DENTAL PANORAMIC AND BITEWING X-RAY IMAGE SEGMENTATION USING U-NET AND TRANSFORMER NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

METE CAN KAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2023

Approval of the thesis:

DENTAL PANORAMIC AND BITEWING X-RAY IMAGE SEGMENTATION USING U-NET AND TRANSFORMER NETWORKS

submitted by **METE CAN KAYA** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Gözde Bozdağı Akar
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering, METU _____

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering, METU _____

Prof. Dr. M. Volkan Atalay
Computer Engineering, METU _____

Assoc. Prof. Dr. Elif Vural
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. İmam Şamil Yetik
Electrical and Electronics Engineering, TOBB ETÜ _____

Date: 23.01.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Mete Can Kaya

Signature :

ABSTRACT

DENTAL PANORAMIC AND BITEWING X-RAY IMAGE SEGMENTATION USING U-NET AND TRANSFORMER NETWORKS

Kaya, Mete Can
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Gözde Bozdağı Akar

January 2023, 68 pages

With the advancement in medical imaging systems, and the underlying software platforms, diagnostics success in medicine improved significantly. Even though automatized systems are essential tools for diagnostic success, medical professional opinion is still used a lot, especially in dentistry. In the area of dentistry, x-ray images are wildly used for diagnostic purposes, i.e. to find caries, the location of embedded wisdom teeth, the health of the bone structure, etc. The dentist uses contrast and region-based information to evaluate these images. However, evaluation can be time-consuming, and it is not foolproof. In the literature, several studies exist on automatic detection from dental panoramic or bitewing images separately. Different then these studies, in this thesis, a transformer-based model is used for the segmentation of teeth using both panoramic and bitewing images. The proposed model achieved similar results on a panoramic dataset with state-of-the-art models while achieving %90 accuracy on the bitewing dataset.

Keywords: Dental image, segmentation, transformers, U-net, Data augmentation

ÖZ

U-NET VE TRANSFORMER AĞLARINI KULLANARAK DENTAL PANORAMİK VE ISIRMA X-RAY GÖRÜNTÜ BÖLÜTLEME

Kaya, Mete Can
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Gözde Bozdağı Akar

Ocak 2023 , 68 sayfa

Tıbbi görüntüleme sistemlerinde ve altta yatan yazılım platformlarındaki ilerlemeyle, tıpta teşhis başarısı önemli ölçüde arttı. Otomatik sistemler teşhis başarısı için temel araçlar olmasına rağmen, tıp uzmanlarının görüşleri, özellikle diş hekimliğinde hala yaygın olarak kullanılmaktadır. Diş hekimliği alanında röntgen görüntüleri teşhis amaçlı yani çürüğün konumu, gömülü 20 yaş dişleri, kemik yapısının sağlığı vb. görüntüler için kullanılmaktadır. Ancak, değerlendirme zaman alıcı olabilir ve kusursuz olmayabilir. Literatürde dental panoramik veya ısırma görüntülerini ayrı ayrı ele alan birçok çalışma bulunmaktadır. Bu çalışmalardan farklı olarak bu tezde dental panoramik ve bitewing görüntülerinden çok modelli bir veri tabanı oluşturulmuş ve dişlerin segmentasyonu için transformatör tabanlı bir model kullanılmıştır. Önerilen model panoramik veri setinde son teknoloji modellerle benzer sonuçlar elde ederken, bitewing veri setinde %90 doğruluk elde etmiştir.

Anahtar Kelimeler: Dental Görüntü, bölütleme, transformers, U-net, veri artırma

I dedicate this research to my brother, who was there when I need.

ACKNOWLEDGMENTS

I would like to acknowledge my supervisor Gözde Bozdağı Akar who made this research possible. From the start of my bachelor's education, she was there to guide me through courses, and research projects.

I would also like to thank my family for their unconditional support during this research. I cannot do this without them.

Finally, I would like to thank my friends at the Multimedia laboratory group, for helping me through my education, and being my friend, mentor, and student. I have learned from their successes and failures.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AA	Auto Augmentation
AI	Artificial Intelligence
ANN	Artificial Neural Networks
ASPP	Atrous Spatial Pyramid Pooling
BCE	Binary Cross Entropy
CNN	Convolution Neural Network
CPU	Center Processing Unit
CCE	Categorical Cross Entropy
CV	Computer Vision
FCN	Full Convolutional Network
FCNN	Fully Connected Neural Network
FDI	Fédération Dentaire Internationale
FN	False Negative
FP	False Positive
FSAA	Full-Scale Axial Attention Layer
GPU	Graphics Processing Unit
ISBI	International Symposium on Biomedical Imaging
JPU	Joint Pyramid Upsampling
LR	Learning Rate
LSCE	Label Smoothing Cross Entropy
LSTM	Long Short Time Memory
LTP	Local Ternary Pattern
LTPEDN	Local Ternary Pattern encoder-decoder
MAE	Mean Absolute Error

ML	Machine Learning
MLP	Multilayer Perception
MSA	Multi-Head Attention
MSE	Mean Square Error
MSLP	Multi-scale Location Perception
NLP	Natural Language Processing
PBA	Population-Based Augmentation
RA	Randaugmentation
RAM	Random Access Memory
RCNN	Region-Based Convolution Neural Network
SGD	Stochastic Gradient Descent
SOTA	State of the Art
SW-MSA	Shifted Window Multi-Head Attention
TL	Transfer Learning
TN	True Negative
TP	True Positive
U-net	U Shaped Neural Network
VIT	Vision Transformer
W-MSA	Window Multi-Head attention

CHAPTER 1

INTRODUCTION

X-ray imaging is an essential data source for diagnosis in dentistry. The images are generated by X-rays that travel through the body, and they are absorbed in different amounts by different tissues, depending on the radiological density of the tissues they pass through. X-ray images contain information on teeth, gums, jaws, and bone structure of the mouth. Since the tissue structure of the human mouth is known, unexpected situations can be detected. Without these images, many dental problems cannot be detected in the early stages as the only other resource is patient complaints. Also, different fields and application areas in density require these images. Examples can be given in fields like dental surgery, implantology, and forensic identification.

In dentistry, different imaging techniques were developed depending on the requirements. There are two main categories of imaging: intra-oral radiographic and extra-oral radiographic imaging. The former is usually used before treatment. In this imaging technique, the receiver sensor is placed inside the patient's mouth. It shows a smaller area with more detailed information. However, it creates custom shooting angles and custom regions of interest. In extra-oral radiography, the receiver sensor is outside of the patient body. This technique has standard shooting angles and regions of interest. The patient is placed in a seated position to stabilize the movement. Therefore, extra-oral radiographic images are more consistent images than intra-oral radiographic images.

Bitewing, periapical, and panoramic imaging techniques are the most used imaging techniques in dentistry. Bitewing and periapical images are intra-oral images that show more detail compared to panoramic images. While Bitewing images are

used for analysis and treatment of the crown, periapical images are used for root region. In these images, it is easier to observe caries compared to panoramic images. Panoramic imaging is an extra-oral radiographic imaging technique. The panoramic images show the patient's whole mouth region hence the resolution per tooth is lower. However, these images can contain more information and can be used to find impacted teeth like wisdom teeth or early disease detection.

X-ray images require time and professional education to understand and examine. These are time-consuming and expensive requirements. Hence, there are several types of research to improve these requirements. Some researchers focus on image enhancement to improve the content or remove the unnecessary part of the image. However, the result of this research still requires a dentist's experience and visual perception. Hence, developing an automated tool is much more effective. However, developing an algorithm for this task is challenging because of difficulties such as variations of patient-to-patient teeth, artifacts used for restorations and prostheses, poor image qualities caused by certain conditions (such as noise, low contrast, homogeneity in regions close to objects of interest), space existing by a missing tooth, and limitation of acquisition methods [1]. In the literature, there are several studies based on both traditional and learning-based approaches but as expected supervised deep learning solutions outperform traditional image processing solutions [55].

[50] showed that using 40 bitewing images is enough to train a U-net architecture [51]. This neural network wins the ISBI 2015 challenge. Then, [55] published an open panoramic dataset for segmentation, and [29] achieve great results with this dataset. [34, 48, 62, 10] work on same dataset to improve the segmentation result. [53] improve the dataset and added the numbering to the dataset. These researches are an important step for the development of automatic dental analysis tools.

1.1 Contributions and Novelties

This thesis aims to achieve teeth segmentation on X-ray dental images using deep learning methods. The contributions are as follows:

- A single deep learning model that can work with bitewing and panoramic images is proposed. Even though the teeth structure is obtained with the same methodology between bitewing and panoramic, the teeth structure in these two methods has different scales, orientations, and positions. One deep-learning method performs on bitewing and panoramic images without any rotation or scale in the testing phase. The experiments show that the proposed model can achieve state of art results in panoramic data while achieving similar results for bitewing teeth segmentation.
- Significant improvement on bitewing teeth segmentation is achieved with a low number of images.

1.2 Structure of the Thesis

The outline of thesis is organized as follows. Problem definition, recent studies, and contributions are given in Chapter 1: Introduction. In Chapter 2, the literature review is given. Recent studies are divided by their tasks, and important ones are explained in detail. In Chapter 3, background information about X-ray dental datasets, deep learning models, loss functions, and metrics are given respectively. In chapter 4, the proposed method is explained. The chapter starts with the motivation and reasoning behind the solution. Then, the proposed method is explained. The results of data augmentation methods and neural network selection are explained in chapter 5. Also, the used hardware and compression with SOTA models are also given in this chapter. Finally, the conclusion is given in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

In this chapter, we will give the basics of current literature on neural network-based solutions for dental images. We separated the literature into two groups; classification and segmentation. The classification will be discussed briefly just to show the usage of neural networks on different structures of the jaw like bone or gum. The segmentation part will be discussed in more detail, and it will be focused on teeth. Then, we explain the rationale for our algorithm selection and our problem selection according to the literature.

For classification, we can see several recent works in table 2.1. The classification studies are more successful compared to other studies as they work with larger datasets, and the task is simpler. In these studies, they separate images into two or three classes, and a CNN network that ends with the FCN layer predicts which class the images belong to. These researches also present solutions for different areas. Lee et al [36] and Kim et al[32] show the usage of periapical dental images for attention on teeth roots. Also, Geetha et al [21] shows that a smaller dataset and preprocessing are enough for accurate caries classification.

For segmentation, the task is defined as the process of separating pixels into distinct groups. These groups can be teeth, caries, implants, and so on. The baseline solution for segmentation of dental images dates back to 2015, where Ronneberger et al[50] won ISBI2015 Grand Challenges in Dental X-ray Image Bitewing Radiography, using their U-net architecture [51]. Their proposed architecture overcomes the gradient vanishing problem and over-fitting due to the small dataset. This achievement is a milestone for this area where a large dataset is difficult to generate. However, their

solution cannot solve the class imbalance problem. Kaya and Akar [30] used octave convolution [14] and focal loss [42] to solve this problem as octave convolutions are smaller than normal convolution layer, and focal loss is known to be effective with class imbalances. However, it shows a little improvement as caries and dental treatments were less in number compared to the main parts of the tooth in the given dataset [58].

There are several segmentation of teeth studies with their own intra-oral dental image datasets. In Ari et al [2], they use a dataset with size 1169 periapical radiographs. However, they train separate U-net models for each tooth structure hence they do not solve the problem of generating a single model that segments all different tooth structures. Haghanifar et al [26], Lee et al[38], Bayrakdar et al [4] and Zhu et al [63] improves the performance on caries segmentation as they used balanced and larger data set for this task. Brief information on their performance can be seen in table 2.2. Haghanifar et al [26] merge CNN architecture with the capsule network since the Capsule network is capable of learning the geometrical relationships between features generated by the CNN part. Lee et al[38] show that the U-net used in [50] is under-fit and can be trained more with the larger dataset. Zhu et al [63] propose a solution that passes the U-net that trained with their dataset by %13 in accuracy. They modify the decoder levels to a network that they proposed as a full-scale axial attention layer (FSAA). Different from CNN layers, this layer merges FCN output with CNN output to obtain better global attention. This network holds the best caries segmentation performance at this moment. Ying et al [60] achieve a very close performance improvement to [63]. They improve the accuracy of [50] by %11. They achieve this performance with a much smaller dataset compared to [63]. They used a modified version of TransUnet [11]. They change the encoder to ResNet-v2 [20] and Identity-ASPP (In-ASPP) layers are used as skip connection parts between the encoder and decoder. These layers are used for increasing the encoding of multi-scale global information. These modules are similar to ASPP in [12]. This research is one of the first research on dental image segmentation that uses transformer networks. They also obtain similar results to other state of art studies.

There are also semantic segmentation of teeth studies on dental panoramic image

datasets. These studies can be separated into two by their neural network architectures; encoder-decoder and Region-Based CNN architectures. The latter is usually used as Mask R-CNN and the research are made with different back-bone, hybrid loss calculation, and different mask layer. Silva et al [53] work on compare MaskR-CNN. Pinheiro et al [48] adds PointRend to MaskRCNN with to improve performace of [53]. Other studies that used encoder-decoder networks used either U-net or a modified version of U-net. Koch et al [34] uses U-net with the dataset used in [29]. The U-net model outperforms the MaskRCNN in F1-score and Recall metrics. Salih et al [52] use U-net with non-trainable layers called local ternary pattern (LTP) [9] which they proposed as The local ternary pattern encoder–decoder neural network (LTPEDN). This network achieves a very close score to normal U-net with half of the size of the network. Chen et al [13] proposed a network called Multi-scale location perception network(MSLPnet). This network adds FCN layers between the encoder and decoder which improves the network's global-level perception. This research also uses a new metric called PFOM [40] that is used for objective boundary measure of performance. They also use a structural similarity loss to improve their score of PFOM. Even though this solution gives a %2 increase in accuracy the segmentation results look more appealing. Zhao et al [62] achieve similar visual and metric performance as [13]. They use a double U-net architecture. The first U-net generates an attention map while the second U-net generates the segmentation result. They modify the first U-net with custom layers called global attention and local attention module that are similar to layers defined in [43]. The global attention module uses the LSTM network whereas other researchers like [13] use FCN for processing global context information. The result of these studies can be found in table 2.2.

Instance Segmentation and numbering of teeth on the panoramic images are also popular research topics. These tasks are not studied in this thesis. However, these studies work on how to improve the existing solution. Some of the methods also improve semantic segmentation performances. For example, Pinheiro et al [48] performs improves the work of [53] and added the numbering of teeth. The instance segmentation performance of studies can be seen in table 2.3. The studies on the numbering of teeth are tries to predict teeth numbers defined in FDI World Dental Federation notation.

Results of these studies can be found in table 2.4.

There are some interesting studies on the numbering of teeth that uses anatomical structures of the human jaw. Lin et al [41] use the structural information on the jaw since human jaws have symmetric structures and teeth are in order. They also use several image enhancement methods to achieve successful numbering without accurate segmentation. Chen et al [10] uses a neural network for segmentation and initial numbering guessing. However, they also use the structural information to fix wrong predictions or conflicting ones like premolar after moral tooth. Chung et al [15] are focused on just numbering without segmentation part. They try to estimate the center points of the tooth, and they use a two-layered network for this task. The first layer predicts the center points and the second layer predicts the BBox around the predicted center points. They use the anatomical structure in post-processing to improve their results.

Table 2.1: A table of classification studies on Dental images with their brief work and performances.

Authors	Year	Network	Task	Data Set	Metrics						
					Acc.	Prec.	Rec.	F1	AUC	Sens.	Spec.
Lee et al [35]	2018	Inception v3	Caries	3000 Periapical	0.82	-	-	-	0.845	-	-
Geetha et al [21]	2020	Preprocess+FCN	Caries	105 Periapical	0.971	-	-	-	0.987	-	-
Bergner et al [5]	2021	EMIL	Caries	38k bitewing	0.736	-	-	0.779	-	0.694	0.778
Lee et al [36]	2018	VGG-19 + SVM	Compromised teeth	1044 periapical	0.734	-	-	-	0.734	-	-
Kim et al [31]	2019	DentNet	Bone Loss	11,189 panoramic				0.75	0.95	0.77	0.95
Kim et al [32]	2020	Resnet-50	Implant Fixture	901 periapical	0.98	0.98	0.98	0.98	-	-	-

Table 2.2: A table of recent Semantic Segmentation studies on Dental images with their brief work and performances.

Authors	Year	Method	Task	Data Set	Metrics						
					Acc.	Spec.	Prec.	Recall	F1	meanIOU	mAP
Ronneberger et al [50]	2015	U-net	Teeth Structure	120 Bitewing	-	-	0.453	-	-	-	-
Kaya and Akar [30]	2020	Octave U-net	Teeth Structure	120 Bitewing	-	-	0.48	-	-	-	-
Haghanifar et al [26]	2020	PaXNet	Caries	400 Panoramic	0.86	-	0.89	0.86	-	-	-
Lee et al [38]	2021	U-net	Caries	354 Bitewing	-	-	0.633	0.65	0.64	-	-
Zhu et al [63]	2022	CariesNet	Caries	1159 Panoramic	0.936	-	0.941	0.86	0.929	-	-
Ying et al [60]	2022	TransUnet	Caries	153 Bitewing	-	-	0.74	-	-	-	-
Jader et al [29]	2018	MaskRCNN	Teeth	1500 Panoramic	0.98	0.99	0.94	0.84	0.88	-	-
Chen et al [10]	2019	Faster RCNN + DNN	Teeth	1250 Periapical	-	-	0.988	0.985	-	0.91	-
Koch et al [34]	2019	U-net	Teeth	1500 Panoramic	0.946	0.954	0.9226	0.94	0.93	-	-
Silva et al [53]	2020	MaskRCNN	Teeth	1500 panoramic	0.96	0.986	0.941	0.866	0.902	-	0.664
Zhao et al [62]	2020	U-net	Teeth	1500 Panoramic	0.969	-	-	0.938	-	-	-
Sivagami et al [56]	2020	U-net	Teeth	1171 Panoramic	0.97	0.95	0.93	0.94	0.93	-	-
Chen et al [13]	2021	MSLPNet	Teeth	1500 Panoramic	0.973	0.98	0.93	0.93	-	-	-
Pinheiro et al [54]	2021	MaskRCNN + PointRend	Teeth	1500 Panoramic	-	-	-	-	-	-	0.73
Salih and Duffy [52]	2022	LTPEDN	Teeth	11,000 Panoramic	0.943	-	-	-	-	-	-
Ari et al [2]	2022	U-net	Caries	1169 Periapical	-	-	0.82	0.82	0.82	-	-
Bayrakdar et al [4]	2022	U-net	Caries	621 Panoramic	-	-	0.84	0.81	0.84	-	-
Çaylak et al [8]	2022	InceptionResV2-U-net	Teeth	131 Panoramic	0.976	-	-	-	0.90	0.82	-

Table 2.3: A table of recent instance segmentation studies on Dental images with their brief work and performances.

Authors	Year	Method	Task	Data Set	Metrics					
					Acc.	Spec.	Prec.	Recall	F1.	MeanIoU
Jader et al [29]	2018	Mask RCNN	teeth	1500 Panoramic	0.98	0.99	0.94	0.84	0.88	-
Lee et al [37]	2020	Mask RCNN	teeth	1024 Panoramic	-	-	0.858	0.89	0.875	0.877

Table 2.4: A table of recent studies on the automatic numbering of teeth.

Authors	Year	Network	Task	Data Set	Metrics					
					Acc.	Prec.	Recall	F1.	mAP	MeanIoU
Lin et al [41]	2010	Image Proccesing&structural geometry	Teeth	47 Bitewing	0.93	-	-	-	-	-
Chen et al [10]	2019	Faster RCNN + DNN	Teeth	1250 Periapical	-	0.917	0.914	-		-
Silva et al [53]	2020	MaskRCNN	Teeth	1500 Panoramic	-	-	-	-	0.70	0.877
Pinheiro et al [48]	2021	MaskRCNN + PointRend	Teeth	1500 Panoramic	-	-	-	-	0.71	-
Kılınç et al [33]	2021	Faster RCNN	Teeth	421 Panoramic	-	0.957	-	0.968	-	-
Chung et al [15]	2020	2 Stage Resnet	Teeth	818 Panoramic	0.997	-	0.972	-	-	-

CHAPTER 3

BACKGROUND INFORMATION

3.1 Dental Datasets

There are three types of imaging in dentistry; bitewing, periapical, and panoramic. However, bitewing and panoramic imaging types have open datasets with labeled teeth. Therefore, bitewing image and panoramic image datasets are used in this thesis. In this section, these datasets will be explained in detail.

3.1.1 Bitewing Dataset

The bitewing dataset used in this thesis is from ISBI 2015 challenge [58]. The dataset consists of 40 train, 40 validation, and 40 test images. Each image has a ground truth image. However, the accessible data does not contain the test images. An example set of image and ground truth pair can be given in figure 3.1.

The bitewing dataset is an unbalanced dataset. The image's resolution is 710x512 pixels. Enamel, dentin, and pulp classes have dominance in the dataset. Since they are the main structure of the tooth, they exist on both healthy and unhealthy teeth except for implants which do not exist in our dataset. The difference between per average pixel in classes is presented in table 3.1 and in figure 3.2. Hence, this dataset needs preprocessing, augmentation, and special loss functions that work with data imbalance.

No.	Important Properties Parts
1	caries (blue color)
2	enamel (green color)
3	dentin (yellow color)
4	pulp (red color)
5	crown (skin color)
6	restoration (orange color)
7	root canal treatment (cyan color)

Figure 3.1: From left to right, input image, label names and color code, and ground truth image.

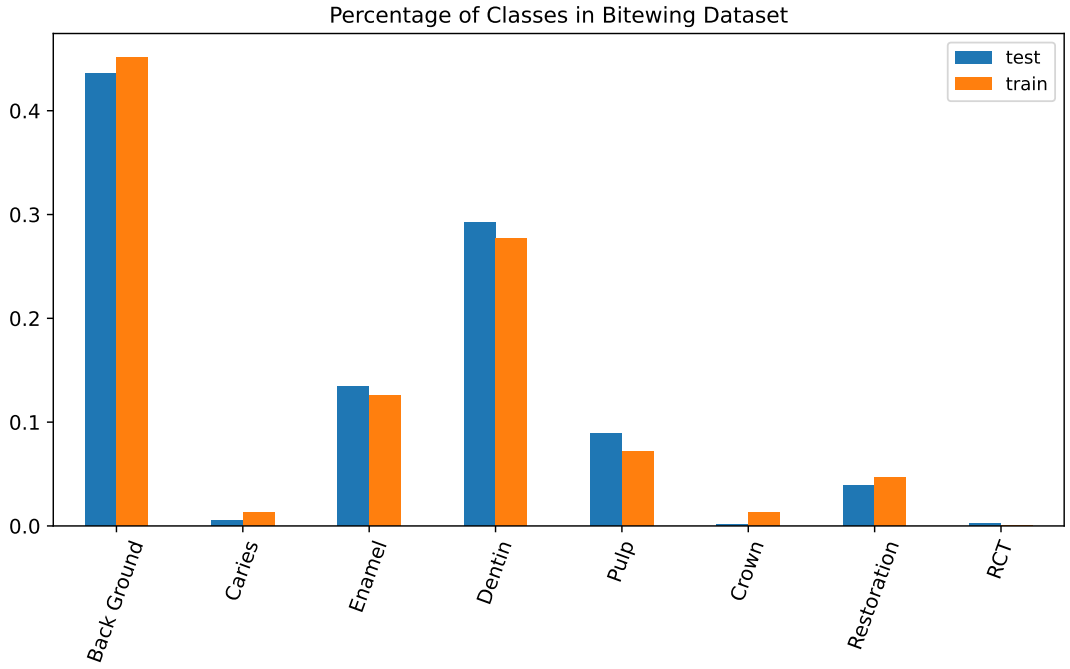


Figure 3.2: Graphical difference between each class in bar representation. Both the train and the test datasets are represented.

3.1.2 Panoramic Dataset

The panoramic dataset is made of 1500 images with a resolution of 1991x1127 pixels. The dataset is shared first in 2018 [55]. This version of the dataset contains a tooth mask for each tooth in this dataset. Then, it improved at [53]. The annotations become more accurate and they add COCO formatted boundary boxes for each tooth in an image. The final version was released in [48], the number of images was reduced to 450, and images are cropped to a resolution of 1876x1036 pixels. This version

Table 3.1: Percent values of classes in bitewing dataset for training and testing.

Class	Train Set Percentage	Test Set Percentage
Back Ground	0.452	0.435
Caries	0.013	0.005
Enamel	0.125	0.133
Dentin	0.277	0.292
Pulp	0.071	0.089
Crown	0.013	0.002
Restoration	0.046	0.039
Root Canal Treatment	0.001	0.002

contains each tooth boundary box with the number that is defined in the FDI World Dental Federation notation.

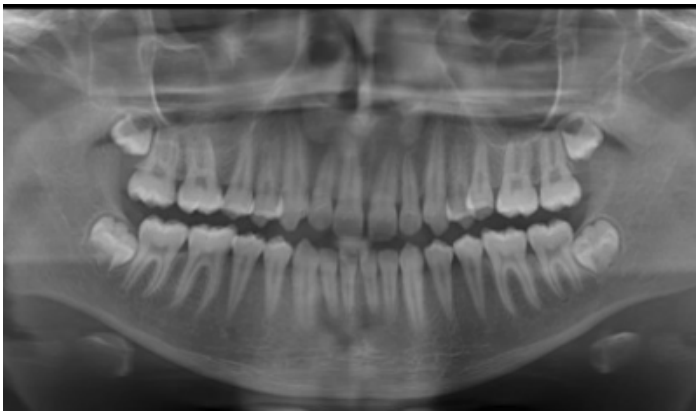
In chapter 2, there are many good studies that achieved accuracy higher than $\%90$ accuracy. However, panoramic images are easier to work on since the patient is in a somewhat locked position and the image is not generated from a single X-ray shot. However, finding caries on a panoramic image is harder since the amount of dose falling on a single tooth is low. Hence, these images lack the necessary knowledge to make an accurate assessment.

Table 3.2: Percent values of classes in Panoramic dataset for training and testing.

Class	Train Set Percentage	Test Set Percentage
Back Ground	0.805	0.773
Teeth	0.194	0.227

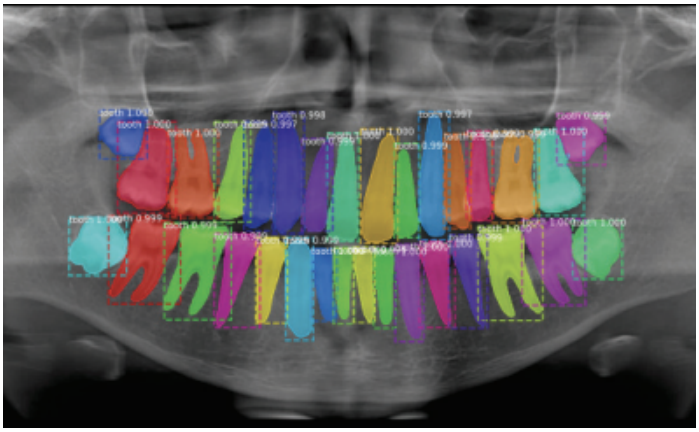
3.2 Neural Network Models in Medical Image Processing

There are different neural network models that are used for medical image processing. The use of these models usually depends on their tasks: classification, semantic



(a) Input panoramic image

(b) Segmentation mask of input image



(c) Instance segmentation mask and bound box of input image

Figure 3.3: (a) A sample from the panoramic dataset, (b) ground truth mask that shared in [55], (c) instance segmentation and numbering information shared in [48]. The amount of information in the panoramic dataset increased drastically.

segmentation, and intrinsic segmentation. However, the dataset is also an essential part of the model selection. For example, transformer-based models require more im-

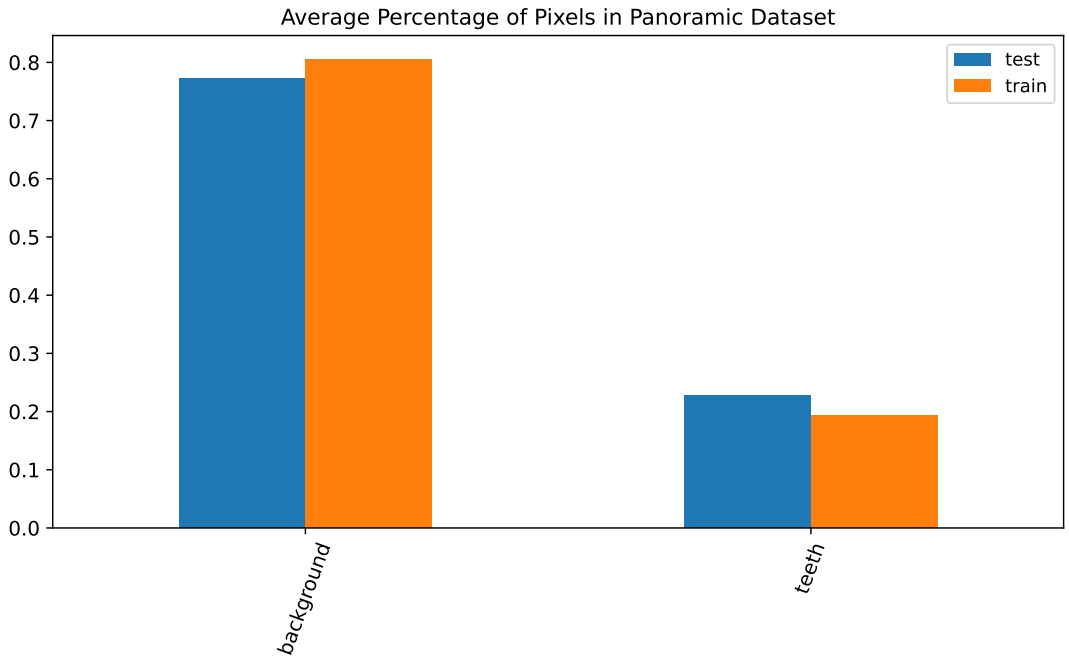


Figure 3.4: Graphical difference between teeth and background in bar representation. Test and Train sets presentation are also given in a color code presented on the plot.

ages than CNN-based counterparts. Hence, the performance of a transformer-based model on dataset with fewer images will be worse compared to CNN based model. The networks utilized in this thesis are shown in Figure 3.5. The selected networks are separated into their basic layers; transformer-based and convolution-based. Then, they separated into tasks: classification, semantic segmentation, and intrinsic segmentation.

3.2.1 CNN based models

CNN-based models are neural network models that used convolution layers as their learnable layers. Convolution layers are the most commonly used learnable layers in neural network models. They are efficient and easily scalable. Convolutional layers have a filter that has the size $c_{in} * c_{out} * w * h$. c_{in} and c_{out} are the dimensions of input and output respectively, and w and h are the width and height of the filter. These filters work by convolving with the input matrix. Since they are independent of input dimensions, they can be used for any image size. One can also add a bias factor to

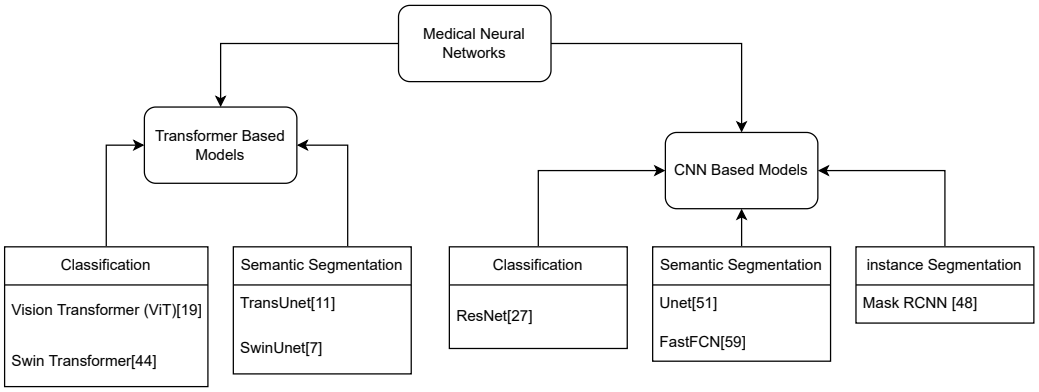


Figure 3.5: Schematic representation of neural networks separation by their architecture and task.

the equation as an addition after the convolution operation. This adds non-linearity to the operation and therefore improves the backpropagation of the neural networks.

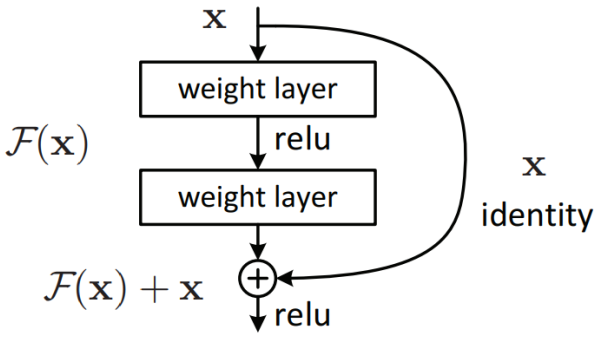


Figure 3.6: Residual learning: a building block taken from [27].

3.2.1.1 Classification

Classification is a task where the model predicts a label for an input image. Classification applications are simpler compared to segmentation tasks as data labeling is much simpler. As mentioned in chapter 2, there are classification researches done on X-ray dental images. The classification studies usually use end-to-end models as seen in [35, 21, 5].

ResNet

ResNet is instructed in [27]. ResNet is a full CNN network. ResNet can be trained to deeper levels because skip connections that solve the gradient vanishing problem. The skip connections pass a few layers forward in the model as seen in Figure 3.6. This allows a gradient to pass from starting layers of the network.

The ResNet is very popular in transfer learning applications. There are several versions of ResNets like ResNet-18, ResNet-34, ResNet-50, etc. These models are trained on large open datasets. The FastFCN and TransUnet use a ResNet model in their backbone stage since this will reduce the amount data of needed to fully train the model.

3.2.1.2 Semantic Segmentation

Semantic segmentation is a task where the prediction is done on the pixel level. Compared to classification, it is a harder task as it is time-consuming to create a dataset. In this thesis research, the datasets that are used are used for semantic segmentation. The research done on these datasets is given in table 2.2.In this part, U-net and Fast-FCN models are explained.

U-net

Skip Connection

U-net is a neural network model that is used for medical image segmentation. U-net can be thought of as a milestone in medical image segmentation since it can learn on small datasets and outperform other models.
The model has a U-shape as seen in figure 3.7. The model block can be separated into encoder and decoder blocks. The encoder is used to obtain dense feature values at the lower level, and the decoder is used to predict the segmentation mask. The model uses skip connections for fast training and original features with fine-grained features from the lower level. This skip connection is between the encoder and decoder at the same level. Also, loss values can be obtained from each level.

Fast FCN

FastFCN [59] is a neural network model that is used for semantic segmentation as seen in figure 3.8. The model is a modified version of [45]. It has three parts: backbone, Joint Pyramid Upsampling (JPU) layer, and segmentation head. A backbone network is used for feature extraction. In [59], the ResNet model that is trained with ImageNet is used. Backbones like efficient or inception cannot be used as they do not have a hierarchical structure like ResNet. JPU is a custom layer that has a structure as seen in figure 3.9. This layer uses feature maps from the backbone at different scales and levels. This layer improves the extract multi-scale context information from multi-level feature maps which results in better performance. The head structure

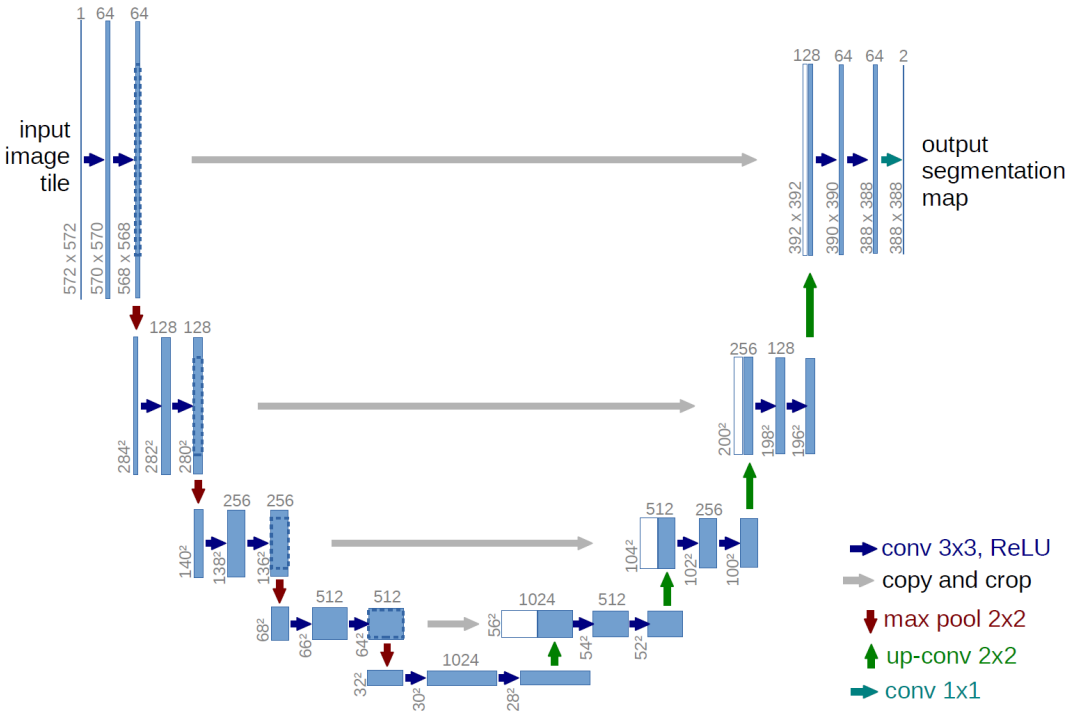


Figure 3.7: U-net model overview [51].

of FastFCN is used to create the segmentation predictions. One can use DeepLabv3 [12], single convolution layer or ENCNet [61]. For model selection, testing may be required as their performance differs.

Deeplabv3 uses dilated convolution and Depthwise Separable Convolution in order

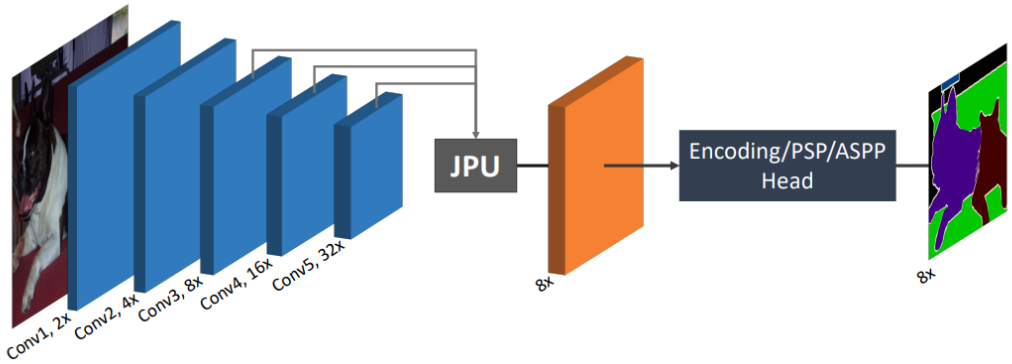


Figure 3.8: Framework Overview of FastFCN Method taken from [59].

to improve its performance and reduce its size.

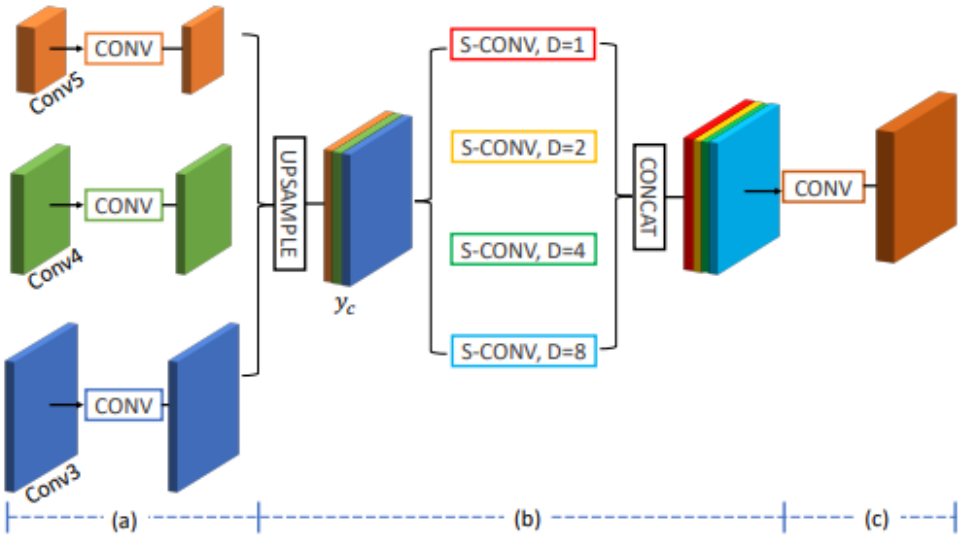


Figure 3.9: The Proposed Joint Pyramid Upsampling (JPU) taken from [59].

Dilated Convolution

Dilated convolution is a convolution layer that works with a larger area with the same filter size as seen in figure 3.10.

Depthwise Separable Convolution

Depthwise convolutions are two stages convolutions, and they are more efficient than normal convolutions without a significant loss in performance. The first stage of this convolution is the depthwise stage. In this stage, separated convolution operations are done on each input channel. Then this channel concatenated. The second stage is a pointwise convolution. In this part, one by one layer is used to a weighted average of the input channels.

Instrinsic Segmentation

Intrinsic segmentation is a task similar to semantic segmentation. Additional to pixel-wise prediction, it also distinguishes the same class object and separates them from

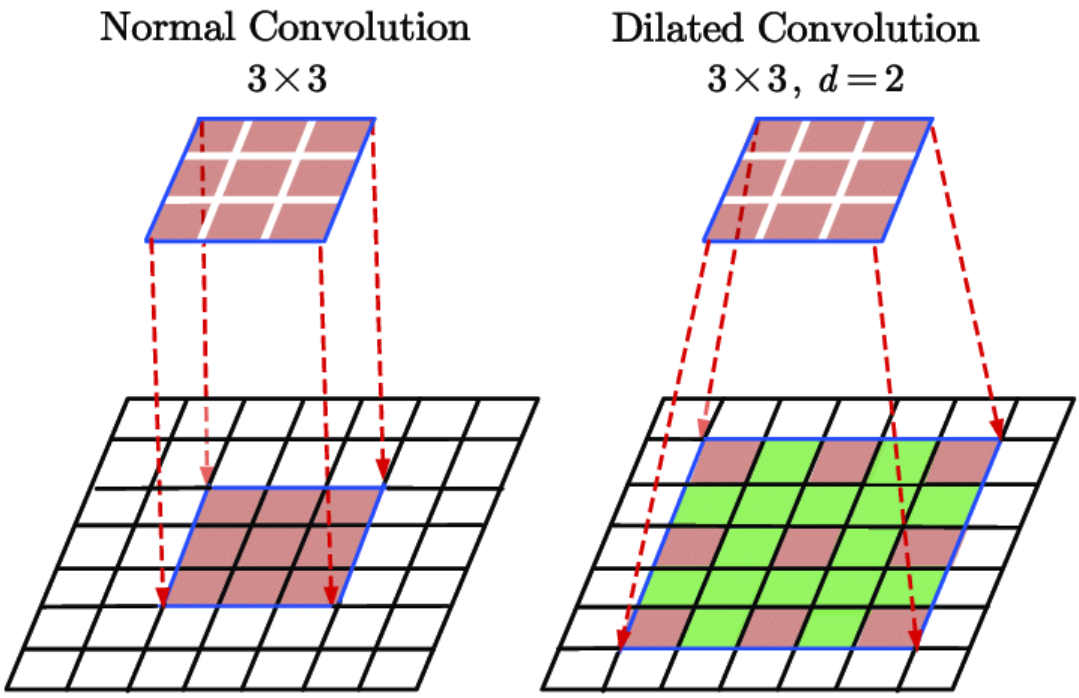


Figure 3.10: Normal convolution multiplication and dilated convolution multiplication range. Green regions are not multiplied with kernel in that instance.

each other. For example, counting the number of people in an image while predicting the people on the pixel level. The dataset shared in [53] is applicable for intrinsic segmentation as it is given in COCO format and each tooth is separately segmented.

The most common models that are used in intrinsic segmentation are Fast RCNN and Mask RCNN. The research done on intrinsic segmentation of X-ray images is given in table 2.3.

Mask RCNN

Mask RCNN is a neural network model that works on instance semantic segmentation. The model is an extension of Fast R-CNN [24] by adding a branch for predicting an object mask. The model is easy to use and can be used for instance segmentation, bounding-box object detection, and person keypoint detection.

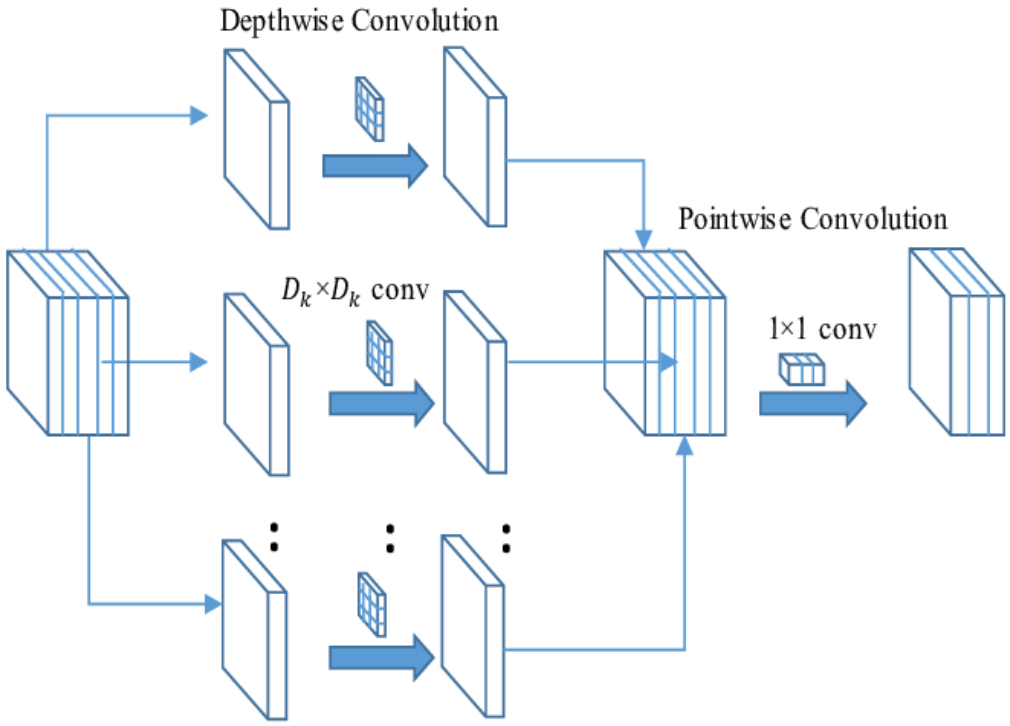


Figure 3.11: Schematic representation of depth-wise separable convolution

The research done on the dental dataset usually uses this architecture as seen in table 2.3. The aim of this research is to generate an automatic diagnosis for each tooth. The model is used for the separation of each tooth from the other and may be used to find dental caries or other cases on each tooth.

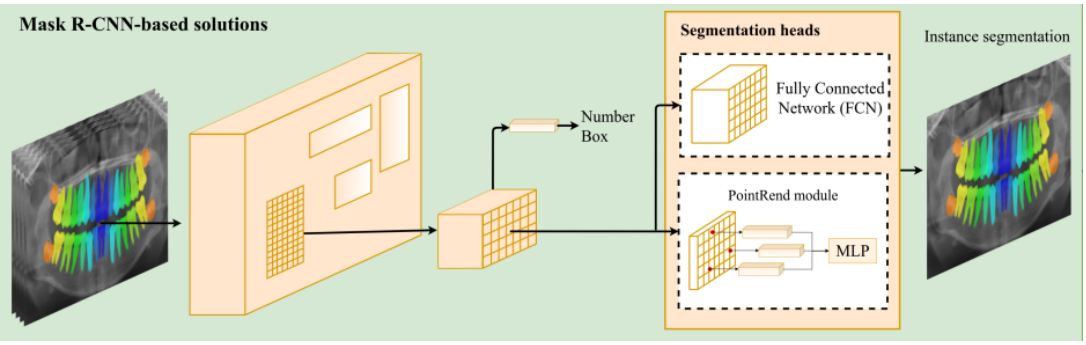


Figure 3.12: Schematic representation of Mask RCNN [48].

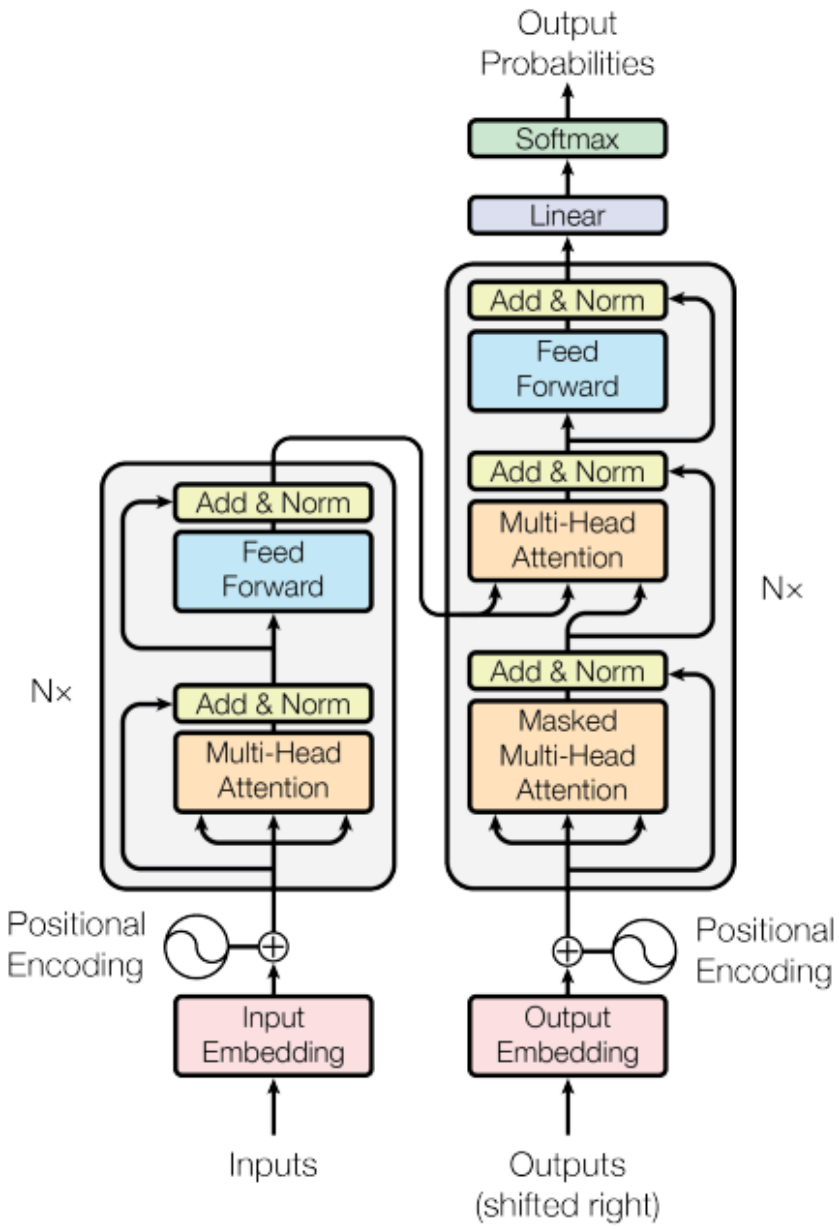


Figure 3.13: Model architecture of transformers

3.2.2 Tranformer Based Models

Transformer-based models are a recent development in deep learning applications compared to CNN models. Transformer models adopt the self-attention mechanism. They become popular and successful in natural language processing (NLP) applications. They also have computer vision applications that outperform CNN models on large datasets.

Before going passing these models' applications on the dental dataset. First basic transformer models and vision transformers will be explained.

Transformers

Transformers are neural network layers that only use attention instead of recurrence and convolutions. These models aim to improve parallelization. An encoder-decoder model made with transformer layers can be seen in figure 3.13.
A single transformer layer consists of Normalization (Norm), multi-head attention, and multilayer perception (MLP). The Norm and MLP have known layers. The norm subtracts the mean and divides the result by the standard deviation, and MLP is a multi-layered FCNN layer. The multi-head attention is a breakthrough part of the transformer hence it will be explained in more detail by beginning with the attention mechanism.

Attention Mechanism

In simple terms, the Attention mechanism is a function that takes a weighted sum of an input value. The input values are defined as V vector. The weights are calculated as

$$W = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Given Q and K are two input vectors called queries and keys. $\frac{1}{\sqrt{d_k}}$ is a scaling factor. d_k is the dimension of the keys. Hence, the attention function is represented as;

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Attention

Instead of performing a single attention function, it is more beneficial to linearly project the queries, keys, and values h times with different, learned linear projections to dk, dk and dv dimensions, respectively [57]. The output of each attention function with size dv is then concatenated and given as input to a fully connected layer. The

multi-head attention calculation is given as:

$$\begin{aligned}\text{MultiHead}(Q,K,V) &= \text{Concat}\left(\text{head}_1,\ldots,\text{head}_\text{h}\right)W^O \\ \text{where head } &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)\end{aligned}$$

Where weights used for head calculation $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}, W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}, W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and fully connected layer represented as weight $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ A single attention head average all the input value, hence using multi-head attention allows the model to work on subsets. This is done by each head working on the different input information and generating m different representation subspaces. The final output comes from the weighted averages of these subspaces.

Self-Attention

When the queries, keys, and values input for attention are generated from the same input, the operation is called Self-attention. The encoder layer of transformer models usually uses a multi-head self-attention mechanism. The decoders use both multi-head self-attention and attention. In [57], the decoder model uses self-attention then the attention model that takes queries and keys from the encoder and values from self-attention.

Position Encoding

Transformers are designed to work in parallel hence they do not receive the information in a serial sequence. However, the relative order of data is important as it can help with a model to solve the relation between close values. Position encoding is a solution to solve this problem.

There are two ways to try to solve this problem: adding incremental value that increases the serial order and added with a periodic function. Adding value creates a problem for different-sized data and causes problems for long sequences as addiction can cause a loss of information. Hence, addition with a periodic function can solve

this problem as it helps with long sequences. [57] uses sine and cosine functions:

$$PE_{(\text{pos},2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right)$$
$$PE_{(\text{pos},2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right)$$

where pos is the position and i is the dimensions. The idea is to use both position of data and the index of the embedding vector to create a position value. Hence, there will be no repetition in the position encoding matrix generated by PE.

One can also use learnable position embeddings [22] however it does not show any improvements according to [57].

Vision Transformers

Vision transformers are a variant of normal transformer layers that are used for images instead of text. The aim is to obtain a similar performance as transformers show in Natural language processing. Vision transformers can be trained with self-supervised learning. They are better at parallelization than CNN models. This research shows better performance on tooth segmentation on both bitewing and panoramic images.

Self-supervised learning is a machine learning process where the model trains itself to learn one part of the input from another part of the input. It is also known as predictive or pretext learning. This learning style has advantages in the use of datasets. Creating a dataset is a costly task and the data preparation lifecycle is a lengthy process in deep learning. For medical data, it can be hard to obtain the data to start with. Vision transformers have good Self-supervised learning performance compared to CNN models. BEIT [3] is a good example of such a model. This model shows that transformers can learn features with annotation data.

Vision transformers work by converting the normal image into small patches. These patches are entered into the vision transformer layers with position embedding as seen in figure 3.14. The image $\mathrm{x} \in \mathbb{R}^{H\times W\times C}$ reshapes into patches $\mathbf{x}_p \in \mathbb{R}^{N\times\left(P^2\cdot C\right)}$ where (H, W) image resolution, C is channel number, (P, P) is the dimensions of im-

age patches.

Similar to [57], position encoding is used. The methods that can be used for 1D data can be used for 2D data since it is flattened before entering the transformer layer. The vertical relation between patches becomes important with the use of images. Therefore, using 2D position embedding becomes an option. However, testing different position encoding methods for the VIT model shows similar performance [19].

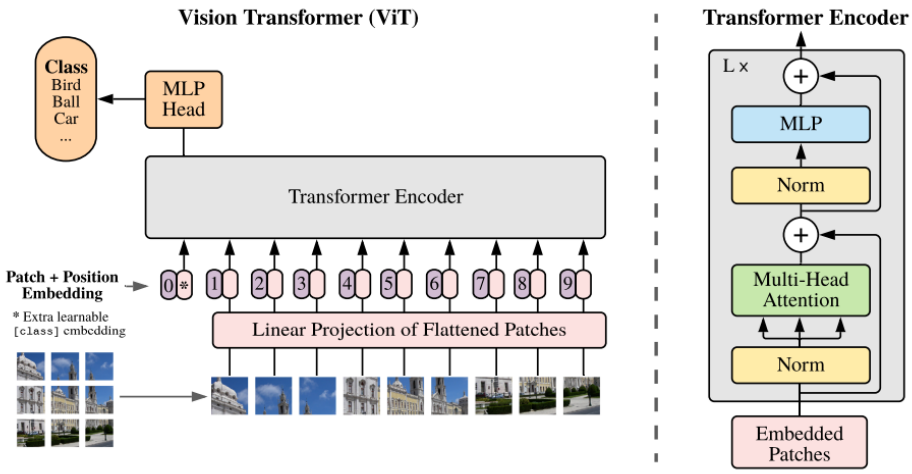


Figure 3.14: Model overview of Vision Transformer Network. The Transformer Encoder is the same as used in figure 3.13. The vision transformers convert and split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder [19].

3.2.2.1 Classification

Many of the transformer-based models have some benchmark in public datasets like ImageNet. These models are can be used as the backbone for more complicated tasks like segmentation tasks. Hence, two transformer models are feasible to use as the backbone. These are Vision transformer (VIT) [19] and Swin transformer[44].

Swin Transformer

Swin Transformer [44] is a hierarchical transformer model as seen in figure 3.15. It works on image patches and uses position embedding the same as VIT. The swin transformer uses an operation called patch merging to decrease the size of the image while increasing the deep of the image. These create levels similar to ResNet models and create fine-grained feathers. Swin transformers also use layers called Window multi-head attention (W-MSA) and Shifted Windows multi-head attention (SW-MSA) in layers as shown in figure 3.16. The windows-based multi-head attention layers also improve the computation performance over the normal multi-head attention layer of VIT as it requires less multiplication. Moreover, Swin transformer architecture also allows it to replace existing models that use ResNet backbone like FastFCN and MaskRCNN because of hierarchical structure.

While VIT works on the same non-overlapping image patches in layers, the Swin transformer uses cyclic shift to change the image patches after each layer as seen in figure 3.17. It also proposes windowed-based attention which improves the computation performance compare to the VIT attention layer. Swin transformer architecture also allows it to replace existing models that use ResNet backbone like FastFCN and Mask RCNN. Swin Transformer has three additional operations to VIT layers: patch merging, Window multi-head attention (W-MSA), and shifted windows multi-head attention (SW-MSA).

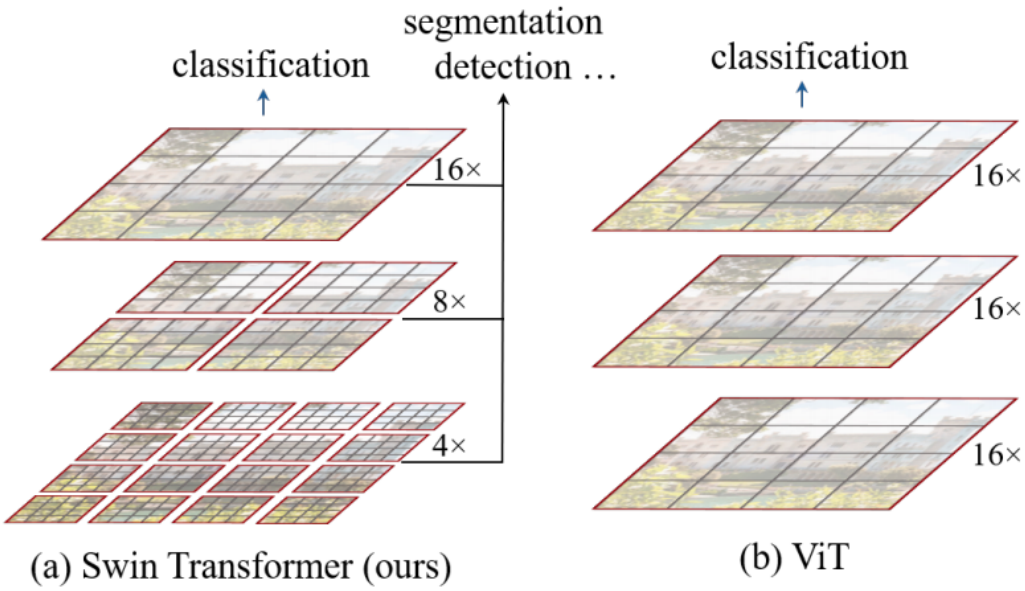


Figure 3.15: (a) The Swin transformer hierarchical feature maps. (b) VIT feature maps.

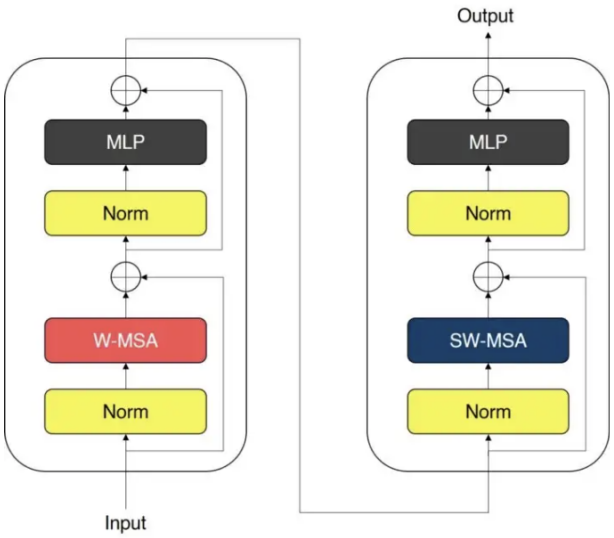


Figure 3.16: Two successive Swin transformer blocks. W-MSA and SW-MSA are multi-head self-attention modules with regular and shifted windowing configurations, respectively. [44]

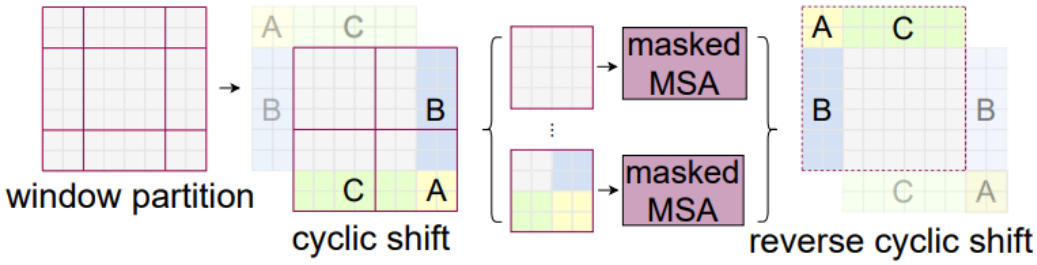


Figure 3.17: Cyclic shift of Swin transformer layer.

Patch Merging

Patch merging is an operation that makes the Swin transformers hierarchical. In basic terms, it splits the input image into patches and stacks patches depth-wise. Then, merge these stacked patches to create an image with a smaller resolution but a higher channel. The operation downsamples the image by n as the number of patches. The image transforms from $H \times W \times C$ to H to $(H/n) \times (W/n) \times (n^{2*}C)$.

Window-based Multi-Head Self Attention

In VIT, multi-head attention helps to improve the score and obtains a better understanding of the model by dividing the attention operation. W-MSA achieves this by reducing total the computation. The W-MSA layer uses images that are processed by patch merging operation. The layer creates windows on image patches, and patches in the same window enter the attention operation. These reduce the complexity of the multi-head operation and ease its use for high-dimension images.

Shifted Window-based Multi-Head Self Attention (SW-MSA)

In W-MSA, self-attention is restricted to window regions. This reduces the model's global understanding of input. To solve this problem, SW-MSA is proposed. SW-MSA uses cyclic shift to shift the patches as seen in figure 3.17. Then, it uses masked MSA only to work on the original feature map. The final windows that are used for MSA are shown in figure 3.18.

3.2.2.2 Semantic Segmentation

The semantic segmentation with transformers is no different than CNN models on function. They are made of the same layers as used in classification. However, they do not fully transform. Their architecture is mixed with CNN layers. In this part, two transformer models will be explained; TransUnet [11] and SwinUnet [7].

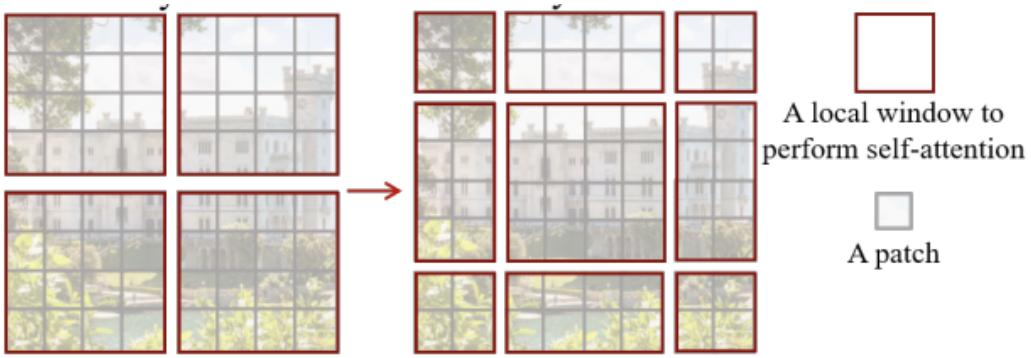


Figure 3.18: An illustration of the shifted window approach for computing self-attention in the proposed Swin Transformer architecture [44]

TransUnet

TransUnet is a hybrid architecture that uses both CNN layers and VIT layers. The architecture is based on the U-net model. To improve the fine grain feature extraction at lower layers. It uses a Transformer block at the lowest layer of the U-net. They also keep the decoder as the CNN layers and show that it performs better them a full transformer network. The overall architecture can be seen in figure 3.19

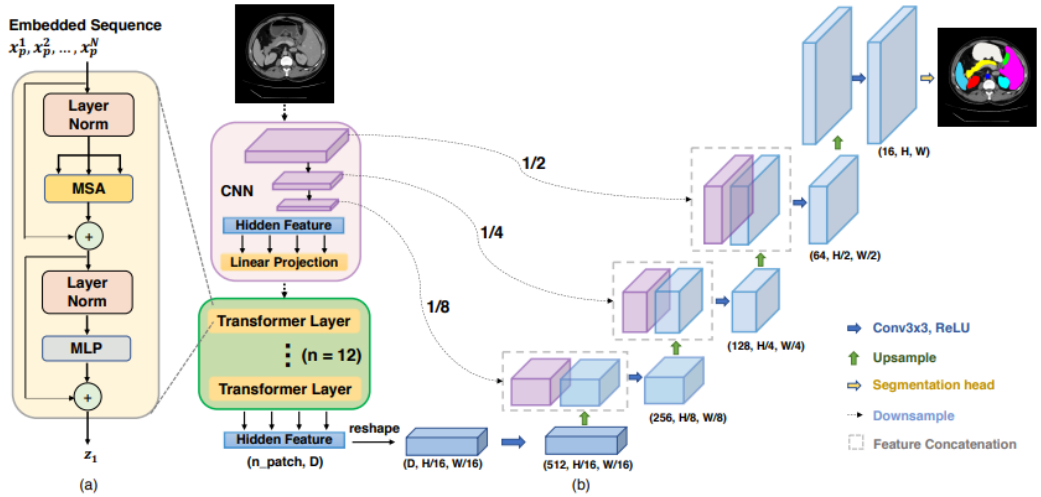


Figure 3.19: Overview of the TransUnet. [11] (a) schematic of the Transformer layer; (b) the architecture of the proposed TransUnet.

SwinUnet

SwinUnet is a U-net-like pure transformer model. They use Swin transformers' hierarchical structure to create levels. Each Patch merging layer downsizes the image and increases the feature map dimensions. In the decoder stage, they use patch merging that concatenates the dimension and increases image size while reducing the feature dimensions. The model architecture can be seen in figure 3.20 This model outperforms other U-net-like models.
The full transformer architecture has two advantages over other models observed during this model's training. Firstly, it backpropagates faster than full CNN models however it still needs more time to train compared to CNN models. Secondly, the model can take full advantage of the self-supervised learning that self-attention layers provide.

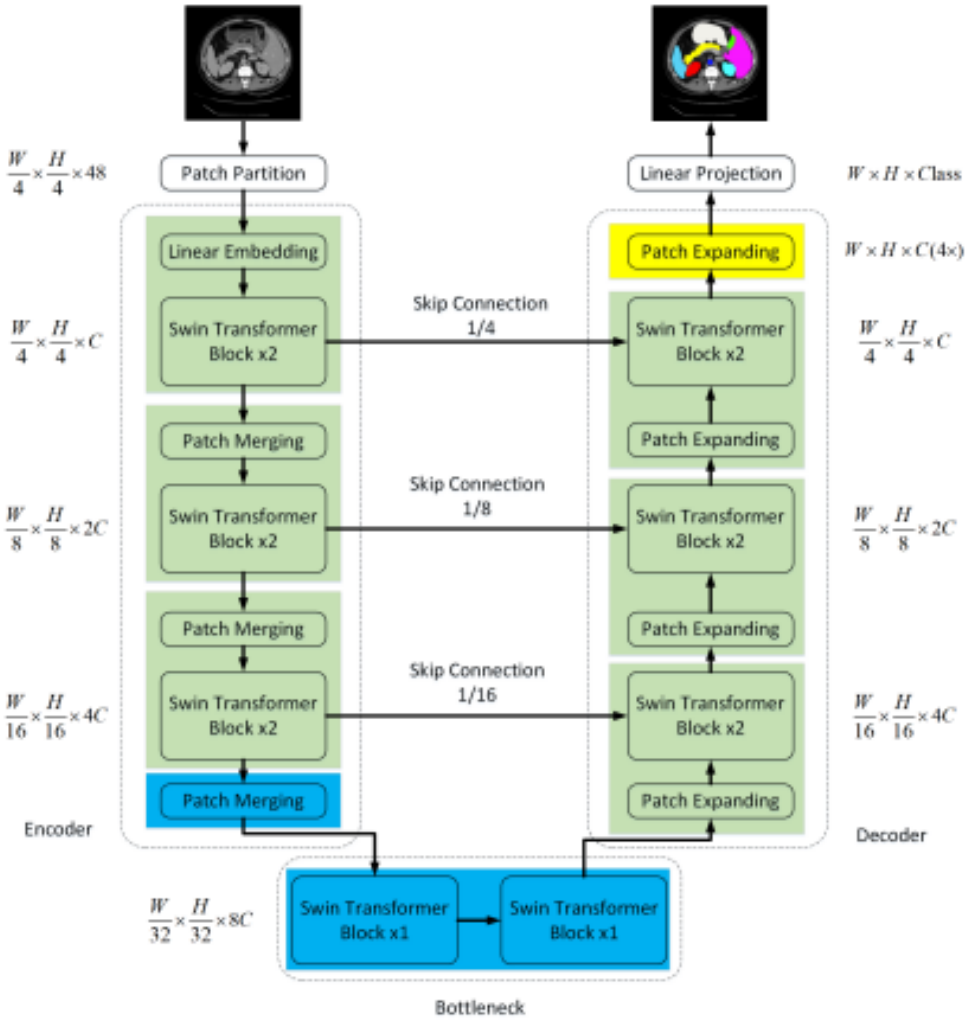


Figure 3.20: Overview of the SwinUnet. [7]

3.3 Loss Functions

Loss functions are used for the training of neural networks. During the training stage of a neural network, parameters are updated based on the loss function's output value. The loss function can be represented by equation 3.1. x and $\mathbf{y}$ are the input and the ground truth respectively. $f()$ is the neural network represented as a function that output is expected to be $f(x) == \mathbf{y}$. The final value of the loss function is a single scalar value. This value can obtainable from a single loss function or a weighted sum of several loss functions.

$$\mathbf{L}_{loss} = L(f(x), \mathbf{y}) \tag{3.1}$$

Selecting a loss function is not an easy task. There are several loss functions that perform better on performance metrics for different tasks. For example, one may prefer cross-entropy loss over dice loss for a classification task. However, dice loss may perform better than cross-entropy loss on a segmentation task. Therefore, the selection of a loss function requires a lot of iteration with a selected model and dataset. Custom loss functions can be used for different cases. For example, [16] aims to multi-class segmentation on Fundus images. The medical images are high-resolution images and the target labels are small and many. Hence, a custom loss performs better than dice or cross-entropy.

The loss functions that have been studied are mean square error (MSE), mean absolute error(MAE), binary cross-entropy (BCE), label smoothing cross-entropy (LSCE), dice loss, and focal loss. Each of these loss functions has an advantage over others. MSE is used as a baseline loss function. The BCE usually performs better than MSE. However, BCE and MSE performance decrease when a dataset is noisy. Hence, LSCE or MAE is used to improve performance. For imbalanced datasets, focal loss is used to improve the results. Dice loss is very useful to guarantee a label prediction, it also improves the performance when used in addition to other loss functions like BCE.

3.3.0.1 Mean Square Error Loss (MSE)

Mean square error is a loss function that is usually used for regression tasks. However, it performs well on semantic segmentation too.

MSE is calculated by averaging the squared differences between the target value and model prediction. MSE is shown in equation 3.2. The $\mathbf{y}$ and $\hat{y}$ are labels and predictions respectively. One can replace $\hat{y}$ with $f(x)$ where x is the input to the model. n is the number of pixels in the output of the model.

$$\mathbf{L}_{mse} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.2}$$

3.3.0.2 Mean Absolute Error (MAE)

Mean Absolute Error is used for noisy labeled dataset [23]. It outperforms BCE and MSE loss functions, as these loss functions are affected by noise in the training set. MAE is calculated by averaging the absolute difference between the target value and model prediction. Different from MSE, MAE does not use square operation hence it does not amplify the large differences between target and prediction values. Also, the use of absolute value operation makes the function symmetrical which makes it noise-robust.

$$\mathbf{L}_{mae} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.3}$$

3.3.0.3 Dice Score

The dice score is a region-based loss function. It shows the similarity between target and prediction classes. It is used for segmentation tasks and performs sufficiently well in many cases. It forces the model to match at least a few pixel values with the target otherwise loss value will be higher compared to other losses like CCE.
The calculation of this score is done as shown in equation 3.4. p_{true} and p_{pred} stand for target class values and prediction values. ϵ value is added to prevent zero division. The dividend part shows two times the correct prediction, and the divisor part shows

the sum of predictions and targets.

$$\mathbf{L}_{dice} = 1 - \frac{2 * \sum p_{\text{true}} * p_{\text{pred}}}{\sum p_{\text{true}}^2 + \sum p_{\text{pred}}^2 + \epsilon} \tag{3.4}$$

3.3.0.4 Binary Cross-Entropy(BCE)

Binary Cross-Entropy can be used with tasks with two classes. It's a Distribution-based Loss function. It uses a cross-entropy function, and common use cases are for object classification and binary pixel-level classification. BCE is also called Log loss.

BCE is defined as:

$$\mathbf{L}_{bce} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(p) + (1 - y_i) log(1 - p)) \tag{3.5}$$

p and y are the predictions and target values. For BCE, the target value is either 1 or 0. The prediction is the output of an activation function like the softmax layer which also normalizes the prediction values between $[0, 1]$. In the segmentation task, equation 3.5 is computed for all pixels in the output image.

3.3.0.5 Categorical cross-entropy (CCE)

Unlike BCE, categorical cross-entropy can be used for more than two classes. CCE is defined as:

$$\mathbf{L}_{cce} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(\hat{y}_i)) \tag{3.6}$$

Here, i shows the label index. For any pixel value, only one of them has the value 1, and the loss value for that pixel is the calculated log value of the prediction of the model for that pixel. The negative mean of these loss values is CCE loss.

3.3.0.6 Label Smoothing Cross Entropy(LSCE)

Label smoothing cross-entropy is used for noisy labeled data sets. The idea is to make the model less confident in its predictions. Thus, the model becomes obtains a more

generalized solution for a given noisy dataset compared to CCE.
The equation 3.6 is changed by modifying the p to given equation;

$$p' = (1 - \epsilon)p + \frac{\epsilon}{n} \tag{3.7}$$

Here, ϵ is used to increase zero values and decrease the ones. Hence, the model cannot just predict ones and zeros. Hence, the final equation is given as:

$$\begin{aligned} \mathbf{L}_{lsce} &= -\sum_{i=1}^{n} (p'_i \log(\hat{y}_i)) \\ &= \sum_{i=1}^{n} \left((1 - \epsilon) p_i \log(\hat{y}_i) + (\frac{\log(\hat{y}_i)}{n}) \right) \end{aligned} \tag{3.8}$$

3.3.0.7 Focal Loss

Focal loss[42] is designed to be used on imbalanced datasets. It has two hyper-parameters that can be tuned to improve the model performance. The function uses a dynamically calculated weight parameter to balance the dataset. Hence, the dominant classes' effect on the loss can be diminished, and minority classes can have a higher weight.

Focal loss is defined as:

$$p_{\mathrm{t}} = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \tag{3.9}$$

$$\mathbf{L}_{fl} = -\frac{1}{n} \sum_{t=1}^{n} \left(\alpha_{\mathrm{t}} \left(1 - p_{\mathrm{t}} \right)^{\gamma} \log \left(p_{\mathrm{t}} \right) \right) \tag{3.10}$$

α_{t} and γ parameters are hyper-parameters that can be tuned manually. The $(1 - p_{\mathrm{t}})^{\gamma})$ part reduces the value loss of the successfully predicted classes and diminishes its effect on the updated model.

3.4 Metric Definitions

The following metrics are used to evaluate the neural network model performance.

3.4.0.1 Confusion Matrix

The confusion matrix is a matrix that use to compare the performance of the model on target classes. The confusion matrix shows raw performance like TP, FP, FN, and TN scores for each class as seen in table **??**. It shows a good representation of model performance and it can be transferred to other metrics.

A confusion matrix is also useful for multi-class tasks. Since it shows models performance between each class. This helps to tune the model on similar classes or analysis of the dataset on these classes.

Table 3.3: A basic binary confusion matrix.

		True Class	
	Total Population = P + N	Positive(P)	Negative(N)
Predicted Class	Positive (PP)	TP	FP
	Negative (NN)	FN	TN

3.4.0.2 Accuracy, Precision, Sensitivity, Specificity, F1 score

There are several scores that can be obtained from the confusion matrix. Accuracy (3.11) shows correct predictions over the total amount of predictions.

$$\text{accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.11}$$

Precision (3.12) is a measure to show the quality of positive predictions of the model.

$$\text{precision } = \frac{TP}{TP + FP} \tag{3.12}$$

Sensitivity (3.13) is a measure to show the percentage of positive has been identified correctly.

$$\text{sensitivity } = \text{recall} = \frac{TP}{TP + FN} \tag{3.13}$$

Specificity 3.14 is a measure to show the quality of negative predictions of the model.

$$\text{specificity} = \frac{TN}{TN + FN} \tag{3.14}$$

F1-score is a harmonic mean of precision and recall. It is a robust and useful metric compared to precision and recall. Since threshold value can be used to increase the precision or lowering it increases recall. For highest F1-score threshold values for classes had to be fine-tuned.

$$F1 = \frac{2 \times \text{ precision } \times \text{ recall}}{\text{precision } + \text{ recall}} \tag{3.15}$$

CHAPTER 4

PROPOSED METHOD

4.1 Motivation

For teeth segmentation on both bitewing and panoramic images, the following problems have to be resolved:

- Background-foreground imbalance for panoramic images.
- Difference scale, distribution, and resolution between two datasets.
- Class imbalances between bitewing image and panoramic image count.

As seen in section 3.1, on average %77 percent of the image is background region. This ratio is %45 percent in bitewing images. Also, panoramic images contain more bone structure than bitewing. Center cropping is a method used in [48] to solve this problem. However, using center cropping reduces the results from bitewing and panoramic images. Hence, this method is not adapted. Random cropping is used in the augmentation pipeline to create samples with different background foreground ratios.

Another problem is the difference between the two datasets. These datasets have different imaging techniques. Even though, teeth can be seen clearly to the human eye in both datasets. Without any change in datasets, the neural network models do not learn both of them. One may try to solve this problem with deep multimodal learning methods like multimodal regularization and fusion structure learning and optimization [49]. However, the two datasets are similar enough to solve by using image augmentation. The problem is simpler than multimodel use cases.

A custom batch sampler is written to solve the problem of the imbalanced data count. This gives additional control over model training. With a custom batch sampler, one can adjust the ratio of images taken from each dataset used in the batch.

4.2 Image Augmentation

Augmentation is a method used to increase the search space. It helps to balance the difference between classes and improves the model's robustness. This effect also helps with model over-fitting. In image augmentation, there are many transformations that can be used. These transformations can be separated into two classes: Geometric and color based. The transformations that fall under these classes are given in table 4.1.

Table 4.1

Geometrical Transformation	Color Based Transformation
Rotate	Equalize
Vertical Shift	Solarize
Horizontal Shift	Random Noise
Vertical Shear	Posterize
Horizontal Shear	Brightness

4.2.1 Center Cropping

Center cropping is a solution for solving the foreground and background imbalances in the panoramic dataset. During the training and testing stages, center cropping is always applied. This method is the robust and basic method that works unless the panoramic machine changes. Cases like improved resolution or orientation may cause problems for future use.

4.2.2 Rotation

Rotation is a geometric transformation and it is the most important one. In [18], rotation achieves the highest effect on average improvements by %0.01 percent. In this thesis, the effect of rotation is %0.15 percent. The proposed idea is rotation breaks the static structure of panoramic images. Hence, the neural network does not depend on the orientation of the image and learns the features of a tooth.

4.2.3 Rand-Augmentation

There are several methods and libraries that can be used to design an image augmentation pipeline. There are four different methods that are considered:

- Using independent probabilities values for all transformations to decide apply or not.
- AutoAugmentation (AA) [17],
- Population-Based Augmentation (PBA)[28],
- Randaugmentation (RA) [18],

The fastest method for developing an augmentation pipeline is to use independent probabilities for each transformation. This method is easy to use however it lacks stability. The transformed image can vary and some outputs do not look like a dental image at all.

AA is a method already implemented in the albumentation library [6]. The AA work by designing the augmentation parameters automatically. However, this method is designed for classification tasks and it works slowly. PBA is an alternative that works faster and more dynamically. However, it has a higher search space of parameters, and it consumes additional computational resources.

RA is a better alternative to the other three methods. It is simple to implement and

deploy. It also does not have any additional computational cost. RA work by randomly selecting the augmentation operation. It selects N operation in total for each iteration. The magnitude of these operations is randomly decided in each iteration. The maximum range of magnitude is determined by the parameter M. Hence, RA has two important hyperparameters N and M. N and M were selected as 3 and 4 respectively. Since the most effective transformation is rotation, it is used with an increased probability to be selected more often than other transformations.

4.3 Neural Network Training Pipeline

Selecting correct methods and hyperparameters for a training neural network is an essential part of the segmentation problem. From dataset to dataset, different setups can have different performances. Hence, testing different training settings is important. There are five setting decisions made in the training pipeline: neural network model, optimizer, loss scheduler, loss function, and batch sampler.

4.3.1 Model Selection

In section3.2, neural network models are described in detail. Both the transformer and CNN-based models are tested on panoramic and bitewing datasets. The results showed the transformer models performed better than the CNN-based model on both of the datasets. This result is compatible with the fact at transformers have a better fine-grain feature extraction performance than CNN models when the dataset is large enough. Due to this, transformer-based models are used for experimental evaluation.

At the time, swin transformers [44] is the best model that achieved the highest score in the public classification and segmentation datasets like coco and cityscapes. However, for bitewing and panoramic datasets, transUnet [11] performs better than the swinUnet [7]. However, The swinUnet is a faster and lighter model. In this thesis, transUnet is selected as the base model since metric performance is more important than speed in medical research fields. The performance difference between models will be given in chapter 5.

4.3.2 Optimizer Selection

There are two popular optimizers; ADAM and Stochastic Gradient Descent (SGD). In [19], ADAM with weight decay shows better performance compare to SGD which is the opposite for CNN-based models like ResNet.
For dental datasets, ADAM and SGD are both tested for the proposed model, and ADAM shows better performance. It also converges much faster than SGD which improves the number of hyperparameter tests that can be contacted.

4.3.3 Scheduler Selection

The learning rate (LR) is a value that determines the speed at which a machine-learning model is able to learn and make updates to its parameters. It affects the model's ability to accurately find the optimal solution and can have a significant impact on its performance. If the learning rate is too high, the model may make large, inaccurate updates to its parameters. If the learning rate is too low, the model may take a long time to learn and may not perform as well. The learning rate has to high enough. It is important to choose an appropriate learning rate for the selected model. There are two approaches used to find an appropriate learning rate: Warm-up, and cosine annealing. Warm-up[25] is a procedure that starts with a very low learning rate and increases it gradually for a few epochs. An example learning rate that uses warm-up can be seen in figure 4.1. This method is successful at preventing early over-fitting and it returns the learning rate to a higher value. Using a high learning rate at in the training is important as it helps the model to get a better generalization of data[39]. The Cosine annealing [46] is a successful scheduling method that outperforms the other tested options like Step LR and exponential LR. An example plot of the learning rate by these three schedulers is given in figure 4.2. The use of warm-up start and cosine annealing gives the best result.

4.3.4 Loss Function Selection

There are many loss functions that can be used for different cases. Depending on the dataset and neural network model loss functions selection becomes essential. For

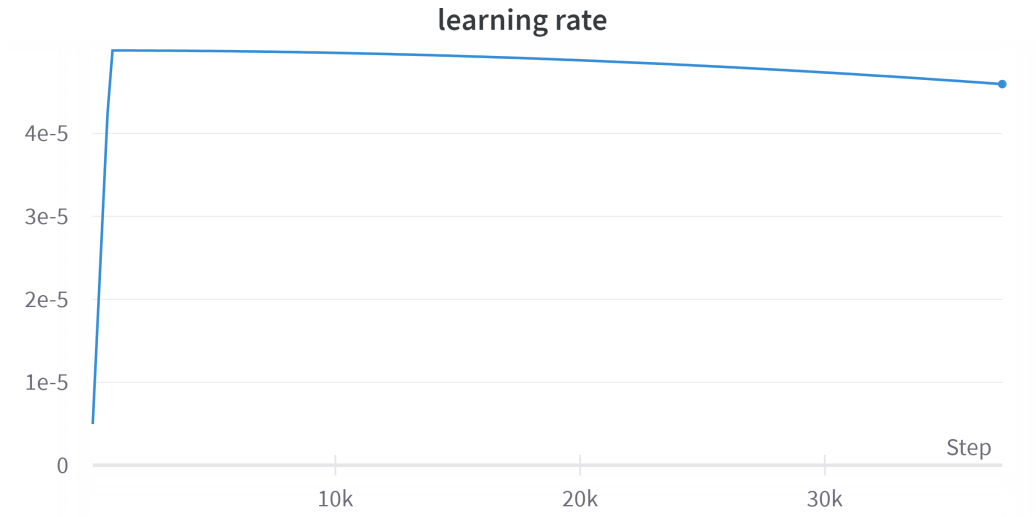


Figure 4.1: An example learning rate graph with a warm-up start.

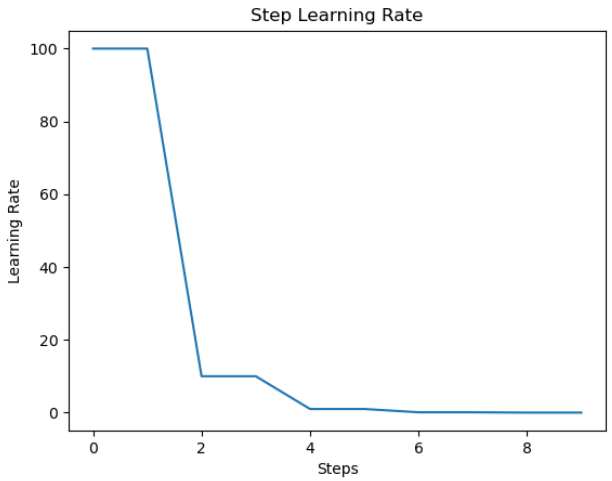
example, Dice loss is essential for a highly imbalanced dataset. However, dice loss does not get the best results in datasets like ImageNet. Hence, mixed loss usage is practiced in this study. The use of mixed loss is a know application in medical studies [11, 7].

The mean of Dice loss and BCE loss is used as a loss function as this method merges the pixel-based performance and entropy-based performance. A similar combination of loss functions was also tested like MSE and focal loss which works similarly to Dice and BCE losses, respectively. These experiments showed that the use of mixed loss improves the metrics around %0.1.

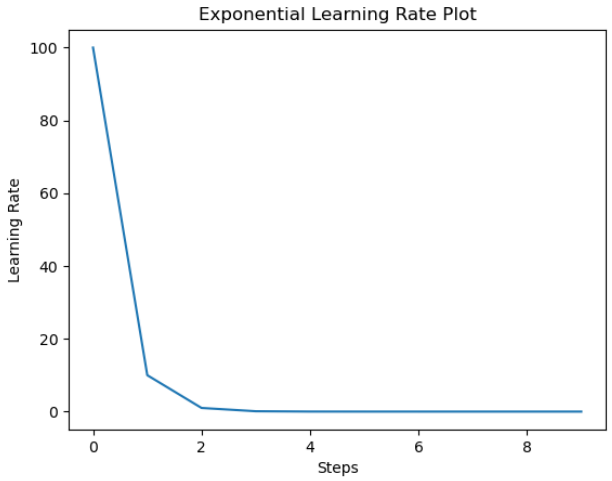
4.3.5 Batch Sampler

The data is fed to the neural network in batches. The batches are a selected set from the dataset. In the PyTorch library, the default process of batch sampling is to use random index numbers to create the batch. This works very well for the balanced dataset. However, in the combined set of bitewing and panoramic datasets, the panoramic data set has a probability of 0.96 of being selected as a sample. For batch size 4, the probability of having a bitewing image in a batch is 0.15. Hence, the neural network does not learn the bitewing as it seldomly exists in a batch.

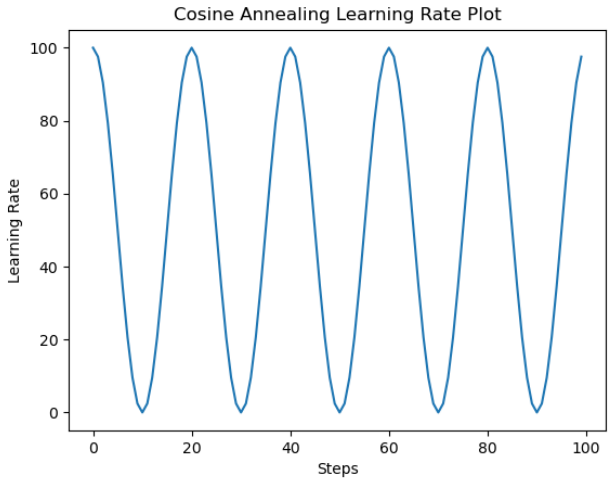
To solve this problem, there are two solutions in literature: weighted loss or balanced batch sampler. In theory, the effect of both must be the same since both solutions affect overall loss similarly. However, since the model is updated per batch, the effect of weighted loss becomes inefficient. Hence balanced batch sampler solution is favored and implemented. New batch sampler works by creating a batch with a weighted rate from bitewing and panoramic. The selected weights are 0.75 panoramic and 0.25 bitewing. This ratio gives the best score.



(a) The plot of change in learning rate with step LR function.



(b) The plot of change in learning rate with exponential LR function.



(c) The plot of change in learning rate with cosine annealing LR function.

Figure 4.2: Change is LR for different LR schedulers.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Hardware and Software Specifications

All the experiments were performed on two computers. The computer's specifications are given in table 5.1. The two pieces of hardware were used to speed up the experiments. Tests for the effect of a hyperparameter were made on the same machine. The same batch size was used for both machines. Only for batch size test, only PC 2 used as Quadro P5000 has 16 Gb of dram while Gtx 1080Ti has 11 Gb of dram. Both machines have the same Cuda library, GCC compiler, python, and Pytorch. For reproducibility of experiment results, PyTorch cudnn benchmark parameter is set to false, and the experiment was done with constant random seed values for python language, NumPy, and PyTorch libraries.

Table 5.1: Test computers hardware specifications.

Hardware	PC 1	PC 2
GPU	NVIDIA GTX 1080TI	NVIDIA QUADRO P5000
CPU	INTEL i7770	INTEL XEON
RAM	32 GB DDR4	32 GB DDR4
Operating System	Ubuntu 20.04 LTS	Ubuntu 20.04 LTS

The training time and performance can change with versions of the library. Less optimized functions or an unknown bug in software may cause different results when libraries are improved or fixed. Hence, it is important to think of the software used in these experiments. All experiments, including training, testing, and benchmarking neural network models, are done with the following versions; PyTorch 1.11 [47],

CUDA 11.3, and python 3.10. The source code can be found in https://github.com/metcan/pano_bite_segmenta

5.2 Training and Implementation Details

In this section, the effect of loss, model, batch size, image size, and augmentation selection are shown. However, some parameters are kept the same. These settings are:

- The custom batch sampler gives 3 panoramic images for each bitewing image.
- ADAM optimizer used for all experiments. The optimizer parameters are Weight Decay: 1e-5, and learning rate: 5e-5. The rest of the parameters are used in the default configuration.
- The mixed loss functions are balanced. The total loss is the mean of loss functions.

5.2.0.1 Effect of Batch Size

Batch size is an effective parameter in the training stage. It shows how many input images enter the model at the same time. Usually, the higher the batch size is better since it improves the model's generalization on the dataset. However, it is usually limited by hardware.
In this thesis, batch size 16 is used as the proposed model. As seen in table 5.2, increasing the batch size does not give the best result for both bitewing and panoramic. Batch size 48 gives the best result for a panoramic image. However, this setting does not give the best result for the bitewing dataset. Since these images are very limited. Most of the bitewing images appeal in the same batch with seems to reduce the performance. Therefore, batch size 16 is selected since the best bitewing results obtain at this batch size.

Table 5.2: The effect of batch size for validation metrics.

Batch Size	Panoramic					Bitewing				
	Acc.	Prec.	Recall	F1	Spec	Acc.	Prec.	Recall	F1	Spec
4	0,966	0,966	0,966	0,965	0,978	0,906	0,906	0,906	0,906	0,886
8	0,967	0,967	0,967	0,967	**0,979**	0,907	0,907	0,908	0,906	0,890
16	0,967	0,967	0,967	0,967	0,975	**0,908**	**0,909**	**0,908**	**0,908**	**0,907**
32	0,967	0,967	0,967	0,967	0,978	0,902	0,901	0,902	0,901	0,884
48	**0,968**	**0,968**	**0,968**	**0,968**	0,978	0,901	0,900	0,901	0,900	0,880

5.2.0.2 Effect of Augmentation and Batch Sampler

Table 5.3 shows the effect of augmentations and batch sampler on validation performance. The following step of the tests are concluded: proposed method(PM), without sampler (WS), without rotation (WR), with the center crop(WC), and without augmentation(WA). Each augmentation improves the result of bitewing performance. Their individual effect on the panoramic dataset is not significant. Using a center crop reduces the panoramic results. In [48] is suggested to improve panoramic results. This may be because of the difference in model architectures and other augmentations that are not used by [48].

5.2.0.3 Effect of Dataset Selection

In table 5.4, using both datasets (BD), only using bitewing (B) and panoramic datasets(P) are tested. They represented as As seen in this table, just using one dataset does perform worse than using two datasets. The result in just bitewing is expected as there is a low amount of these images, and the model may not fully train just by bitewing images. However, using both datasets performs on panoramic images better than just panoramic dataset. This shows that bitewing images create a challenge to model with the help of the validation dataset.

Table 5.3: The effect of augmentation and sampler.

Method	Panoramic					Bitewing				
	Acc.	Prec.	Recall	F1	Spec	Acc.	Prec.	Recall	F1	Spec
PM	**0,967**	**0,967**	**0,967**	**0,967**	**0,975**	**0,908**	**0,909**	**0,908**	**0,908**	**0,907**
WS	0,965	0,967	0,965	0,966	0,968	0,876	0,875	0,874	0,870	0,849
WR	0,967	0,967	0,967	0,967	0,975	0,893	0,893	0,893	0,893	0,878
WC	0,955	0,956	0,955	0,956	0,965	0,88	0,88	0,88	0,88	0,86
WA	0,967	0,968	0,967	0,967	0,976	0,887	0,888	0,887	0,886	0,843

Table 5.4: The effect of the training datasets.

Methods	Panoramic					Bitewing				
	Acc.	Prec.	Recall	F1	Spec	Acc.	Prec.	Recall	F1	Spec
BD	**0,967**	**0,967**	**0,967**	**0,967**	**0,975**	**0,908**	**0,909**	**0,908**	**0,908**	**0,907**
B	-	-	-	-	-	0,746	0,718	0,781	0,798	0.756
P	0,964	0,966	0,964	0,964	0,964	-	-	-	-	-

5.2.0.4 Effect of Loss Function

In table 5.5, several loss functions and their mixed versions are tested. The experiments are shown that weighted merge of DICE and BCE loss performs best for the panoramic dataset and bitewing. Even though, loss functions perform close to each other on the panoramic dataset, their performance on the bitewing dataset changes. Mixed loss of DICE and BCE also obtain good robustness as the specificity metric for both datasets is highest for DICE+BCE for the loss function.

Table 5.5: The effect of the loss functions on validation performance.

Loss	Panoramic					Bitewing				
	Acc.	Prec.	Recall	F1	Spec	Acc.	Prec.	Recall	F1	Spec
DICE+MSE	0,968	0,968	0,968	0,968	0,976	0,907	0,907	0,907	0,907	0,891
DICE+BCE	0,967	0,967	0,967	0,967	0,975	0,908	0,909	0,908	0,908	0,907
DICE	0,967	0,967	0,967	0,967	0,975	0,905	0,905	0,905	0,905	0,887
FOCAL	0,968	0,968	0,968	0,968	0,978	0,898	0,898	0,898	0,898	0,88
BCE	0,967	0,967	0,967	0,967	0,977	0,895	0,895	0,895	0,895	0,869
MSE	0,967	0,967	0,967	0,967	0,976	0,892	0,893	0,893	0,891	0,889

5.2.0.5 Effect of Image Size

The image size is an effective parameter in training. With a smaller size, some of the details on the image may be lost. However, with larger sizes, the hardware will limit the batch size, and the training will be longer.

In table 5.6, 128x128, 256x256, and 512x512 image sized are tested. The image size does not tested beyond 512x512 since training image dimensions become bigger than all bitewing images. Table 5.6 shows larger the image size better the results for panoramic images. However, the training cost of 512 by 512 images is 21 hours while 256 by 256 is 4 hours and 46 minutes. Due to resource limitations, using 256 by 256 becomes more efficient. Also, the model's bitewing performance starts to decline at 512x512.

Table 5.6: The changes in performance for different image sizes.

Image Size	Panoramic					Bitewing				
	Acc.	Prec.	Recall	F1	Spec	Acc.	Prec.	Recall	F1	Spec
128	0,959	0,96	0,959	0,959	0,97	0,907	0,907	0,907	0,907	0,903
256	0,967	0,967	0,967	0,967	0,975	0,908	0,909	0,908	0,908	0,907
512	0,972	0,972	0,972	0,972	0,98	0,911	0,912	0,911	0,911	0,902

5.3 Performance Comparison of TransUnet, SwinUnet, U-net, FastFCN, and state-of-the-art models

There are four models that tested for semantic segmentation: FastFCN, U-net, SwinUnet, and TransUnet. U-net, SwinUnet, and TransUnet all have U-shaped architecture and skip connections, While Fastfcn has a JPU layer that merges the outputs of different levels of the model.
The FastFCN is the worst-performed model in experiments. The model does not separate the teeth in the bitewing image and does not separate the empty region between the upper and lower jaw in panoramic images. This problem does not exist in U-shaped architects. The U-net model predicts the teeth boundary on both bitewing and panoramic images. The visual performance difference between U-net, SwinUnet, and TransUnet is not very clear. All models predict the edge boundaries correctly. All models have problems with bitewing images when the label is on the edges of the image. This problem is caused by the bitewing dataset as these images are converted from the real film which creates the black region around the image and writings on the image. This noise affects the model's performance.

The difference between U-shaped model appeal in metrics scores. Even though their visual performances are close to each other, the transUnet model has the best metrics scores for both panoramic and bitewing as seen in table 5.7. The results from 2 are also given in table 5.7 to show that the TransUnet model can get %90 results on the bitewing dataset while it performs close to state-of-the-art model on the panoramic images. In table 5.6, 512x512 images achieve a similar result to state-of-the-art performance in accuracy and outperform them in Recall and F1 scores. However, working with this image size is not feasible with the resources at hand.

There are no studies to compare thesis results in bitewing teeth segmentation. Bitewing images are usually used for caries detection. However, these results show that using the dental image dataset as a simple multimodel dataset can improve the overall performance. Also, the results were achieved with a small number of bitewing images. The bitewing images also include teeth with treatments. This model can be

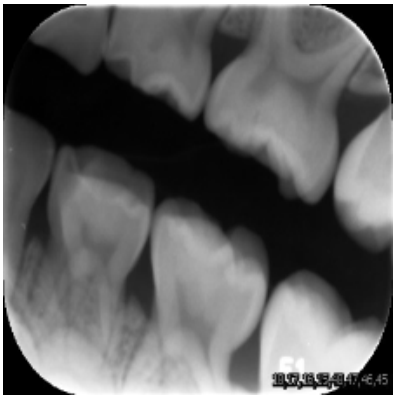
used to extract teeth from bitewing images to process by a second network or post-processing to extract more information like caries or a treatment. This accurate teeth extraction in bitewing images can result in reducing the required dataset for a more detailed analysis of bitewing images.

Table 5.7: comparison table of tested neural network models and SOTA studies on panoramic images.

Models	Panoramic				
	Acc.	Prec.	Recall	F1	Spec
TransUnet	0,967	0,967	0,967	0,967	0,975
SwinUnet	0,959	0,958	0,958	0,958	0,972
U-net	0,967	0,967	0,967	0,967	0,978
FastFCN	0,897	0,904	0,897	0,9	0,912
Chen et al [13]	0.973	0.93	0.93	-	0.98
Caylak et al [8]	0.976	-	-	0.9	-
Jader et al [29]	0.98	0.94	0.84	0.88	0.99

Table 5.8: Model Comparison comparison on bitewing settings.

Models	Bitewing				
	Acc.	Prec.	Recall	F1	Spec
TransUnet	0,908	0,909	0,908	0,908	0,907
SwinUnet	0,879	0,876	0,876	0,876	0,858
U-net	0,854	0,86	0,854	0,855	0,89
FastFCN	0,819	0,821	0,819	0,817	0,724



(a) Input bitewing image.

(b) Ground-Truth image.

(c) FastFCN prediction image.

(d) U-net prediction image.

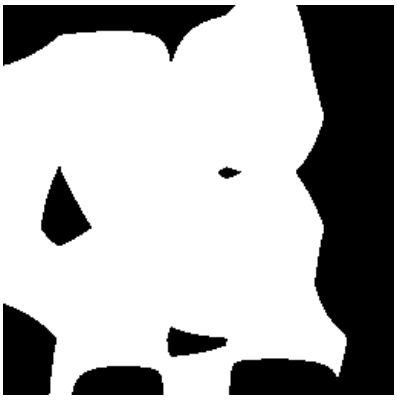
(e) SwinUnet prediction image.

(f) TransUnet prediction image.

Figure 5.1: Predictions of FastFCN, U-net, SwinUnet, and TransUnet on a healthy bitewing image.

(a) Input bitewing image.

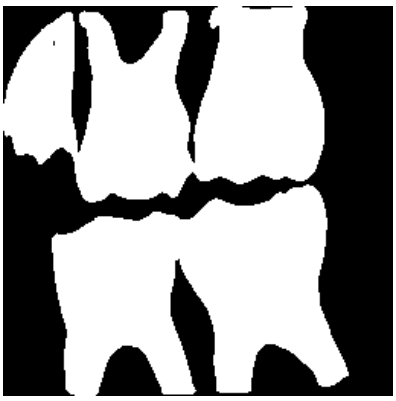
(b) Ground-Truth image.



(c) FastFCN prediction image.

(d) U-net prediction image.

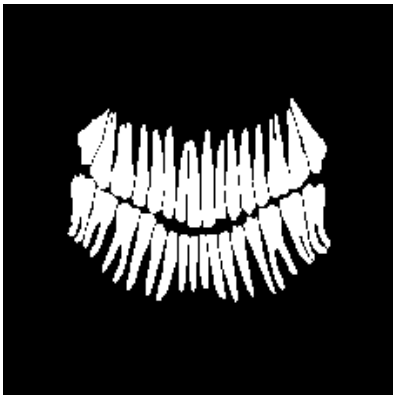
(e) SwinUnet prediction image.



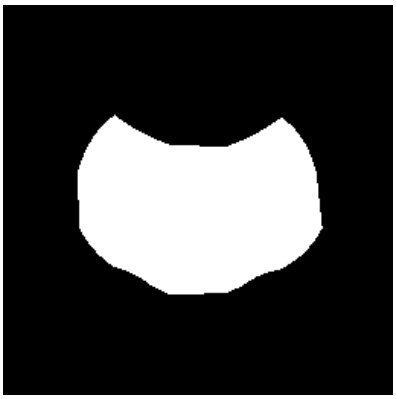
(f) TransUnet prediction image.

Figure 5.2: Predictions of FastFCN, U-net, SwinUnet, and TransUnet on a bitewing image with treatments.

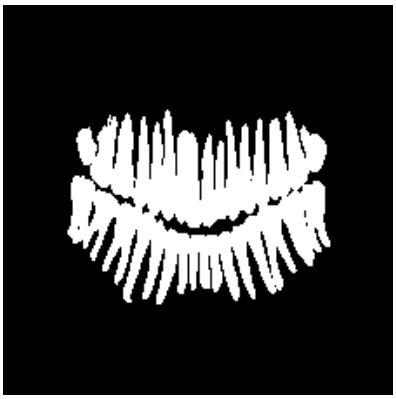
(a) Input panoramic image.



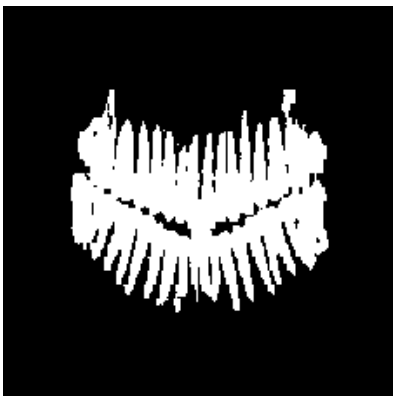
(b) Ground-Truth image.



(c) FastFCN prediction image.



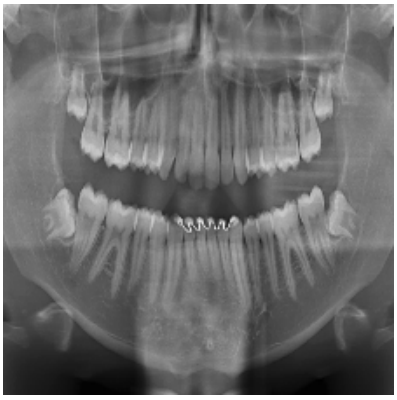
(d) U-net prediction image.



(e) SwinUnet prediction image.

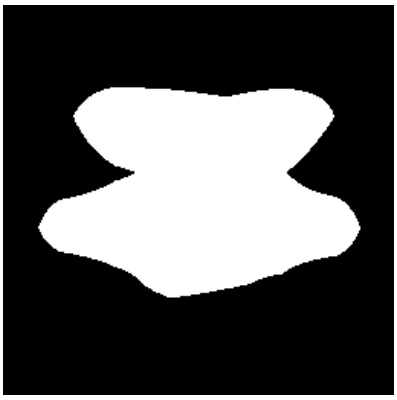
(f) TransUnet prediction image.

Figure 5.3: Predictions of FastFCN, U-net, SwinUnet, and TransUnet on a panoramic image with the closed jaw.



(a) Input panoramic image.

(b) Ground-Truth image.



(c) FastFCN prediction image.

(d) U-net prediction image.

(e) SwinUnet prediction image.

(f) TransUnet prediction image.

Figure 5.4: Predictions of FastFCN, U-net, SwinUnet, and TransUnet on the Panoramic with wide distance between upper and lower jaw.

CHAPTER 6

CONCLUSION

In this study, a neural network training method was proposed to segment teeth on both panoramic and bitewing images. The proposed method achieves state-of-the-art results on panoramic images and achieves over %90 percent on the accuracy, precision, recall, F1-score, and specificity metrics on bitewing images.

There are three problems that are required to be solved to achieve these results. Firstly, the panoramic dataset has an imbalanced distribution of background and foreground. Secondly, the panoramic dataset is much larger than bitewing. Finally, bitewing and panoramic images have differences in scale, orientation, and resolution.

The solution to the first problem is using random cropping to create images with different background and foreground rations. The second problem is solved by using the custom batch sampler. The aim is to generate each batch with an adjustable amount of bitewing and panoramic images. This way loss generated from each batch is based on both bitewing and panoramic images. The use of image augmentation solves the final problem. The augmentation on bitewing increases the amount of bitewing number and the augmentation on panoramic images forces the model to learn teeth features, not position.

There are several experiments concluded to select the best neural network model, optimizer, scheduler, and loss function. For the model, the use of two datasets will require a more robust model. Transformers-based models are selected as these models can learn more about input data because of self-attention. The optimizer and scheduler are selected based on testing, and other transformer implementation and

source codes. The loss function is selected based on testing. These choices result in performance similar to state of art models in panoramic images and %90.7 accuracy performance in the bitewing dataset.

The successful segmentation of teeth in bitewing images can lead to many benefits. The studies on panoramic images show that with increasing segmentation performance, the numbering of teeth also increases. Numbering in the bitewing images can have many benefits like auto treatment registration. This means it will be easier to hold the patient's treatment history or register the patient. In the bitewing dataset, the model also predicts the large caries regions, root canal treatments, and crowns. It can also be used for future annotation tools to separate each tooth.

As future work, a periapical dataset can be added to the datasets. This way, all dental X-ray images can be segmented by one model. Also, segmentation of teeth structure on all dental image types can be added to improve the functionality of current models. Caries detection can be added to the current model with a large enough bitewing or periapical dataset. Then, a self-supervised train can be done to find caries on panoramic images.

REFERENCES

[1]Yusra Y Amer and Musbah J Aqel. “An efficient segmentation algorithm for panoramic dental images”. In: *Procedia Computer Science* 65 (2015), pp. 718–725.

[2]Tugba Ari et al. “Automatic Feature Segmentation in Dental Periapical Radiographs”. In: *Diagnostics* 12.12 (2022), p. 3081.

[3]Hangbo Bao, Li Dong, and Furu Wei. “Beit: Bert pre-training of image transformers”. In: *arXiv preprint arXiv:2106.08254* (2021).

[4]Ibrahim Sevki Bayrakdar et al. “Deep-learning approach for caries detection and segmentation on dental bitewing radiographs”. In: *Oral Radiology* 38.4 (2022), pp. 468–479.

[5]Benjamin Bergner et al. “Interpretable and Interactive Deep Multiple Instance Learning for Dental Caries Classification in Bitewing X-rays”. In: *arXiv preprint arXiv:2112.09694* (2021).

[6]Alexander Buslaev et al. “Albumentations: Fast and Flexible Image Augmentations”. In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[7]Hu Cao et al. “Swin-unet: Unet-like pure transformer for medical image segmentation”. In: *arXiv preprint arXiv:2105.05537* (2021).

[8]Tulin Caylak et al. “Automated Dental Panoramic Image Segmentation Using Transfer Learning Based CNNs”. In: *2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE. 2022, pp. 1–4.

[9]A Chahi, Y Ruichek, R Touahni, et al. “Local directional ternary pattern: A new texture descriptor for texture classification”. In: *Computer vision and image understanding* 169 (2018), pp. 14–27.

[10]Hu Chen et al. “A deep learning approach to automatic teeth detection and numbering based on object detection in dental periapical films”. In: *Scientific reports* 9.1 (2019), pp. 1–11.

[11]Jieneng Chen et al. “Transunet: Transformers make strong encoders for medical image segmentation”. In: *arXiv preprint arXiv:2102.04306* (2021).

[12]Liang-Chieh Chen et al. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[13]Qiaoyi Chen et al. “MSLPNet: multi-scale location perception network for dental panoramic X-ray image segmentation”. In: *Neural Computing and Applications* 33.16 (2021), pp. 10277–10291.

[14]Yunpeng Chen et al. “Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3435–3444.

[15]Minyoung Chung et al. “Individual tooth detection and identification from dental panoramic x-ray images via point-wise localization and distance regularization”. In: *Artificial Intelligence in Medicine* 111 (2021), p. 101996.

[16]Elif K Çontar. “SEGMENTATION OF MULTI CLASS RETINAL LESIONS FROM FUNDUS IMAGES”. MA thesis. Middle East Technical University, 2022.

[17]Ekin D Cubuk et al. “Autoaugment: Learning augmentation policies from data”. In: *arXiv preprint arXiv:1805.09501* (2018).

[18]Ekin D Cubuk et al. “Randaugment: Practical automated data augmentation with a reduced search space”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 702–703.

[19]Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[20]Shang-Hua Gao et al. “Res2Net: A New Multi-Scale Backbone Architecture”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.2 (2021), pp. 652–662. DOI: 10.1109/TPAMI.2019.2938758.

[21]V Geetha, KS Aprameya, and Dharam M Hinduja. “Dental caries diagnosis in digital radiographs using back-propagation neural network”. In: *Health Information Science and Systems* 8.1 (2020), pp. 1–14.

[22]Jonas Gehring et al. “Convolutional sequence to sequence learning”. In: *International conference on machine learning*. PMLR. 2017, pp. 1243–1252.

[23]Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. “Robust loss functions under label noise for deep neural networks”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017.

[24]Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[25]Priya Goyal et al. “Accurate, large minibatch sgd: Training imagenet in 1 hour”. In: *arXiv preprint arXiv:1706.02677* (2017).

[26]Arman Haghanifar, Mahdiyar Molahasani Majdabadi, and Seok-Bum Ko. “Paxnet: Dental caries detection in panoramic x-ray using ensemble transfer learning and capsule classifier”. In: *arXiv preprint arXiv:2012.13666* (2020).

[27]Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[28]Daniel Ho et al. “Population based augmentation: Efficient learning of augmentation policy schedules”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2731–2741.

[29]Gil Jader et al. “Deep instance segmentation of teeth in panoramic X-ray images”. In: *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE. 2018, pp. 400–407.

[30]Mete Can Kaya and Gözde Bozdağı Akar. “Dental X-ray Image Segmentation using Octave Convolution Neural Network”. In: *2020 28th Signal Processing and Communications Applications Conference (SIU)*. IEEE. 2020, pp. 1–4.

[31]Jaeyoung Kim et al. “DeNTNet: Deep Neural Transfer Network for the detection of periodontal bone loss using panoramic dental radiographs”. In: *Scientific reports* 9.1 (2019), pp. 1–9.

[32]Jong-Eun Kim et al. “Transfer learning via deep neural networks for implant fixture system classification using periapical radiographs”. In: *Journal of clinical medicine* 9.4 (2020), p. 1117.

[33]Münevver Coruh Kılıc et al. “Artificial intelligence system for automatic deciduous tooth detection and numbering in panoramic radiographs”. In: *Dentomaxillofacial Radiology* 50.6 (2021), p. 20200172.

[34]Thorbjørn Louring Koch et al. “Accurate segmentation of dental panoramic radiographs with U-Nets”. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE. 2019, pp. 15–19.

[35]Jae-Hong Lee et al. “Detection and diagnosis of dental caries using a deep learning-based convolutional neural network algorithm”. In: *Journal of dentistry* 77 (2018), pp. 106–111.

[36]Jae-Hong Lee et al. “Diagnosis and prediction of periodontally compromised teeth using a deep learning-based convolutional neural network algorithm”. In: *Journal of periodontal & implant science* 48.2 (2018), pp. 114–123.

[37]Jeong-Hee Lee et al. “Application of a fully deep convolutional neural network to the automation of tooth segmentation on panoramic radiographs”. In: *Oral surgery, oral medicine, oral pathology and oral radiology* 129.6 (2020), pp. 635–642.

[38]Shinae Lee et al. “Deep learning for early dental caries detection in bitewing radiographs”. In: *Scientific reports* 11.1 (2021), pp. 1–8.

[39]Yuanzhi Li, Colin Wei, and Tengyu Ma. “Towards explaining the regularization effect of initial large learning rate in training neural networks”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[40]Dong Hoon Lim. “Robust edge detection in noisy images”. In: *Computational Statistics & Data Analysis* 50.3 (2006), pp. 803–812.

[41]Phen-Lan Lin, Yan-Hao Lai, and Po-Whei Huang. “An effective classification and numbering system for dental bitewing radiographs using teeth region and contour information”. In: *Pattern Recognition* 43.4 (2010), pp. 1380–1392.

[42]Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[43]Nian Liu, Junwei Han, and Ming-Hsuan Yang. “Picanet: Learning pixel-wise contextual attention for saliency detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3089–3098.

[44]Ze Liu et al. “Swin transformer v2: Scaling up capacity and resolution”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12009–12019.

[45]Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[46]Ilya Loshchilov and Frank Hutter. “Sgdr: Stochastic gradient descent with warm restarts”. In: *arXiv preprint arXiv:1608.03983* (2016).

[47]Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[48]Laıs Pinheiro et al. “Numbering permanent and deciduous teeth via deep instance segmentation in panoramic x-rays”. In: *17th International Symposium on Medical Information Processing and Analysis*. Vol. 12088. SPIE. 2021, pp. 95–104.

[49]Dhanesh Ramachandram and Graham W Taylor. “Deep multimodal learning: A survey on recent advances and trends”. In: *IEEE signal processing magazine* 34.6 (2017), pp. 96–108.

[50]Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “Dental X-ray image segmentation using a U-shaped Deep Convolutional network”. In: *International Symposium on Biomedical Imaging*. Vol. 1. 2015, p. 3.

[51]Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[52]Omran Salih and Kevin Jan Duffy. “The local ternary pattern encoder–decoder neural network for dental image segmentation”. In: *IET Image Processing* (2022).

[53]Bernardo Silva et al. “A study on tooth segmentation and numbering using end-to-end deep neural networks”. In: *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE. 2020, pp. 164–171.

[54]Bernardo Silva et al. “OdontoAI: A human-in-the-loop labeled data set and an online platform to boost research on dental panoramic radiographs”. In: *arXiv preprint arXiv:2203.15856* (2022).

[55]Gil Silva, Luciano Oliveira, and Matheus Pithon. “Automatic segmenting teeth in X-ray images: Trends, a novel data set, benchmarking and future perspectives”. In: *Expert Systems with Applications* 107 (2018), pp. 15–31.

[56]S Sivagami et al. “Unet architecture based dental panoramic image segmentation”. In: *2020 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*. IEEE. 2020, pp. 187–191.

[57]Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[58]Ching-Wei Wang et al. “A benchmark for comparison of dental radiography analysis algorithms”. In: *Medical image analysis* 31 (2016), pp. 63–76.

[59]Huikai Wu et al. “Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation”. In: *arXiv preprint arXiv:1903.11816* (2019).

[60]Shunv Ying et al. “Caries Segmentation on Tooth X-ray Images with a Deep Network”. In: *Journal of Dentistry* (2022), p. 104076.

[61]Hang Zhang et al. “Context encoding for semantic segmentation”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 7151–7160.

[62]Yue Zhao et al. “TSASNet: Tooth segmentation on dental panoramic X-ray images by Two-Stage Attention Segmentation Network”. In: *Knowledge-Based Systems* 206 (2020), p. 106338.

[63]Haihua Zhu et al. “CariesNet: a deep learning approach for segmentation of multi-stage caries lesion from oral panoramic X-ray image”. In: *Neural Computing and Applications* (2022), pp. 1–9.