

AMERICAN UNIVERSITY OF BEIRUT

SEGMENTATION AND MOTION ANALYSIS
OF TEXTURED THREE-DIMENSIONAL
SCANS OF TEETH

by

ABDUL REHMAN HASSAN EL BSAT

A thesis

submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Mechanical Engineering
of Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
April 2021

AMERICAN UNIVERSITY OF BEIRUT

SEGMENTATION AND MOTION ANALYSIS
OF TEXTURED THREE-DIMENSIONAL
SCANS OF TEETH

by
ABDUL REHMAN HASSAN EL BSAT

Approved by:

Dr. Elie Shammas, Associate Professor
Mechanical Engineering

Advisor



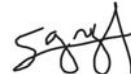
Dr. Daniel Asmar, Associate Professor
Mechanical Engineering

Co-Advisor



Dr. Georges Sakr, Assistant Professor
Computer Engineering

Co-Advisor



Dr. Joseph Ghafari, Professor
Otorhinolaryngology

Member of the Committee



Date of thesis defense: April 22, 2021

Acknowledgements

I would like to thank my advisor, Dr. Elie Shammas, for his invaluable supervision and expertise that helped in formulating the research questions. My gratitude extends to the Faculty of Engineering for the funding opportunity to undertake my studies at the Department of Mechanical Engineering at the American University of Beirut. I would like to thank my co-advisors, Dr. Daniel Asmar and Dr. Georges Sakr, as well as Professor Joseph Ghafari for their valuable guidance throughout this research. Your guidance lead me down the right direction and helped complete my dissertation successfully. I would like to thank my friends and lab mates for stimulating discussions as well as happy distractions to rest my mind outside of my research. I appreciate my family and friends for their encouragement and support throughout my studies.

Finally, I would like to thank the Orthodontics and Dentofacial Surgery Department at the American University of Beirut Medical Center for collecting and assisting in labeling and annotating the data used in this work.

An Abstract of the Thesis of

Abdul Rehman Hassan El Bsat for Master of Engineering
Major: Mechanical Engineering

Title: Segmentation and Motion Analysis of Textured Three-dimensional Scans of Teeth

Teeth movement is an important process for a dentist that helps in gauging the progress of the treatment. However, the lack of a stable reference with respect to which one could measure the teeth movement makes this a challenging problem. In this work, the Rugae are used as stable reference on which a segmentation and motion measurement of all individual teeth in the upper jaw is performed. The approach in this work is to utilize deep learning Convolutional Neural Networks (CNNs) to segment the rugae and the individual teeth. Building upon the robustness of two-dimensional image semantic segmentation, this work develops a method to convert three-dimensional textured scans of the upper palate to two-dimensional data on which the semantic segmentation is applied. Moreover, the achieved two-dimensional segmentation is pulled-back to segment the original three-dimensional textured mesh. After the segmentation of two three-dimensional scans of the same patient before and after an orthodontic treatment, an algorithm is developed to match the scans at the stable rugae region from which the three-dimensional, translation and rotation, motion of the individual teeth is computed.

Contents

Acknowledgements	v
Abstract	vi
1 Introduction	1
2 Semantic Segmentation of two-dimensional Images	5
2.1 Dataset Collection	5
2.2 Dataset Labeling Methods	7
2.2.1 Semantic Labeling	7
2.2.2 Label Statistics	8
2.3 Machine learning and Semantic Segmentation	8
2.3.1 Dataset Split	9
2.3.2 Network Architectures	10
2.3.3 Network Assessment	11
2.4 Results	13
2.4.1 Semantic Segmentation Training Results	13
2.4.2 Semantic Segmentation Application Results	15
2.4.3 Limitations	18
2.5 Conclusion	19
3 Semantic Segmentation of Three-dimensional Textured Scans	23
3.1 Data collection, Annotation and Augmentation	23
3.1.1 Dataset Collection	23
3.1.2 Dataset Annotation	24
3.1.3 Dataset Augmentation	25
3.2 Deep Network Design and Training	27
3.2.1 Network Architecture	27
3.2.2 Training Process	28
3.3 Rasterization	29
3.4 Results	29
3.5 Conclusion	32

4	Motion Measurement of Teeth	35
4.1	Coordinate Frame Definition	35
4.2	Motion Estimation	35
4.3	Results	37
4.4	Conclusion	37
5	Conclusion	42
A	Network Architectures	43
B	Abbreviations	46

List of Figures

2.1	Acquiring image of the maxillary teeth.	5
2.2	Image taken at occlusal view showing the rugae	6
2.3	First two columns depict the original dataset while the last column depicts a pair of images from the expanded dataset.	6
2.4	Labeling for semantic segmentation and the two labeling schemes.	7
2.5	Statistics of the labeling schemes	8
2.6	First two columns depict the original dataset while the last column depicts a pair of images from the expanded dataset.	9
2.7	IoU, Accuracy and Precision Representation	12
2.8	Family of Teeth average sample result for DenseNet in 2.8a, 2.8b, and 2.8c. Family of Teeth average sample result for SegNet in 2.8d, 2.8e, and 2.8f	14
2.9	Data augmentation via rotation and perspective shrinking.	14
2.10	Individual Teeth Labeling results for DenseNet including rotation dataset displaying worst result in the first row of figures 2.10a, 2.10b, and 2.10c; average result in the second row of figures 2.10d, 2.10e, and 2.10f ;and Best result in the third row of figures 2.10g, 2.10h, and 2.10i	17
2.11	individual Teeth Labeling results for SegNet including rotation dataset displaying worst result in the first row of figures 2.11a, 2.11b, and 2.11c; average result in the second row of figures 2.11d, 2.11e, and 2.11f ;and Best result in the third row of figures 2.11g, 2.11h, and 2.11i	18
2.12	Graphs of the label distribution over the set analyzed	21
2.13	The first column shows the original image from the pre-treatment set followed by its ground truth label in the row below and its prediction from the model in the final row. The second column shows the same order for an image from the post-treatment of the same patient.	22
2.14	Miss labeling due to missing teeth	22
3.1	3D scan samples	23
3.2	Image of the maxillary teeth	24

3.3	Meshmixer steps of selection of vertices on the left, single segmentation in the middle, and the final separation on the right	24
3.4	Number of pixels for each label compared with number of vertices for each label	25
3.5	Ratio of pixels to vertices for each label	26
3.6	3D Scan Samples showing different light conditions	26
3.7	3D Scan Samples showing different camera proximities	27
3.8	Flow Chart of the Annotation and 2D projections	28
3.9	Sample 3D mesh with Camera locations surrounding it	29
3.10	Flow Chart of the 3D segmentation process	30
3.11	Attention Layer Comparison on Unseen Projection of a trained scan	31
3.12	Attention Layer Comparison on Unseen Projection of an Unseen scan	31
3.13	Final Training results displaying worst result in the first row of figures 3.13a, 3.13b, and 3.13c; average result in the second row of figures 3.13d, 3.13e, and 3.13f ;and Best result in the third row of figures 3.13g, 3.13h, and 3.13i	33
3.14	3D Scan Segmented on the left and a sample raster projection onto the mesh on the right	34
4.1	Mesh with sphere placement on the left and generated coordinate frame on the right	36
4.2	Motion Measurement process applied on one tooth as an example	36
4.3	This is a diagram showing the process of alignment being done on a manually changed scan. Scans are imported, aligned on the reference rugae, followed by individual alignment of every tooth with respect to the rugae aligned reference for measurement purposes	39
4.4	This is a diagram showing the process of alignment being done on a pre-treatment and post-treatment scans of a real patient. Scans are imported, aligned on the reference rugae, followed by individual alignment of every tooth with respect to the rugae aligned reference for measurement purposes	41
A.1	The DenseNet and MobileNet architectures are depicted.	43
A.2	The Segnet and AdapNet architectures are depicted.	45

List of Tables

2.1	The image count and labeling scheme of the original and expanded datasets.	10
2.2	Model Architecture Comparison on Tooth Family Labels	13
2.3	Model Architecture Comparison on Specific Tooth Labels	15
2.4	Model Architecture Comparison on Specific Tooth Labels with Rotation for all Architecture Candidates	16
2.5	Dataset Analysis Results	19
2.6	Pre and Post IoU Values for teeth	20
3.1	Model Architecture Comparison on Test Set	30
3.2	Model Architecture Comparison on Unseen Scan set	30
3.3	Model Training Results on the Final Dataset	32
4.1	Translation Motion Results comparing actual ground truth movement with respect to the movement achieved from the software	38
4.2	Rotation Motion Results comparing actual ground truth movement with respect to the movement achieved from the software	38
4.3	Translation Motion Results comparing actual ground truth movement with respect to the movement achieved from the software	40

Chapter 1

Introduction

Digital dentistry has been evolving in recent years. The main goal of digital dentistry is to segment 3D teeth models for treatment planning and tooth movement measurement for orthodontic diagnosis. However, there is no general robust model that can define the parameters of teeth for all humans due to the high variability in teeth appearance between people. This document intends to use the robustness of two-dimensional image semantic segmentation to segment 3D texture-colored models of teeth followed by tooth measurement with respect to a stable reference.

In [1], using the integral intensity profile from 2D X-ray images, a dental arch represented as a four-degree polynomial is created to separate tooth region in the image. This is done by placing orthogonal planes drawn in the local minima of the intensity profile along the arch curve to separate the teeth. This method was implemented on both the upper and lower jaws, and was validated on segmenting teeth on 2D X-rays. This method requires intervention to refine the placement the planes which separate the teeth and additionally, it does not semantically segment the teeth. In other words, the regions bounded by the separation planes include both teeth and gingiva regions.

Oktay [2] implemented object detection machine learning algorithms on panoramic X-ray images. Pre-processing had to be performed on the X-ray imaged to identify the mouth gap. This was done iteratively to identify regions in X-ray images where teeth were expected to be found. The located mouth gap area, that is, the area of high probability of teeth being there, was split into three sub-regions: molars, premolars, and anterior teeth (canines and incisors). Then, within each region, each tooth was labeled by a bounding box belonging to a specific class. The union of the similarly labeled regions was used to define a bounding area for a certain label which in turn was used during the training processes. Additionally, a data augmentation procedure was employed. It involved using the horizontal and vertical symmetry to divide the image into four quadrants. The individual quadrants were then mirrored and rotated to match up the bounding region for the different labels. In this work, the Alexnet network architecture was used and

trained to perform the object detection task. Note that, the object detection was performed on the three bounding subregions defined earlier. Beyond the required pre-processing need in this method, object detection identifies whether a tooth exists in the X-ray image and highlights it with an encompassing rectangular shape. Again, there rectangular area includes both teeth and gingiva regions.

Other machine learning methods which are more relevant to our proposed methods focused on teeth segmentation rather than teeth detection. Mikia et al. [3] applied deep learning to classify teeth. The input to the network consisted of the two-dimensional X-ray images of pre-segmented teeth which was done manually. The networks were trained to classify the segmented teeth. The authors acknowledged that their dataset consisted of 52 images was insufficient for training the network. Hence, the authors implemented data augmentation such as image rotation and light intensity manipulation to address the sample size limitation. Note that, this approach classified the pre-segmented X-ray images rather than semantically segmenting the teeth.

Switching from two-dimensional to three-dimensional images, minimum curvature was used to initiate the segmentation process of teeth in [4]. While this method was able to segment the individual teeth ins three-dimensions, it required user interaction at multiple stages to exclude the undesirable areas picked by the curvature-based algorithm.

Similarly, Raith et al. [5] classified teeth features in three-dimensional scan, using an Artificial Neural Network (ANN). Note that ANN's are typically employed in tasks involving pattern recognition in digital image analysis. In this work, cusps of the teeth are the desired features to be detected. According, the locations of these cusps were manually labeled and fed to the as an input to the ANN as a feature vector. The ANN as trained to identify and located the cusps.. Three cusp detection approaches were evaluated and compared. The first relied on the "Cusp Distance", in reference to the distance between each cusp with the neighboring cusps. The second was the "range image" approach, whereby images required additional range data; the cusps were uniquely manually labeled, and used in a post-processing algorithm to classify the teeth. The third approach was a combination of the above two methods. This method only classifies the cusps of the teeth and does not segment the individual teeth.

Another machine learning approach was adopted by Xu et al. [6] who performed segmentation directly on a three-dimensional model. They classified mesh faces on a two-level segmentation, the first separates the teeth from the gingiva and the second segments individual teeth. The input to their network was a 600-dimensional feature vector which was defined [7]. A label optimization methods was additionally employed to correct wrongly predicted labels. Even with label optimization, "sticky teeth", that is, pairs of adjacent teeth similarly labeled after optimization, were sometimes falsely predicted. Principal Component Analysis was implemented to address the problem of sticky teeth. The authors reduced the number of faces in the model through mesh simplification to reduce the required

processing power needed to train the network. This method required pre- and post-processing step.

Another relevant prior work is done by Cui et al.[8] who performed 3D segmentation using CBCT images that are regressed back into the dental model from the scan. Their process involved having the CBCT image go through edge extraction via deep network as a pre-processing step to the segmentation network. The resulting image contained only the boundaries at which the teeth are located in the CBCT image. The next step involved concatenating the original CBCT image with the segmented edges into a similarity matrix of the region proposal network. This network then branched into four segments: classification, segmentation, 3D box regressor and identification. The segmentation and classification branches were applied onto the CBCT image. Using the help of the edges of teeth that were extracted, the teeth in the image were segmented and classified into their respective classes. The identification branch helped with the spatial relation that was needed for the 3D box regressor to regress the segmentation onto the dental model. Hence, a 3D model with segmented teeth was generated using the 4 branches. This method required pre-processing methods to facilitate the segmentation process of the 3D model.

Another work done by Tian et al. [9] involved performing 3D segmentation using three-level hierarchical deep learning along with pre-processing and post processing to the data. Their process started by performing pre-processing on the dental model to enhance the resolution of it by generating an octree model before being inputted into the three-level hierarchical neural network. The input goes through the first level of the hierarchy which focused on separating the gingiva from the teeth. This was followed by boundary refinement using dense-CRF technique as post processing to the data. Following that, the teeth were inputted into a 4-label classification network that classifies teeth into incisors, canines, premolars, and molars. Finally, the third level was a 2- label network that separated incisors to central and lateral, separated premolars into first and second, and separated molars into first and second. Finally, point cloud reconstruction was done to recombine the segmented gingiva with the classified teeth recreating the dental model with segmented teeth. This method required multiple pre- and post processing steps to enhance the segmentation results achieved.

In the work of Ashmore et al. [10], the authors first defined a coordinate frame to create a reference coordinate system to evaluate motion of molar tooth throughout a headgear treatment. To reproduce the coordinate frame in a consistent manner, certain points of reference in each cast were taken to generate it. The points include: a single point where the median raphe meets incisive papillae, set of points traced along the median raphe, 4 unique anatomic rugae points on each side of the palate, 4 unique anatomic points on the first permanent molar of each side, 3 points where the mandibular teeth meet with the maxillary teeth (usually cusp of molar and cusp of incisor) to be used to estimate the occlusal plane. The 4 points selected on each molar were averaged to create a centroid,

which was used to examine translations along the X, Y, and Z axes. Rotation was calculated by applying Procrustes onto the 4 point pairs selected on each molar tooth. In this process, the authors had to pre-select points representing the molar tooth (manually segment) on a cast. The number of points is small and the distance between the selected points is small as well since they are based on unique details in the occlusal anatomy. Due to the proximity of the 4 points, minor measurement errors could have a large impact on the calculated rotations. In our process, the method utilizes the entire body instead of using a handful of points and that removes the error of selection of points from the process and provides more accurate rotational measurements of teeth.

This thesis work proposes a completely autonomous system for teeth segmentation in two-dimensional color images without any manual intervention or pre-processing. Deep learning was implemented through a Fully Convolutional Neural Network (F-CNN) architecture to semantically segment individual teeth and palatal rugae in color images. A benchmark using different architectures was done to assess the effect of data augmentation on the semantic segmentation of teeth. Accuracy and associated metrics were defined to identify the best network architecture for semantic segmentation of maxillary teeth and rugae. Then, a labeled dataset consisting of 800 photographic images of actual patients taken at the occlusal view will be made publicly available along with the trained network.

Furthermore, this thesis work proposes 3D tooth segmentation that requires no pre- or post- processing of the data to further enhance the prediction results. A dataset consisting of 100 3D texture-colored models of actual patients with their projections is collected. The 3D meshes were cleaned, segmented manually, and labeled. An algorithm is developed to produce 100 2D projections of each scan with various lighting and perspective parameters. The generated 10,000 image set is labeled automatically using the projected anti-aliases coloring. Deep learning was implemented using a Fully Convolutional Neural Network (F-CNN) architecture designed for two-dimensional images to utilize its robustness on semantic segmentation. Utilizing the trained model, 2D segmentation of projections of unseen model is achieved. These segmented 2D images are used to back the segmented region on the 3D mesh, thus achieving 3D segmentation. Finally, motion measurement was performed on each individual tooth using rugae as a novel stable reference from which motion can be measured.

Chapter 2

Semantic Segmentation of two-dimensional Images

2.1 Dataset Collection

The proposed dataset consists of images compiled by the Orthodontics and Dentofacial Surgery Department at the American University of Beirut Medical Center (AUBMC). Two-dimensional (2D) images of the maxillary palate were taken at the occlusal plane with a single-lens reflex camera using a mirror as shown in Fig. 2.1. A typical image highlighting the palatal rugae is depicted in Fig. 2.2. The images were taken at different distances to generate diversity in the dataset and make the training more robust. Moreover, images with dental appliances were included in the data set to robustify the training as shown in Fig. 2.3. Finally, images of the same patient before and after treatment were also included to validate the network capability to segment teeth before and after treatment.

In contrast to previous related work, the images used in the paper not X-ray images, but RGB colored images. The images were saved in the PNG file format and with 480x320 pixel resolution. Two datasets were acquired, one original and one expanded.



Figure 2.1: Acquiring image of the maxillary teeth.

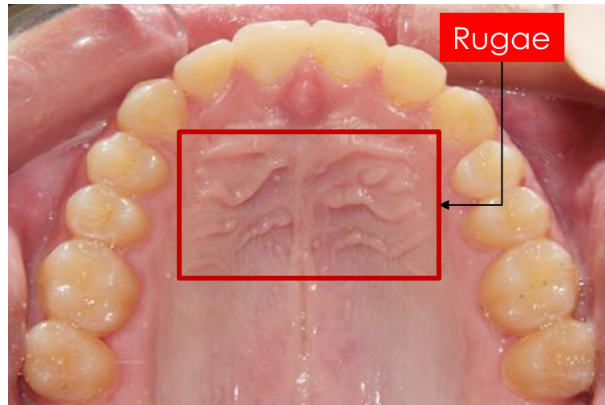


Figure 2.2: Image taken at occlusal view showing the rugae

The original dataset comprised 797 images. This set was split into two subsets 719 images and 78 images. The first subset was used to train a network to semantically segment family teeth, whereas the entire set of 797 images was used to semantically segment individual teeth. Note that, both networks were also trained to segment palatal rugae.

An additional image dataset composed of 47 pair of images is utilized solely to test the robustness of the trained network. Each pair of images is taken from the same patient before and after an orthodontic treatment as shown Fig. 2.6.

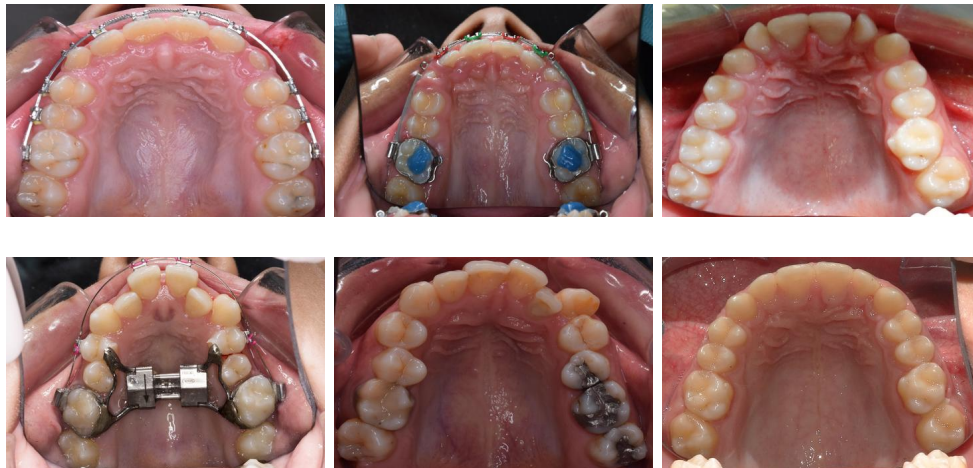


Figure 2.3: First two columns depict the original dataset while the last column depicts a pair of images from the expanded dataset.

2.2 Dataset Labeling Methods

Image labels serve as the ground truth for the training, validation, and testing of various neural networks architectures. The proposed labeling method labels the pixels in an image in the form of polygons drawn by the user to fit the shape of the object of interest as shown in Fig. 2.4a. The labeling was applied to the entire data set of $(797 + 2 \cdot 47 = 891)$ images.

2.2.1 Semantic Labeling

Labeling for semantic segmentation is done by assigning a class to every pixel in an image. The labeling of all the pixels in an image was performed in Matlab application [11] by creating polygons manually following the contour of the regions of interest. For each image in the dataset similar to the one shown in Fig. 2.4a, an associated image of similar dimensions was created to identify the labels by assigning different color to each label (Fig. 2.4b). In Figure 2.4c, the label is superimposed on the original image to show how the labels capture the family teeth. All labels and contours of the teeth and the rugae area were verified by orthodontists.

In this paper, two labeling schemes were used. The first scheme is comprised of 5 labels label to segment the rugae and the family of teeth: molars, premolars, canine, and incisors. The second labeling scheme consists of 23 defined labels: the rugae and the individual teeth number 1 through 22. Figure 2.4 depicts both labeling schemes.

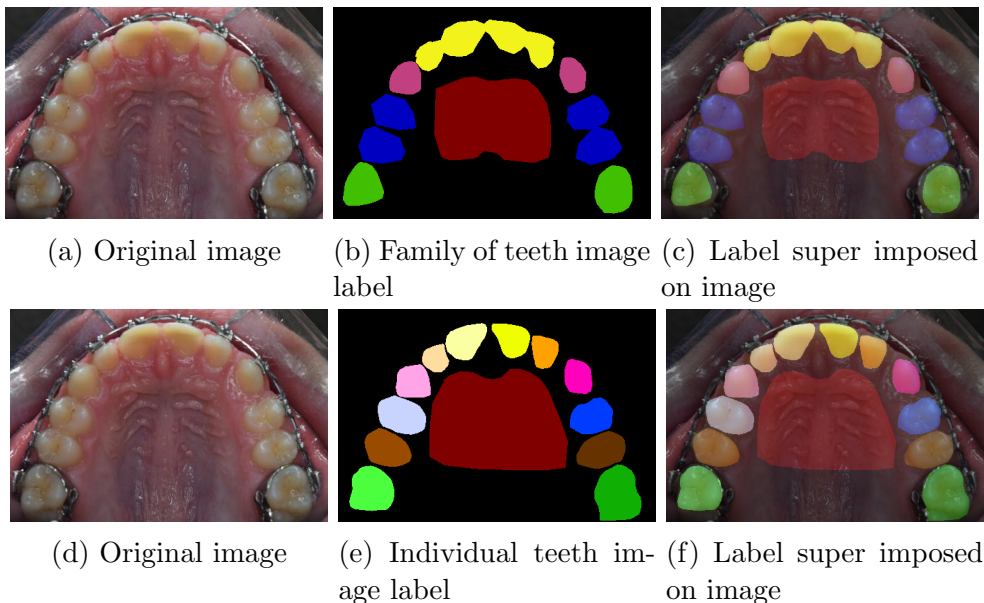


Figure 2.4: Labeling for semantic segmentation and the two labeling schemes.

2.2.2 Label Statistics

Semantically segmenting teeth and rugae is challenging due to variability in the size of the labels. It is clear from Fig. 2.4, that the typical size of the rugae label is substantially larger than the labels of individual teeth.

To gauge the variability of size of the labeled, the reader is referred to Fig. 2.5. For each labeling scheme and dataset combination, the number of pixels associated with each label were tallied. A bar chart depicting the count of pixels for each label is shown in Fig. 2.5. In Fig. 2.5a, the statistics of the family of teeth labeling scheme clearly shows rugae labels are the highest whereas the canine labels were the lowest. The number of labeled pixels of the molars, premolar, and incisors were comparable. In Fig. 2.5b, the statistics of the individual labeling scheme is depicted. Again, it’s clear that the number rugae pixels is the most, however, the number of pixels associated with individual adult teeth were comparable. One can also note that, the number of pixels associated with primary teeth and third molars are insignificant. Accordingly, one should expect low accuracy in segmenting the primary teeth and the third molars. Finally, in Fig. 2.5c, the statistics of the 47 pairs of images which was used for testing only is depicted. It is evident that the distribution of the teeth labels is similar to the training dataset in Fig. 2.5b. This will eliminate any possible data bias towards any label.

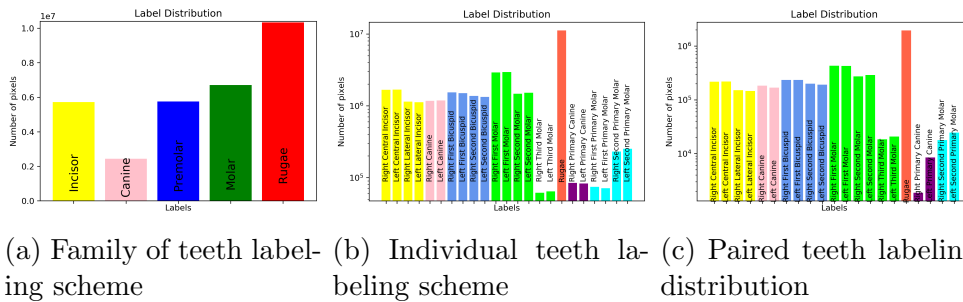


Figure 2.5: Statistics of the labeling schemes

2.3 Machine learning and Semantic Segmentation

Deep Learning is a model that is designed to analyze data similar to how a human would draw conclusions. Humans can identify features and patterns from huge amounts of data. The deep learning model aims to imitate that ability by using a layered structure of algorithms called an artificial neural network. The design of such networks were inspired by the biological neural network of the human brain. Using such networks, the algorithms are trained to find and identify patterns and



Figure 2.6: First two columns depict the original dataset while the last column depicts a pair of images from the expanded dataset.

features in massive amounts of data which allows it to make predictions on its own.

In typical semantic segmentation machine learning applications, a labeled dataset comprised of images and their associated labels are needed. These labeled images are fed into the chosen network with a specific architecture as input. The network in turn will produce a prediction of the label as an output. To assess the accuracy of the network, the predicted label is compared to the input label which is considered as ground truth. In most cases, the training process starts with initial values or what is referred to as a pre-trained network. This was the case for the semantic segmentation application Siam et al [12] and [13]. However, in our case, there exist no pre-trained network that is relevant to teeth segmentation. Accordingly, the main architecture is trained from scratch and without using any pre-trained front-end architectures.

Typical network architectures used for semantic segmentation comprise of encoding and decoding layers. The encoding layers are utilized for the feature extraction from the input image while the decoding layers are utilized for the pixel-wise predictions. Since to the best of the authors knowledge, no prior semantic segmentation for teeth exists, multiple architectures were investigated to determine the best performing semantic segmentation teeth.

2.3.1 Dataset Split

In typical machine learning applications, the data is split into three categories. Most of the data is used for training the network, that is, solving for the inter-

nal network parameters. Then, throughout the training process, the algorithm validated the training progress against a validation set which is disjoint from the training set. Finally, after the training is done, the trained network is testing against a verification set with is also disjoint from the previous two sets. Accordingly, the data is typically split into three sets: training, validation, and testing.

In this paper, the dataset with the family of teeth labeling scheme was split into 89% for training and validation (639 images) and 11% for testing (80 images). Moreover, the training and validation sets were split into 92% for training (586 images) and 8% for validation (53 images), respectively. As for the dataset with the individual teeth labeling scheme, it was split into 88% for training and validation (705 images) and 12% for testing (92 images). Moreover, the training and validation were split into 91% for training (641 images) and 9% for validation (64 images), respectively. This dataset split statistics are depicted in Table 2.1. Note that the dataset comprised of 47 pairs of before and after images of patients was entirely used for testing to assess the robustness of the trained model.

Table 2.1: The image count and labeling scheme of the original and expanded datasets.

	Train	Validate	Test	Total
Family of teeth labeling scheme				
Semantic Segmentation				
Count	586	53	80	719
Percent	82%	7%	11%	100%
Individual teeth labeling scheme				
Semantic Segmentation				
Count	641	64	92	797
Percent	80%	8%	12%	100%

2.3.2 Network Architectures

Up to our knowledge, there has been no network that was solely trained on semantically segmenting teeth. Hence, multiple architectures were tested to find the most suitable for our dataset. Our approach is similar to the network architecture study which was performed for urban scenes in [14].

The first architecture candidate was MobileUnet. This network is comprised of a small number of layers and hence it is relatively fast to train. This network has been wildly in many medical applications.

The second architecture which was tested is AdapNet. The architecture of this network designed to adapt to the environmental changes and focus less on the environment when predictions are made. For example, images taken in different

lighting conditions would not affect the predictive ability of this network. Hence, this architecture was a good candidate to include in the pool of networks to test. This network could give good results on our dataset since the images are taken at varying proximity which could be interpreted as changes in the environment with respect to the rest of the classes.

The third architecture to be considered was DenseNet. It is a model that uses features of various complexity levels. This architecture predicts smooth boundaries which enables the network to deal with limit datasets. The dataset used in this paper is not a large dataset when compared to known datasets which could include up to hundreds of thousands of images.

The final architecture which was considered is Segnet. The architecture of the Segnet that is designed to be efficient and to use a limited amount of memory. The network was designed primarily for understanding road scene understanding which requires the ability to perceive spatial-relationships. This capability is important in our case since our dataset has a variety of proximity from which the teeth are captured.

A brief description of the network architectures can be found in the Appendix.

2.3.3 Network Assessment

Semantic segmentation predictions are typically evaluated using Average mean Intersection over Union (Average mIoU). Note that for semantic segmentation, given two images representing ground truth and its associated prediction, one can define the IoU for a given class c such that

$$IoU(c) = \frac{\sum_i o_i == c \wedge y_i == c}{\sum_i o_i == c \vee y_i == c}, \quad (2.1)$$

where o_i is for predictions pixels, y_i for targets or labels pixels, \wedge is a logical *and* operation, and \vee is a logical *or* operation. This is summed over all the pixels i of the image pair. This is similar to how it was defined in [15]. The equation is also visually represented in Fig. 2.7a. The mean IoU averages the values of all the IoU's for all the classes in an image pair. For a dataset comprised of several image pairs, the Average mIoU is the average of the mean IoU for all image pairs.

In addition to the Average mIoU, several other metrics were computed to assess the accuracy of the trained model. These metrics are: pixel accuracy, and pixel precision. Pixel accuracy is the percentage of pixels in an image that are correctly classified with respect to the input ground truth pixels. This measure can be evaluated for specific classes or averages for all classes in an image or averages for a single for the entire dataset. Note that, per-class pixel accuracy can provide more information on the ability of the network to accurately segment especially for classes that occupy small regions in the image. Finally, precision is defined by the ratio of all the correctly detected pixels with respect to predicted

pixels. This metric describes how many correct predictions are compared to total predictions generated by the model.

Note that for semantic segmentation, given two images representing ground truth and its associated prediction, one can define the pixel accuracy for a given class c such that

$$Accuracy(c) = \frac{\sum_i o_i == c \wedge y_i == c}{\sum_i y_i == c}, \quad (2.2)$$

where o_i is for predictions pixels, y_i for targets or labels pixels, \wedge is a logical *and* operation, and \vee is a logical *or* operation. This is summed over all the pixels i of the image pair. The equation is also visually represented in Fig. 2.7b 7b.

Similarly, given two images representing ground truth and its associated prediction, one can define the precision for a given class c such that

$$Precision(c) = \frac{\sum_i o_i == c \wedge y_i == c}{\sum_i o_i == c}, \quad (2.3)$$

where o_i is for predictions pixels, y_i for targets or labels pixels, \wedge is a logical *and* operation, and \vee is a logical *or* operation. This is summed over all the pixels i of the image pair. The equation is also visually represented in Fig. 2.7c.

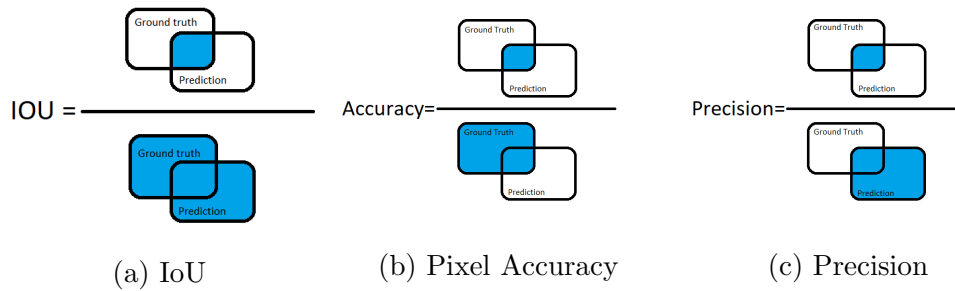


Figure 2.7: IoU, Accuracy and Precision Representation

It is worth noting that since the dataset exhibits class imbalance, that is, dissimilar class sizes, the Average mIoU is a better metric to assess the network prediction accuracy than the pixel accuracy and pixel precision. The class imbalance is exaggerated by the fact that the background (gingiva and non-teeth regions) and the rugae labels cover relatively larger areas than the rest of the classes, that is, the teeth; hence, as expect the high pixel accuracy does not translate to a more accurate semantic segmentation as was explained [16].

2.4 Results

2.4.1 Semantic Segmentation Training Results

Family of teeth labeling scheme

The four network architectures were trained on the original dataset with the family of teeth labeling scheme. In this case, not data augmentation was performed. The training results are depicted in Table 2.2. The SegNet and DenseNet exhibited the highest accuracy in terms of Average mIoU. The Average mIoU is 55.99 and 54.95 for the Segnet and DenseNet, respectively. Prediction results on the actual test images are shown in Fig. 2.8. The first row depicts the prediction using DenseNet, whereas the second row depicts the prediction by SegNet. For both networks, the predicted labels exhibited spatial shifts. This is an indication that the network model memorized the spatial position of the teeth rather than segmenting them. To mitigate this issue, data augmentation was employed.

Table 2.2: Model Architecture Comparison on Tooth Family Labels

Model	FC-DenseNet56	SegNet	MobilUNet Skip	AdapNet
Label Name	Score	Score	Score	Score
Incisor	70.23	71.32	69.15	69.7
Canine	56.14	55.66	51.65	49.59
Premolar	67.36	70.36	66.76	57.76
Molar	59.54	66.02	66.89	53.17
Rugae	82.92	82.67	80.17	81.12
Void	89.25	88.47	88.15	90.88
Average Accuracy	83.50	83.49	82.63	83.43
Average Precision	83.80	83.47	82.80	84.58
Average mean IoU score	54.95	55.99	53.68	53.23

Individual teeth labeling scheme

Given the networks trained using the family of teeth scheme exhibited spatial memory, data augmentation is required to improve the network’s accuracy. In this paper, two data augmentation methods were employed: the first involved rotating images (and their labels) and then adding them to the original set (Fig. 2.9). The second data augmentation targeted changing the perspective of the images (and their labels) by shearing them as shown in Fig. 2.9e.

To assess the value of the data augmentation the top two performing architectures were re-trained on the full dataset. Recall that the training, validation, and testing data split is shown in the last two rows of Table 2.1. The two sets of data augmentation were tested individually and incrementally giving rise to six

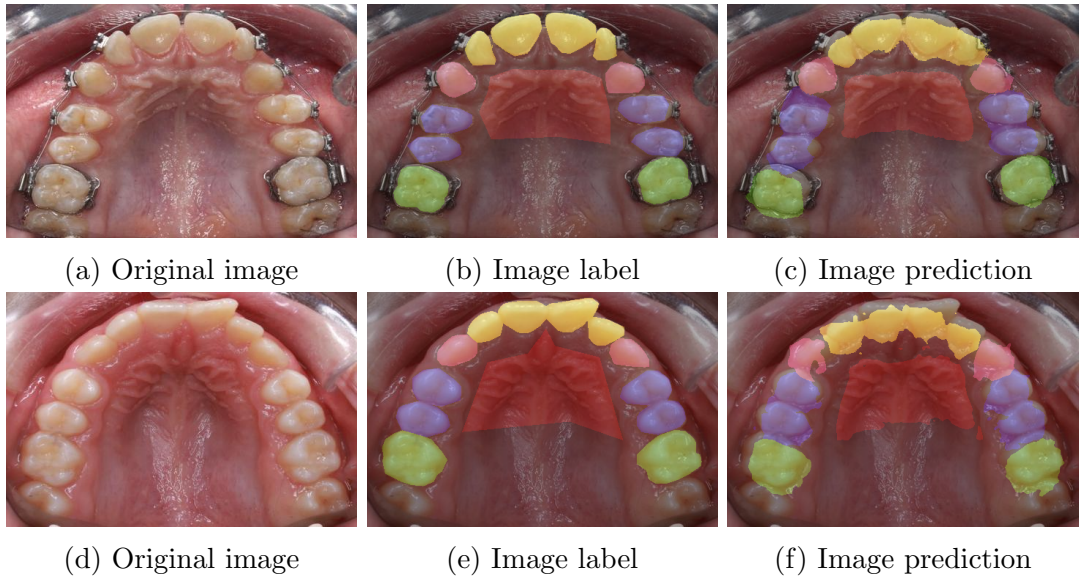


Figure 2.8: Family of Teeth average sample result for DenseNet in 2.8a, 2.8b, and 2.8c. Family of Teeth average sample result for SegNet in 2.8d, 2.8e, and 2.8f

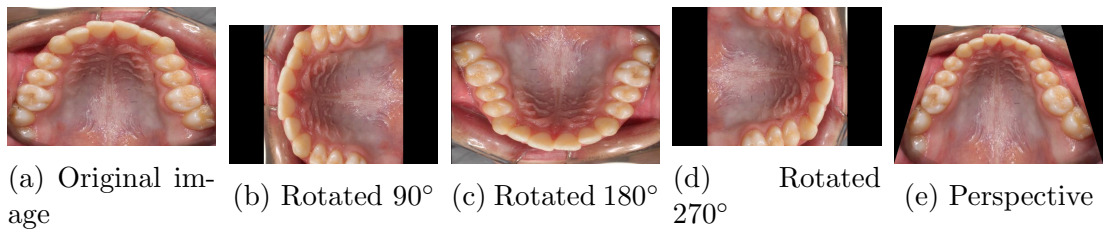


Figure 2.9: Data augmentation via rotation and perspective shrinking.

training combinations which are shown in Table 2.3. The highest Average mIoU was on the dataset that used rotation data augmentation only. The perspective data augmentation did not improve the training accuracy. This could be possible due to the fact that the images already have perspective variability since they were taken using actual cameras. Accordingly, only the rotation data augmentation was used in the dataset for the final training. Consequently, the four original architectures were retrained on the full dataset (including the rotation data augmentation) using the individual teeth labeling scheme and.

The results of the trained architectures are shown in Table 2.4. SegNet remained the best architecture with an Average mIoU of 86.66 and a 95.19% accuracy. By using the per-class accuracy of teeth, we can deduce that on average 1/20th of a tooth will be miss-labeled.

The results of DenseNet and SegNet performed on the test dataset using the rotation augmented dataset are shown in Figs. 2.10 and 2.11, respectively. In both figures, the first row depicts the worst prediction, the middle row depicts an

average prediction, and the best prediction result is displayed in the third row.

Table 2.3: Model Architecture Comparison on Specific Tooth Labels

Model	Avg. Accuracy	Avg. Precision	Avg. mean IoU score
DenseNet	81.84	82.41	45.89
SegNet	82.32	81.93	49.53
DenseNet-Rotated	95.00	95.23	85.40
SegNet-Rotated	95.19	95.40	86.66
DenseNet-Rotation & Perspective	94.68	94.77	84.19
SegNet-Rotation & Perspective	65.09	81.28	7.90

2.4.2 Semantic Segmentation Application Results

Having identified that SegNet is the most accurate network to semantically segment images of teeth, the robustness of the trained model is testing on a third dataset. Recall that this data set is comprised of 47 pairs of images that capture the before and after treatment images of a patients.

Network Accuracy

Statistical analysis done on the labels of this third dataset similar to the analysis done on the previous two datasets. The number of pixels per class in each is computed as shown in Fig. 2.12a for the Right Central Incisor class. The pixel distribution of each class is in Fig. 2.12. It is evident that the primary teeth and the third molars classes had the fewest pixels (close to 10% only) in all the images as shown in Fig. 2.12o, 2.12p, 2.12q, 2.12r, 2.12s, and 2.12t.). Accordingly, two Average mean IoU’s were computed, one that included all classes and the second that ignore the low-occurrence classes. Additionally, to focus on the teeth segmentation accuracy of the network, the rugae class is not included in the second computation of the Average mean IoU. of the This second Average mean IoU is referred to as the “Teeth Only IoU” in Table 2.5.

The Teeth Only IoU value for the 47 pair dataset is 86.2% which only includes the teeth labels mentioned in Table 2.5. The average mean IoU of all teeth (including primary and third molars) and rugae is 82.9%. This value is expected to be lower due to the rugae boundaries not being consistently defined during the labeling process.

Network Robustness

To validate the robustness of the trained network, the accuracy of prediction is gauged for the pre-treatment and post treatment images separately. The two

Table 2.4: Model Architecture Comparison on Specific Tooth Labels with Rotation for all Architecture Candidates

<i>Model</i>	FC-DenseNet56	SegNet	MobilUNet Skip	AdapNet
Right Central Incisor	94.12	93.75	93.45	93.27
Left Central Incisor	94.14	94.39	93.04	92.48
Right Lateral Incisor	91.42	91.61	91.02	90.99
Left Lateral Incisor	93.24	92.28	91.53	91.09
Right Canine	92.06	92.36	92.70	91.78
Left Canine	92.44	92.73	91.62	90.81
Right 1st Bicuspid	93.28	94.18	93.19	92.88
Left 1st Bicuspid	94.23	93.25	92.36	91.96
Right 2nd Bicuspid	88.86	91.49	89.84	90.87
Left 2nd Bicuspid	93.04	92.03	90.99	90.06
Right 1st Molar	93.29	94.58	94.20	92.80
Left 1st Molar	92.72	92.62	94.46	90.70
Right 2nd Molar	93.03	92.39	92.66	91.32
Left 2nd Molar	93.55	94.46	92.30	91.00
Right 3rd Molar	96.76	97.22	96.70	97.38
Left 3rd Molar	95.51	96.20	95.74	95.90
Right Primary Canine	98.09	99.68	97.83	99.77
Left Primary Canine	96.69	98.84	98.26	97.99
Right 1st Primary Molar	98.70	99.23	98.73	98.64
Left 1st Primary Molar	97.08	98.54	96.77	97.26
Right 2nd Primary Molar	98.61	99.58	98.76	98.17
Left 2nd Primary Molar	97.55	98.81	98.60	97.40
Rugae	88.65	88.77	89.36	88.08
Void	97.17	97.07	96.68	96.82
Average Accuracy	95.00	95.19	94.82	94.55
Average Precision	95.23	95.40	95.03	94.76
Average mean IoU score	85.40	86.66	84.92	84.42

results were checked if they were statistically different using the chi-square goodness of fit test.

Recall that, the chi-square goodness of fit test is a non-parametric test that is used to compare the observed sample distribution with the expected distribution. This test determines if the distribution of the accuracies between the set of pre-treatment images and the set of post-treatment images are statistically similar or not. The x^2 test statistic is calculated by:

$$x^2 = \sum \frac{(O - E)^2}{E}, \quad (2.4)$$

where O corresponds to the observed sample and E corresponds to the ex-

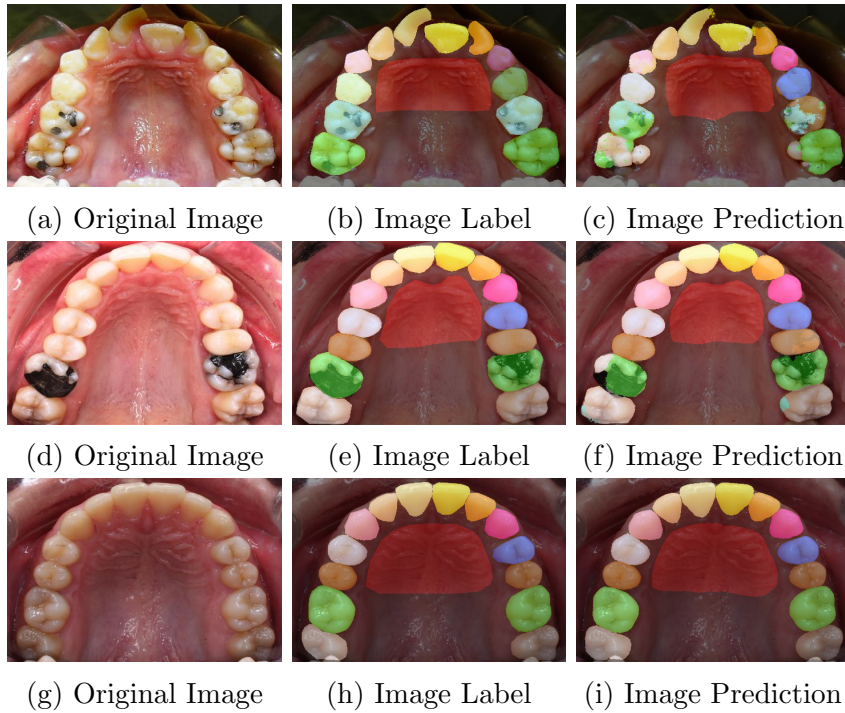


Figure 2.10: Individual Teeth Labeling results for DenseNet including rotation dataset displaying worst result in the first row of figures 2.10a, 2.10b, and 2.10c; average result in the second row of figures 2.10d, 2.10e, and 2.10f ;and Best result in the third row of figures 2.10g, 2.10h, and 2.10i

pected sample. This gives us the chi square statistic. Using the chi square statistic and the number of the samples used, the p-value which is the probability that any difference between the observed distribution and the expected distribution is due to random chance can be calculated. The common acceptable threshold for the p-value is 0.05. A value greater than 0.05 would mean that the difference is not statistically significant, otherwise the two distributions are deemed different.

The Average mean IoU was calculated for the Teeth labels for both groups, pre and post treatment, separately and displayed them in Table 2.6. Given that there are 14 classes, then there will be 14 values for each set. The pre-treatment was considered to be as the observed set while the post-treatment as the expected set. After applying the chi-square test and checking the p-value, the p-value was calculated to be 0.9989 which is higher than the significance level chosen. Hence, the prediction values in both sets are not statistically different. This shows that our model is robust regardless of the input images were taken from pre-treatment set or the post-treatment set.

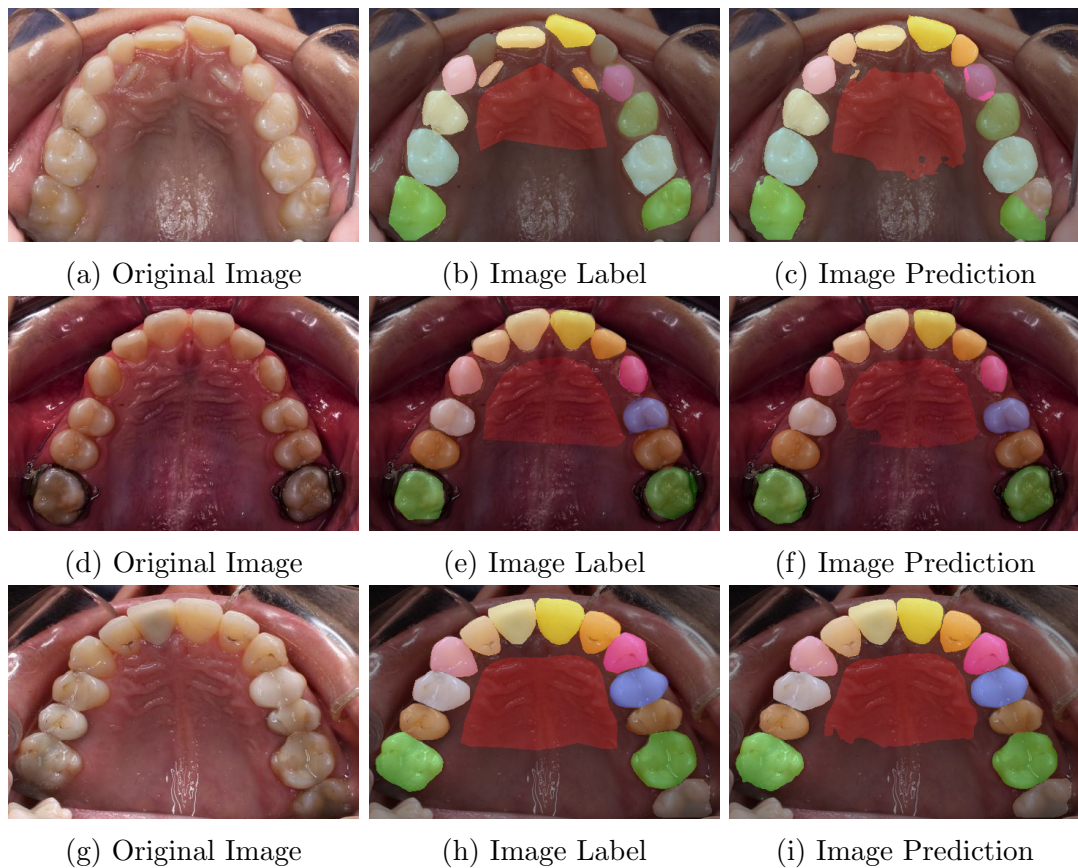


Figure 2.11: individual Teeth Labeling results for SegNet including rotation dataset displaying worst result in the first row of figures 2.11a, 2.11b, and 2.11c; average result in the second row of figures 2.11d, 2.11e, and 2.11f ;and Best result in the third row of figures 2.11g, 2.11h, and 2.11i

2.4.3 Limitations

It is worth noting that the rugae label was the lowest in accuracy. This was due to the fact that the rugae area is not well-defined area as opposed to the teeth. This fact is a reason why the labels for the rugae class in every image varies especially that several people performed the ground truth labels, and the defining boundary can vary between individuals.

For the sample image exhibiting missing teeth, the trained model mislabeled the existing teeth. Specifically, if one of the premolars is missing, the existing premolar could be mislabeled. For instance, in Fig. 2.14b the left premolar (situated on the right side of the image) is labeled correctly as the first premolar and colored in blue while the right premolar (situated on the left) is labeled incorrectly as a second premolar colored as brown. Orthodontists are able to label these teeth correctly because of the small spacing between the left canine and premolar, which hints that there was a premolar previously in that space and

Table 2.5: Dataset Analysis Results

Name	Value
Right Central Incisor mIoU	86.2%
Left Central Incisor mIoU	85.8%
Right Lateral Incisor mIoU	84.7%
Left Lateral Incisor mIoU	82.6%
Right Canine mIoU	86.5%
Left Canine mIoU	81.6%
Right 1st Bicuspid mIoU	90.6%
Left 1st Bicuspid mIoU	90.9%
Right 2nd Bicuspid mIoU	87.0%
Left 2nd Bicuspid mIoU	86.7%
Right 1st Molar mIoU	91.6%
Left 1st Molar mIoU	89.1%
Right 2nd Molar mIoU	81.4%
Left 2nd Molar mIoU	83.2%
Rugae mIoU	78.6%
Average Accuracy	93.8%
Average Precision	94.3%
All Teeth and Rugae Average mIoU	82.9%
Teeth Only Average mIoU	86.2%

has been removed as opposed to the right side. The prediction of both teeth is depicted in Fig. 2.14c. The right premolar was predicted correctly owing to the spacing; however, the left premolar was predicted falsely and was classified as a second premolar.

The robustness check confirms that there is no bias of predictions towards the post treatment. This is a valuable attribute of the trained model since most of the images in the pre-treatment have misaligned and crowded teeth, yet the trained model was able to correctly segment them.

2.5 Conclusion

In this document, a semantically labeled maxillary teeth dataset taken at the occlusal view was introduced. The dataset consisted of colored images in contrast to previous work that used X-ray images. Machine learning methods were applied to identify the best network architecture to be used in semantically segmenting these images. The best network to segment maxillary teeth and rugae is SegNet which yielded an accuracy of 95.19% and an Average mIoU of 86.66%. It is worth

Table 2.6: Pre and Post IoU Values for teeth

Labels	Pre- Treatment Average mIoU Values	Post Treatment Average mIoU Values
Right Central Incisor	86.9%	85.5%
Left Central Incisor	86.0%	85.6%
Right Lateral Incisor	84.5%	84.9%
Left Lateral Incisor	82.2%	83.0%
Right Canine	84.5%	88.5%
Left Canine	78.4%	84.7%
Right 1st Bicuspid	89.3%	92.0%
Left 1st Bicuspid	90.7%	91.2%
Right 2nd Bicuspid	87.5%	86.5%
Left 2nd Bicuspid	89.3%	84.2%
Right 1st Molar	91.8%	91.4%
Left 1st Molar	87.5%	90.6%
Right 2nd Molar	79.1%	83.5%
Left 2nd Molar	78.1%	87.6%

noting that the developed method required no post-processing nor pre-training. The model robustness was also verified by applying to a test set consisting of pre-treatment images and post-treatment images. The robustness results yielded an Average mIoU value of 86.2% for the teeth only classes.

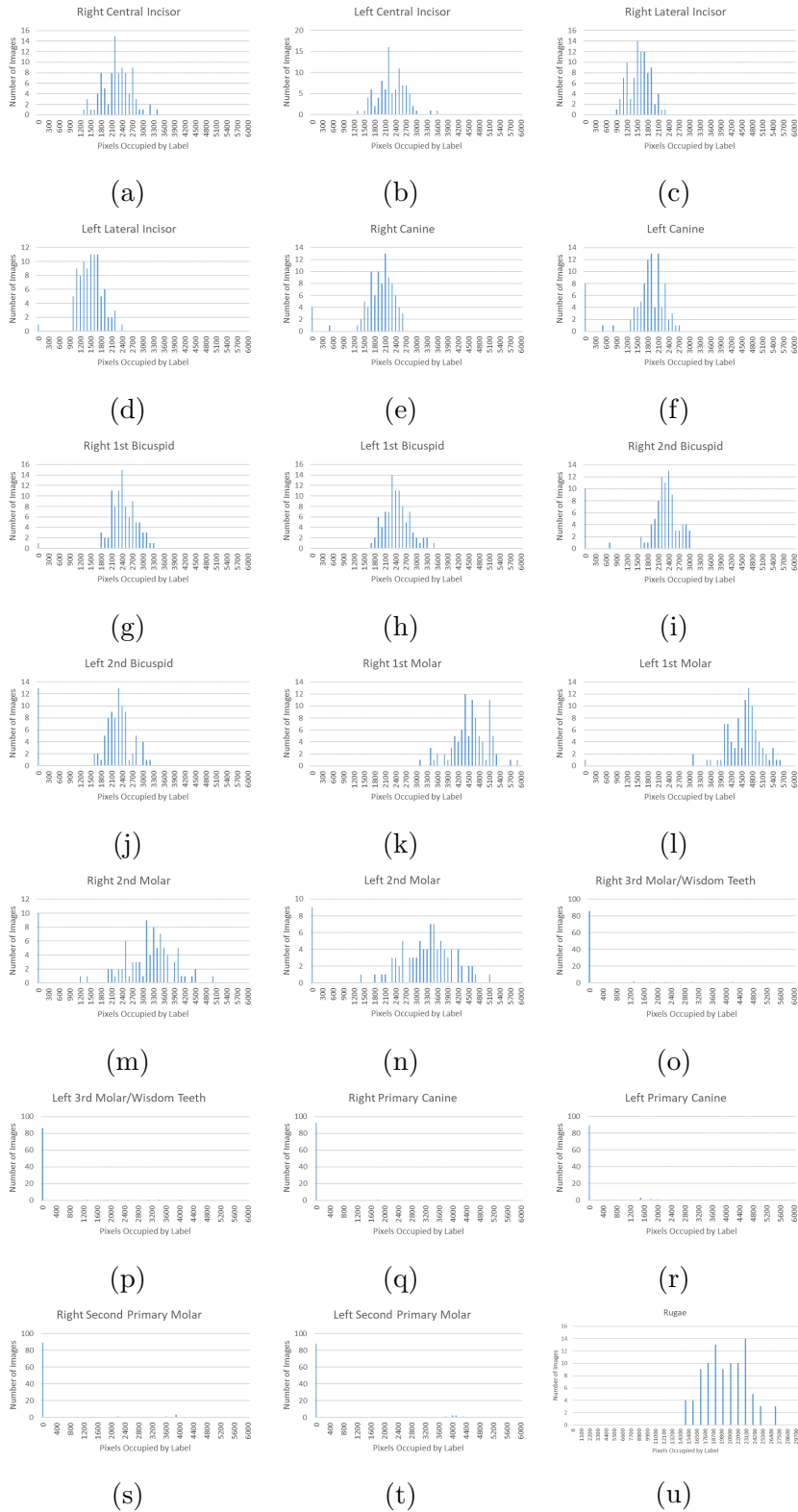


Figure 2.12: Graphs of the label distribution over the set analyzed



Figure 2.13: The first column shows the original image from the pre-treatment set followed by its ground truth label in the row below and its prediction from the model in the final row. The second column shows the same order for an image from the post-treatment of the same patient.

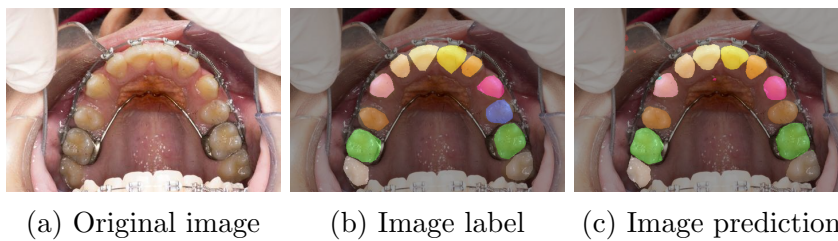


Figure 2.14: Miss labeling due to missing teeth

Chapter 3

Semantic Segmentation of Three-dimensional Textured Scans

3.1 Data collection, Annotation and Augmentation

3.1.1 Dataset Collection

In this proposed work, a dataset is generated, labeled and augmented from the set of texture-colored 3D meshes, see Fig. 3.1. Texture-colored 3D meshes are provided by the Orthodontics and Dentofacial Surgery department at the American University of Beirut Medical Center (AUBMC). The 3D dataset consists of 3D scans of the upper maxillary jaw of patients, see Fig. 3.2. The scans were taken using an orthodontic 3D scanner probe and amount to a total of 100 colored meshes in the PLY format.

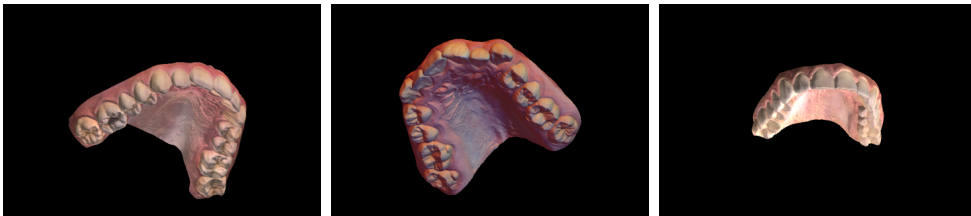


Figure 3.1: 3D scan samples

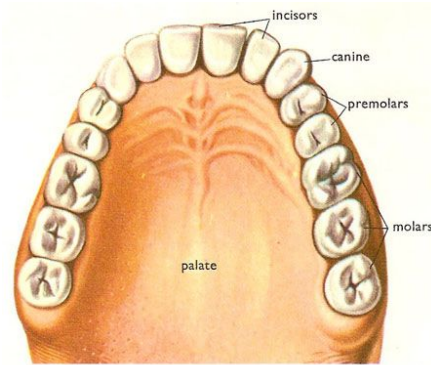
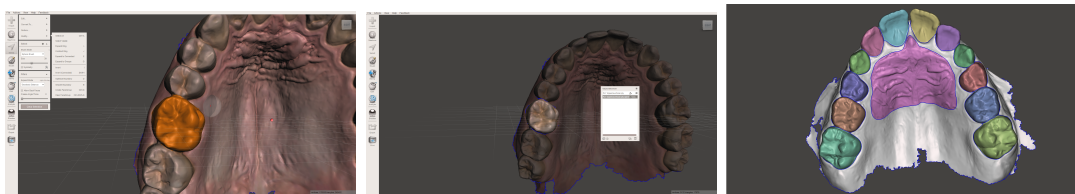


Figure 3.2: Image of the maxillary teeth

3.1.2 Dataset Annotation

Annotating 3D mesh for semantic segmentation is done by assigning a class to every vertex in the mesh. The three-dimensional meshes were segmented manually with the help of a program called Meshmixer. Using this program, the user can select the vertices they desire to separate into a different component along with the option of creating a smooth boundary separator, see Fig. 3.3. This was done to every single group that is differentiated as a separate label in the next step.



(a) Selection of vertices (b) Separation of selected objects (c) Segmented mesh

Figure 3.3: Meshmixer steps of selection of vertices on the left, single segmentation in the middle, and the final separation on the right

After having the mesh contain separated components, the vertices in each component were assigned to a class using a program called Meshlab. Each label group is identified by color using this method. This scheme consists of a total of 27 defined labels.

Statistics of the distribution of labels throughout the dataset are shown in Fig. 3.4. A graph displaying the ratio of pixels to vertices is shown in Fig. 3.5. This shows that the ratios are averaged at a value of 25. It is evident from the ratio distribution that there is no misrepresentation of any label (ratio too low compared to the other ratios) when projected from 3D to 2D. Finally, the

varying ratios are due to the randomness of the projection viewpoints. It is also evident that most of the labels that are low fall under the primary teeth category and third molars. Hence, it is expected that these labels might generate lower accuracy results than the rest of the labels.

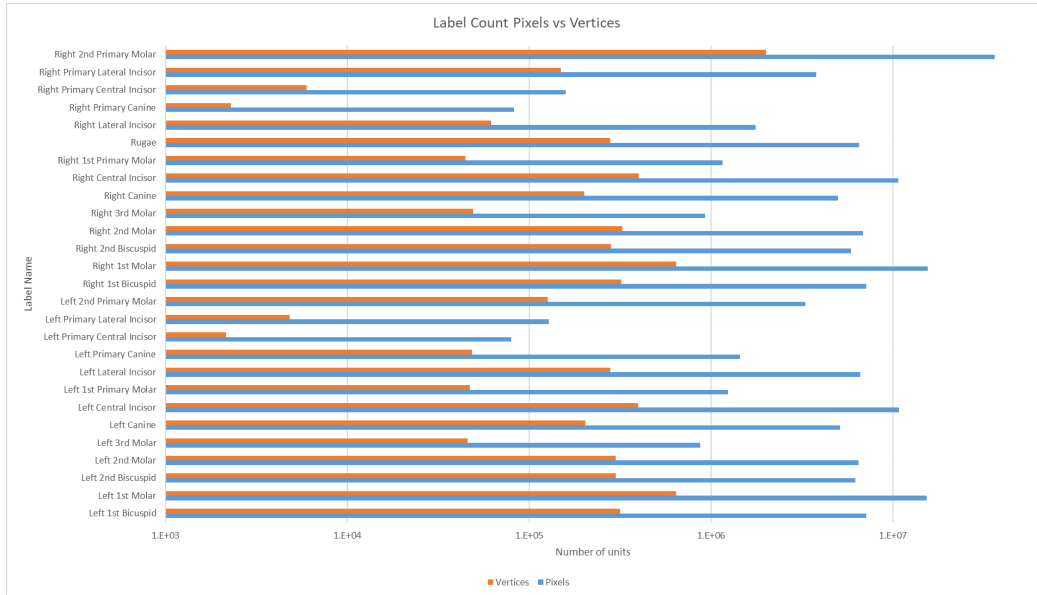


Figure 3.4: Number of pixels for each label compared with number if vertices for each label

3.1.3 Dataset Augmentation

The 3D textured mesh is transformed into 2D snapshots taken from various angles using Mathematica. The rotation angle covers a span of 180 degrees that is generated randomly at every iteration. Hence, it is restricted to the frontal face of the teeth instead of taking snapshots of the back side of the mesh. This was done so that the model goes through various angles and avoids learning predictions on a set sequence of angles only.

Furthermore, the 2D generation process was augmented so that the code takes the images under various lighting conditions. The lighting conditions included natural default light by the software, ambient light which is a bright light onto the mesh, and varying intensity with location change from which the light can shine from. For the third choice of lighting, two positions were chosen for the light location; however, the light intensity was randomly generated at every iteration. This was decided to be done since each software renders the mesh differently. Hence, disregarding this slight change in rendering could affect the results if the model learns under a single rendered color. Therefore, this trained the model on having different shadings and lighting of meshes from different softwares and

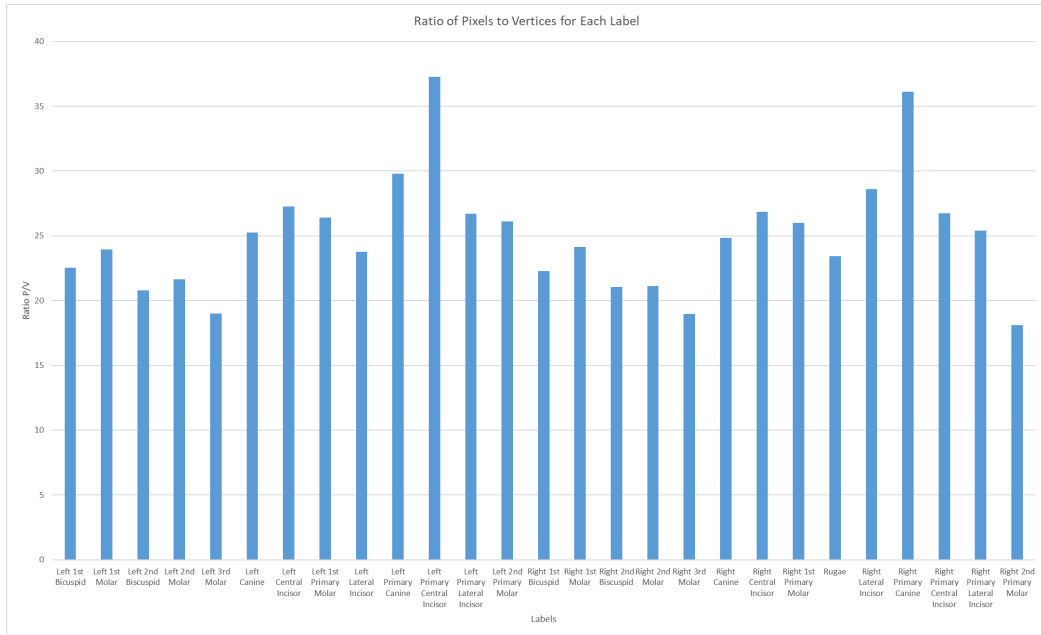


Figure 3.5: Ratio of pixels to vertices for each label

makes it more robust and independent of the lighting from the rendering software. (Fig. 3.6).

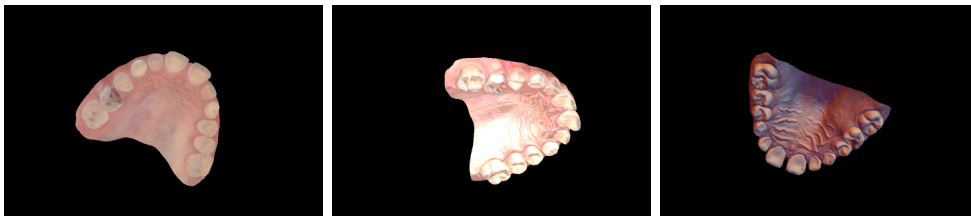


Figure 3.6: 3D Scan Samples showing different light conditions

Finally, another added augmentation was having the proximity of the camera to the mesh also randomly generated with the camera angle. This would allow the mesh to be taken at various distances in various angles for more randomness and general robustness of the model (Fig. 3.7). For the dataset, the program generated 100 two-dimensional images per mesh which counts for a total of 10000 images to be trained on.

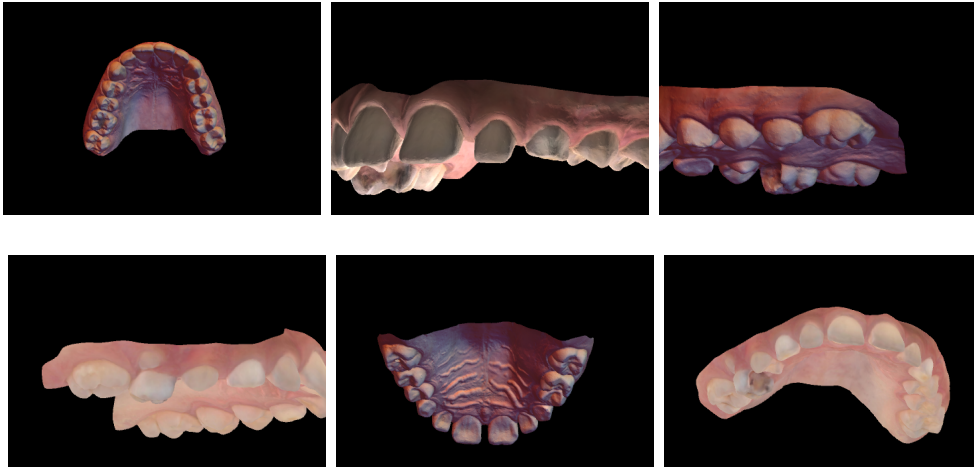


Figure 3.7: 3D Scan Samples showing different camera proximities

3.2 Deep Network Design and Training

3.2.1 Network Architecture

The architecture which was considered is Segnet. A brief description of the network architectures can be found in the Appendix. This architecture was chosen since it yielded the best results with our two-dimensional dataset. Hence, this architecture was concluded to be suitable for segmentation of teeth. Building upon the robustness of this network, the segmented two-dimensional images will be back projected onto the three-dimensional mesh.

In addition, two attention layers were implemented which are channel attention and spatial attention. The effect of implementing these layers was compared with the original network to check if the results are enhanced with their addition. The channel attention aims to learn a one-dimensional weight and assign it to a corresponding channel by going through the relationship between each channel of the feature map. The spatial attention uses the relationships between different spatial positions to learn and assign a two-dimensional spatial weight to a corresponding spatial position. This helps the model to learn more representative features. Two different implementation augmentation were made and tested on. The first one was in a way such that the feature map was fed initially into a channel attention module to refine the features in channels. Following that, the refined channel feature map was fed into the spatial attention module for spatial axis refinement. In the end, the final layer layer processes (Softmax) are applied and the predictions are generated.[17]. The second implementation was having the attention layers at the transition between the encoding layers and decoding layers (center of structure instead of at the end).

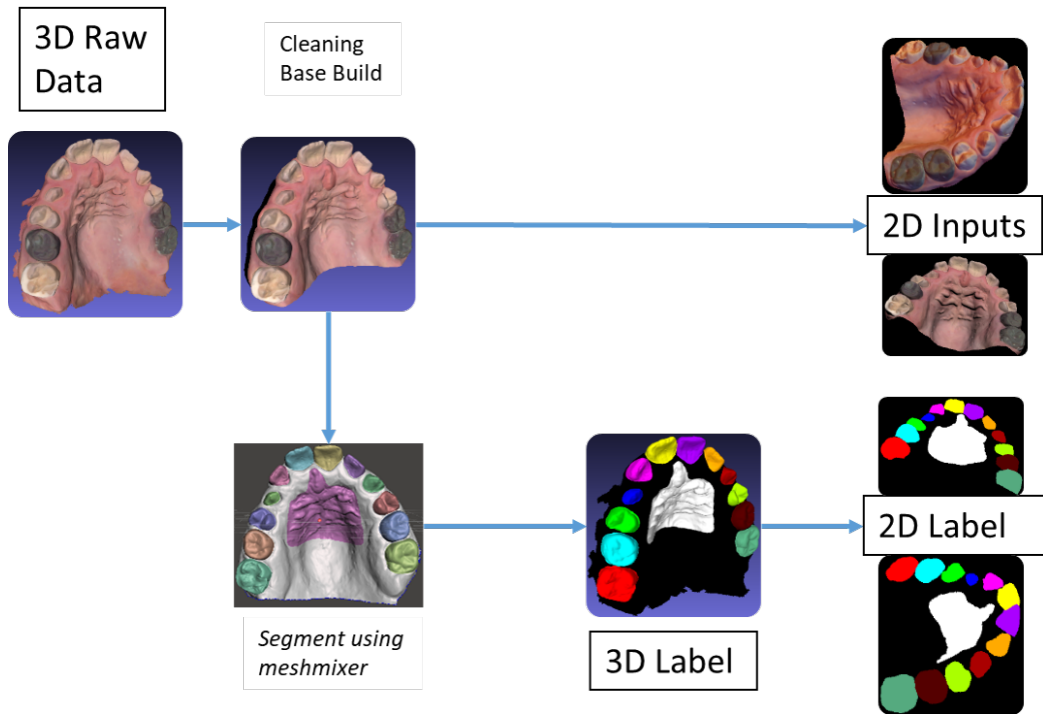


Figure 3.8: Flow Chart of the Annotation and 2D projections

3.2.2 Training Process

Two separate training sets were done while testing for the effect of the attention layers on the architecture. The training was done on a partial dataset of 5000 images that is part of the 10000 images dataset (the full dataset). This was done to have faster training while comparing attention layer implementations. Once the conclusion is reached, a final training with the full dataset was done on the architecture best for the dataset.

For the training on the partial dataset, the dataset was split into 83% for training and validation (4167 images) and 17% for testing (833 images). Furthermore, the training and validation set were split into 92% for training (3819 images) and 8% for validation (348 images). The hyper parameters used for the architecture training were 150 epochs, batch size of 1, learning rate of 0.0001, and decay of 0.995.

For the training on the final dataset, the dataset was split into 80% for training (8000 images), 10% for validation (1000 images) and 10% for testing (1000 images). Furthermore, the test and validation sets were complete unseen scans. In the previous training, the test and validation were unseen projections of scans already in the training. However, the final training with the final dataset had unseen mesh scans in the test and validation which is even harder for the model to predict. The hyper parameters used for the architecture training were 120

epochs, batch size of 1, learning rate of 0.0001, and decay of 0.995.

3.3 Rasterization

After generating the predictions from the trained model, the two-dimensional predictions are back projected onto the three-dimensional mesh. Using a software called Meshlab, 2D projections were generated from a fixed set of camera positions that capture the entire mesh, see Fig. 3.9. The projections were entered into the trained model to generate their two-dimensional segmented predictions on them. The predictions on the images, known as rasters, can be back-projected onto the mesh using the saved camera locations. The new generated mesh will be segmented based on color, see Fig. 3.10. Each class can be identified using the color designated to it.

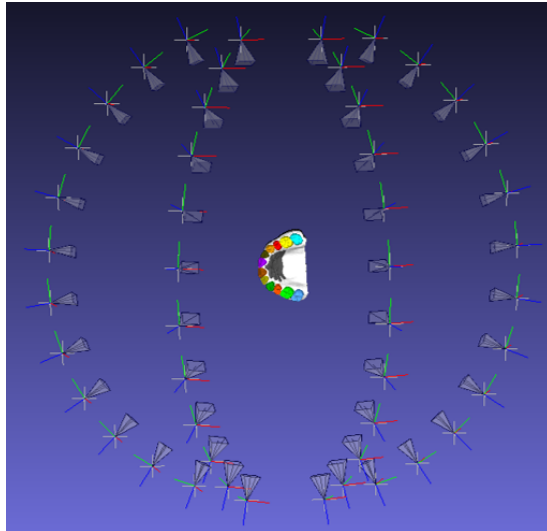


Figure 3.9: Sample 3D mesh with Camera locations surrounding it

3.4 Results

To assess the value of all three architectures(original and both attention implementations), the architectures were trained on the augmented dataset. The training results are shown in tables 3.1 and 3.2. The first table shows results done on unseen projections of scans already in the training. The architecture that had the attention placed at the end had the best results with an average accuracy of 99.37 % and an average mIoU of 88.45. The second table shows prediction results done on unseen projections of unseen 3D scans. The architecture that had the attention placed at the end had the best results as well with an

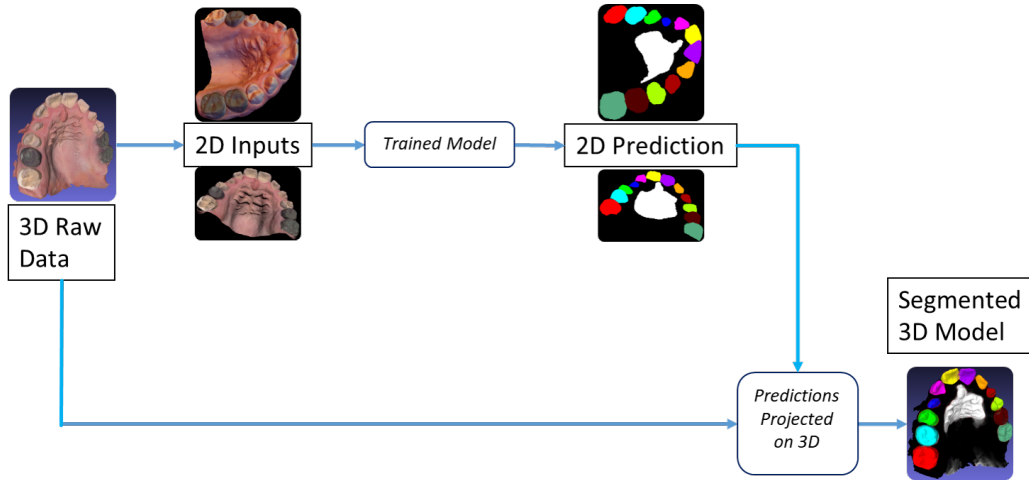


Figure 3.10: Flow Chart of the 3D segmentation process

average accuracy of 98 % and an average mIoU of 79.01. Prediction results of the first table are shown in Fig. 3.11. In these results, it is evident that in both architectures that include attention the prediction result is enhanced. Prediction results of the second table are shown in Fig. 3.12. In those results, it was within our expectations that the predictions will not be fully accurate since the dataset used for training was still insufficient in size. However, it can be seen that the architecture that has the attention layers similar to paper [17] are showing more promising results in terms of refinement of the labels. Hence, it was concluded that the architecture that has attention layers included in the end was the best one to be used for training the final dataset.

Table 3.1: Model Architecture Comparison on Test Set

Model	SegNet Original	Segnet with Attention Layers (End)	Segnet with Attention Layers (Centered)
Avg. Accuracy	99.29	99.37	99.34
Avg. mean IoU score	87.50	88.45	88.29

Table 3.2: Model Architecture Comparison on Unseen Scan set

Model	SegNet Original	Segnet with Attention Layers (End)	Segnet with Attention Layers (Centered)
Avg. Accuracy	97.80	98.00	97.60
Avg. mean IoU score	76.68	79.01	76.35

After identifying the best architecture to use, the final dataset was trained on that architecture. The prediction results were displayed in Fig. 3.13. The first

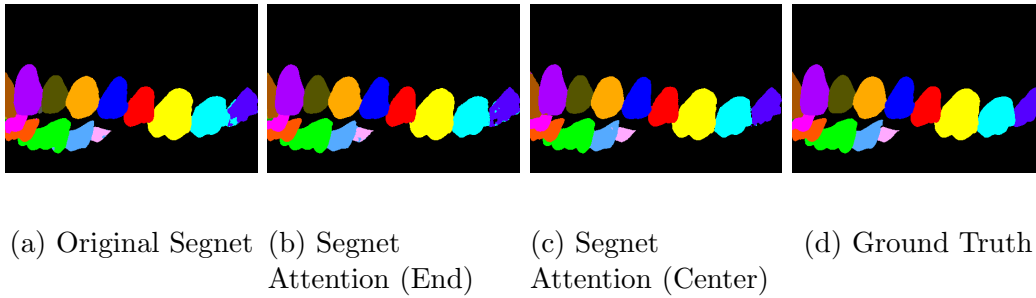


Figure 3.11: Attention Layer Comparison on Unseen Projection of a trained scan

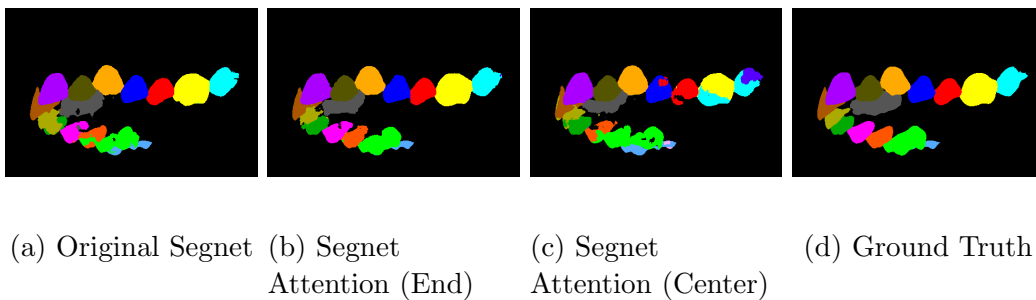


Figure 3.12: Attention Layer Comparison on Unseen Projection of an Unseen scan

row depicts the worst prediction, the middle row depicts an average prediction, while the third row depicts the best prediction result. The training results were shown in 3.3. It can be seen that there are certain labels that are of low value compared to the rest. These labels are mostly the labels that were less commonly found compared to other labels. Most of these labels fall under the category of primary teeth and the third molars. Hence, an additional term was included in the table which was the Adult Teeth Average Mean IoU Score which is similar in calculation to the average mean IoU score but excludes third molars, primary teeth, rugae and background from the calculation. This metric has a value of 84.26 and is close to the average mean IoU score of 85.41. The final training has an average accuracy of 98.69 % and Average mIoU score of 85.41. It is important to note that these results were done on unseen projection of unseen scans by the model.

The remaining step for the segmentation is applying the predictions onto the 3D mesh using the rasterization process. The needed projections are generated, entered into the trained model and have the predictions applied on them. Following that, the predictions were back-projected on the mesh using the camera locations linked to the images, see Fig. 3.14.

Table 3.3: Model Training Results on the Final Dataset

<i>Model</i>	Mean IoU Scores
right 3rd molar	70.91
right 2nd molar	68.22
right 1st molar	83.12
right 2nd bicuspid	89.52
right 1st bicuspid	89.19
right canine	81.47
right lateral incisor	83.72
right central incisor	94.76
left central incisor	94.12
left lateral incisor	78.93
left canine	87.23
left 1st bicuspid	85.71
left 2nd bicuspid	84.66
left 1st molar	86.26
left 2nd molar	72.72
left 3rd molar	66.16
right 2nd primary molar	77.25
right 1st primary molar	72.57
right primary canine	73.27
right primary lateral incisor	0.05
right primary central incisor	-
left primary central incisor	-
left primary lateral incisor	0.00
left primary canine	80.10
left 1st primary molar	59.21
left 2nd primary molar	77.51
rugae	85.87
background	99.34
Average Accuracy	98.69
Average Precision	98.71
Average mean IoU score	85.41
Adult Teeth Average Mean IoU Score	84.26

3.5 Conclusion

This document introduces a maxillary teeth dataset of 3D mesh projections and motion measurement of tooth using the rugae as a stable reference. The dataset consisted of 100 texture-colored scans which generated a total of 10,000 projec-

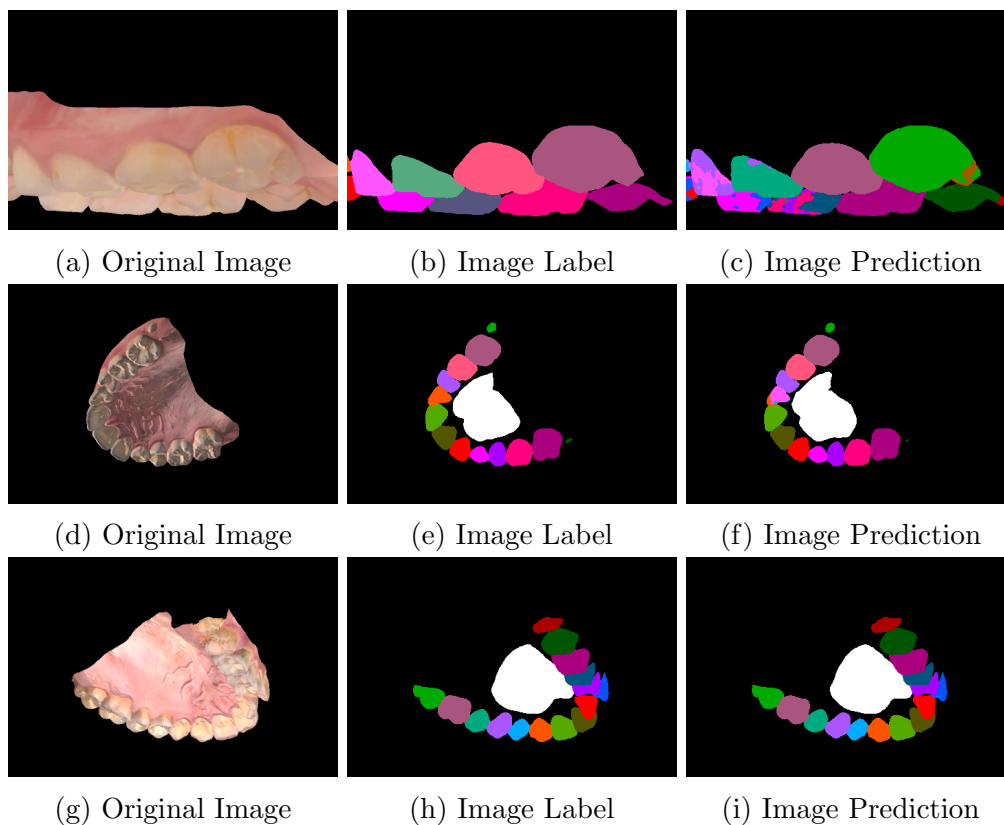


Figure 3.13: Final Training results displaying worst result in the first row of figures 3.13a, 3.13b, and 3.13c; average result in the second row of figures 3.13d, 3.13e, and 3.13f ;and Best result in the third row of figures 3.13g, 3.13h, and 3.13i

tions to be used for training. Attention layers were tested and added onto the architecture enhancing the precision of the results. The best network yielded in an accuracy of 98.69 % and Average mIoU of 85.41. It is also worth noting that the developed method required no post-processing nor pre-training as compared to related work in the 3D segmentation domain.

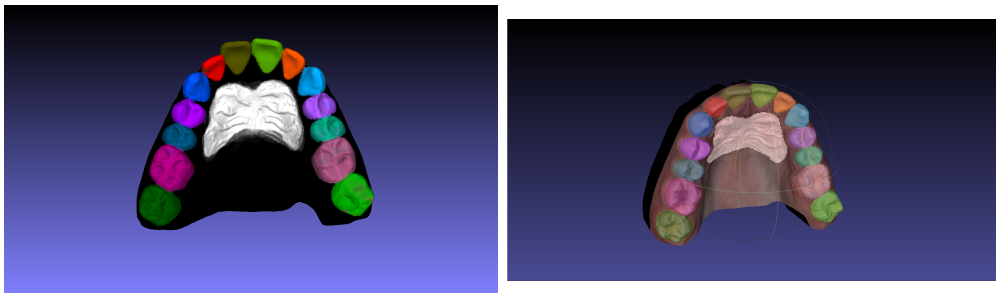


Figure 3.14: 3D Scan Segmented on the left and a sample raster projection onto the mesh on the right

Chapter 4

Motion Measurement of Teeth

4.1 Coordinate Frame Definition

Before calculating the motion, a designated fixed coordinate frame was needed to be established. This was needed since the 3D scans are not oriented or aligned after each scan. Hence, a shared coordinate frame definition was required to have a consistent coordinate frame reference for all scans in order to measure motion with respect to it.

The coordinate frame was done using a placement of 6 spheres onto the mesh, see Fig. 4.1a. Two planes can be generated with this set of spheres. The first plane that defines the y-axis known as occlusal plane was generated using the spheres located at the cusps of the molars and incisors. The second plane that defines the z-axis known as sagittal plane was generated using the spheres located along the median raphae passing through the middle of rugae. Finally, the x-axis was defined along the normal of both planes. The finalized coordinate frame with the axis is shown in Fig. 4.1b with z-direction being along the sagittal plane, y-direction along the occlusal plane, and x-along the normal between both planes.

4.2 Motion Estimation

After performing the 3D segmentation, measuring the motion of the teeth between two related three-dimensional meshes becomes possible. In this process, two meshes are required and have to be segmented scans taken from the same patient. The first scan will be called pre-treatment scan and the second scan will be post-treatment scan. Using a program called Polyworks, the complete mesh representing pre-treatment is imported while the segmented teeth and rugae from the mesh representing the post-treatment is imported, see Fig. 4.2a. The process is composed of two main steps. First, the segmented objects from the post-treatment scan are aligned with the pre-treatment scan using the trans-

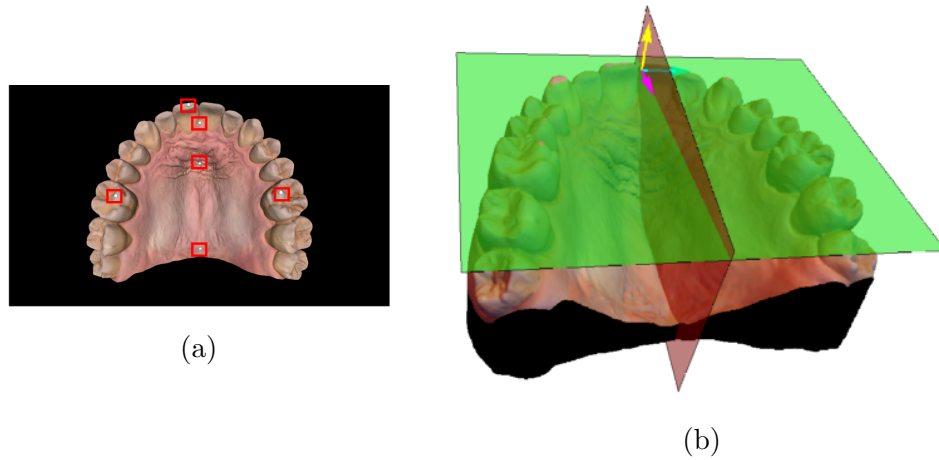


Figure 4.1: Mesh with sphere placement on the left and generated coordinate frame on the right

formation from rugae alignment of both scans as a stable reference. This aligns the segmented objects with their counterparts from the pre-treatment scan for a more accurate motion estimation. This was possible since the rugae is known for it's unique shape for every patient and acts like a fingerprint. Hence, after aligning the rugae using ICP, the transformation is applied to the rest of the teeth from the post-treatment mesh which will produce two sets of segmented teeth that are aligned based on the palatal rugae region, see Fig. 4.3c.

The final step involves aligning separately every single tooth with its counterpart from the other mesh. This process fixates both models with a reference position from which the motion assessment can be performed with respect to. Then, a second set of alignments are performed between each desired tooth from one scan and its corresponding tooth from the second scan, see alignment of Tooth 1 in Fig. 4.3d. The second set of alignments generate a set of transformation matrices. Using the set of matrices, the rotation and translation values across each axis are calculated for every tooth. These values are in reference to the stable palatal rugae region due to the first alignment performed.

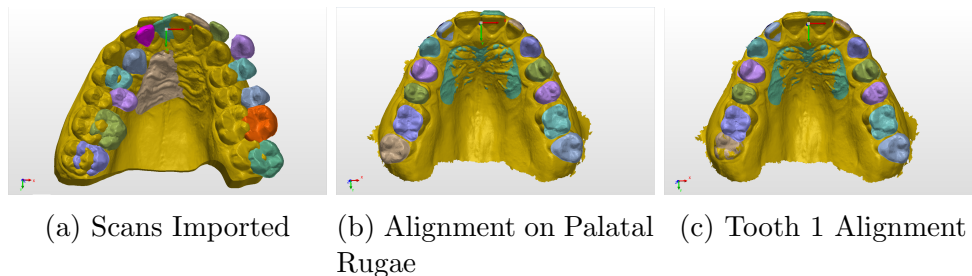


Figure 4.2: Motion Measurement process applied on one tooth as an example

4.3 Results

After 3D segmentation, motion measurement process becomes applicable. The pre scan and the segmented parts from the post scan are imported. The post scan imported was a scan which had its segmented parts manually shifted in order to gain a ground truth for the movement and compare with the results from the process using the software. This was done since the real pre-treatment and post-treatment scans have no ground truth to compare which is needed to assess the process's accuracy.

A stable reference using alignment of the palatal rugae was done first. The resulting transformation was applied onto the remaining teeth. At this point, alignment of teeth could be done to measure the motion. Each tooth was aligned individually and their resulting transformation was calculated.

The transformation was in the form

$$Transformation = T_z T_y T_x R_z(\alpha) R_y(\beta) R_x(\zeta) = \begin{pmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where angle β is found using $-\sin^{-1}(R_{31})$, angle ζ is found using $-\tan^{-1}(\frac{R_{32}}{R_{33}})$, angle α is found using $-\tan^{-1}(\frac{R_{21}}{R_{11}})$. Rotation around y-axis has two solutions; however, we can exclude second solution since it is large for the application of orthodontics (greater than 90 degrees in value).

Using that information, the results for the motion estimation were generated. Two tables were displayed showing the translation and rotation of the teeth with respect to the ground truth, see Tab. 4.1 and 4.2. It can be seen that the results from the process are almost identical except in two instances in the translation part. The first instance is the model detecting movement in the Z direction when there wasn't any. The second instance is the model detecting movement in the X direction of a value that is negligible.

After performing the manually moved scans test successfully, the motion measurement process was applied on a real pair of scans that represent pre-treatment and post-treatment scans of the same patient, see Fig. 4.4. In addition, the motion captured for each individual tooth was displayed in Table 4.3.

4.4 Conclusion

This work was able to perform motion measurement successfully using our novel rugae area as a stable reference on a manually moved model for comparison purposes and a real pair of models that have undergone an orthodontic treatment.

Table 4.1: Translation Motion Results comparing actual ground truth movement with respect to the movement achieved from the software

<i>Model</i>	Translation in X-Axes(mm)		Translation in Y-Axes(mm)		Translation in Z-Axes(mm)	
Tooth Number	Actual	Experimental	Actual	Experimental	Actual	Experimental
Tooth 1	-2	-2	0	0	0	0
Tooth 2	0	0	1	1	0	0
Tooth 3	2	2	0	0	0	0.07
Tooth 4	-2	-2	0	0	0	0
Tooth 5	1	1	2	2	0	0
Tooth 6	-1	-1	-1	-1	0	0
Tooth 7	0	0	0	0	0	0
Tooth 8	0	-0.00034	-3	-2.999	0	0
Tooth 9	0	0	-1	-1	0	0
Tooth 10	2	2	-2	-2	2	2
Tooth 11	0	0	-2	-2	0	0
Tooth 12	2	2	0	0	0	0
Tooth 13	2	2	0	0	0	0
Tooth 14	0	0	1	1	0	0

Table 4.2: Rotation Motion Results comparing actual ground truth movement with respect to the movement achieved from the software

<i>Model</i>	Rotation in X-Axes (Deg)		Rotation in Y-Axes (Deg)		Rotation in Z-Axes (Deg)	
Tooth Number	Actual	Experimental	Actual	Experimental	Actual	Experimental
Tooth 1	0	0	0	0	0	0
Tooth 2	0	0	0	0	0	0
Tooth 3	0	0	-2	-2	0	0
Tooth 4	3	3	0	0	0	0
Tooth 5	0	0	0		0	0
Tooth 6	0	0	0	0	0	0
Tooth 7	0	0	0	0	20	20
Tooth 8	0	0	0	0	-20	-20
Tooth 9	0	0	3	3	0	0
Tooth 10	0	0	0	0	0	0
Tooth 11	0	0	0	0	0	0
Tooth 12	0	0	0	0	0	0
Tooth 13	0	0	0	0	0	0
Tooth 14	0	0	0	0	0	0

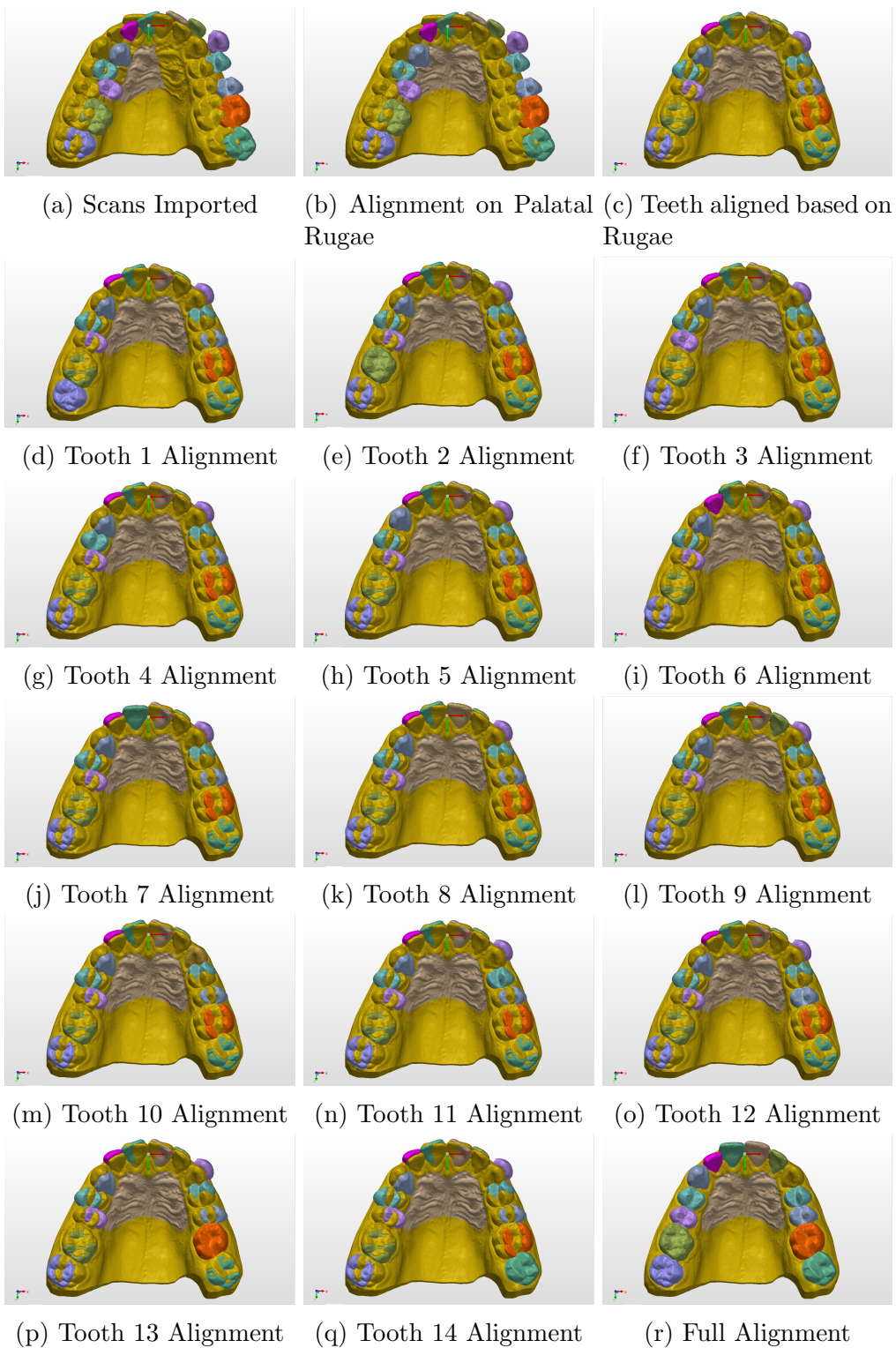


Figure 4.3: This is a diagram showing the process of alignment being done on a manually changed scan. Scans are imported, aligned on the reference rugae, followed by individual alignment of every tooth with respect to the rugae aligned reference for measurement purposes

Table 4.3: Translation Motion Results comparing actual ground truth movement with respect to the movement achieved from the software

<i>Model</i>	Trans in X(mm)	Trans in Y(mm)	Trans in Z(mm)	Rot in X(Deg)	Rot in Y(Deg)	Rot in Z(Deg)
Tooth Number	Value	Value	Value	Value	Value	Value
Tooth 1	4.066	-0.366	6.514	-3.036	-5.325	2.494
Tooth 2	-0.122	-2.948	4.505	-8.474	80.83	1.5
Tooth 3	0.698	-0.588	-1.006	5.734	1.756	0.337
Tooth 4	0.490	-0.016	2.312	-1.09	-3.18	0.379
Tooth 5	-0.079	0.241	-0.143	-0.13	3.6	1.25
Tooth 6	0.461	0.270	0.460	1.37	-1.1	0.277
Tooth 7	0.192	0.689	0.127	2.68	-0.21	0.675
Tooth 8	0.068	0.916	0.035	3.55	0.75	-0.070
Tooth 9	-0.107	0.381	0.111	2.4	0.489	0.43
Tooth 10	0.306	-0.062	0.099	0.792	-2	0.796
Tooth 11	-0.946	0.914	2.110	-0.77	2.758	-2.76
Tooth 12	-1.027	0.223	-0.954	4	-3	-2.5
Tooth 13	-1.500	-0.901	2.235	-5.1	-6.769	-3
Tooth 14	-4.035	0.035	5.653	-3.18	2.9	-3.7

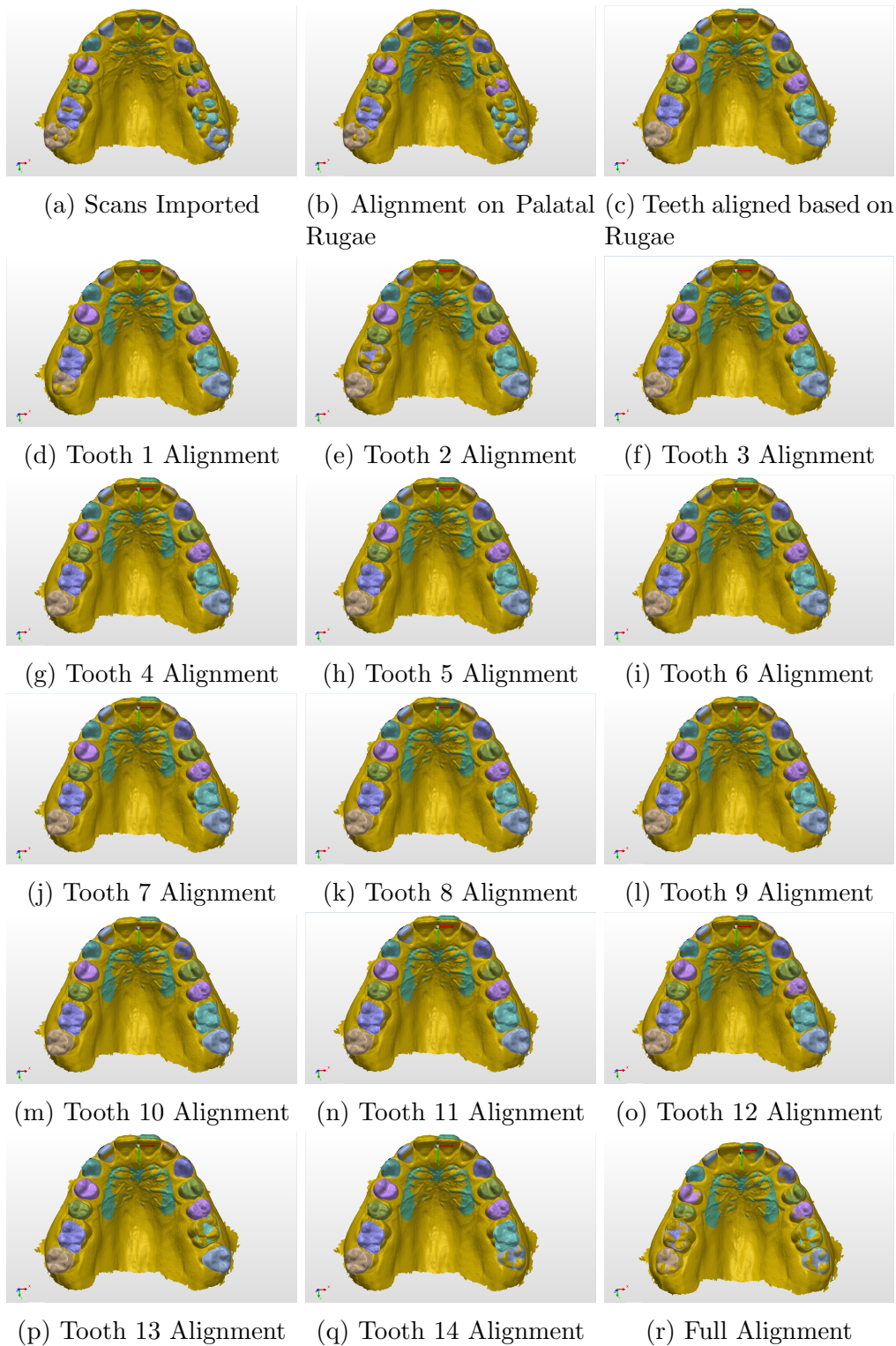


Figure 4.4: This is a diagram showing the process of alignment being done on a pre-treatment and post-treatment scans of a real patient. Scans are imported, aligned on the reference rugae, followed by individual alignment of every tooth with respect to the rugae aligned reference for measurement purposes

Chapter 5

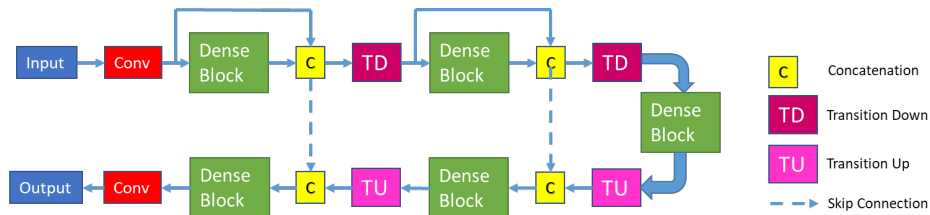
Conclusion

This work was able to produce a method that segments individual teeth in three-dimensional textured scans using the robustness of two-dimensional semantic segmentation. Furthermore, this method required no pre- or post- processing of the data to enhance the segmentation of the three-dimensional prediction result as compared to previous related work in the three-dimensional segmentation. Additionally, motion measurement was performed using our novel rugae area as a stable reference on a manually moved model for comparison purposes and a real pair of models that have undergone an orthodontic treatment. In both measurement processes, the method utilized the entire body instead of using a handful of points which removed the error of selection of points from the process and provided more accurate rotation and translation measurements of teeth.

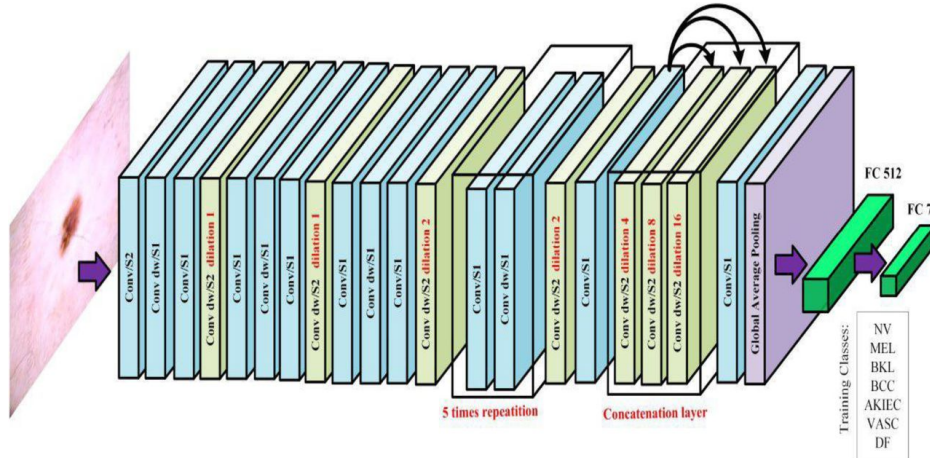
Appendix A

Network Architectures

In this work, four main architectures were included in our benchmark analysis. All of the network architectures share the same down-sampling factor of 32 which ensures a unified down-sampling factor that allows the proper assessment of the decoding method. The networks include:



(a) FC-DenseNet Architecture [15].



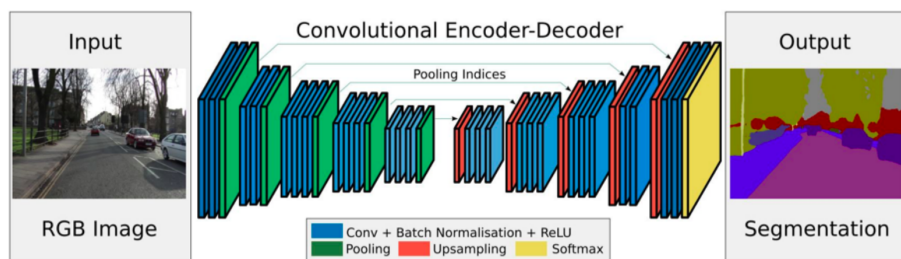
(b) MobileUNet Architecture [18].

Figure A.1: The DenseNet and MobileNet architectures are depicted.

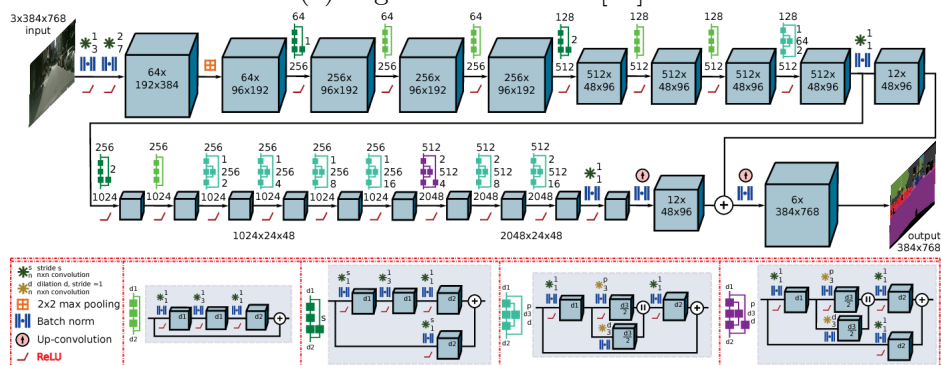
1. **FC-DenseNet56**[15]: This network uses a downsampling-upsampling style encoder-decoder network. As the name suggests, it consists of 56 layers. In

this architecture, denseblocks (DB) consisting of 4 layers are used. Each layer consists of a batch normalization, followed by ReLU, a 3×3 convolution, and dropout with probability $p = 0.2$. The DB has a growth rate of 12 as shown in Fig. A.1a. In addition, on the downsampling side, each DB is followed by a Transition Down (TD) block that consists of batch normalization, followed by ReLU, a 1×1 convolution, a dropout with $p = 0.2$, and a non-overlapping max pooling of size 2×2 . On the upsampling side, each DB is preceded by a Transition Up (TU) layer that has 3×3 transposed convolution with stride of 2 to compensate for the pooling operation. The network is terminated by a 1×1 convolution and a softmax layer.

2. **MobileUNet-Skip**[18]: This architecture is comprised of 28 layers such that we have two types of convolution layer blocks. The first type is a convolution layer that consists of a regular convolution followed by batch normalization and ReLU. The second type consists of a depth wise convolution followed by batch normalization and ReLU. Then, the layer is followed by 1×1 point-wise convolution, batch normalization, and ReLU. Finally, the architecture goes through a fully connected layer that feeds into a softmax layer for classification as shown in Fig. A.1b.
3. **Encoder-Decoder-Skip based on SegNet**[19]: This network uses a VGG-style encoder-decoder, where the upsampling in the decoder is done using transposed convolutions. The encoder network consists of 13 convolutional layers. For each encoder layer there exists a convolution with a filter bank to produce a set of feature maps. This is followed by batch normalization and an element-wise ReLU. Then, max-pooling with a 2×2 window and stride of 2 is performed. For every encoder layer there exists a decoder layer; hence, the decoder network consists of 13 layers similar to the encoder layers but differ in replacing the maxpooling by upsampling the input feature map followed by batch normalization. The architecture is shown in Fig. A.2a. In addition, the architecture employs additive skip connections from the encoder to the decoder.
4. **Adapnet**[20]: This architecture is a modified version of ResNet50 that uses bilinear upscaling instead of transposed convolutions as shown in Fig. A.2b. In addition, lower resolution processing is performed using a multi-scale strategy with atrous convolutions.



(a) Segnet architecture [19].



(b) Adapnet architecture[20].

Figure A.2: The Segnet and AdapNet architectures are depicted.

Appendix B

Abbreviations

Average mIoU	Average mean Intersection over Union
ANN	Artificial Neural Network
CBCT	Cone Beam Computed Tomography
CNN	Convolutional Neural Network
F-CNN	Fully Convolutional Neural Network
ICP	Iterative Closest Point
IoU	Intersection over Union

Bibliography

- [1] H. Gao and O. Chae, “Automatic tooth region separation for dental ct images,” in *2008 Third International Conference on Convergence and Hybrid Information Technology*, vol. 1, pp. 897–901, IEEE, 2008.
- [2] A. B. Oktay, “Tooth detection with convolutional neural networks,” in *2017 Medical Technologies National Congress (TIPTEKNO)*, pp. 1–4, IEEE, 2017.
- [3] Y. Miki, C. Muramatsu, T. Hayashi, X. Zhou, T. Hara, A. Katsumata, and H. Fujita, “Classification of teeth in cone-beam ct using deep convolutional neural network,” *Computers in biology and medicine*, vol. 80, pp. 24–29, 2017.
- [4] M. Zhao, L. Ma, W. Tan, and D. Nie, “Interactive tooth segmentation of dental models,” in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 654–657, IEEE, 2006.
- [5] S. Raith, E. P. Vogel, N. Anees, C. Keul, J.-F. Güth, D. Edelhoff, and H. Fischer, “Artificial neural networks as a powerful numerical tool to classify specific features of a tooth based on 3d scan data,” *Computers in biology and medicine*, vol. 80, pp. 65–76, 2017.
- [6] X. Xu, C. Liu, and Y. Zheng, “3d tooth segmentation and labeling using deep convolutional neural networks,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 7, pp. 2336–2348, 2018.
- [7] K. Guo, D. Zou, and X. Chen, “3d mesh labeling via deep convolutional neural networks,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 1, p. 3, 2015.
- [8] Z. Cui, C. Li, and W. Wang, “Toothnet: Automatic tooth instance segmentation and identification from cone beam ct images,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6361–6370, 2019.
- [9] S. Tian, N. Dai, B. Zhang, F. Yuan, Q. Yu, and X. Cheng, “Automatic classification and segmentation of teeth on 3d dental model using hierarchical deep learning networks,” *IEEE Access*, vol. 7, pp. 84817–84828, 2019.

- [10] J. L. Ashmore, B. F. Kurland, G. J. King, T. T. Wheeler, J. Ghafari, and D. S. Ramsay, “A 3-dimensional analysis of molar movement during headgear treatment,” May 2002.
- [11] Z. W. Wuzheng-Sjtu, “wuzheng-sjtu/instance-segment-label-tool-matlab,” Nov 2018.
- [12] M. Siam, M. Gamal, M. Abdel-Razek, and S. Yogamani, “Real-time semantic segmentation benchmarking framework,”
- [13] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, and M. Jagersand, “Rtseg: Real-time semantic segmentation comparative study,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 1603–1607, 2018.
- [14] GeorgeSeif, “”georgeseif/semantic-segmentation-suite”,” 3 2019.
- [15] S. Jégou, M. Drozdal, D. Vazquez, A. Romero, and Y. Bengio, “The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1175–1183, 7 2017.
- [16] E. Tiu, “Metrics to evaluate your semantic segmentation model,” 8 2019.
- [17] H. Li, K. Qiu, and L. Chen, “Scattnet: Semantic segmentation network with spatial and channel attention mechanism for high-resolution remote sensing images,” May 2020.
- [18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Figure 6 from mobilenets: Efficient convolutional neural networks for mobile vision applications: Semantic scholar,” 1 1970.
- [19] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 12 2017.
- [20] A. Valada, J. Vertens, A. Dhall, and W. Burgard, “Adapnet: Adaptive semantic segmentation in adverse environmental conditions,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4644–4651, 5 2017.

