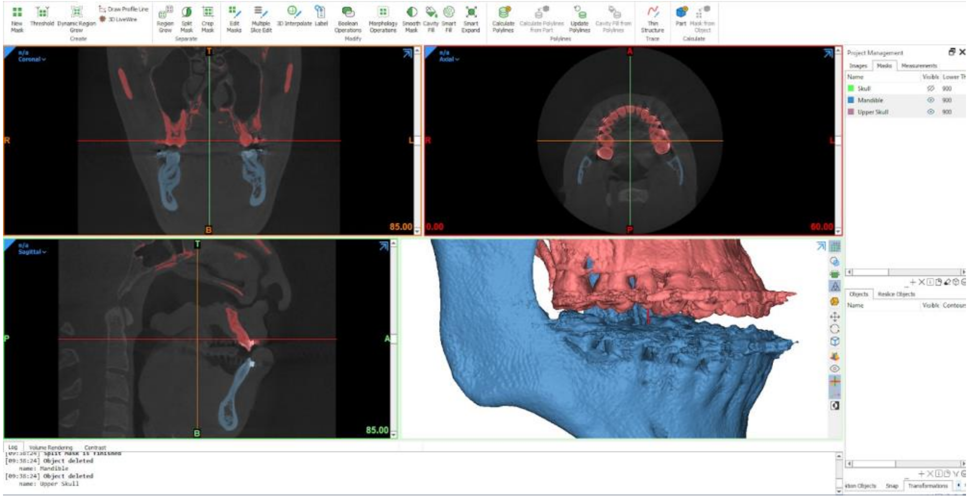
Next, six CBCT images were segmented for training and reviewed for sufficiency by the technical physicians. Areas of attention were pointed out after which the training was completed and the construction of the protocol could begin.

Protocol

Below a protocol is described for segmentations of the mandible, the maxilla including the cranial bones. For the mandible a total of 5 different masks will be created. The mandibular body, mandibular dentition, mandibular ramus and finally masks for the left and right condyles. The maxilla will consist of three different masks. The frontal upper skull, upper dentition and the distal upper skull. Furthermore the mandibular canals and skin will be segmented.

Mandibular Body and Frontal Skull

1. Import the CBCT scan by selecting ”New Project”.
2. Adjust the contrast by holding the right mouse button until the mandibular body is clearly visible in the sagittal plane or by selecting the ”contrast” tab in the bottom of the screen. Drag the circle to around 1900 HU.
3. Go to ”Segment” and choose ”New Mask”. Adjust the lower threshold until the contour of the mandibular body is highlighted (around 825 HU). Set the upper threshold at its upper limit. Name this mask ”Skull”.
4. Create the 3D model of the mask ”Skull” by clicking the upper right icon of the 3D preview window.
5. To split the mask go to ”Region Grow” if Mandible and Maxilla are not connected. Otherwise got to ”Split Mask” Select the mask ”Skull”, rename ”Region A” ”Mandible” and ”Region B” ”Upper Skull”. Carefully use the brush in all three planes to split the mask ”Skull” and click ”OK”.



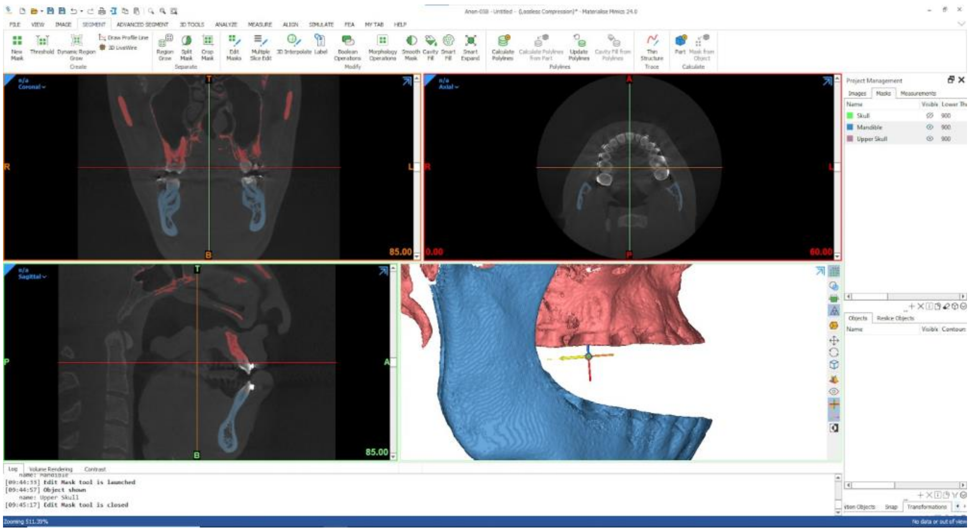
Name	Visible	Lower Th
🟩 Skull		900
🟦 Mandible		900
🟪 Upper Skull		900

```
[09:38:24] Split Mask is finished
[09:38:24] Object deleted
    name: Mandible
[09:38:24] Object deleted
    name: Upper Skull
```

If the patient has braces or other highly dense objects around the dentition that causes noise, follow steps 6-12. Otherwise continue at step 13

Dentition

6. Rotate the 3D preview of the mask ”Mandible” to create a side view. Make sure the most distal molars directly overlap.
7. Go to ”Edit Masks” Select the mask ”Mandible”, choose ”Remove” and ”Lasso”. Carefully cut out the braces and noise caused by the dense objects by cutting directly around the braces. Try to spare the roots of the dentition.
8. Repeat the previous step for the mask ”Upper Skull”.

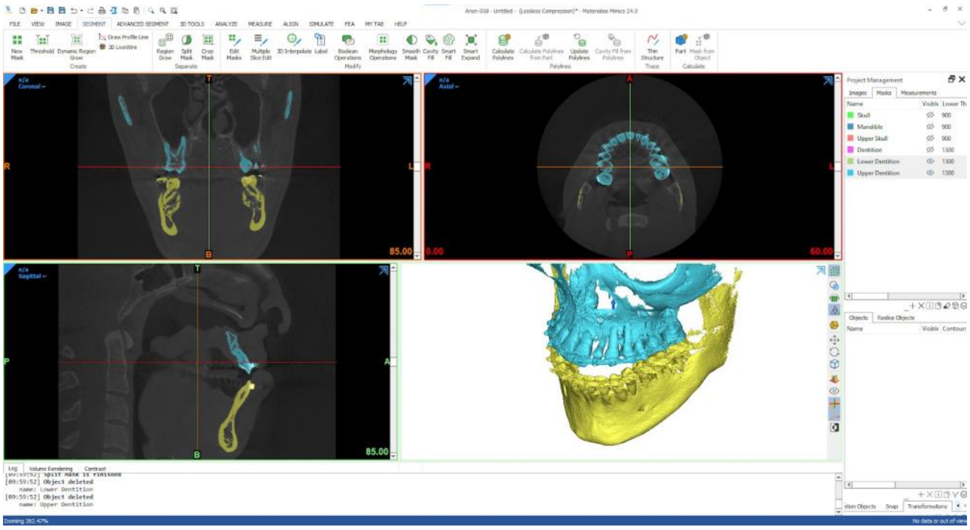


Project Management

Name	Visible	Lower Th
🟩 Skull		900
🟦 Mandible		900
🟪 Upper Skull		900

```
    name: Mandible
[09:44:33] Edit Mask tool is launched
[09:44:57] Object shown
    name: Upper Skull
[09:45:17] Edit Mask tool is closed
```

9. Go to "New Mask". Set the lower threshold so the outlines of the dentition are completely visible and the noise caused by dense objects is minimal. This lower threshold will be higher than the previous mask "Skull". Set the upper threshold at its upper limit. Name this mask "Dentition".



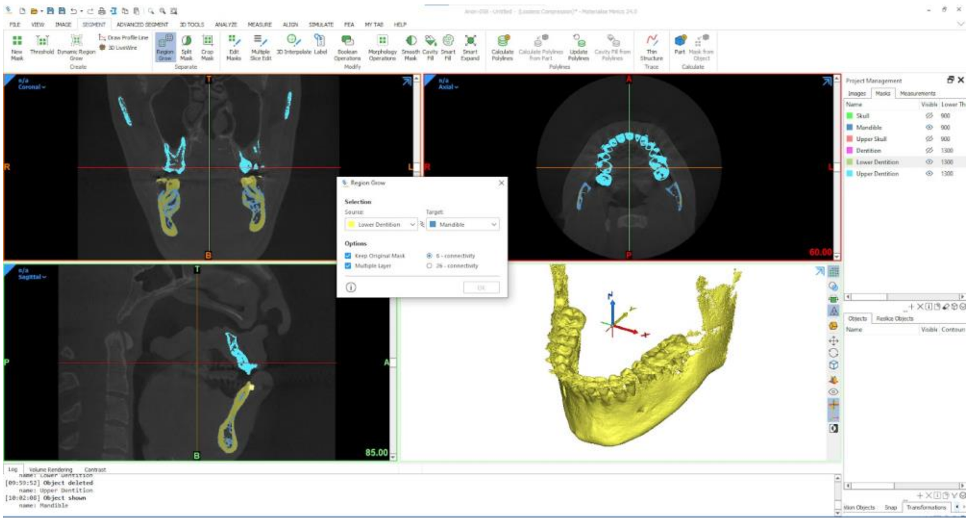
Project Management

Name	Visible	Lower Th
🟩 Skull		900
🟦 Mandible		900
🟥 Upper Skull		900
🟪 Dentition		1300
🟩 Lower Dentition		1300
🟦 Upper Dentition		1300

```
[09:59:52] Split Mask is finished
[09:59:52] Object deleted
    name: Lower Dentition
[09:59:52] Object deleted
    name: Upper Dentition
```

10. Go to "Split Mask" Select the mask "Dentition", rename "Region A" "Lower Dentition" and "Region B" "Upper Dentition". Carefully use the brush in all three planes to split the mask "Dentition" and click "OK".

11. Go to "Region Grow", set the mask "Lower Dentition" as "Source" and "Mandible" as "Target". Click in the 3D preview on the mask "Lower Dentition" To unite the masks "Lower Dentition" and "Mandible".



Region Grow

Selection

Source: Lower Dentition

Target: Mandible

Options

☑ Keep Original Mask

☑ Multiple Layer

● 6 - connectivity

○ 26 - connectivity

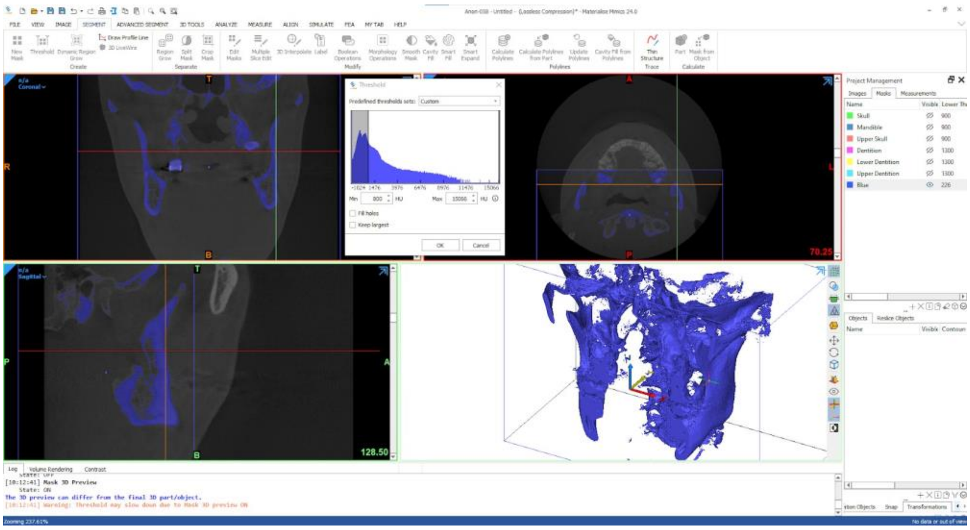
Name	Visible	Lower Th
Skull		900
Mandible	👁	900
Upper Skull		900
Dentition		1300
Lower Dentition	👁	1300
Upper Dentition	👁	1300

```
    name: Lower Dentition
[09:59:52] Object deleted
    name: Upper Dentition
[10:02:00] Object shown
    name: Mandible
```

12. Repeat the previous step to unite the masks ”Upper Dentition and ”Upper Skull”.

Mandibular Ramus and Distal Skull

13. Go to ”New Mask”, adjust the anterior border of the region of interest in the sagittal plane to ensure the most anterior coronoid process is just within the anterior border of the segmentation. Adjust the lower threshold until the ramus of the mandible is marked without much noise surrounding the ramus (around 650 HU). Set the upper threshold at its upper limit. Name the resulting mask ”Ramus and Upper Skull”.



Threshold

Predefined thresholds sets: Custom

Min 800 HU Max 15066 HU

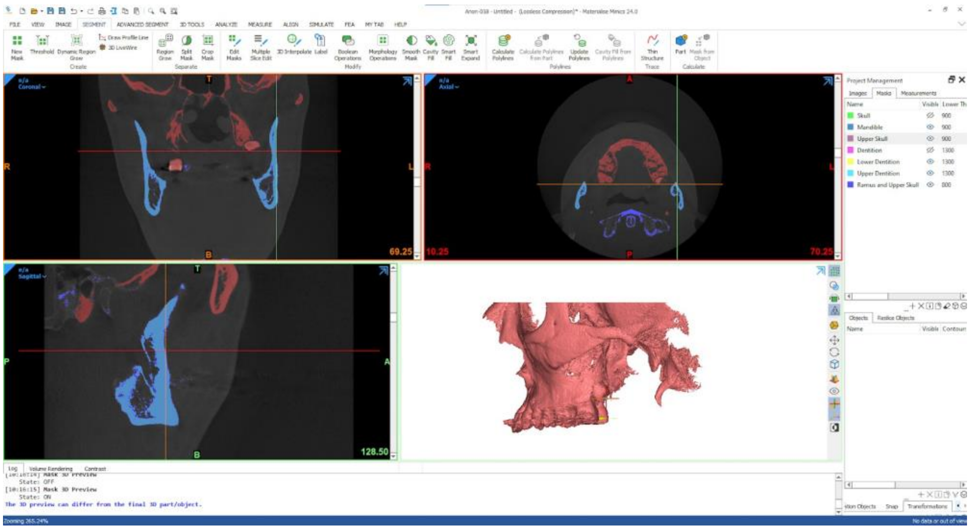
□ Fill holes

□ Keep largest

Name	Visible	Lower Th
Skull		900
Mandible		900
Upper Skull		900
Dentition		1300
Lower Dentition		1300
Upper Dentition		1300
Blue	👁	226

```
    State: OFF
[10:12:41] Mask 3D Preview
    State: ON
The 3D preview can differ from the final 3D part/object.
[10:12:41] Warning: Threshold may slow down due to Mask 3D preview ON
```

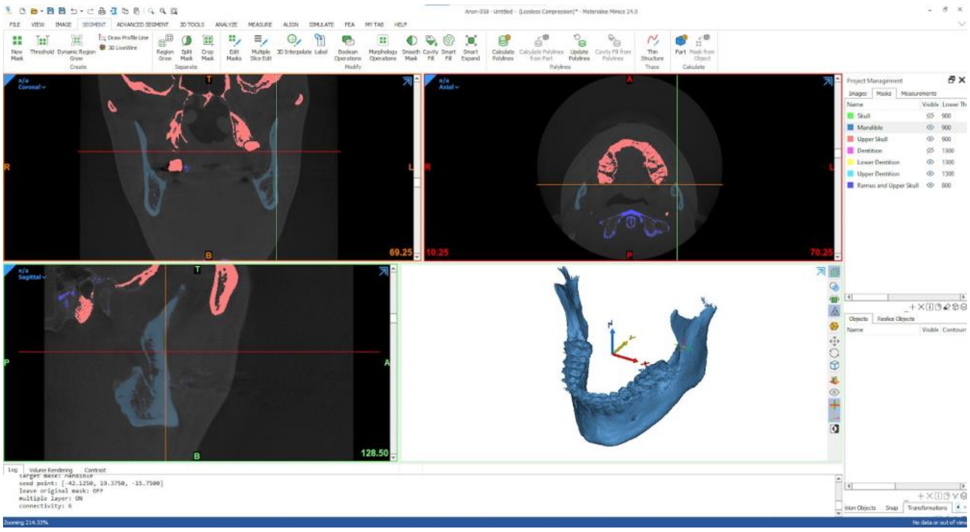
14. Go to ”Region Grow” set as source the mask ”Ramus and Upper Skull” and set as target ”Upper Skull”. Click on the temporal bone and left and right zygomatic bone to unite the masks in the mask named ”Upper
Skull”.



Name	Visible	Lower Th
Skull		900
Mandible		900
Upper Skull		900
Dentition		1300
Lower Dentition		1300
Upper Dentition		1300
Ramus and Upper Skull		800

```
[10:16:14] Mask 3D Preview
    State: OFF
[10:16:15] Mask 3D Preview
    State: ON
The 3D preview can differ from the final 3D part/object.
```

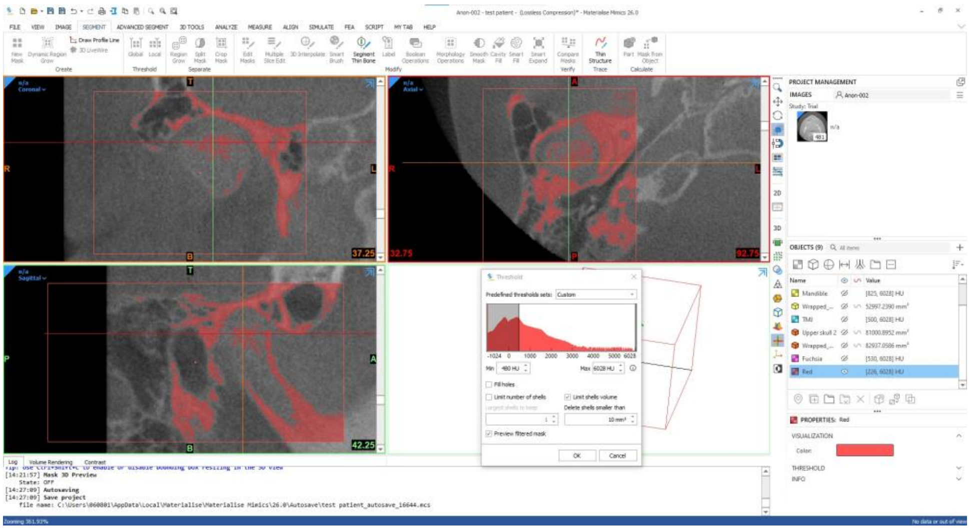
15. Go to ”Region Grow”, set as source the mask ”Ramus and Upper Skull”and set as target ”Mandible”. Click on the left and right ramus of the mandible to unite the masks in the mask named ”Mandible”.



```
    target mask: Mandible
    seed point: [-42.1250, 19.3750, -15.7500]
    leave original mask: OFF
    multiple layer: ON
    connectivity: 6
```

Temporomandibular joint (TMJ) and Mandibular Condyles

16. Go to ”New Mask”, adjust the borders of the segmentation in the axial,saggital and coronal view so that the zygomatic process of the temporal bone, the mandibular fossa of the temporal bone and the most caudal part of the sigmoid notch of the mandible are marked. Adjust the lower threshold so that the contour of the left condylar head is visible (480-540 HU). Set the upper threshold at its upper limit. Name this mask ”Condyle L”.



Threshold

Predefined thresholds sets: Custom

Min 480 HU   Max 6028 HU

☐ Fill holes

☐ Limit number of shells   ☑ Limit shells volume

Delete shells smaller than 10 mm³

☑ Preview filtered mask

OBJECTS (9)

Name	Value
Mandible	[825, 6028] HU
Wrapped_...	52997.2390 mm³
TMJ	[500, 6028] HU
Upper skull 2	81000.8952 mm³
Wrapped_...	82937.0586 mm³
Fuchsia	[530, 6028] HU
Red	[226, 6028] HU

PROPERTIES: Red

```
[14:21:57] Mask 3D Preview
    State: OFF
[14:27:09] Autosaving
[14:27:09] Save project
    file name: C:\Users\060001\AppData\Local\Materialise\Materialise Mimics\26.0\Autosave\test patient_autosave_16644.mcs
```

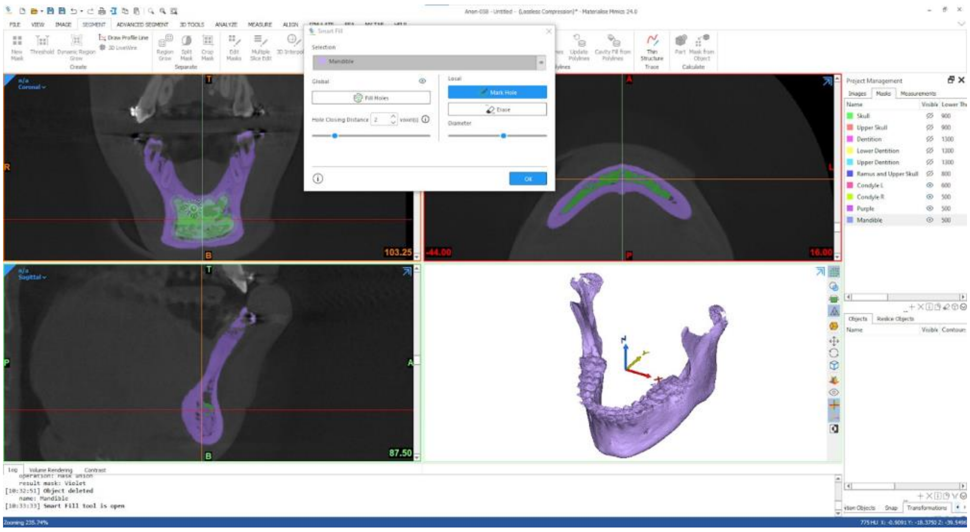
17. Go to ”Region Grow”, set as source the mask ”Condyle L” and set as target ”Mandible”. Click on the condyle to unite the masks in the mask named ”Mandible”. Use ”Split Mask” if the condyle and mandibular fossa of the temporal bone are connected.

18. Go to ”Region Grow”, set as source the mask ” Condyle L” and set as target ”Upper skull”. Click on the zygomatic arch to unit the masks in the mask named ”Upper skull”.

19. Repeat the previous steps 17 & 18 for the right condyle. Name the resulting mask ”Condyle R”.

Fill Holes and Finish Mandibular Mask

20. Go to ”Smart Fill”, use ”Mark Hole” and fill up the hollow structures within the condylar head, ramus, mandibular body and the teeth by using all three views. Make sure to leave the enterance of the mental foramen and mandibular foramen open. Name the resulting mask ”Mandible” and remove the previous mask ”Mandible”.



Smart Fill

Selection: Mandible

Hole Closing Distance 2 voxel(s)

Name	Visible	Lower Th
Skull		900
Upper Skull		900
Dentition		1300
Lower Dentition		1300
Upper Dentition		1300
Ramus and Upper Skull		800
Condyle L		600
Condyle R		500
Purple		500
Mandible		500

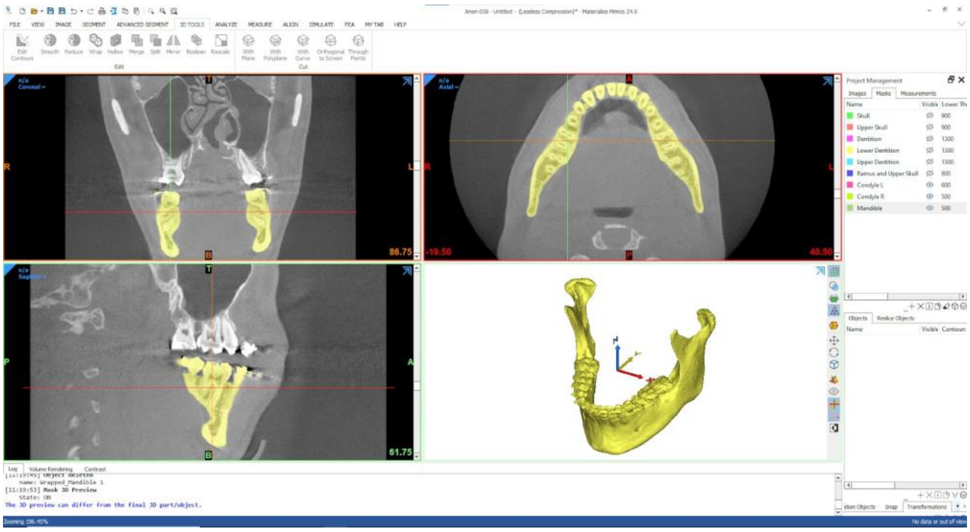
```
    operation: Mask union
    result mask: Violet
[10:32:51] Object deleted
    name: Mandible
[10:33:33] Smart Fill tool is open
```

21. Check all three views if the contours of the mandible, ramus, condylar heads and teeth (with braces if applicable) are correct. Adjust with ”Edit Mask” if necessary.

22. Select the mask ”Mandible” and click on ”Smooth Mask”.

23. Go to ”Region Grow”, select as source and as target the mask ”Mandible”. This ensures that the mask consists of one part.

24. Go to ”Smart Fill”, select ”Fill Holes”, set ”Hole Closing Distance” at 2 voxels. Click on ”OK”. Name the resulting mask ”Mandible” and delete the previous mask ”Mandible”.



Project Management

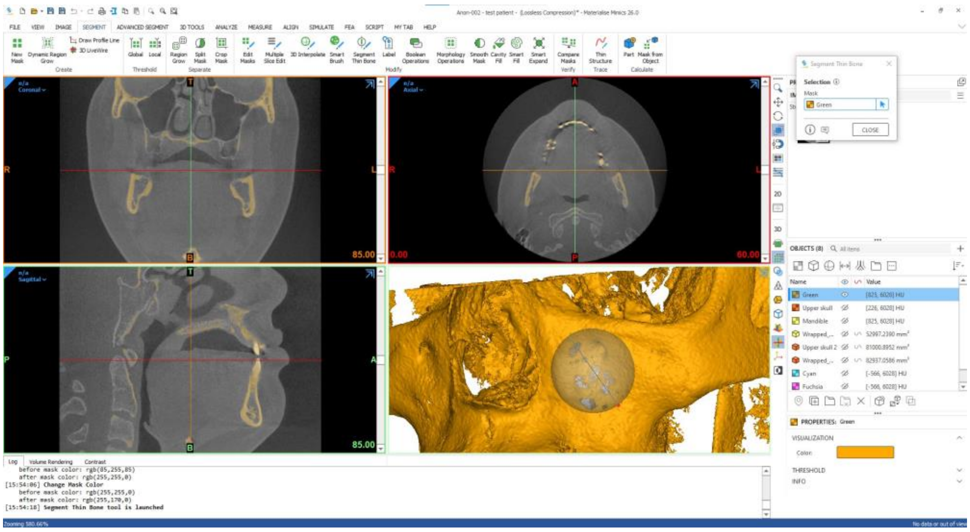
Name	Visible	Lower Th
Skull		900
Upper Skull		900
Dentition		1300
Lower Dentition		1300
Upper Dentition		1300
Ramus and Upper Skull		800
Condyle L		600
Condyle R		500
Mandible		500

```
[11:19:45] Object deleted
    name: Wrapped_Mandible 1
[11:19:53] Mask 3D Preview
    State: ON
The 3D preview can differ from the final 3D part/object.
```

Add thin bones and Finish Upper skull Mask

25. Select the mask ”Upper skull” and Go to ”Smart Fill”, use ”Mark Hole” and fill up the hollow structures in the maxilla. Do not fill the sinuses. Name the resulting mask ”Upper skull” and remove the previous mask ”Upper skull”.

26. Check the anterior surface of the maxilla in the ”Upper skull” mask for holes. If holes are present select ”Segment Thin Bone” and draw a line in 2D or 3D to indicate the hole.



OBJECTS (8)

Name	Value
Green	[825, 6028] HU
Upper skull	[226, 6028] HU
Mandible	[825, 6028] HU
Wrapped_...	52997.2390 mm²
Upper skull 2	81000.8952 mm²
Wrapped_...	82937.0586 mm²
Cyan	[-566, 6028] HU
Fuchsia	[-566, 6028] HU

PROPERTIES: Green

```
    before mask color: rgb(85,255,85)
    after mask color: rgb(255,255,0)
[15:54:06] Change Mask Color
    before mask color: rgb(255,255,0)
    after mask color: rgb(255,170,0)
[15:54:18] Segment Thin Bone tool is launched
```

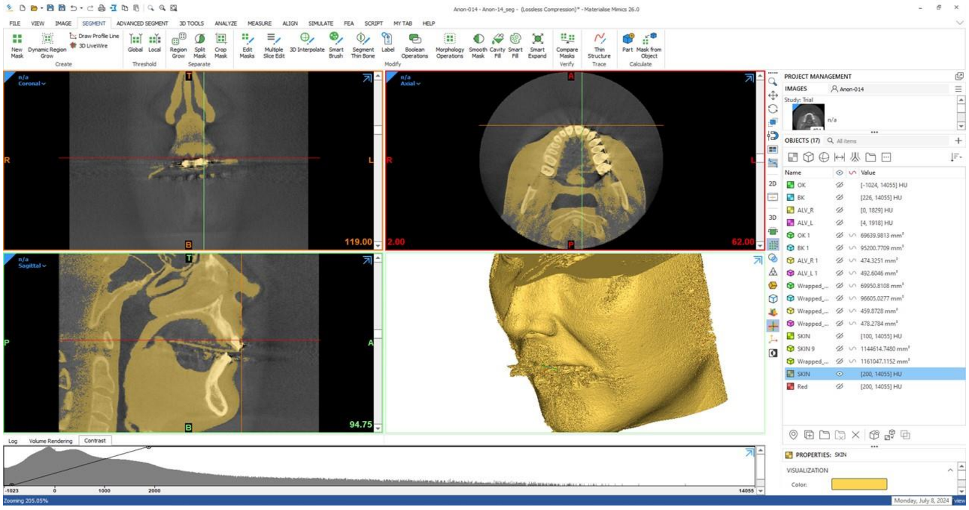
27. Select the mask ”Upper skull” and click on ”Smooth Mask”.

28. Go to ”Region Grow”, select as source and as target the mask ”Upper skull”. This ensures that the mask consists of one part.

29. Go to ”Smart Fill”, select ”Fill Holes”, set ”Hole Closing Distance” at2 voxels. Click on ”OK”. Name the resulting mask ”Upper skull” and delete the previous mask ”Upper skull”.

Create skin segmentation

30. Go to ”Segment” and choose ”New Mask”. Adjust the lower threshold until most of the soft tissue is included and the noise of the background is minimal (around 200 HU). Set the upper threshold at its upper limit. Name this mask ”skin”.
31. Remove most of the noise from the mask using the ”Region Grow” functionality.

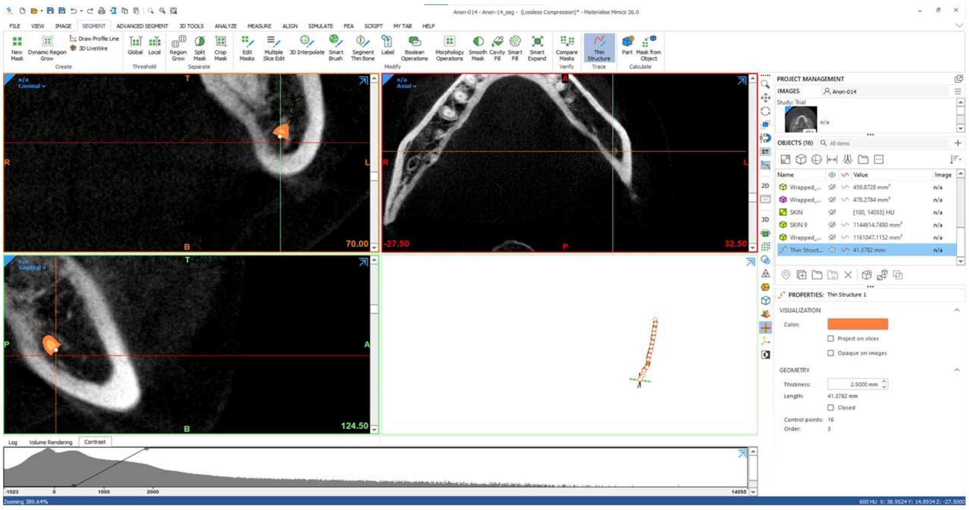


Name	Value
OK	[-1024, 14055] HU
BK	[226, 14055] HU
ALV_R	[0, 1829] HU
ALV_L	[4, 1918] HU
OK 1	69639.9813 mm³
BK 1	95200.7709 mm³
ALV_R 1	474.3251 mm³
ALV_L 1	492.6046 mm³
Wrapped_...	69950.8108 mm³
Wrapped_...	96605.0277 mm³
Wrapped_...	459.8728 mm³
Wrapped_...	478.2784 mm³
SKIN	[100, 14055] HU
SKIN 9	1144614.7480 mm³
Wrapped_...	1161047.1152 mm³
SKIN	[200, 14055] HU
Red	[200, 14055] HU

32. Go to ”Smart Fill” and use ”Fill holes” with hole closing distance set to2 voxels.
33. Remove any of the remaining noise using the ”Lasso” under ”Edit mask” by carefully selecting the noise in either the 3D viewer or the 2D views.

Create mandibular canal segmentations

34. For segmenting both the mandibular canal left and right, scroll to the entrance of the inferior alveolar nerve into the mandible. Go to ”Thin Structure” and select the center of the mandibular canal every 5 - 10 slices in the mandibular ramus, where the canal is mainly vertical until the apex of the third molar is reached. To better visualize the canal through the body of the mandible the contrast could be adjusted for better visualization. Follow the contour through the mandibular body. By using ”Shift + left mouse click” the other views automatically jump to the same slice, to provide better visualization. Press ”Esc” when the last point is placed to create the object. Do this for both the left and right mandibular canal.



PROJECT MANAGEMENT

IMAGES Anon-014

Study: Trial

OBJECTS (16)

Name			Value	Image
Wrapped_...			459.8728 mm³	n/a
Wrapped_...			478.2784 mm³	n/a
SKIN			[100, 14055] HU	n/a
SKIN 9			1144614.7480 mm³	n/a
Wrapped_...			1161047.1152 mm³	n/a
Thin Struct...			41.3782 mm	n/a

PROPERTIES: Thin Structure 1

VISUALIZATION

Color:

□ Project on slices

□ Opaque on images

GEOMETRY

Thickness: 2.5000 mm

Length: 41.3782 mm

□ Closed

Control points: 16

Order: 3

35. Go to the just created ”Thin structure” and under properties, check if thethickness under geometry is set to 2.5 mm.

36. Select ”Mask from Object” and select the thin line structure to create amask of the mandibular canal. Rename the segmentation to MC left or MC right.

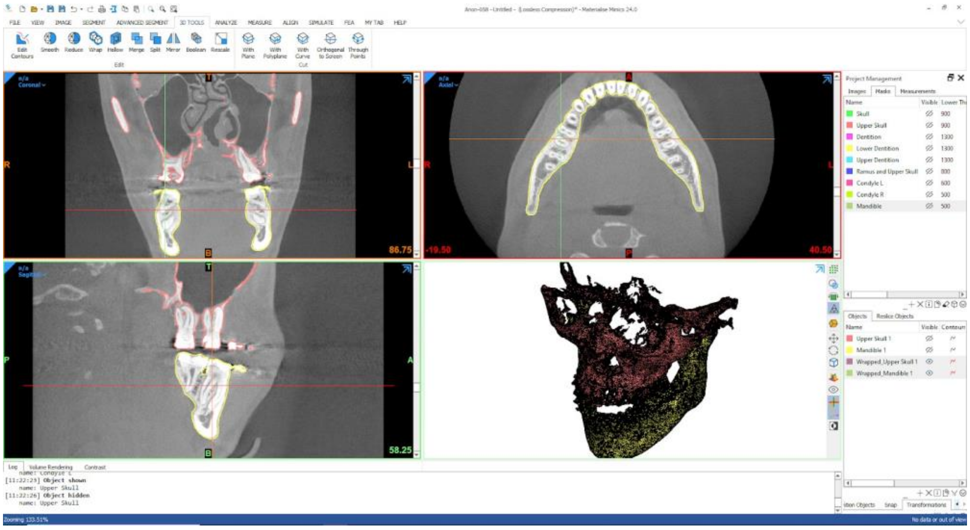
37. Select the mask of the Mandible and subtract the MC left and MC rightusing a boolean operation.

Create and export as STL file

38. Select the masks and right mouse click on the masks and go to ”CalculatePart”. Set ”Quality” to ”Optimal*” and click ”OK”. This creates the parts.

39. Go to ”3D Tools”, select ”Wrap” set ”Smallest detail” to 0.5 mm, ”GapClosing Distance” to 0.5 mm and click ”OK”.

40. Make the contours of the part of for example the ”Wrapped Mandible 1” visible by clicking on the ”Contours” icon in the ”Objects” screen. Follow the contours one final time in all three views to assess if they are sufficient.



Project Management

Images | Masks | Measurements

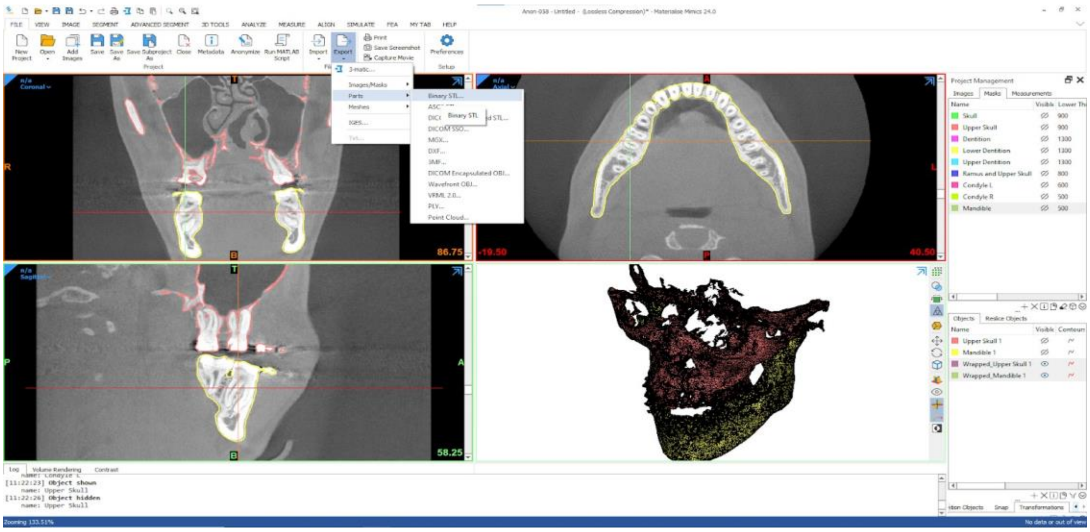
Name	Visible	Lower Th
Skull		900
Upper Skull		900
Dentition		1300
Lower Dentition		1300
Upper Dentition		1300
Ramus and Upper Skull		800
Condyle L		600
Condyle R		500
Mandible		500

Objects | Reslice Objects

Name	Visible	Contour
Upper Skull 1		
Mandible 1		
Wrapped_Upper Skull 1		
Wrapped_Mandible 1		

```
    name: Condyle L
[11:22:23] Object shown
    name: Upper Skull
[11:22:26] Object hidden
    name: Upper Skull
```

41. Go to ”File”, select ”Export”, select ”Parts”, ”Binary STL” and select allmasks. Choose the correct output file.



Name	Visible	Lower Th
🟩 Skull	⊘	900
🟥 Upper Skull	⊘	900
🟪 Dentition	⊘	1300
🟨 Lower Dentition	⊘	1300
🟦 Upper Dentition	⊘	1300
🟦 Ramus and Upper Skull	⊘	800
🟥 Condyle L	⊘	600
🟩 Condyle R	⊘	500
🟩 Mandible	⊘	500

Name	Visible	Contour
🟥 Upper Skull 1	⊘	~
🟨 Mandible 1	⊘	~
🟪 Wrapped_Upper Skull 1	👁	~
🟩 Wrapped_Mandible 1	👁	~

```
    name: Condyle L
[11:22:23] Object shown
    name: Upper Skull
[11:22:26] Object hidden
    name: Upper Skull
```

42. Go to ”File”, select ”Export”, select ”Parts”, ”Binary STL” and select allparts. Choose the correct output file.

Results

A total of sixty patients who were referred to the Erasmus Medical Center for orthognathic surgery from a period between March 2017 - September 2022 were included. Three patients were excluded due to missing CBCT data. This resulted in a total of 57 complete segmentations.

Creating the segmentations for every patient costed approximately 60 to 90 minutes each depending on the amount of slices, scan quality, contrast to noise ratio and deformities of the skeleton.

Discussion

The aim of this article was to provide a quick and easy to use protocol for CMF segmentations using the software Mimics 26.0.

During the segmentation of the data multiple problems emerged. First of all, nearly all patients either had braces, implants, dental filling or other dense object within the dental region which resulted in noise throughout the oral region. This protocol compensates for this noise by creating a separate mask for the dentition, but often the noise is still visible in other parts of the skull and mandible.

Furthermore the HU values of the CBCT images have higher contrast between different tissues around the iso-centre of the image, located in the center of the mouth, than towards the edges of the image. Therefore additional local thresholding steps were needed to segment bones near the posterior edge of the image.

A third limitation of the dataset was low image quality of the CBCT. Some CBCT images only had 121 axial slices, what resulted in poor segmentation quality. Sharp corners in the final segmentations are clearly visible and differentiating details such as the mental foramen or mandibular foramen becomes difficult.

Finally a small portion of the patients were not completely still during the CBCT image acquisition. This resulted in overlapping dense structures in the image. This is a problem for this segmentation protocol, since the parts of the bone that appeared to be captured twice on the image, the gray values are much higher than the surrounding bone tissue that only appeared once on the image. Finding a threshold for these images is extremely difficult and it is questionable how usable the resulting segmentations of these images are for future research.

Despite these difficulties a data set of 57 patient with their respective CBCT images, mandible segmentations and upper skull segmentations was created that can have multiple applications in future research in the field of orthognathic surgery.

References

1. Cheng LH, Roles D, Telfer MR. Orthognathic surgery: the patients' perspective. Br J Oral Maxillofac Surg. 1998;36(4):261-263. doi:10.1016/s02664356(98)90709-4

2. Alkhayer A, Piffk´o J, Lippold C, Segatto E. Accuracy of virtual planning in orthognathic surgery: a systematic review. Head Face Med. 2020;16(1):34. Published 2020 Dec 4. doi:10.1186/s13005-020-00250-2

3. Stokbro K, Aagaard E, Torkov P, Bell RB, Thygesen T. Virtual planningin orthognathic surgery. Int J Oral Maxillofac Surg. 2014;43(8):957-965. doi:10.1016/j.ijom.2014.03.011

4. Baan F, Sabelis JF, Schreurs R, et al. Validation of the OrthoGnathicAnalyser 2.0-3D accuracy assessment tool for bimaxillary surgery and genioplasty. PLoS One. 2021;16(1):e0246196. Published 2021 Jan 26. doi:10.1371/journal.pone.0246196

5. Swennen GR, Mollemans W, Schutyser F. Three-dimensional treatmentplanning of orthognathic surgery in the era of virtual imaging. J Oral Maxillofac Surg 2009;67:2080–2092.

6. O'Neil M, Khambay B, Moos KF, Barbenel J, Walker F, Ayoub A. Validation of a new method for building a three-dimensional physical model of the skull and dentition. Br J Oral Maxillofac Surg 2012;50:49–54.

7. Ahmad M, Jenny J, Downie M. Application of cone beam computed tomography in oral and maxillofacial surgery. Aust Dent J. 2012;57 Suppl 1:82-94. doi:10.1111/j.1834-7819.2011.01661.x

8. Li, C., Lin, L., Zheng, Z., & Chung, C. H. (2021). A User-Friendly Protocol for Mandibular Segmentation of CBCT Images for Superimposition and Internal Structure Analysis. Journal of clinical medicine, 10(1), 127. https://doi.org/10.3390/jcm10010127

Appendix B – Ground truth landmark annotations

Landmarks are colored based on their group: mandible (green), maxilla and cranial bones (yellow), dental (orange) and surgical (blue). The “y” or “n” indicate appearance in the ground truth dataset.

Landmark	ma_001	ma_002	ma_003	ma_004	ma_005	ma_006	ma_007	ma_008	ma_009	ma_010
B Point	y	y	y	y	y	y	y	y	y	y
Gnation	y	y	y	y	y	y	y	y	y	y
Condyle L	y	y	y	y	y	y	y	y	y	y
Corornoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale L	y	y	y	y	y	y	y	y	y	y
Gonion L	y	y	y	y	y	y	y	y	y	y
Sogmoid Notch L	y	y	y	y	y	y	y	y	y	y
Menton	y	y	y	y	y	y	y	y	y	y
Pogonion	y	y	y	y	y	y	y	y	y	y
Condyle R	y	y	y	y	y	y	y	y	y	y
Coronoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale R	y	y	y	y	y	y	y	y	y	y
Gonion R	y	y	y	y	y	y	y	y	y	y
Sigmoid Notch R	y	y	y	y	y	y	y	y	y	y
Anterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Basion	y	y	y	y	y	y	y	y	y	y
Infraorbitale L	y	y	y	y	y	y	y	y	y	y
Infraorbitale R	y	y	y	y	y	y	y	y	y	y
Nasion	y	n	n	n	n	y	n	n	y	y
A Point	y	y	y	y	y	y	y	y	y	y
Porion L	y	y	y	n	y	y	y	y	y	y
Porion R	y	y	y	n	y	y	y	y	y	y
Posterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Sella	y	y	n	y	y	y	y	y	y	y
Supraorbitale L	n	n	n	n	n	n	n	n	n	n
Supraorbitale R	n	n	n	n	n	n	n	n	n	n
13	y	y	y	n	y	y	y	y	y	y
16	y	y	y	y	y	y	y	y	y	y
23	y	y	n	y	y	y	y	y	y	y
26	y	y	y	n	y	y	y	y	y	y
33	y	y	y	y	y	y	y	y	y	y
36	y	y	y	y	y	y	y	y	y	y
43	y	y	y	y	y	y	y	y	y	y
46	y	y	y	y	y	y	y	y	y	y
IsL1	y	y	y	y	y	y	y	y	y	y
IsU1	y	y	y	y	y	y	y	y	y	y
Inferior border L	y	y	y	y	y	y	y	y	y	y
Inferior border R	y	y	y	y	y	y	y	y	y	y
Lingula L	y	y	y	y	y	y	y	y	y	y
Lingula R	y	y	y	y	y	y	y	y	y	y
M2-line L	y	y	y	y	y	y	y	y	y	y
M2-line R	y	y	y	y	y	y	y	y	y	y
Nasal notch L	y	y	y	y	y	y	y	y	y	y
Nasal notch R	y	y	y	y	y	y	y	y	y	y
Zygomatic Process L	y	y	y	y	y	y	y	y	y	y
Zygomatic Process R	y	y	y	y	y	y	y	y	y	y

Landmark	ma_011	ma_012	ma_013	ma_014	ma_015	ma_016	ma_017	ma_018	ma_019	ma_020
B Point	y	y	y	y	y	y	y	y	y	y
Gnation	y	y	y	y	y	y	y	y	y	y
Condyle L	y	y	y	y	y	y	y	y	y	y
Corornoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale L	y	y	y	y	y	y	y	y	y	y
Gonion L	y	y	y	y	y	y	y	y	y	y
Sogmoid Notch L	y	y	y	y	y	y	y	y	y	y
Menton	y	y	y	y	y	y	y	y	y	y
Pogonion	y	y	y	y	y	y	y	y	y	y
Condyle R	y	y	y	y	y	y	y	y	y	y
Coronoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale R	y	y	y	y	y	y	y	y	y	y
Gonion R	y	y	y	y	y	y	y	y	y	y
Sigmoid Notch R	y	y	y	y	y	y	y	y	y	y
Anterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Basion	y	y	y	y	y	y	y	y	y	y
Infraorbitale L	y	y	y	y	y	y	y	y	y	y
Infraorbitale R	y	y	y	y	y	y	y	y	y	y
Nasion	n	n	n	y	n	y	n	n	n	n
A Point	y	y	y	y	y	y	y	y	y	y
Porion L	n	n	y	y	y	y	y	y	y	y
Porion R	y	n	y	y	y	y	y	y	y	y
Posterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Sella	y	y	y	y	y	y	y	y	y	y
Supraorbitale L	n	n	n	n	n	n	n	n	n	n
Supraorbitale R	n	n	n	n	n	n	n	n	n	n
13	y	y	y	y	y	y	y	y	y	y
16	y	y	y	y	y	y	y	y	y	y
23	y	y	y	y	y	y	y	y	y	y
26	y	y	y	y	y	y	y	y	y	y
33	y	y	y	y	y	y	y	y	y	y
36	y	y	y	y	y	y	y	y	y	y
43	y	y	y	y	y	y	y	y	y	y
46	y	y	y	y	y	y	y	y	y	y
IsL1	y	y	y	y	y	y	y	y	y	y
IsU1	y	y	y	y	y	y	y	y	y	y
Inferior border L	y	y	y	y	y	y	y	y	y	y
Inferior border R	y	y	y	y	y	y	y	y	y	y
Lingula L	y	y	y	y	y	y	y	y	y	y
Lingula R	y	y	y	y	y	y	y	y	y	y
M2-line L	y	y	y	y	y	y	y	y	y	y
M2-line R	y	y	y	y	y	y	y	y	y	y
Nasal notch L	y	y	y	y	y	y	y	y	y	y
Nasal notch R	y	y	y	y	y	y	y	y	y	y
Zygomatic Process L	y	y	y	y	y	y	y	y	y	y
Zygomatic Process R	y	y	y	y	y	y	y	y	y	y

Landmark	ma_021	ma_022	ma_023	ma_024	ma_025	ma_026	ma_027	ma_028	ma_029	ma_030
B Point	y	y	y	y	y	y	y	y	y	y
Gnation	y	y	y	y	y	y	y	y	y	y
Condyle L	y	y	y	y	y	y	y	y	y	y
Corornoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale L	y	y	y	y	y	y	y	y	y	y
Gonion L	y	y	y	y	y	y	y	y	y	y
Sogmoid Notch L	y	y	y	y	y	y	y	y	y	y
Menton	y	y	y	y	y	y	y	y	y	y
Pogonion	y	y	y	y	y	y	y	y	y	y
Condyle R	y	y	y	y	y	y	y	y	y	y
Coronoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale R	y	y	y	y	y	y	y	y	y	y
Gonion R	y	y	y	y	y	y	y	y	y	y
Sigmoid Notch R	y	y	y	y	y	y	y	y	y	y
Anterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Basion	n	y	y	y	y	y	y	y	y	y
Infraorbitale L	y	y	y	y	y	y	y	y	y	n
Infraorbitale R	y	y	y	y	y	y	y	y	y	n
Nasion	n	n	n	n	y	n	n	n	n	n
A Point	y	y	y	y	y	y	y	y	y	y
Porion L	n	y	y	y	y	y	y	y	y	y
Porion R	n	y	y	y	y	y	y	y	y	y
Posterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Sella	n	y	y	n	y	y	y	y	y	n
Supraorbitale L	n	n	n	n	y	n	n	n	n	n
Supraorbitale R	n	n	n	n	y	n	n	n	n	n
13	y	n	y	y	y	y	y	y	y	y
16	y	y	y	y	y	y	n	y	y	y
23	y	n	y	y	n	y	n	y	y	y
26	y	y	y	y	y	y	n	y	y	y
33	y	y	y	y	y	y	y	y	y	y
36	y	y	y	y	y	y	y	y	y	y
43	y	y	y	y	y	y	y	y	y	y
46	y	y	y	y	y	y	y	y	y	y
IsL1	y	n	y	y	y	y	y	y	y	y
IsU1	y	y	y	y	y	y	y	y	y	y
Inferior border L	y	y	y	y	y	y	y	y	y	y
Inferior border R	y	y	y	y	y	y	y	y	y	y
Lingula L	y	y	y	y	y	y	y	y	y	y
Lingula R	y	y	y	y	y	y	y	y	y	y
M2-line L	y	y	y	y	y	y	y	y	y	y
M2-line R	y	y	y	y	y	y	y	y	y	y
Nasal notch L	y	y	y	y	y	y	y	y	y	y
Nasal notch R	y	y	y	y	y	y	y	y	y	y
Zygomatic Process L	y	y	y	y	y	y	y	y	y	y
Zygomatic Process R	y	y	y	y	y	y	y	y	y	y

Landmark	ma_031	ma_032	ma_033	ma_034	ma_035	ma_036	ma_037	ma_038	ma_039	ma_040
B Point	y	y	y	y	y	y	y	y	y	y
Gnation	y	y	y	y	y	y	y	y	y	y
Condyle L	y	y	y	y	y	y	y	y	y	y
Corornoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale L	y	y	y	y	y	y	y	y	y	y
Gonion L	y	y	y	y	y	y	y	y	y	y
Sogmoid Notch L	y	y	y	y	y	y	y	y	y	y
Menton	y	y	y	y	y	y	y	y	y	y
Pogonion	y	y	y	y	y	y	y	y	y	y
Condyle R	y	y	y	y	y	y	y	y	y	y
Coronoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale R	y	y	y	y	y	y	y	y	y	y
Gonion R	y	y	y	y	y	y	y	y	y	y
Sigmoid Notch R	y	y	y	y	y	y	y	y	y	y
Anterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Basion	y	y	y	y	y	y	y	y	y	y
Infraorbitale L	y	y	n	y	y	y	y	y	y	y
Infraorbitale R	y	y	n	y	y	y	y	y	y	y
Nasion	n	n	n	y	y	n	n	y	n	n
A Point	y	y	y	y	y	y	y	y	y	y
Porion L	y	y	n	y	y	y	y	n	y	y
Porion R	y	y	n	y	y	y	y	n	y	y
Posterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Sella	y	y	n	y	y	y	y	y	y	y
Supraorbitale L	n	n	n	n	n	n	n	n	n	n
Supraorbitale R	n	n	n	n	n	n	n	n	n	n
13	y	y	y	y	y	y	y	n	n	y
16	y	y	y	y	y	y	y	n	y	y
23	y	y	y	y	y	y	y	y	n	y
26	y	y	y	y	y	y	y	y	y	y
33	y	y	y	y	y	y	y	y	y	y
36	n	y	y	y	y	y	y	y	y	y
43	y	y	y	y	y	y	y	y	y	y
46	n	y	y	y	y	y	y	y	y	y
IsL1	y	y	y	y	y	y	y	y	y	y
IsU1	y	y	y	y	y	y	y	y	y	y
Inferior border L	y	y	y	y	y	y	y	y	y	y
Inferior border R	y	y	y	y	y	y	y	y	y	y
Lingula L	y	y	y	y	y	y	y	y	y	y
Lingula R	y	y	y	y	y	y	y	y	y	y
M2-line L	y	y	y	y	y	y	y	y	y	y
M2-line R	y	y	y	y	y	y	y	y	y	y
Nasal notch L	y	y	y	y	y	y	y	y	y	y
Nasal notch R	y	y	y	y	y	y	y	y	y	y
Zygomatic Process L	y	y	y	y	y	y	y	y	y	y
Zygomatic Process R	y	y	y	y	y	y	y	y	y	y

Landmark	ma_041	ma_042	ma_043	ma_044	ma_045	ma_046	ma_047	ma_048	ma_049	ma_050
B Point	y	y	y	y	y	y	y	y	y	y
Gnation	y	y	y	y	y	y	y	y	y	y
Condyle L	y	y	y	y	y	y	y	y	y	y
Corornoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale L	y	y	y	y	y	y	y	y	y	y
Gonion L	y	y	y	y	y	y	y	y	y	y
Sogmoid Notch L	y	y	y	y	y	y	y	y	y	y
Menton	y	y	y	y	y	y	y	y	y	y
Pogonion	y	y	y	y	y	y	y	y	y	y
Condyle R	y	y	y	y	y	y	y	y	y	y
Coronoid R	y	y	y	y	y	y	y	y	y	y
Foramen mentale R	y	y	y	y	y	y	y	y	y	y
Gonion R	y	y	y	y	y	y	y	y	y	y
Sigmoid Notch R	y	y	y	y	y	y	y	y	y	y
Anterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Basion	y	y	y	y	y	y	y	y	y	y
Infraorbitale L	y	y	y	y	y	y	y	y	y	y
Infraorbitale R	y	y	y	y	y	y	y	y	y	y
Nasion	n	n	n	y	n	n	y	n	n	n
A Point	y	y	y	y	y	y	y	y	y	y
Porion L	y	y	y	y	y	y	y	y	y	n
Porion R	y	y	y	y	y	y	y	y	y	n
Posterior Nasal Spine	y	y	y	y	y	y	y	y	y	y
Sella	y	y	y	y	n	y	y	y	y	y
Supraorbitale L	n	n	n	n	n	n	n	n	n	n
Supraorbitale R	n	n	n	n	n	n	n	n	n	n
13	y	y	y	y	y	y	y	y	y	y
16	y	y	y	y	y	y	y	y	y	y
23	y	y	n	y	y	y	y	y	y	y
26	y	y	y	y	y	y	y	y	y	y
33	y	y	y	n	y	y	y	y	y	y
36	y	n	y	y	y	y	y	y	y	y
43	y	y	y	n	y	y	y	y	y	y
46	y	n	y	y	y	y	y	y	y	y
IsL1	y	y	y	n	y	y	y	y	y	y
IsU1	y	y	y	y	y	y	y	y	y	y
Inferior border L	y	y	y	y	y	y	y	y	y	y
Inferior border R	y	y	y	y	y	y	y	y	y	y
Lingula L	y	y	y	y	y	y	y	y	y	y
Lingula R	y	y	y	y	y	y	y	y	y	y
M2-line L	y	y	y	y	y	y	y	y	y	y
M2-line R	y	y	y	y	y	y	y	y	y	y
Nasal notch L	y	y	y	y	y	y	y	y	y	y
Nasal notch R	y	y	y	y	y	y	y	y	y	y
Zygomatic Process L	y	y	y	y	y	y	y	y	y	y
Zygomatic Process R	y	y	y	y	y	y	y	y	y	y

Landmark	ma_051	ma_052	ma_053	ma_054	ma_055	ma_056	ma_057
B Point	y	y	y	y	y	y	y
Gnation	y	y	y	y	y	y	y
Condyle L	y	y	y	y	y	y	y
Corornoid R	y	y	y	y	y	y	y
Foramen mentale L	y	y	y	y	y	y	y
Gonion L	y	y	y	y	y	y	y
Sogmoid Notch L	y	y	y	y	y	y	y
Menton	y	y	y	y	y	y	y
Pogonion	y	y	y	y	y	y	y
Condyle R	y	y	y	y	y	y	y
Coronoid R	y	y	y	y	y	y	y
Foramen mentale R	y	y	y	y	y	y	n
Gonion R	y	y	y	y	y	y	y
Sigmoid Notch R	y	y	y	y	y	y	y
Anterior Nasal Spine	y	y	y	y	y	y	y
Basion	y	y	y	y	y	y	y
Infraorbitale L	y	y	y	y	y	y	y
Infraorbitale R	y	y	y	y	y	y	y
Nasion	n	n	n	n	n	n	n
A Point	y	y	y	y	y	y	y
Porion L	y	y	y	n	y	y	y
Porion R	y	y	y	n	y	y	y
Posterior Nasal Spine	y	y	y	y	y	y	y
Sella	y	y	y	y	n	y	n
Supraorbitale L	n	n	n	n	n	n	n
Supraorbitale R	n	n	n	n	n	n	n
13	y	y	y	y	y	y	y
16	y	y	y	y	y	y	y
23	y	y	y	y	y	y	y
26	y	y	y	y	y	y	y
33	y	y	y	y	y	y	y
36	y	y	y	y	y	y	y
43	y	y	y	y	y	y	n
46	y	y	y	y	y	y	n
IsL1	y	y	y	y	y	y	y
IsU1	y	y	y	y	y	y	y
Inferior border L	y	y	y	y	y	y	y
Inferior border R	y	y	y	y	y	y	n
Lingula L	y	y	y	y	y	y	y
Lingula R	y	y	y	y	y	y	n
M2-line L	y	y	y	y	y	y	y
M2-line R	y	y	y	y	y	y	n
Nasal notch L	y	y	y	y	y	n	n
Nasal notch R	y	y	y	y	y	n	n
Zygomatic Process L	y	y	y	y	y	y	n
Zygomatic Process R	y	y	y	y	y	y	n

Appendix C – Landmark groups

Landmarks that showed overlap or had the potential to overlap adjacent landmarks were extracted from group A and placed in group B.

Landmark group A	Landmark group B
B Point	Pogonion
Condyle left & right	Sigmoid Notch left & right
Coronoid left & right	Anterior Nasal Spine
Foramen mentale left & right	Porion left & right
Gonion left & right	13
Menton	16
Basion	23
Infraorbitale left & right	26
A point	Upper incisal midpoint
Posterior nasal spine	
Sella	
33	
36	
43	
46	
Lower incisal midpoint	
Inferior border left & right	
Lingula left & right	
Nasal notch left & right	
M2-line left & right	
Zygomatic process left & right+	

Appendix D – Inference workflow

Preprocessing

This python file handles the preprocessing of the images for the segmentation and landmark localization models. The script is executed by the inference.sh file.

```
import os
import SimpleITK as sitk
import numpy as np
from tqdm import tqdm
import scipy.ndimage


def copy_image(image, original_spacing):
    """
    Copy the relevant metadata of the image to the new image and change the voxel size
    The voxel size after downloading from: https://ditto.ing.unimore.it/toothfairy2/
    is 1.0 x 1.0 x 1.0 mm, while the paper describing the dataset describe the
    original voxel size as 0.3 x 0.3 x 0.3 mm

    Parameters
    ----------
    image : SimpleITK.Image
        The original image
    new_spacing : list of float
        The new voxel size

    Returns
    -------
    SimpleITK.Image
        The image with the changed voxel size
    """
    # Create a new image with the same pixel data but different spacing
    copy_image = sitk.GetImageFromArray(sitk.GetArrayFromImage(image))
    copy_image.SetSpacing(original_spacing)
    copy_image.SetOrigin(image.GetOrigin())
    copy_image.SetDirection(image.GetDirection())

    return copy_image


def resample_image(image, new_spacing, is_mask=False):
    """
    Resample the image to the new spacing using cubic or nearest neighbor interpolation

    Parameters
    ----------
    image : SimpleITK.Image
        The original image
    new_spacing : list of float
        The new voxel size
    is_binary : bool
        Whether the image is a binary mask

    Returns
    -------
    SimpleITK.Image
        The resampled image
    """
    # Get the original spacing and size
    original_spacing = np.array(image.GetSpacing(), dtype=np.float32)
    original_size = np.array(image.GetSize(), dtype=np.float32)

    # Calculate the new size
    new_size = original_size * original_spacing / new_spacing
    new_size = np.round(new_size).astype(int)  # Round to the nearest integer

    # Convert the image to a numpy array
    image_array = sitk.GetArrayFromImage(image)

    # Resample the image
    if is_mask:
        print(f"\nResampling image using Nearest Neighbor interpolation")
        # Use nearest neighbor interpolation for binary masks
         resampled_array = scipy.ndimage.zoom(image_array, original_spacing / new_spacing,
order=0)
    else:
        print(f"Resampling image using Cubic interpolation")
        #  Use cubic interpolation for other images
         resampled_array = scipy.ndimage.zoom(image_array, original_spacing / new_spacing,
order=3)

    # Create a new image from the resampled array
    resampled_image = sitk.GetImageFromArray(resampled_array)
    resampled_image.SetSpacing(new_spacing)
    resampled_image.SetDirection(NEW_DIRECTION)

    return resampled_image


def pad_image(image, new_size):
    """
    Pad images to global variables NEW_SIZE with the minimum value in the image
    """
    # Get the original size
    original_size = np.array(image.GetSize(), dtype=int)
    # Calculate the padding size
    padding_size = np.array(new_size) - original_size
    # Make sure the padding size is not negative
    padding_size = np.maximum(padding_size, 0)
    # Calculate the padding for each dimension
    lower_padding = padding_size // 2
    upper_padding = padding_size - lower_padding
    # Convert numpy arrays to lists of integers
    lower_padding = lower_padding.tolist()
    upper_padding = upper_padding.tolist()
    # Get the minimum value in the image
    min_pixel_value = float(np.min(sitk.GetArrayFromImage(image)))
    # Create the padded image
    padded_image = sitk.ConstantPad(image, lower_padding, upper_padding, min_pixel_value)
    # Set the origin of the padded image
    padded_image.SetOrigin(NEW_ORIGIN)

    return padded_image


def resize_images(input_folder_path, output_folder_path, new_size, new_spacing, mask=False):
    """
    Resize and pad the images in the input folder to the output folder

    Parameters
    ----------
    input_folder_path : str
        The path to the input folder
    output_folder_path : str
        The path to the output folder
    """
    # Loop over all the files in the input folder
    for file_name in tqdm(os.listdir(input_folder_path)):
        file_path = os.path.join(input_folder_path, file_name)
        if file_name.endswith(".nii.gz"):
            # Read the image
            image = sitk.ReadImage(file_path)
           if image.GetSpacing() != tuple(new_spacing) or image.GetSize() != tuple(new_size):
                # # Change the voxel size for MICCAI dataset
                # print(f"Copying metadata of {file_name}")
                # image = copy_image(image, ORIGINAL_SPACING)
                # Resample the image
                       print(f"{file_name} has spacing: {image.GetSpacing()}, and size:
{image.GetSize()}")
                resampled_image = resample_image(image, new_spacing, mask)
                print(f"Resampled {file_name}")
                # Pad the image
                padded_image = pad_image(resampled_image, new_size)
                print(f"Padded {file_name}")
                # Save the padded image
                output_file_path = os.path.join(output_folder_path, file_name)
                print(f"Saving {file_name} to {output_file_path}")
                sitk.WriteImage(padded_image, output_file_path)
            else:
                print(f"{file_name} already has the desired spacing and size.")
                output_file_path = os.path.join(output_folder_path, file_name)
                print(f"Saving {file_name} to {output_file_path}")
                sitk.WriteImage(image, output_file_path)


""" Preprocessing images for Segmentation model """

# Environment variable, Change original spacing if needed. DO NOT change the other variables
for optimal performance
# ORIGINAL_SPACING = [0.3, 0.3, 0.3] # for MICCAI dataset
NEW_ORIGIN = [85.125, -84.875, -60]
NEW_DIRECTION = [-1, 0, 0, 0, 1, 0, 0, 0, 1]
new_spacing = [0.25, 0.25, 0.25]
new_size = [681, 681, 481]
mask = False

# TODO: Change the input and output paths
input_path = "/trinity/home/r060801/test_space/nnUNet/nnUNet_raw/raw_images"
output_path                                                                               =
"/trinity/home/r060801/test_space/nnUNet/nnUNet_preprocessed/prep_segmentation_model"
resize_images(input_path, output_path, new_size, new_spacing, mask)

""" Preprocessing images for Landmarking models """

# Environment variable, Change original spacing if needed. DO NOT change teh other variables
for optimal performance
NEW_ORIGIN = [85.125, -84.875, -60]
NEW_DIRECTION = [-1, 0, 0, 0, 1, 0, 0, 0, 1]
new_spacing = [0.5, 0.5, 0.5]
new_size = [340, 340, 240]
mask = False

# TODO: Change the input and output paths
input_path = "/trinity/home/r060801/test_space/nnUNet/nnUNet_raw/raw_images"
output_path                                                                               =
"/trinity/home/r060801/test_space/nnUNet/nnUNet_preprocessed/prep_landmark_model"
resize_images(input_path, output_path, new_size, new_spacing, mask)
```

Postprocessing

This python files handles the post-processing and translates landmark volumes into precise coordinates and pairs these with the corresponding segmentation files. Each set of landmarks and segmentation data is then organized and saved in a dedicated patient folder for easy access and further analysis. The script is executed by the inference.sh file.

```
import os
import SimpleITK as sitk
from scipy.ndimage import center_of_mass
import numpy as np
import scipy.ndimage
import json
import shutil

""" Save segmentation prediction in patient folder """

def save_segmentation_to_patient_dir(input_path, output_path):

    """
    Copies segmentation files from an input directory to a patient-specific directory in the
output path.

    Parameters:
    - input_path: The directory containing the segmentation files.
    - output_path: The base directory where patient-specific directories will be created and
files copied into.
    """

    # Ensure the input directory exists
    if not os.path.exists(input_path):
        print("Input directory does not exist")
        return

    # List all files in the input directory
    for filename in os.listdir(input_path):
        if filename.endswith(".nii.gz"):
            # Extract the base name without extension
            base_name = filename.split(".")[0]
            print(f"Processing segmentation of patient: {base_name}")

            # Create the output directory path for each patient
            patient_output_dir = os.path.join(output_path, base_name)

            # Ensure the output directory exists
            os.makedirs(patient_output_dir, exist_ok=True)

            # Define the full path for the input and output files
            input_file = os.path.join(input_path, filename)
            output_file = os.path.join(patient_output_dir, filename)

            # Copy the file
            shutil.copy(input_file, output_file)
            print(f"Copied {filename} to {output_file}")

""" Postprocessing of landmark prediction, segmentation to voxel """

# Function to load image data and extract useful parameters
def load_patient_segmentation_data(input_image):

    """
    Loads a segmentation image and extracts useful parameters such as segmentation array,
shape, voxel size, origin, and direction.

    Parameters:
    - input_image: Path to the segmentation image file.

    Returns:
    A dictionary containing the segmentation array, image shape, voxel size, origin, and
direction matrix.
    """

    # Load the image
    image = sitk.ReadImage(input_image)
    segmentation = sitk.GetArrayFromImage(image)
    return {
        'segmentation': segmentation,
        'shape': image.GetSize(),
        'voxel_size': image.GetSpacing(),
        'origin': image.GetOrigin(),
        'direction': np.array(image.GetDirection()).reshape(3, 3)
    }

# Function to convert voxel coordinates to RAS coordinate system
def voxel_to_RAS(voxel_coords, voxel_size, origin, direction):

    """
    Converts voxel coordinates to RAS coordinates using voxel size, origin, and direction
matrix.

    Parameters:
    - voxel_coords: The voxel coordinates to convert.
    - voxel_size: The size of the voxels.
    - origin: The origin of the image.
    - direction: The direction matrix of the image.

    Returns:
    The RAS coordinates corresponding to the given voxel coordinates.
    """

    # Convert voxel coordinates to RAS coordinates
    RAS_coords = np.array(voxel_coords) * np.array(voxel_size)
    return np.dot(direction, RAS_coords) + origin

def process_single_landmark(segmentation, label, label_value, voxel_size, origin,
direction):

    """
    Processes a single landmark by finding its center of mass and converting it to RAS
coordinates.

    Parameters:
    - segmentation: The segmentation array.
    - label: The label name.
    - label_value: The integer value of the label in the segmentation.
    - voxel_size: The size of the voxels.
    - origin: The origin of the image.
    - direction: The direction matrix of the image.

    Returns:
    The RAS coordinates of the landmark's center of mass, or None if the label is not found.
    """

    # Find the coordinates of the label in the segmentation
    coords = np.argwhere(segmentation == label_value)
    if coords.size == 0:
        return None  # Skip if no coordinates found for the label

    # Calculate the center of mass of the label
    labeled_array, num_features = scipy.ndimage.label(segmentation == label_value)
    if num_features > 1:
        largest_label = np.argmax(np.bincount(labeled_array.flat)[1:]) + 1
        coords = np.argwhere(labeled_array == largest_label)

    # Calculate the center of mass in RAS
    center_coordinates = center_of_mass(segmentation == label_value)[::-1]
    landmark_coordinate = voxel_to_RAS(center_coordinates, voxel_size, origin, direction)
    return landmark_coordinate.tolist()

def update_json_template(template, position, label):

    """
    Updates a JSON template with the position and label of a landmark.

    Parameters:
    - template: The JSON template to update.
    - position: The RAS coordinates of the landmark.
    - label: The label of the landmark.
    Returns:
    The updated JSON template.
    """

    # Update the template with the position and label
    template['markups'][0]['controlPoints'][0]['position'] = position
    template['markups'][0]['controlPoints'][0]['label'] = label
    return template

def landmark_volume_to_point(input_path, output_path, landmark_labels):

    """
    Converts landmark volumes to point annotations and saves them in the specified output
path.

    Parameters:
    - input_path: The directory containing the landmark volume files.
    - output_path: The base directory where the point annotations will be saved.
     - landmark_labels: A dictionary mapping label names to their integer values in the
segmentation.
    """

    # Ensure the input directory exists
    os.makedirs(output_path, exist_ok=True)
    landmark_predictions = os.listdir(input_path)

    # Process each landmark prediction
    for landmark_prediction in landmark_predictions:
        if landmark_prediction.endswith(".nii.gz"):
            image_path = os.path.join(input_path, landmark_prediction)
            try:
                # Load the segmentation data
                data = load_patient_segmentation_data(image_path)
            except Exception as e:
                print(f"Error processing {landmark_prediction}: {str(e)}")
                continue

            # Create a directory for the patient
            base_name = landmark_prediction.split(".")[0]
            output_base_path = os.path.join(output_path, base_name)
            os.makedirs(output_base_path, exist_ok=True)

            # Load the template for the landmark.mrk.json file
            with
open('/trinity/home/r060801/test_space/nnUNet/nnUNet_raw/template_landmark/template.mrk.jso
n', 'r') as file:
                template_json = json.load(file)
            print(f"Saving landmarks for patient {landmark_prediction}...")

            # Process each landmark label
            for label, label_value in landmark_labels.items():
                landmark_coordinate = process_single_landmark(data['segmentation'], label,
label_value, data['voxel_size'], data['origin'], data['direction'])
                if landmark_coordinate:
                                  updated_template  =  update_json_template(template_json,
landmark_coordinate, label)

                    # Save the updated template as a new .mrk.json file
                    output_file_path = os.path.join(output_base_path, f"{label}.mrk.json")
                    with open(output_file_path, 'w') as outfile:
                        json.dump(updated_template, outfile, indent=4)

# TODO: Change the input and output paths
input_path_one                                                                            =
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset001_CMF/predictions_pp"
input_path_two                                                                            =
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset002_CMF_LM/predictions_pp"
input_path_three                                                                          =
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset003_CMF_LM/predictions_pp"

output_path = "/trinity/home/r060801/test_space/nnUNet/nnUNet_results/output_patients/"

# Landmark label mappings
landmark_labels_part_one = {
    "33": 1, "36": 2, "43": 3, "46": 4, "IsL1": 5,
    "B-Point": 6, "l-Condyl": 7, "l-Coronoid": 8, "l-Foramen Mentale": 9,
    "l-Gonion": 10, "Menton": 11, "r-Condyle": 12, "r-Coronoid": 13,
    "r-Foramen Mentale": 14, "r-Gonion": 15,
    "Nasion": 17, "Basion": 18, "Posterior Nasal Spine": 19,
    "Infraorbitale L": 20, "Infraorbitale R": 21, "Sella": 22,
    "Lingula L": 23, "Lingula R": 24, "M2-line L": 25, "M2-line R": 26,
    "Inferior border L": 27, "Inferior border R": 28, "Point A": 29,
    "Zygomatic Process L": 30, "Zygomatic Process R": 31,
    "Nasal notch L": 32, "Nasal notch R": 33
}

# Landmark label mappings
landmark_labels_part_two = {
    "13": 1, "16": 2, "23": 3, "26": 4, "Anterior Nasal Spine": 5,
    "IsU1": 6, "l-Sigmoid Notch": 7, "Porion L": 8, "Porion R": 9,
    "Pogonion": 10, "r-Sigmoid Notch": 11
}

# Process the segmentation and landmark predictions
print(f'saving all data to {output_path}')
```

```
save_segmentation_to_patient_dir(input_path_one, output_path)
landmark_volume_to_point(input_path_two, output_path, landmark_labels_part_one)
landmark_volume_to_point(input_path_three, output_path, landmark_labels_part_two)
```

Inference script

The inference.sh script manages the execution of the image analysis pipeline on the GPU cluster of the Erasmus MC. It sets up the job environment, including allocating resources and setting up logging. The script loads necessary Python modules, activates a virtual environment, and sets environment variables for nnUNet paths.

It checks for the presence of specific machine learning models and installs them if missing. The script then proceeds to preprocess the data, run nnUNet predictions for segmentation and landmark detection, and applies postprocessing to refine these predictions. Each major step is logged.

```
#!/bin/bash
#SBATCH --ntasks=12
#SBATCH --mem=24G
#SBATCH --gres=gpu:1
#SBATCH --exclude=gpu002,gpu003
#SBATCH -p short
#SBATCH -t 1-22:99:99
#SBATCH -o log/output_%j.log
#SBATCH -e log/error_%j.log
#SBATCH --job-name=inference_CMF

#TODO: Change all paths

log() {
    echo "$(date '+%Y-%m-%d %H:%M:%S') - $1"
}

log '# Starting job'

# Load modules
module load Python/3.11.5-GCCcore-13.2.0
log '# Loaded Python module'

source /trinity/home/r060801/venv_CMF_nnUNetv2_py3.11/bin/activate
log '# Activated virtual environment'

# Define the nnUNet paths as environment variables
export nnUNet_raw="/trinity/home/r060801/test_space/nnUNet/nnUNet_raw"
export nnUNet_preprocessed="/trinity/home/r060801/test_space/nnUNet/nnUNet_preprocessed"
export nnUNet_results="/trinity/home/r060801/test_space/nnUNet/nnUNet_results"
```

```
# Define the path to the zipped models ready for installation
model_zip_path_1="/trinity/home/r060801/test_space/models_zip/Segmentation_model"
model_zip_path_2="/trinity/home/r060801/test_space/models_zip/landmarking_model_4_A"
model_zip_path_3="/trinity/home/r060801/test_space/models_zip/landmarking_model_4_B"

# Define the installed model directories
model_dir_1="$nnUNet_results/Dataset001_CMF"
model_dir_2="$nnUNet_results/Dataset002_CMF_LM"
model_dir_3="$nnUNet_results/Dataset003_CMF_LM"

# Function toA check and install a model from a zip file
install_model_if_not_present() {
    local model_path=$1
    local model_dir=$2
    local model_name=$(basename "$model_dir")
    if [ -d "$model_dir" ]; then
        log "# $model_name already installed"
    else
        log "# Installing $model_name"
        nnUNetv2_install_pretrained_model_from_zip "$model_path"
        log "# Installed $model_name"
    fi
}

# Check and install models if not already present
install_model_if_not_present "$model_zip_path_1" "$model_dir_1"
install_model_if_not_present "$model_zip_path_2" "$model_dir_2"
install_model_if_not_present "$model_zip_path_3" "$model_dir_3"

log '# Installed all models'

# Start timer
start_time=$(date +%s)

# Run preprocessing on the data
python preprocessing.py

log '# Predicting segmentations...'

# Run segmentation prediction
nnUNetv2_predict \
-i "/trinity/home/r060801/test_space/nnUNet/nnUNet_preprocessed/prep_segmentation_model/" \
-o "/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset001_CMF/predictions" \
-d 001 \
-c 3d_fullres \
-tr nnUNetTrainer_onlyMirror01

log '# Postprocessing...'
```

```
# Apply postprocessing
nnUNetv2_apply_postprocessing \
-i "/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset001_CMF/predictions" \
-o "/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset001_CMF/predictions_pp" \
-pp_pkl_file
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset001_CMF/nnUNetTrainer_onlyMi
rror01__nnUNetPlans__3d_fullres/crossval_results_folds_0_1_2_3_4/postprocessing.pkl" \
-np 8 \
-plans_json
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset001_CMF/nnUNetTrainer_onlyMi
rror01__nnUNetPlans__3d_fullres/crossval_results_folds_0_1_2_3_4/plans.json"

log '# Predicting landmarks group A...'

# Run landmark model_A_4mm prediction
nnUNetv2_predict \
-i "/trinity/home/r060801/test_space/nnUNet/nnUNet_preprocessed/prep_landmark_model/" \
-o "/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset002_CMF_LM/predictions" \
-d 002 \
-c 3d_fullres \
-tr nnUNetTrainer_onlyMirror01

log '# Postprocessing...'

# Apply postprocessing
nnUNetv2_apply_postprocessing \
-i "/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset002_CMF_LM/predictions" \
-o
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset002_CMF_LM/predictions_pp" \
-pp_pkl_file
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset002_CMF_LM/nnUNetTrainer_onl
yMirror01__nnUNetPlans__3d_fullres/crossval_results_folds_0_1_2_3_4/postprocessing.pkl" \
-np 8 \
-plans_json
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset002_CMF_LM/nnUNetTrainer_onl
yMirror01__nnUNetPlans__3d_fullres/crossval_results_folds_0_1_2_3_4/plans.json"

log '# Predicting landmarks group B...'

# Run landmark model_B_4mm prediction
nnUNetv2_predict \
-i "/trinity/home/r060801/test_space/nnUNet/nnUNet_preprocessed/prep_landmark_model/" \
-o "/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset003_CMF_LM/predictions" \
-d 003 \
-c 3d_fullres \
-tr nnUNetTrainer_onlyMirror01
```

```
log '# Postprocessing...'

# Apply postprocessing
nnUNetv2_apply_postprocessing \
-i "/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset003_CMF_LM/predictions" \
-o
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset003_CMF_LM/predictions_pp" \
-pp_pkl_file
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset003_CMF_LM/nnUNetTrainer_onl
yMirror01__nnUNetPlans__3d_fullres/crossval_results_folds_0_1_2_3_4/postprocessing.pkl" \
-np 8 \
-plans_json
"/trinity/home/r060801/test_space/nnUNet/nnUNet_results/Dataset003_CMF_LM/nnUNetTrainer_onl
yMirror01__nnUNetPlans__3d_fullres/crossval_results_folds_0_1_2_3_4/plans.json"

# Run postprocessing on the data and export to output_patients folder
python postprocessing.py

# End timer
end_time=$(date +%s)
elapsed=$(( end_time - start_time ))

log "Postprocessing and export time was: $((elapsed / 60)) minutes and $((elapsed % 60))
seconds"

# Deactivate virtual environment
deactivate

log '# Deactivated virtual environment'

log '# Job completed'
```

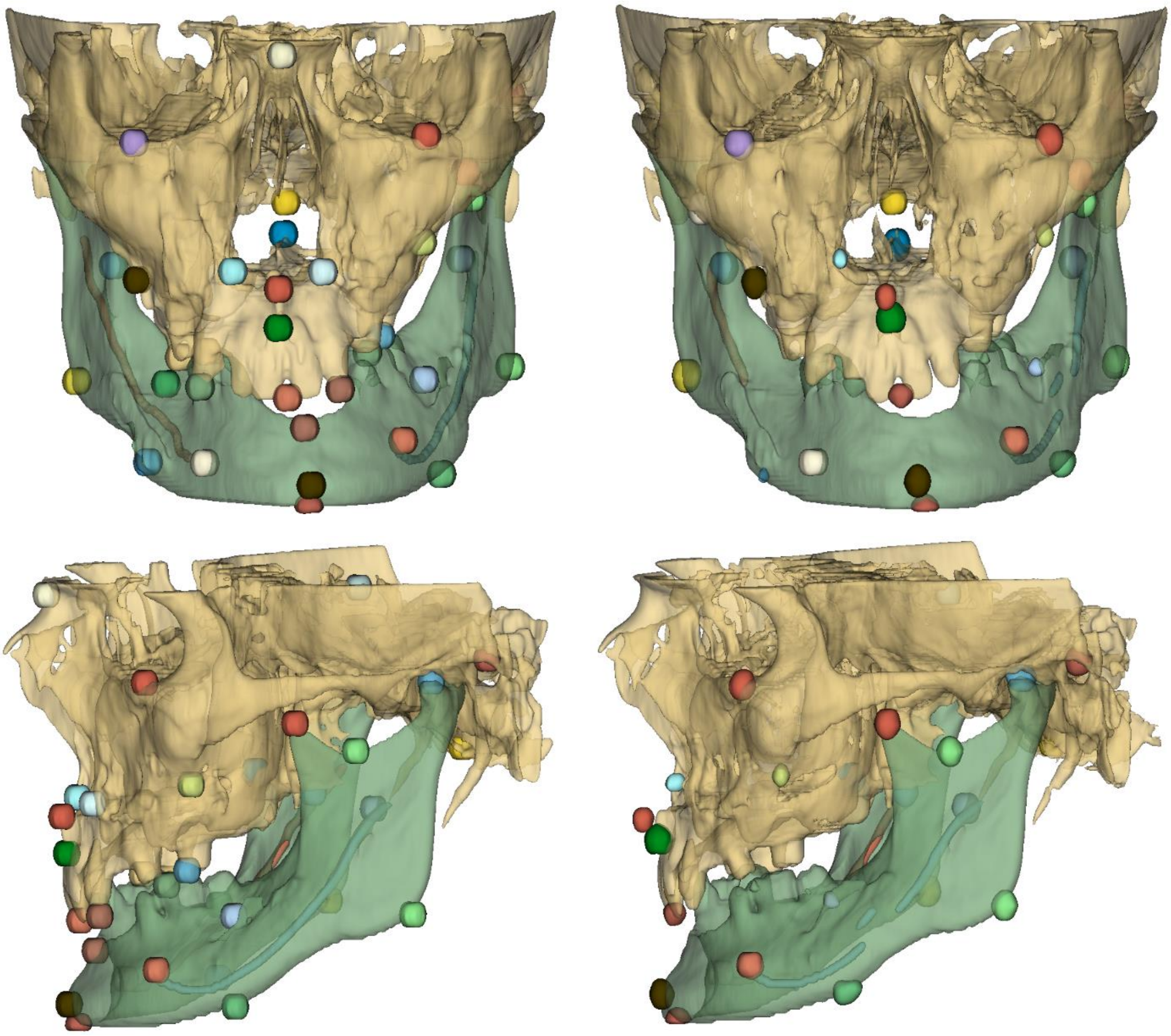
Appendix E – Additional metrics segmentation model

This table presents additional metrics, IoU and HSD95 that can be used to compare the models performance with those reported in other studies.

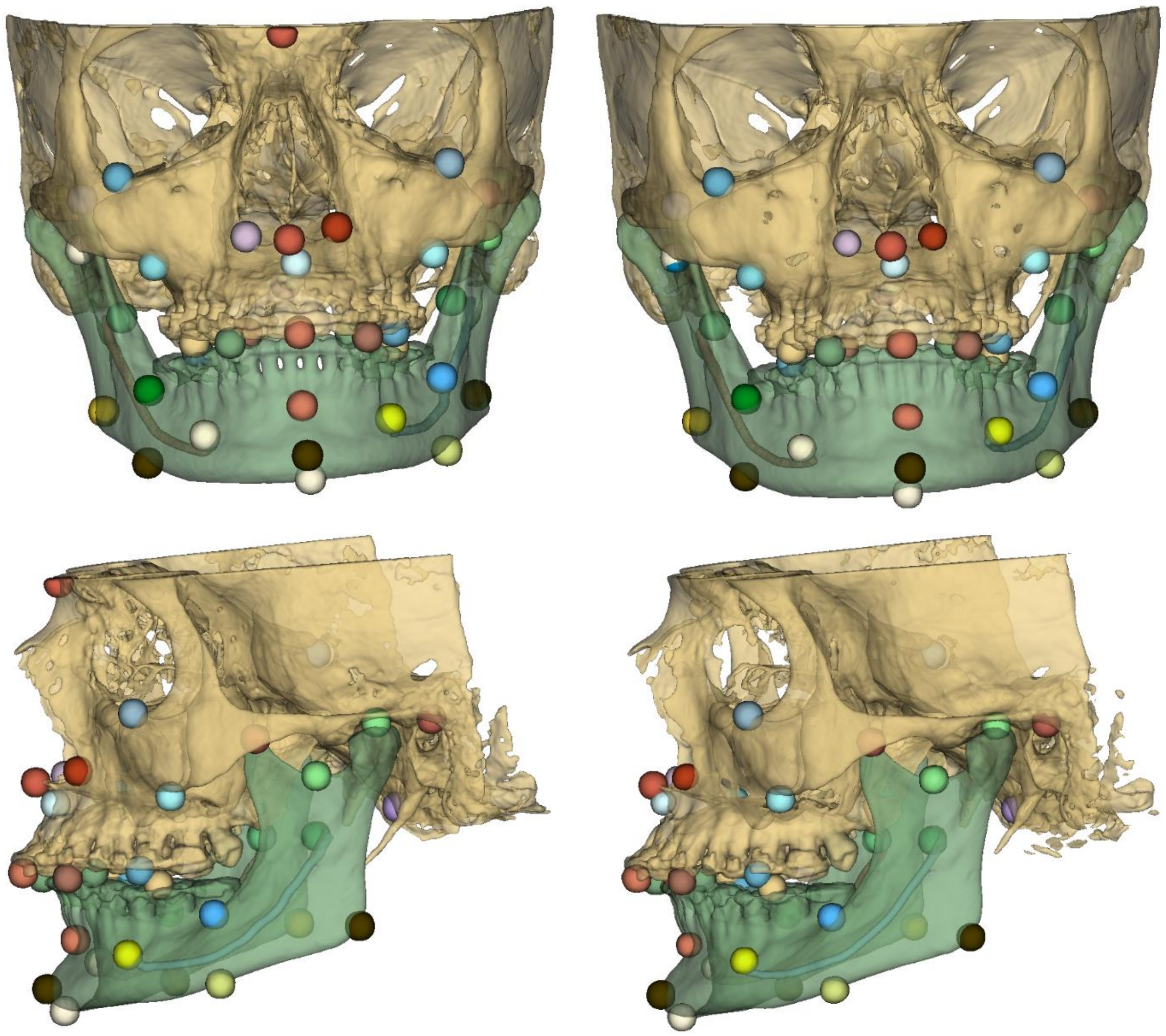
	Metric	Test dataset (n=11) Median [Q1 -Q3]	TF2F (n=30) Median [Q1 -Q3]	TF2P (n=30) Median [Q1 -Q3]
Mandible	IoU	0.96 [0.96 - 0.97]	0.94 [0.94 - 0.95]	0.94 [0.92 - 0.94]
	HSD95	0.25 [0.25 - 0.35]	0.50 [0.50 - 0.54]	0.56 [0.50 - 0.82]
Maxilla	IoU	0.84 [0.81 - 0.87]	0.86 [0.83 - 0.90]	0.53 [0.40 - 0.72]
	HSD95	1.62 [1.09 - 3.58]	1.93 [1.39 - 3.10]	38.9 [6.22 - 44.6]
Mandibular canal left	IoU	0.63 [0.59 - 0.65]	0.67 [0.62 - 0.70]	0.69 [0.63 - 0.73]
	HSD95	0.75 [0.66 - 1.04]	0.66 [0.56 - 1.29]	0.59 [0.50 - 1.17]
mandibular canal right	IoU	0.59 [0.48 - 0.66]	0.70 [0.63 - 0.72]	0.70 [0.66 - 0.72]
	HSD95	0.79 [0.73 - 2.35]	0.56 [0.50 - 0.71]	0.56 [0.50 - 0.78]
Skin	IoU	0.90 [0.89 - 0.92]	-	-
	HSD95	3.48 [3.22 - 3.81]	-	-
Overall	IoU	0.84 [0.65 - 0.92]	0.75 [0.68 - 0.91]	0.72 [0.60 - 0.91]
	HSD95	0.94 [0.59 - 3.20]	0.61 [0.50 - 1.76]	0.66 [0.50 - 4.39]

Appendix F – Patient cases

The first patient case was the worst predicted patient form the test dataset. On the left the ground truth segmentations are shown in two positions and on the right the predictions. In the prediction, a step or discontinuous mandibular canal could be seen. Furthermore, several landmarks are missing, such as the nasal notch left, the nasion, element 13, 16, 23, 26. Notably the predicted landmarks are not perfectly spherical (anterior nasal spine) or have visible volumetric deviations from the ground truth (M2-line left, Nasal notch right).



The second patient case reflects the average testcase in test set. No apparent errors in the segmentation of the anatomy or the landmarks were observed. Small imperfections can be observed on the surface of the maxilla and some landmark segmentations such as the B point might not be perfectly spherical.

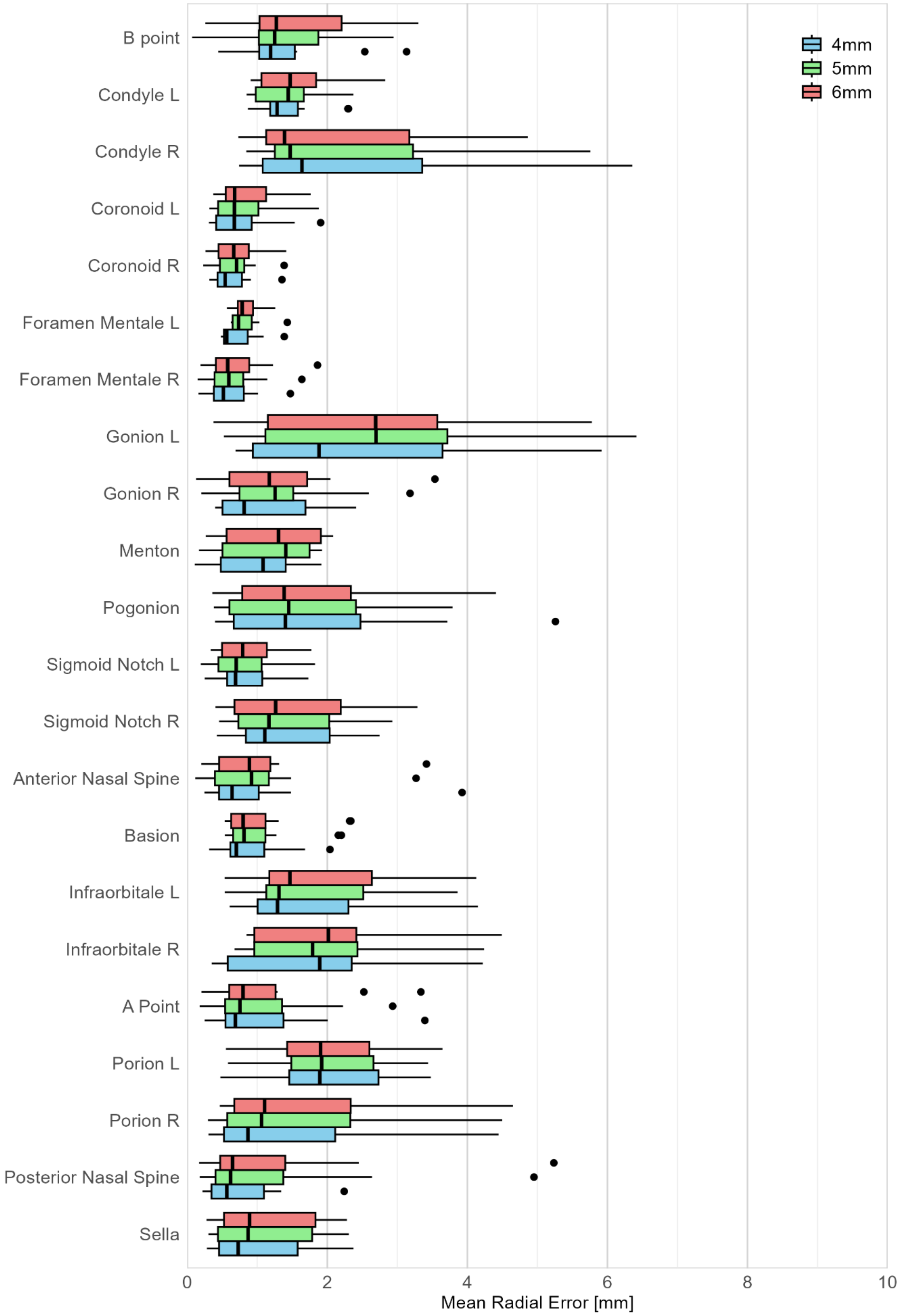


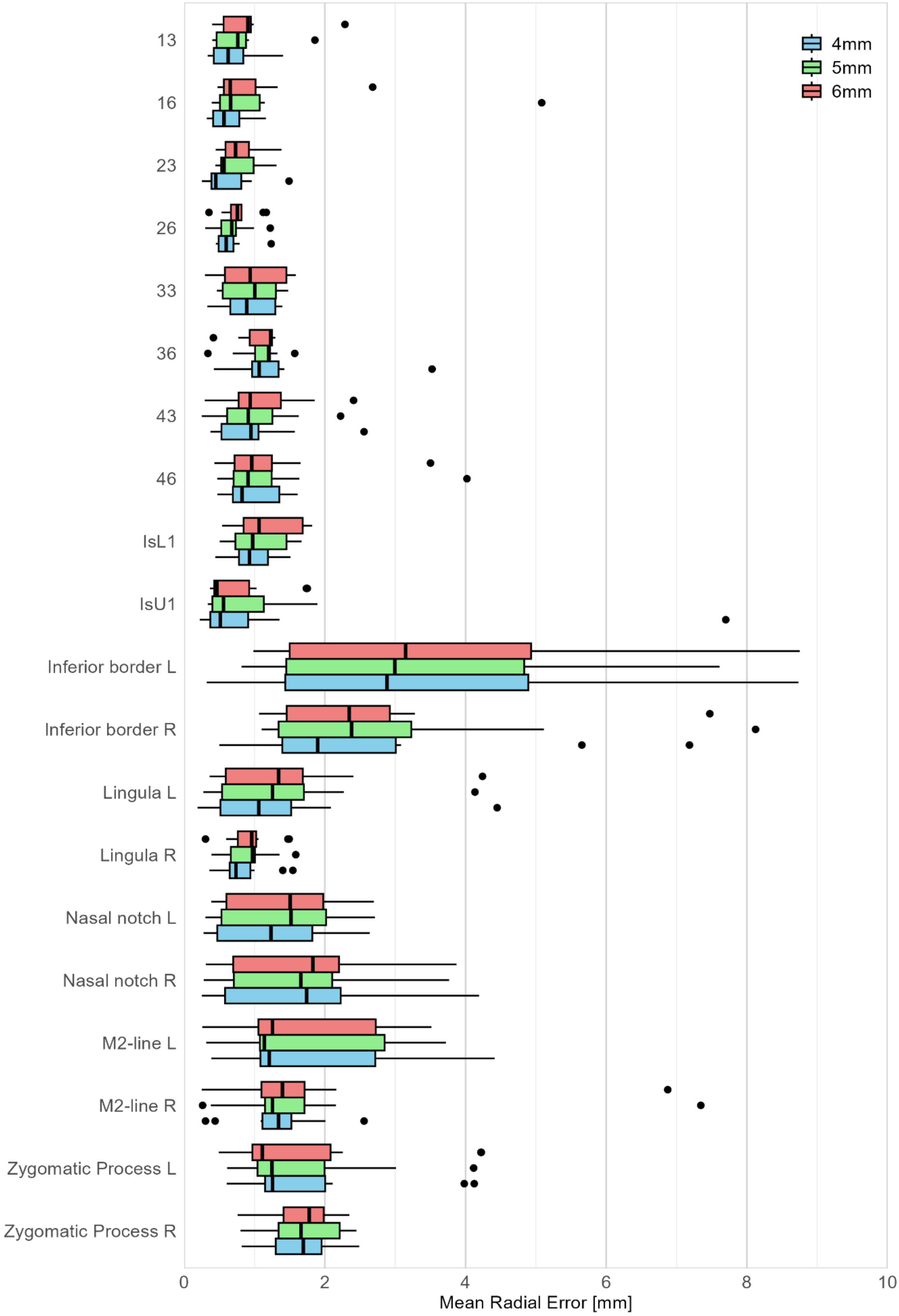
Appendix G – Individual landmark results

Landmarking results for each configuration (4, 5 and 6 mm sphere radius), reported in median and successful detection rate (SDR) within 2 mm. NA = not available.

	Model_4_A&B		Model_5_A&B		Model_6_A&B	
Landmark	**Median [Q1 - Q3] (mm)**	**SDR < 2mm (%)**	**Median [Q1 - Q3] (mm)**	**SDR < 2mm (%)**	**Median [Q1 - Q3] (mm)**	**SDR < 2mm (%)**
B Point	1.19 [1.03 - 1.54]	80	1.25 [1.02 - 1.87]	70	1.27 [1.03 - 2.20]	63.6
Condyle L	1.28 [1.18 - 1.58]	81.8	1.44 [0.98 - 1.67]	81.8	1.47 [1.06 - 1.84]	90.9
Condyle R	1.64 [1.08 - 3.35]	54.5	1.47 [1.25 - 3.22]	54.5	1.39 [1.13 - 3.17]	54.5
Coronoid L	0.67 [0.41 - 0.92]	100	0.67 [0.44 - 1.02]	100	0.68 [0.55 - 1.13]	100
Coronoid R	0.54 [0.43 - 0.78]	100	0.70 [0.47 - 0.81]	100	0.66 [0.45 - 0.88]	100
Foramen Mentale L	0.56 [0.52 - 0.86]	100	0.73 [0.65 - 0.92]	100	0.79 [0.72 - 0.94]	100
Foramen Mentale R	0.52 [0.38 - 0.81]	100	0.59 [0.39 - 0.80]	100	0.58 [0.41 - 0.89]	100
Gonion L	1.88 [0.94 - 3.64]	54.5	2.70 [1.12 - 3.71]	45.5	2.69 [1.15 - 3.57]	45.5
Gonion R	0.84 [0.54 - 2.02]	72.7	1.40 [0.76 - 2.06]	72.7	1.27 [0.71 - 1.91]	72.7
Menton	1.08 [0.48 - 1.41]	100	1.41 [0.50 - 1.75]	100	1.30 [0.56 - 1.91]	81.8
Pogonion	1.40 [0.66 - 2.47]	63.6	1.45 [0.60 - 2.41]	72.7	1.38 [0.79 - 2.34]	72.7
Sigmoid Notch L	0.69 [0.57 - 1.07]	100	0.70 [0.45 - 1.06]	100	0.79 [0.50 - 1.14]	100
Sigmoid Notch R	1.11 [0.84 - 2.03]	72.7	1.17 [0.73 - 2.03]	72.7	1.26 [0.67 - 2.19]	72.7
A Point	0.69 [0.55 - 1.38]	81.8	0.75 [0.54 - 1.35]	81.8	0.80 [0.60 - 1.26]	81.8
Anterior Nasal Spine	0.64 [0.45 - 1.02]	90.9	0.92 [0.39 - 1.16]	90.9	0.89 [0.46 - 1.19]	90.9
Basion	0.70 [0.61 - 1.10]	90.9	0.81 [0.65 - 1.12]	81.8	0.80 [0.63 - 1.12]	81.8
Infraorbitale L	1.29 [1.01 - 2.30]	72.7	1.31 [1.13 - 2.51]	72.7	1.47 [1.17 - 2.64]	63.6
Infraorbitale R	1.89 [0.58 - 2.35]	54.5	1.79 [0.96 - 2.43]	54.5	2.01 [0.96 - 2.42]	45.5
Nasion	*NA*	0	*NA*	0	*NA*	25
Porion L	1.89 [1.46 - 2.73]	55.6	1.92 [1.49 - 2.66]	55.5	1.90 [1.43 - 2.60]	55.6
Porion R	0.87 [0.52 - 2.11]	66.7	1.06 [0.57 - 2.33]	66.6	1.10 [0.67 - 2.33]	66.7
Posterior Nasal Spine	0.57 [0.35 - 1.09]	90	0.62 [0.40 - 1.37]	81.8	0.65 [0.47 - 1.40]	81.8
Sella	0.73 [0.45 - 1.58]	80	0.87 [0.44 - 1.78]	70	0.89 [0.53 - 1.83]	70
13	0.63 [0.42 - 0.84]	100	0.76 [0.46 - 0.88]	100	0.91 [0.56 - 0.95]	88.9
16	0.57 [0.41 - 0.78]	100	0.66 [0.51 - 1.07]	88.9	0.65 [0.56 - 1.01]	88.9
23	0.45 [0.39 - 0.81]	100	0.57 [0.53 - 0.99]	100	0.73 [0.59 - 0.92]	100
26	0.60 [0.49 - 0.70]	100	0.68 [0.53 - 0.74]	100	0.76 [0.66 - 0.82]	100
33	0.89 [0.66 - 1.30]	100	1.00 [0.55 - 1.31]	100	0.94 [0.58 - 1.45]	100
36	1.07 [0.97 - 1.34]	90	1.20 [1.01 - 1.22]	100	1.23 [0.94 - 1.26]	90
43	0.95 [0.53 - 1.06]	90	0.91 [0.61 - 1.25]	90	0.94 [0.77 - 1.38]	90
46	0.82 [0.69 - 1.35]	100	0.91 [0.70 - 1.24]	90.9	0.96 [0.72 - 1.25]	90.9
IsL1	0.93 [0.78 - 1.19]	100	0.97 [0.73 - 1.46]	100	1.06 [0.84 - 1.69]	100
IsU1	0.51 [0.37 - 0.91]	90.9	0.56 [0.40 - 1.13]	100	0.47 [0.43 - 0.92]	100
Inferior border L	2.88 [1.44 - 4.89]	45.5	2.99 [1.45 - 4.83]	36.4	3.15 [1.50 - 4.93]	36.4
Inferior border R	1.90 [1.39 - 3.01]	54.5	2.38 [1.34 - 3.23]	45.5	2.35 [1.46 - 2.92]	45.5
Lingula L	1.06 [0.52 - 1.52]	81.8	1.25 [0.54 - 1.70]	81.8	1.34 [0.59 - 1.69]	81.8
Lingula R	0.74 [0.65 - 0.94]	100	0.97 [0.66 - 1.01]	100	0.96 [0.76 - 1.02]	100
M2-line L	1.21 [1.08 - 2.72]	63.6	1.14 [1.08 - 2.85]	63.6	1.25 [1.06 - 2.72]	63.6
M2-line R	1.34 [1.11 - 1.52]	80	1.25 [1.15 - 1.71]	81.8	1.40 [1.10 - 1.71]	81.8
Nasal notch L	1.23 [0.47 - 1.82]	80	1.52 [0.53 - 2.02]	72.7	1.51 [0.60 - 1.98]	72.7
Nasal notch R	1.74 [0.58 - 2.22]	63.6	1.66 [0.71 - 2.10]	63.6	1.83 [0.70 - 2.20]	63.6
Zygomatic Process L	1.26 [1.15 - 2.00]	72.7	1.25 [1.04 - 1.99]	72.7	1.11 [0.97 - 2.08]	72.7
Zygomatic Process R	1.69 [1.30 - 1.95]	72.7	1.66 [1.34 - 2.21]	63.6	1.78 [1.41 - 1.98]	72.7
Overall	**0.98 [0.58 - 1.60]**	**88.6**	**1.03 [0.63 - 1.67]**	**85.9**	**1.08 [0.66 - 1.82]**	**83.0**

Additionally, boxplots for visualization for each individual landmark from model_4mm (blue), model_5mm (green) and model_6mm (red).




13
16
23
26
33
36
43
46
IsL1
IsU1
Inferior border L
Inferior border R
Lingula L
Lingula R
Nasal notch L
Nasal notch R
M2-line L
M2-line R
Zygomatic Process L
Zygomatic Process R
4mm
5mm
6mm
0
2
4
6
8
10
Mean Radial Error [mm]